半導体記憶装置のための
バイト誤り検出・訂正符号の研究

昭和59年7月

金田重郎

半導体記憶装置のための
バイト誤り検出・訂正符号の研究

昭和59年 7 月

金　田　重　郎

〔あらまし〕

半導体記憶素子の高集積化とともに，大型計算機システムの記憶素子として，複数ビット出力の記憶素子が用いられる様になって来た。この種の記憶素子では，従来の１ビット出力記憶素子とは異なり，素子故障時に，複数ビットの塊となった誤り（バイト誤り）を生じる。従って，複数ビット出力記憶素子を用いた半導体記憶装置の高信頼化には，バイト単位の誤りを検出・訂正する符号が効果的である。

本論文では，半導体記憶装置に適したバイト誤り検出・訂正符号について，冗長度が小さく，符号化・復号化回路のＬＳＩ化が容易となる，新しい符号構成法を提案する。また，これら符号の高信頼化効果を定量的に明らかにし，符号の適用範囲について述べる。

目　次

略号一覧

本論文の主たる略号を以下に示す。

n　：符号ビット長*　　（ビット）

b　：バイト長　　（ビット）

r　：チェックビット長　（ビット）

k　：データビット長*　（ビット）

H　：パリティチェック行列（H行列）

R　：巡回性符号における巡回オペレーター
（ただし，チェックビット長rがバイト長bの倍数に限定される時には，誤解を招かない範囲でr/b をRとすることがある。）

H_0：巡回性符号のH行列を生成するための単位行列

D　：巡回性符号の巡回度（繰り返しの回数）

e　：誤りパターン

w(e)：誤りパターンの重み

S　：シンドローム

d　：ハミング距離

*）通常，nは「符号長」，kは「データ長」と呼ばれる。しかし，本論文では，符号長がガロア体GF（2）上の長さであり，ガロア体GF（2^b）上での長さで無いことを強調するため，特に，「符号ビット長」「データビット長」と呼ぶ。

第１章 序論

１．１ 研究の目的

近年，情報処理システムの大規模化，故障により生じる社会的影響の増大等により，計算機システムの高信頼化が，ますます重要となっている。

大規模情報処理システムを高信頼化するためには，システムを構成する論理装置，主記憶装置等の各装置の高信頼化が必要であり，なかでも，主記憶装置をはじめとする半導体記憶装置の高信頼化は，装置を構成する部品点数が多いため，特に重要である。

現状の半導体記憶装置は，記憶素子として，１ビット出力の記憶素子を使用して構成され，１ビット誤り訂正・２ビット誤り検出符号（ＳＥＣ－ＤＥＤ符号）による誤り制御を行っている。しかし，近年，記憶素子の高集積化とともに，複数ビットの出力を有する記憶素子が用いられる様になりつつある。この種の複数ビット出力素子では，素子の故障時に，複数ビットの塊となった誤りを生じるため，従来のＳＥＣ－ＤＥＤ符号に代わって，複数ビットの誤りを検出・訂正できる符号が，装置高信頼化のために必要となる。

本研究は，複数ビットの誤りを検出・訂正する新しい符号を考案し，これにより，経済性・信頼性に優れた半導体記憶装置を提供することを目的とする。

１．２ 記憶装置のための誤り検出・訂正符号に関する研究の歴史

誤り検出・訂正符号の研究[(1)-(8)]は，1948年のShannon の通信理論の発表にまでさかのぼることができ，その基礎はHamming [(9)] により1950年に確立された。その後，特にディジタル通信システムへの適用を中心として，目覚ましい発展を遂げてきた。

誤り検出・訂正符号の計算機への適用[(10)-(12)]は，磁気テープ記憶装置，磁気ディスク記憶装置などのファイル記憶装置[(13)-(15)]に始まり，その後，主記憶装置[(16)-(17)]にも使用される様になった。しかしながら，ファイル記憶装置用誤り検出・訂正符号に対する要件と，主記憶装置（半導体記憶装置）用誤り検出・訂正符号に対する要件の間には，大きな差がある。

ファイル記憶装置に対する誤り検出・訂正符号の適用

ファイル記憶装置では，媒体の製造歩留り向上を目的として，誤り訂正符号の機能が活用されており，このため，高い誤り訂正能力が要求される。反面，媒体が比較的安価であるため，符号の冗長度*)を大きく取ることが可能であり，符号化・復号化速度に対する要求も（半導体記憶装置に比較すれば）厳しくない。従って，伝送用符号で使用される巡回シフトレジスタを用いた逐次的な符号化・復号法を用いることができる。

このような観点から，ファイル記憶装置[15]には，ファイア符号等のバースト誤り訂正符号が使用されてきた。更に，近年，6250BPI 高密度磁気テープ記憶装置[13]，超大容量記憶装置（ＭＳＳ）[14]等には，従来のファイア符号に代わって，インターリーブ，Erasure 訂正等の手法により誤り訂正能力を強化した，単一バイト誤り訂正符号（単一隣接誤り訂正符号）が用いられる様になっている。

*）チェックビット長／符号ビット長を，本論文では符号の「冗長度」と呼ぶ。

半導体記憶装置に対する誤り検出・訂正符号の適用

主記憶装置等の半導体記憶装置では，極めて高速の符号化・復号化が要求され，又，記憶素子自体が高価であるため，符号の冗長度を極力抑える必要がある。反面，ファイル記憶装置に比較すると，誤りの検出・訂正能力を比較的低く抑えることができる。誤り検出・訂正符号が適用されている半導体記憶装置は，主として，主記憶装置であり，主記憶装置にたいしては，IBM7030 (1961)，IBM360 ／85 (1968) 以来，１ビット誤り訂正・２ビット誤り検出能力を持つハミング符号（ＳＥＣ－ＤＥＤ符号）[16]が広く用いられている。

ここで，半導体記憶装置のための誤り検出・訂正符号の研究について概観する。

半導体記憶装置に対する誤り検出・訂正符号の研究は，1970年のHsiao による最小重みＳＥＣ－ＤＥＤ符号の提案[16]により一つの転機を迎えた。これ以後，Hsiao をはじめとする米国ＩＢＭ社のグループを中心として，記憶装置用誤り検出・訂正符号の研究は大きく進展する。この最小重みＳＥＣ－ＤＥＤ符号は，H行列（パリティチェック行列）の行間演算により，従来のＳＥＣ－ＤＥＤ符号に一致するため，理論上の新規性には乏しい。しかし，従来のハミングＳＥＣ－ＤＥＤ符号と比較すると，

(1)符号化・復号化遅延が最小

(2)H行列にモジュラな繰り返し性を与えることができ，符号化・復号化回路のLSI 化に適す

等の特徴を有し，今日，ミニコン以上の計算機の主記憶装置には，不可欠のものとなっている。日本国内でも，主要な計算機メーカーは，モジュラな構成を持つ最小重みＳＥＣ－ＤＥＤ符号の符号化・復号化ＬＳＩを製造し，自社装置に適用している。

半導体記憶装置への適用をねらった誤り検出・訂正符号の研究としては，その後

(1)ビット単位の誤りを扱い，更に誤り訂正能力の大きな符号の研究，

(2)複数ビット出力記憶素子に適した，バイト誤り検出・訂正符号の研究，

の２方向に分かれて発展した。

前者のビット単位の誤りを訂正する符号の研究としては，２ビット誤り訂正符号の研究が代表的であり，種々の興味深い符号構成が提案されている[18]-[25]が，大型計算機への適用は，商用機のごく特殊な装置を除いて，報告されていない。２ビット誤り訂正符号については，本論文の主旨からはなれるので，詳しく述べることはしない。

後者のバイト単位の誤りを検出・訂正する符号の研究は，1970年のBossen[26]による，単一バイト誤り訂正符号（単一b 隣接誤り訂正符号）の提案にはじまる。Bossenの単一バイト誤り訂正符号は，チェックビット長がｂの倍数に限定されているが，その後Hong,Patel[27]は，単一バイト誤り訂正符号として任意のチェックビット長を持ち，理論的に最大の符号ビット長を持つ符号の構成法を提案した。

更に符号の誤り検出能力を高め，単一バイト誤りの訂正だけでなく，二重のバイト誤りの検出までも可能とした符号（単一バイト誤り訂正・二重バイト誤り検出符号）は，Reed & Solomon[28]により提案されており，符号化・復号化回路の構成法の研究[54]，多重誤りに対する符号の誤り検出能力についての研究[44]等がなされている。更に，２バイト以上の多重バイト誤りを訂正する符号の研究もなされているが，近い将来に半導体記憶装置に適用される可能性は薄いので，多重バイト誤り訂正符号については議論を省略する。

上記のバイト誤りを訂正できる符号は，訂正能力が高いものの，一方で，符号の冗長度が現状のＳＥＣ－ＤＥＤ符号に比べて大きく，装置の経済性が悪化する問題点を持っている。この問題点を回避するため，Bossen[29]により，バイト誤りに対しては，誤りの検出にとどめて，このバイト誤り検出能力を１ビット誤り訂正符号に賦与する提案がなされた。この種のバイト誤り検出符号については，その後，Reddy [30],Fujiwara [31]らにより，符号の構成法に改良が加えられ，さらに，Varanasi[37]，Chen[38]らにより，すぐれた符号構成が，提案された。

しかし，上記の従来のバイト誤り検出・訂正符号は，かならずしも，複数ビット出力記憶素子の故障救済を対象に提案されたものではなく，複数ビットを読みだすことの出来る記憶カード（プリント板）等の複数ビット単位に物理的境界を持つモジュールの誤りを訂正することを目的に提案された符号である。従って，既存のバイト誤り検出・訂正符号を現実の半導体記憶装置に適用しようとすると，必要とする符号ビット長が確保できない，符号化・復号化回路のＬＳＩ化が困難である，符号化・復号化遅延が現状のＳＥＣ－ＤＥＤ符号に比べて増大する等の問題を生じる。

１．３　研究の動機と主要な成果

本研究の主要な動機は以下に示す通りである。

(1)記憶素子の高集積化に伴い，複数ビット出力の記憶素子が主記憶装置をはじめとする半導体記憶装置に使用される様になってきた。複数ビット出力素子においては，素子の故障時に複数ビットの塊となった誤りを生じる可能性があるため，現状の１ビット誤り訂正・２ビット誤り検出符号（ＳＥＣ－ＤＥＤ符号）に代わって，複数ビットの誤りを検出・訂正するバイト誤り検出・訂正符号が，装置高信頼化に必須となる。

(2)従来提案されているバイト誤り検出・訂正符号は，複数ビット出力記憶素子への適用を前提として開発されていない。このため，複数ビット出力記憶素子を用いた半導体記憶装置に適用しようとすると，必要な符号ビット長が確保できない，チェックビット長が現状のＳＥＣ－ＤＥＤ符号に比べて増大する等の問題点が生じる。従って，複数ビット出力記憶素子に適した，新しいバイト誤り検出・訂正符号を必要とする。

⑶半導体記憶装置に対する誤り検出・訂正符号の適用にあたっては，高信頼化・高速化の観点から，符号化・復号化回路のＬＳＩ化が望まれる。しかし，そのままＬＳＩ化すると，ゲート数が大きいため，ＬＳＩ化のコストが増大する。従って，符号自体に符号化・復号化回路のＬＳＩ化に適したモジュラな性質を与えた，巡回性バイト誤り検出・訂正符号の開発が必要である。

⑷半導体記憶装置の性能は，システム全体の性能に大きな影響を与える。従って，誤り訂正・検出のための時間（符号化・復号化遅延）を短く抑えた符号が必要である。

この様な観点から，本論文では，半導体記憶装置への実用に耐えうる，新しいバイト誤り検出・訂正符号の構成法を追究する。主要な成果を以下に示す。

主要な成果

（Ａ）単一ｂビットバースト誤り検出ＳＥＣ－ＤＥＤ符号の提案[39]-[40]

本符号は，従来，Bossen,Reddy,Fujiwara により提案されていた１ビット誤り訂正・２ビット誤り検出・単一バイト誤り検出符号（ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号）を特殊解として包含する。

（Ｂ）新しい奇数重み列ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の提案[42]-[43]

本符号は，チェックビット長が大きくなるにつれて，その符号ビット長が，（ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の符号ビット長の自明な上界である）ＳＥＣ－ＤＥＤ符号の符号ビット長に漸近し，チェックビット長の長い領域で，現在知られている最長の符号ビット長を持つ。

（Ｃ）最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の提案[45]-[48]

本符号は，冗長度が小さく，しかも符号化・復号化のハードウェア量が最小となる最小重み符号であり，実用性の高い諸元であるチェックビット長ｒ＝ｂ＋２ビットまたはバイト長ｂ＝４ビットで，現在知られている最長の符号ビット長を持つ。

（D）新しい単一バイト誤り訂正・二重バイト誤り検出符号の提案[49]-[52]

従来の単一バイト誤り訂正・二重バイト誤り検出符号（ＳｂＥＣ－ＤｂＥＤ符号）は，符号ビット長を自由に延ばすことができない。これに対して，本符号は，任意の符号ビット長を選択できる。

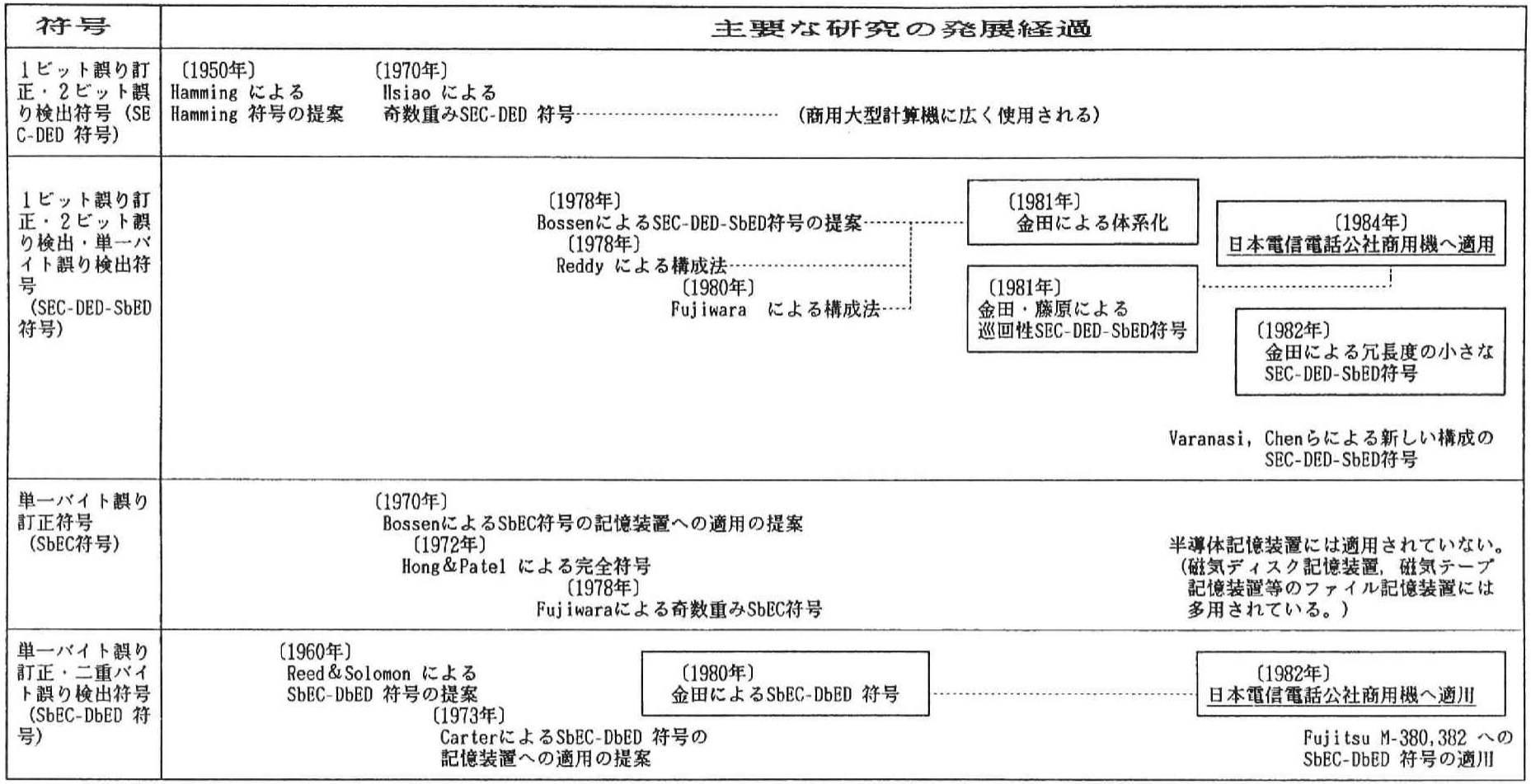
（E）巡回性ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号，巡回性ＳｂＥＣ－ＤｂＥＤ符号の提案[49]-[52]

符号化・復号化回路のＬＳＩ化に適したモジュラな構成を持つ巡回性ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号，および巡回性ＳｂＥＣ－ＤｂＥＤ符号の構成法を提案した。

（F）半導体記憶装置故障率の近似計算法の提案[62] [63] [77]

半導体記憶装置の故障率を見通し良く算出する，近似計算法を提案し，特に，ソフトエラーに対する，誤り検出・訂正符号の効果を定量的に明らかにした。

更に，上記の２種の符号（ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号，ＳｂＥＣ－ＤｂＥＤ符号）を，商用機に導入し，符号の実用性をシステム的観点からも明らかにした。以上の成果の位置づけを図1-1 に示す。



注1）　　で囲ったものは，本研究の成果を示す。

図１－１　本研究の位置づけ

１．４　本論文の概要

以下，第２章では，半導体記憶装置のためのバイト誤り検出・訂正符号に対する要件について詳説する。次に，第３章では，１ビット誤り訂正・２ビット誤り検出・単一バイト誤り検出符号（ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号）の構成法を３通り提案する。ひとつは，従来のBossen,Reddy,Fujiwara によるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を体系化した符号（構成Ａ）であり，これらの符号がＳＥＣ－ＤＥＤ－ＳｂＥＤ機能のみではなく，単一のバースト誤りを検出する機能を持つことを明らかにする。次に，奇数重み列の条件を満たす新しいＳＥＣ－ＤＥＤ－ＳｂＥＤ符号（構成Ｂ）を提案する。そして，最後に，既存のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号にくらべて，冗長度の小さな最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号（構成Ｃ）を提案する。第４章では，単一バイト誤り訂正・二重バイト誤り検出符号（ＳｂＥＣ－ＤｂＥＤ符号）の構成法を提案する。符号の構成法は２種にわかれ，はじめに，理論的な構成法を示し，次に，符号化・復号化回路のＬＳＩ化に適するモジュラな構成を持つＳｂＥＣ－ＤｂＥＤ符号の構成法を示す。第５章では，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号とＳｂＥＣ－ＤｂＥＤ符号の符号化・復号化回路量，遅延等を評価する。第６章では，半導体記憶装置における誤り検出・訂正符号の高信頼化効果を近似的に算出する手法を示す。第７章は，符号の実用化結果について述べる。第８章は，本論文のまとめである。

第２章 誤り検出・訂正符号が満たすべき要件

２．１ 緒言

本章では，主記憶装置[32]-[34]をはじめとする半導体記憶装置のための誤り検出・訂正符号が満たすべき条件について，符号機能，経済性，符号化・復号化速度等の観点から考察する。

２．２ 符号構成上の要件

2. 2. 1 符号機能

半導体記憶装置のための誤り検出・訂正符号は，その対象とする誤りのパターンに対応して，ビット系の符号と，バイト系の符号，及び，その中間的な符号とに区別することが出来る。

ｉ．ビット系符号

１ビット誤り，２ビット誤り等のランダムなビット誤りを訂正・検出する符号を，本論文では，ビット系の符号と呼ぶ。ビット系の符号は，１ビット出力の記憶素子を用いた半導体記憶装置の高信頼化に適している。代表的なビット系の符号を以下に示す。

(1)１ビット誤り訂正符号（ＳＥＣ符号…Single Error Correcting code)

(2)１ビット誤り訂正・２ビット誤り検出符号（ＳＥＣ－ＤＥＤ符号……
……Single Error Correcting-Double Error Detecting code)

(3)２ビット誤り訂正符号（ＤＥＣ符号……Double Error Correcting code)

(4)２ビット誤り訂正・３ビット誤り検出符号（ＤＥＣ－ＴＥＤ符号……
……Double Error Correcting —Triple Error Detecting code)

上記の中で，訂正能力が２ビット以上の符号（ＤＥＣ符号，ＤＥＣ－ＴＥＤ符号等）は研究対象として興味深い。しかし，現状の半導体記憶装置では，信頼度確保のために，２ビット以上の誤り訂正を必要とすることはまれであり，本論文では，これら，多数ビット誤り訂正符号については割愛する。

ＳＥＣ－ＤＥＤ符号は，今日，半導体記憶装置のための誤り検出・訂正符号として広く使用されている。一方，ＳＥＣ符号を適用した場合には，訂正可能な１ビット誤りが生じた後，さらに同一読出し単位に誤りが重畳して合計２ビットの誤りとなった際，この２ビットの誤りを検出できない。従って，一般の商用装置にＳＥＣ符号を適用すると，検出されない誤りの発生を防止する観点から，１ビット誤りが生じた段階で直ちに記憶装置自体の動作を停止する必要が生じ，符号の１ビット誤り訂正能力を生かし得ない。この理由から，商用機の誤り検出・訂正のためにＳＥＣ符号が使用される可能性は小さい。

ii．バイト系符号

複数ビットの塊となった誤りを検出・訂正する能力を持つ符号であり，複数ビット出力記憶素子に適している。まず，「バイト誤り」を以下の様に定義する。

〔定義：バイト誤り〕

符号ビット長ｎの符号語の各ビットを$d_0, d_1, d_2, \ldots, d_{n-1}$とする。ここで，ｎは，２以上の正整数ｂの倍数とする。以下に示す様に，隣接するｂビット毎に符号語をｎ／ｂ個に区分する時，区分された各ｂビット単位の塊を「バイト」と呼び，誤りが，１個のバイトのなかに限定して発生した場合を単一バイト誤り，２個のバイトにまたがって発生した場合を二重バイト誤りと呼ぶ。以下，同様に，Ｍ個のバイトにまたがって発生した場合をＭ重バイト誤りと呼ぶ。

符号語＝ $(d_0, d_1, \ldots, d_{b-1} \mid d_b, d_{b+1}, d_{b+2}, \ldots, d_{2b-1} \mid$

$\ldots\ldots \mid d_{n-b}, d_{n-b+1}, \ldots, d_{n-1})$

上記の定義により，図2-1(a)に示す様に，ｂビット以下の誤りが特定の１バイトに集中

して生じているならば，この誤りを単一バイト誤りとよぶ。更に，２ビット以上，２ｂビット以下の誤りが，２バイトにまたがって発生している時には，これを二重バイト誤りとよぶ。具体的には，この「バイト」は，複数ビット出力記憶素子のｂビットの出力に相当する。

複数ビットの塊となった誤りには，上記のバイト誤りの他に，「バースト誤り」が知られている。バースト誤りでは，バイト毎の境界を考慮することはなく，長さｂビットの隣接した誤りは，符号語のどの位置にあっても，ｂビットのバースト誤りとして扱われる。図2-1(b)には，単一のｂビットバースト誤りの例を図示しておく。

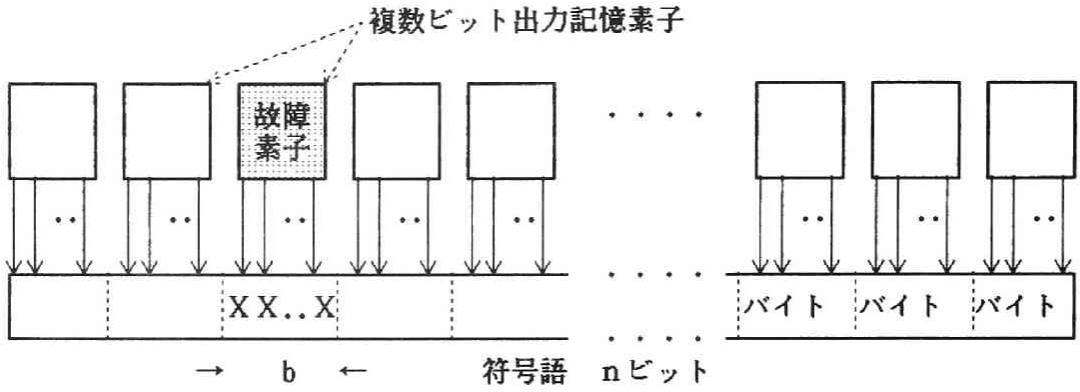
以下に主要なバイト誤り検出・訂正符号を示す。

(1)単一バイト誤り訂正符号（ＳｂＥＣ符号……
…Single b-bit byte Error Correcting code）

(2)単一バイト誤り訂正・二重バイト誤り検出符号（ＳｂＥＣ－ＤｂＥＤ符号……
…Single b-bit byte Error Correcting-Double b-bit byte Error Detecting code)

(3)二重バイト誤り訂正符号（ＤｂＥＣ符号……
…Double b-bit byte Error Correcting code)

(4)二重バイト誤り訂正・三重バイト誤り検出符号（ＤｂＥＣ－ＴｂＥＤ符号……
…Double b-bit byte Error Correcting —Triple b-bit byte Error Detecting code)

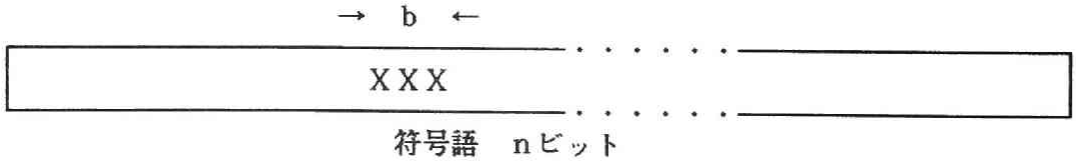


バイト誤り（単一バイト誤り）

注1）
X：誤りビット

注2）
誤りが，M個のバイトの中にまたがって生じている時，これをM重バイト誤りとよぶ。

（a） バイト誤り



注1）
X：誤りビット

注2）
誤りが，隣接したbビットの中に限定されている時，この誤りを単一bビットバースト誤りとよぶ。

（b） バースト誤り

図2-1 バイト誤りとバースト誤りの比較

これらのバイト系符号のなかで，近い将来実用に供される可能性のあるのは，単一バイト誤り訂正符号（ＳｂＥＣ符号），単一バイト誤り訂正・二重バイト誤り検出符号（ＳｂＥＣ－ＤｂＥＤ符号）である。ただし，ＳｂＥＣ符号は，ｂの値が４程度と小さくなると，二重のバイト誤り検出確率が小さくなる。従って，ＳｂＥＣ符号をｂの値が小さい時に使用することは，訂正能力を越える誤りに対する検出能力の観点から，問題を残す。ＳｂＥＣ符号は，バイト長ｂの値が大きい（８，９等）範囲で，実用的である。

一方，ＳｂＥＣ－ＤｂＥＤ符号の既存の構成法は，符号ビット長が限定される問題点を有している。このため，ｂ＝４に対する最大データビット長は，５６ビットにとどまり，データビット長６４ビット，１２８ビット等の実用的な諸元に対しては符号を適用できない。従って，ＳｂＥＣ－ＤｂＥＤ符号に関しては，半導体記憶装置用符号として必要な諸元を確保し得る，新しい構成法が望まれる。

iii．ビット系とバイト系の中間的な符号

上記のバイト系の符号では，バイト誤りの訂正までも可能とした。これに対して，バイト誤りに対しては，誤り検出にとどめ，ビット系の符号にこのバイト誤り検出能力を付加した符号が提案されている。これらの符号は，複数ビット出力記憶素子への適用をねらった符号であり，誤りの訂正能力は，バイト系の符号に劣るが，符号の冗長度が小さく，経済的に装置に適用できる。

(1) １ビット誤り訂正・単一バイト誤り検出符号（ＳＥＣ－ＳｂＥＤ符号……
…Single Error Correcting- Single b-bit byte Error Detecting code ）

(2) １ビット誤り訂正・２ビット誤り検出・単一バイト誤り検出符号（ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号…Single Error Correcting-Double Error Detecting— Single b-bit byte Error Detecting code ）

これらの中間的な符号は，記憶素子のセルやセンスアンプが故障した際の誤りの波及が１ビットに限定される可能性が高い場合に，実用性が高い。ただし，訂正可能な誤りにさらに次の誤りが重畳した時の誤り検出能力については，個々の符号毎に吟味しておく必要がある。これら中間的な符号の構成法は，種々提案されているが，本論文で述べる様に，現状のＳＥＣ－ＤＥＤ符号と比較すると，チェックビット長が長く，装置経済化の観点か

ら，改良余地を残している。

2. 2. 2 経済性・性能からの半導体記憶装置用符号が満たすべき要件

半導体記憶装置のための誤り検出・訂正符号では，特に以下の観点から符号構成を最適化しておく必要がある。

(1)高速な符号化・復号化

主記憶装置をはじめとする半導体記憶装置の性能（アクセスタイム，サイクルタイム）は，直接，システム性能に影響し，誤り検出・訂正のための時間は，装置性能のクリティカルパスに含まれる。このため，半導体記憶装置では，高速な符号化・復号化が要求され，通信用誤り検出・訂正符号等において使用される巡回シフトレジスタによる逐次的な符号化・復号化手法を適用することはできず，組合せ回路（主として排他的ＯＲによるツリーから構成される）による並列的な符号化・復号化を実施する必要がある。さらに，符号の構成上も，並列符号化・復号化に要するハードウェア量・遅延が極力小さくなる様に配慮する必要がある。

(2)小さなチェックビット長

誤り検出・訂正符号を適用するためには，本来のデータビットの他に，冗長な付加ビット（チェックビット）を必要とする。半導体記憶素子は，比較的高価であり，記憶素子数の増加は，そのまま装置価格の上昇を引き起こすため，極力，チェックビットの割合（冗長度）の小さな誤り検出・訂正符号が必要である。言い替えると，同一チェックビット長にたいしては，出来るだけ符号ビット長の大きく取れる符号が望ましい。

2. 2. 3 多重バイト誤りに対する符号の検出能力

ＳｂＥＣ符号，ＳＥＣ－ＳｂＥＤ符号，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号は，訂正可能な誤りが生じた読みだし単位に，さらに誤りが重畳して，（二重のバイトにまたがる）多数ビット誤りとなった時の誤り検出を保証していない。そこで，若干の例を用いて，これらの符号の多数ビット誤りに対する検出率を調べ，半導体記憶装置に適した符号機能を明確

化する。なお，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の多数ビット誤り検出能力については，別途，論文中で議論する。

ＳＥＣ－ＳｂＥＣ符号

ＳＥＣ－ＳｂＥＤ符号の構成法は，種々提案されているが，ここでは，代表的なBossenの構成法[29]を使用する。図2-2 に，符号ビット長ｎ＝４０ビット，バイト長ｂ＝４の場合のBossenによるＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号のパリティチェック行列（Ｈ行列）を示す。

1111	0000	0000	0000	1111	1111	1111	0000	0000	0000
0000	1111	0000	0000	1111	0000	0000	1111	1111	0000
0000	0000	1111	0000	0000	1111	0000	1111	0000	1111
0000	0000	0000	1111	0000	0000	1111	0000	1111	1111
0100	0100	0100	0100	0100	0100	0100	0100	0100	0100
0010	0010	0010	0010	0010	0010	0010	0010	0010	0010
0001	0001	0001	0001	0001	0001	0001	0001	0001	0001

図2-2　(40,33) ＳＥＣ－Ｓ４ＥＤ符号のＨ行列

一般に，符号の誤り検出・訂正能力を越えた，多数ビットの誤りに対する検出能力を理論的に評価することは困難であり，評価には，計算機によるシミュレーションを必要とする。図2-2 の符号について，計算機シミュレーションにより，複数ビットの誤りに対する検出率を求めた結果を表2-1 にしめす。ただし，誤りの生じたバイトをｉ，ｊとする。（ｉとｊは相等しくなってはならない。）

表2-1　シミュレーション結果（検出率）

		バイトｉの誤りビット数			
		1ビット	2ビット	3ビット	4ビット
バイトｊの誤りビット数	4ビット	360／／360	540／／540	0／／360	0／／45
	3ビット	1080／／1440	1080／／2160	420／／720	
	2ビット	1080／／2160	1350／／1620		
	1ビット	420／／720			

（表2-1 で，分母は計算機により発生させた可能な誤りパターンの総数。分子は検出できる誤りパターンの総数。）

ここでは，被符号化データビット長が３３ビットの場合を示したが，３２，６４，１２８ビットでも，検出率は大差無い。この結果から，ＳＥＣ－Ｓ４ＥＤ符号を使用すると，例えば，本来システムダウンとして扱うべき２ビット誤りのなかで，３０％以上が検出できず見過ごされることになる。従って，特にｂの値が４程度と小さい時には，ＳＥＣ－ＳｂＥＤ符号は，実用に適さない。

ＳｂＥＣ符号

ＳｂＥＣ符号[26]の２重バイト誤りに対する検出率は，チェックビット長ｒ＝２ｂビットの時には，理論的に求まり，

$$p = \frac{k}{2^b - 1} \quad \cdots\cdots (2\text{-}1)$$

となる。従って，上記の式から，バイト長ｂ＝８，データビット長ｋ＝６４，１２８等では，二重バイト誤りを９９％以上の確率で検出でき，ＳｂＥＣ－ＤｂＥＤ符号を使用する必要はないことが判る。事実，この様な観点から，6250BPI 磁気テープ記録装置等には，ｂ＝８のＳｂＥＣ符号が使用されている。

一方，ｂ＝４の場合には，データビット長３２ビットの符号で，ｐ＝８／１５となり，二重バイト誤りの５０％近くを検出できない。更に，データビット長ｋ＝６４，１２８ビットでは，チェックビット長を増加させる必要があり，符号のH行列の構成法により，２重バイト誤りに対する検出率が変化する。例えば，（Ｓ４ＥＣ－Ｄ４ＥＤ符号と同一冗長である），１６ビットのチェックビット長を持つデータビット長６４ビットのＳ４ＥＣ符号の２重バイト誤りに対する検出率は，高々８５％程度である（著者による）。

以上の様なシミュレーション結果から，ＳＥＣ－ＳｂＥＤ符号，ＳｂＥＣ符号は，特にｂの値が４程度と小さい時には，実用装置に導入することは困難であり，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号，ＳｂＥＣ－ＤｂＥＤ符号等を必要とすると結論できる。

本節で述べた分析から，著者は，表2-2 に示すバイト誤り・検出符号が半導体記憶装置に効果的・実用的であると考えている。

表2-2　半導体記憶装置のための誤り検出訂正符号

符号略称	誤り検出・訂正能力	特徴	備考
ＳＥＣ－ＤＥＤ符号	１ビット誤り訂正・２ビット誤り検出	チェックビット長が短い。	１ビット出力記憶素子に適し，広く，主記憶装置に使用されている。
ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号	１ビット誤り訂正・２ビット誤り検出・単一ｂビットバイト誤り検出	チェックビット長がＳＥＣ－ＤＥＤ符号とほぼ同等であり，経済的に装置に適用できる。	複数ビット出力記憶素子に適し，特に，単一の記憶素子が故障した時，２ビット以上の誤りをふくむ単一バイト誤りを生じる可能性が低い時に有効である。
ＳｂＥＣ符号	単一ｂビットバイト誤り訂正	ｂが大きい時（例えばｂ＝８）には，二重のバイト誤りに対する検出率が高くなり，実用的である。	複数ビット出力記憶素子に適する。
ＳｂＥＣ－ＤｂＥＤ符号	単一ｂビットバイト誤り訂正・二重ｂビットバイト誤り検出	高信頼度の要求される装置に適し，特に，ｂが４程度と小さい時に，効果的である。	複数ビット出力記憶素子に適する。

2.3 巡回性符号* と奇数重み列符号

半導体記憶装置のための誤り検出・訂正符号の応用では，データビット長が３２，６４，１２８ビット等の値に限定されている。このため，符号は，与えられたチェックビット長にたいする最長の符号ビット長ではなく，短縮化して使用される。このことから，逆に，種々の制限を符号の構成に与え，実用上の利点を得ることが提案されている。

本節では，これらの符号構成技術について述べる。

巡回性符号（モジュラな構成を持つ符号）[35]

前述した様に，半導体記憶装置のための誤り検出・訂正では，高速の誤り検出・訂正が要求され，組合せ回路による並列符号化・復号化が必須である。しかし，誤り検出・訂正のハードウエア量は，数千ゲートに達し，遅延の削減，誤り検出・訂正回路自体の故障率低減の観点から，ＬＳＩ化が望ましい。

しかし，数千ゲートの符号化・復号化回路をそのままＬＳＩ化することは，ＬＳＩのゲート規模，ピン数等の観点から実現が難しく，同一のＬＳＩを複数個使用することにより，符号化・復号化回路を構成することが望まれる。巡回性符号はこの様な観点から，ＩＢＭ社のグループにより提案された手法であり，そのH行列に，モジュラな繰り返し性を持つため，符号化・復号化回路にモジュラな繰り返し（レピータビリティ）を与えることができる。

巡回性符号のH行列を式（2 —2a）に示す。巡回性符号のH行列は，生成行列H_0をr／Dビットずつ列方向に巡回置換した行列としてH行列が与えられる。このため，H_0を巡回置換した行列である各H_j毎に，同一のハードウェアを使用でき，符号化・復号化回路にモジュラな性質を賦与できる。ただし，rはチェックビット長，Dは巡回度であり，Dはrの約数でなければならない。Rをr ／Dビットの巡回置換を実行するオペレーターとして，H_jは式（2-2b）の様に表される。R行列の空白は「０」を意味する。

＊巡回性符号と類似した用語に「巡回符号」がある。この巡回符号は英語のcyclic codeに対応する。本論文では，rotational code の訳語として，「巡回性符号」を使用する。
なお，ＬＳＩ技術の進展に伴い，近年，大きなゲート量を持つＬＳＩを容易に作成できる様になり，第７章で述べる様に，符号化・復号化回路の繰り返しとしては，２程度で十分な場合も多い。しかし，巡回化自体の意義がこれにより失われたわけではなく，ＬＳＩ内部をモジュラ化することは，設計，試験コスト削減上の意義を持つ。

$$H = [H_0, H_1, \cdots, H_{D-1}] \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2\text{-}2a)$$

$$H_j = R^j \cdot H_0 \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2\text{-}2b)$$

$$R = \begin{bmatrix} & & & & 1 \\ & & & & 1 & 1 \\ & & & & & 1 & 1 \\ & & & & & & \ddots & \ddots \\ & & & & & & & 1 & 1 \\ 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & \ddots & & & & \\ & & & & & \ddots & & & \\ & & & & & & \ddots & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \\ & & & & & & & & & 1 \end{bmatrix}$$

r/D

r − r/D

r × r

(注) 空白は「0」。

ここで，巡回性符号のH行列の例を以下にしめす。この符号は3章でのべる最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号のｂ＝3，ｒ＝5の構成例である。H行列全体は，5の繰り返し性を持ち，H_0は5行3列の行列である。

$$H = \left[\begin{array}{ccc|ccc|ccc|ccc|ccc} 1 & 1 & 1 & & & & 1 & & & 1 & & & & & 1 \\ & 1 & & 1 & 1 & 1 & & 1 & & 1 & & & 1 & & \\ 1 & & & & 1 & & 1 & 1 & 1 & & 1 & & 1 & & \\ 1 & & & 1 & & & & 1 & & 1 & 1 & 1 & & 1 & \\ & 1 & & 1 & & & 1 & & & & 1 & & 1 & 1 & 1 \end{array}\right] \quad \cdots\cdots\cdots\cdots\cdots (2\text{-}3)$$

(注) 空白は「0」。

←H→	←H→	←H→	←H→	←H→
0	1	2	3	4

既存の巡回性符号は，容易に巡回化できるＳＥＣ符号，ＳＥＣ―ＤＥＤ符号[35]，ＳｂＥＣ符号[36]のみであり，従来は，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号，ＳｂＥＣ－ＤｂＥＤ符号の巡回化は困難とされていた。

最小重み符号

H行列中の「１」の個数をH行列の重みとよび，符号のH行列の重みが，その能力を持つ符号として最小になる符号を，最小重み符号とよぶ。特に，Hsiao による最小重みＳＥＣ－ＤＥＤ符号[16]は良く知られている。

一般に，最小重み符号は，

⑴符号化・復号化のためのハードウェア量が最小

⑵符号化・復号化のための遅延が最小

等の利点を持つ。しかし，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号，ＳｂＥＣ－ＤｂＥＤ符号については，最小重みの符号は知られていない。最小重み化と巡回性は，全く別個の制限条件であるが，実用上，これらの条件を同時に満足する符号がのぞまれる。

奇数重み列符号

符号のH行列の各列ベクトルがすべて奇数重みとなる符号を奇数重み列符号と呼ぶ。Hsiao による最小重みＳＥＣ－ＤＥＤ符号は，奇数重み列符号でもある。

一般に，符号を奇数重み列化すると，

⑴短縮化して符号を使用した時に，多数ビット誤りの検出率が高い

⑵シンドロームの排他的ＯＲを取るのみで，１ビット誤りの有無を判断でき，誤りの検出回路が簡明

等の利点があり，特に，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の奇数重み列化が望まれる。

２．４　結語

複数ビット出力記憶素子を用いた半導体記憶装置のための誤り検出・訂正符号は，

⑴バイト単位の誤りを検出・訂正するバイト系符号，

⑵バイト単位の誤りは，検出にとどめ，この能力をビット系符号に付加した中間的な符号

に区分できる。このなかで，符号の訂正能力，多数ビット誤りの検出確率等を考慮し，半導体記憶装置に適した符号機能として，以下の符号に注目する。

⑴バイト系の符号としては，単一バイト誤り訂正符号（ＳｂＥＣ符号），単一バイト誤り訂正・二重バイト誤り検出符号（ＳｂＥＣ－ＤｂＥＤ符号）が実用性が高い。特に，ＳｂＥＣ－ＤｂＥＤ符号は，記憶素子の出力ビット数（ｂ）が４程度と小さい範囲で，実用的である。

ＳｂＥＣ符号に関しては，すでに多くの研究があり，半導体記憶装置用符号として研究の余地は少ない。一方，ＳｂＥＣ－ＤｂＥＤ符号に関しては，既存の構成法では符号ビット長が限定され，ｂ＝２，３，４等で，半導体記憶装置として必要なデータビット長を持つ符号を構成できない。このため，任意の符号ビット長に対して符号を構成できる，新しいＳｂＥＣ－ＤｂＥＤ符号を必要とする。

⑵中間的な符号としては，１ビット誤り訂正・２ビット誤り検出・単一バイト誤り検出符号（ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号）が実用的である。しかし，既存のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号は，ＳＥＣ－ＤＥＤ符号に比べて，チェックビット長が増大する問題点を有し，装置経済化の観点から，チェックビット長の削減が望まれる。

⑶さらに，符号化・復号化の高速化，ＬＳＩ化の容易化等の観点から，

符号化・復号化回路にモジュラな性質を持つ巡回性符号

符号化・復号化回路の遅延が最小となる，最小重み符号

の条件を満たすＳｂＥＣ－ＤｂＥＤ符号，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号が望まれる。

第３章　ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号

３．１　緒言

本章では，１ビット誤り訂正・２ビット誤り検出・単一バイト誤り検出符号（ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号）について，以下の新しい符号構成を提案する。

(a)単一バースト誤り検出ＳＥＣ－ＤＥＤ符号[39]-[40]　・・・・・構成Ａ

（この単一バースト誤り検出ＳＥＣ－ＤＥＤ符号は，既存のBossen,Reddy,Fujiwara 等によるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を包含する。）

(b)奇数重み列ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号[42]-[43]　・・・・・構成Ｂ

(c)最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号[45]-[49]　・・・・・・構成Ｃ

上記の符号の中で，(a)の単一バースト誤り検出ＳＥＣ－ＤＥＤ符号は，既存のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号[29]-[31]を体系化した符号である。(b)の奇数重み列ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号は，符号化・復号化回路のＬＳＩ化に適した符号である。(c)の最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号は，実用性の高い諸元であるバイト長ｂ＝４，あるいは，チェックビット長ｒ＝ｂ＋２において，現在知られている最長の符号ビット長を持ち，更に，符号化・復号化回路のＬＳＩ化に適する。

以下，最初に，既存のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の問題点について示し，つぎに，上記の各構成毎に，符号の構成法，最大符号ビット長等を示す。

３．２　既存のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号について

本論に入る前に，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の研究の歴史と動向について，紹介する。

ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号から２ビット誤り検出能力を取り去った，ＳＥＣ－Ｓｂ

ＥＤ符号は，Bossen[29]により提案された。ここで，Bossenによるｂ＝４，ｒ＝６のＳＥＣ－Ｓ４ＥＤ符号のＨ行列を以下に示す。

H＝

1 1 1 1 0 0 0 0 0 0 0 0	0 0 0 0 1 1 1 1 0 0 0 0	0 0 0 0 0 0 0 0 1 1 1 1	1 1 1 1 1 1 1 1 0 0 0 0	1 1 1 1 0 0 0 0 1 1 1 1	0 0 0 0 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1
0 1 0 0 0 0 1 0 0 0 0 1	0 1 0 0 0 0 1 0 0 0 0 1	0 1 0 0 0 0 1 0 0 0 0 1	0 1 0 0 0 0 1 0 0 0 0 1	0 1 0 0 0 0 1 0 0 0 0 1	0 1 0 0 0 0 1 0 0 0 0 1	0 1 0 0 0 0 1 0 0 0 0 1

Bossenによる（28,22）ＳＥＣ－Ｓ4 ＥＤ符号のＨ行列

この例からも分かる様に，Bossenの構成法では，ｂ×ｂの単位行列から最上部の１行を取り去った（b-1 行b 列）行列を，各バイト毎にならべ，そのあとで，バイトを区別するための行を付加してゆく。したがって，最大符号ビット長は，

$$n = b\,(2^{r-b+1} - 1)$$

で与えられる。

上記のBossenの論文では，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号については，触れていない。しかし，このBossenの符号に，all '1' 行を付加すれば，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号となることは自明であり，その意味で，BossenをＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の提案者と見なすこともできる。この場合のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の符号ビット長は，以下の式で与えられる。本論文では，BossenのＳＥＣ－ＳｂＥＤ符号にall '1' 行を付加することにより構成されるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を，BossenのＳＥＣ－ＤＥＤ－ＳｂＥＤ符号と呼ぶ。

$$n = b \cdot 2^{r-b}$$

その後，Reddy[30] は，このBossenの構成法を改良して，ｂが５以上では更に小さなチェックビット長で，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を構成できることを示した。符号ビット長は以下の様になる。

$$n = b\,(2^{r-b+1} - 1) \qquad \text{ただし，} b \geq 5$$

ただし，Bossen,Reddyの符号は，巡回性符号でもなく，また，最小重み符号でもない。

さらに，その後，Fujiwara[31]によりＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の構成法が提案されたが，本論文でのべる様に，数学理論上は，Bossenの符号と同一である。

その後，著者の研究[45]-[48]とほぼ同時に，２つのグループ（Varanasi，Chen）[37][38]から，新しいＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の構成法が提案された。著者の符号を含めて，これらの効率の良い符号は，複雑な構成法を使用しており，従来のBossen,Reddy等の「シンプルな構成」では，あまり効率の良い符号は構成できないものと考えられる。特に著者のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号は，実用性の高い符号諸元で，Varanasi，Chenによる符号に比べて，大きな符号ビット長を持っている。これらＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の各構成法の最大符号ビット長の比較は，本章末で示す。

本論文で提案するＳＥＣ－ＤＥＤ－ＳｂＥＤ符号により，半導体記憶装置用符号として実用的な諸元を持つＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の開発は終焉したと考えられる。ただし，現在のところ，本質的な限界符号ビット長にほぼ達したと考えられるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号は，ｂ＝４（著者），３（Chen）のみであり，ｂ＞４かつｒ≫ｂ＋２の領域でさらに長い符号ビット長を持つ符号を構成することは，理論的課題として残されている。

3.3 単一バースト誤り検出ＳＥＣ－ＤＥＤ符号[39]-[40] ・・（構成Ａ）

本章では，近年一般的となりつつある，ニブル転送方式[74]を用いた記憶素子に適した誤り検出・訂正符号として，グループ分割パリティチェックによりバイト誤り（バースト誤り）を検出できるＳＥＣ－ＤＥＤ符号の構成法[39]-[40]を提案する。

ニブル転送型の記憶素子

近年，２５６Ｋビット素子等のＭＯＳ記憶素子[75]-[76]では，ニブル転送方式[74]の入出力方式を用いた記憶素子が登場してきた。このニブル転送方式では，複数ビットのデータ出力を，１ビットのデータピンから，時分割的に入力（出力）する。従って，ニブル転送方式を用いた記憶素子は，複数ビット出力記憶素子と見なすことができる。

ニブル転送方式を用いた記憶素子において，ｂ回に分けて出力される複数ビットを，それぞれ別の符号化単位にふり分ける事が可能ならば，誤り検出・訂正用符号はＳＥＣ－ＤＥＤ符号で良い。しかし，１個の記憶素子から出力される複数ビットを，異なる符号化単位に分割すると，記憶素子のスループットが低下して，システム設計のネックとなる。

この様な観点から，ニブル転送方式を用いた記憶素子においても，１個の記憶素子から出力されるｂビットを，同一の符号化単位に収容することが望ましい。このばあい，ニブル転送方式の記憶素子には，バイト誤り検出・訂正符号が効果的となる。

ニブル転送型素子に適した符号構成

ニブル転送型読み書きモードを持つ記憶素子を使用した記憶装置において，従来のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号に見られる，符号語全体の中でのバイト誤り検出の手法を適用すると，ｂ個の転送単位を一旦，どこかに記憶して，この読みだし単位全体に対してシンドロームを生成する必要がある。

本節では，ニブル転送動作によって，１個の記憶素子から時分割的にデータが読みだされる場合でも，各転送単位毎のパリティ検査のみで，単一の素子故障が検出可能なＳＥＣ－ＤＥＤ符号の構成法を提案する。本構成法では，符号語（符号ビット長はｂの倍数）の各バイトから１ビットずつ取り出して作成したｂ個のグループ内のパリティ和が被符号化

情報にかかわらず一定となる様に符号を構成する。本符号は，ｂビット以下のバースト誤りを検出する単一バースト誤り検出ＳＥＣ－ＤＥＤ符号であり，Bossen,Reddy,Fujiwaraらにより提案されている既存のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号[29]-[31]を特殊解として包含する。

３．３．１　単一バースト誤り検出ＳＥＣ－ＤＥＤ符号の構成法[39]-[40]

図3-1 に示す様に，転送単位毎のパリティチェックのみで，バイト誤りを検出することを考える。次の用語を定義しておく。

〔定義3-1 〕ｂ分割パリティチェック

符号語（但し，符号ビット長ｎはｂの倍数であるものとする。）を，ｎ／ｂビットずつ，ｂ個のグループに等分する時，各グループのパリティ和が被符号化情報にかかわらず一定であるならば，その符号はｂ分割パリティチェック可能であると呼ぶ。

ｂ分割パリティチェック可能な符号の機能

まず，次の定義3-2 であたえられるグループ化行列Ａを考える。（Ａの階数がｂである理由は後述する。）

〔定義3-2 〕グループ化行列Ａ

ｂをバイト長，ｒをチェックビット長とする。ｂ行ｒ列の２進行列Ａを考え，これをグループ化行列と名づける。行列Ａは任意に定め得るが，その階数（rank）はｂでなければならない。

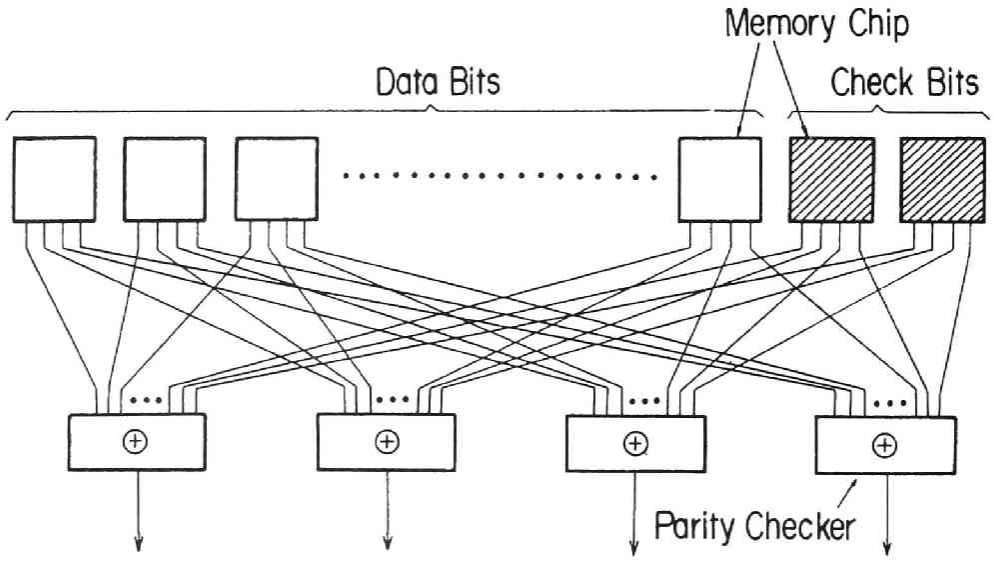


図3-1　4ビット分割パリティチェックの例

このグループ化行列Aに従って，H行列から次式のように，b個の行ベクトルg_0，g_1，....，g_{b-1}を作成する。但し，H行列の行ベクトルをf_0，f_1，....，f_{r-1}，Aのi行j列要素をa_{ij}とする時，g_iは$a_{ij}=1$となるjに対するすべてのf_jのベクトル和である。

$$\begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \\ \cdot \\ \cdot \\ \cdot \\ g_{b-1} \end{bmatrix} = \begin{bmatrix} \\ A \\ \\ \end{bmatrix} \cdot \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ \cdot \\ \cdot \\ \cdot \\ f_{r-1} \end{bmatrix} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (3\text{-}1)$$

図3-2 は，符号のH行列がグループ化行列Aによりグループ化される例である。この場合，バイト長bは4ビット，チェックビット長rは8ビットである。

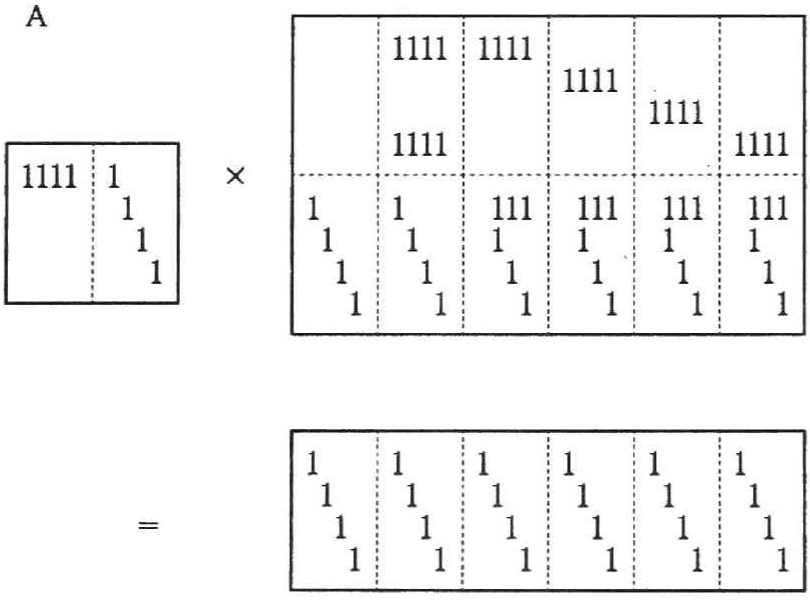


図3-2　グループ化行列Aの例

このグループ化行列Ａを用いて，ｂ分割パリティチェック可能な符号構成を表す。

〔定理3-1〕ｂ分割パリティチェック可能な符号の条件

グループ化行列ＡをＨ行列に乗じた結果得られる行列ＡＨが，ｂ行ｂ列の単位行列を行方向に並べた行列と一致する時，また，その時に限って，そのＨ行列で表される符号はｂ分割パリティチェック可能である。ただし，グループ分割時において，同一グループに属するビットは，各バイト内の同一ビット位置をしめるものとする。

（証明）Ｈ行列の行ベクトルをf_0，f_1，....，f_{r-1}とする。符号語をｎ行１列の列ベクトルＣとする時，Ｃの定義から次式が成立する。

$$f_j \cdot C = 0 \quad \text{for all } j \qquad \cdots\cdots (3\text{-}2)$$

g_iを構成するために用いられる行ベクトルf_jをq_0，q_1，....，q_{pi}とする。ここで，piはg_iを生成するためのf_jの個数である。g_iは次式で表される。

$$g_i = q_0 + q_1 + . . + q_{pi} \qquad \cdots\cdots (3\text{-}3)$$

ベクトルg_iにおいて１が立っているビット位置に対して符号語のパリティ和を作成すると次式の様になる。

$$\begin{aligned} g_i \cdot C &= (q_0 + q_1 + . . + q_{pi}) \cdot C \\ &= q_0 \cdot C + q_1 \cdot C + . . + q_{pi} \cdot C \\ &= 0 \end{aligned} \qquad \cdots\cdots (3\text{-}4)$$

行列ＡＨの各列ベクトルの重みが１であり，かつ，行列ＡＨの行方向の重みがすべて等しいことから，ベクトルg_iにおいて，１が立っているビット位置に対してパリティ和を取ると，このパリティチェックは他グループとは独立に実行しうる。従って，この場合の行列Ｈで与えられる符号はｂ分割パリティチェック可能である。逆も同様にして証明できる。　（Ｑ．Ｅ．Ｄ．）

この定理3-1 は，H行列に対して行間演算（ある行ベクトルを他の行ベクトルに加算する操作を次々と行う。）をほどこし，ｂ行ｂ列の単位行列を行方向に並べた行列をH行列中に生成することができれば，そのH行列で表される符号はｂ分割パリティチェック可能であることを示している。

ここで，式（3-2）は，偶数パリティにより，チェックビットが生成されていることを意味している。この場合，式（3-4）から，各グループ毎のパリティチェックは偶数パリティチェックでよい。これに対して，チェックビット付加が奇数パリティにより行われている時は，(1)ベクトルq_jの個数piが奇数の時は奇数パリティチェック，(2)ベクトルq_jの個数が偶数の時は偶数パリティチェック，を用いて，グループ毎のパリティチェックを実行する必要がある。

以上の準備のもとに，次の定理が成立する。

〔定理3-2〕単一バースト誤り検出ＳＥＣ－ＤＥＤ符号

H行列の列ベクトルが互いに相異なり，しかもｂ分割パリティチェック可能な符号は，単一バースト誤り検出ＳＥＣ－ＤＥＤ符号であり，その最大符号ビット長ｎは次式で与えられる。

$$n = b \cdot 2^{r-b} \quad \text{...(3-5)}$$

（証明）H行列の行ベクトルをf_0，f_1，....，f_{r-1}とする。f_0，f_1，...，f_{r-1}を加算してg_0，g_1，....，g_{b-1}が生成されるから，H行列の列ベクトルとしてａｌｌ‘0’ベクトルは存在しない。従って，１ビットの訂正機能は保証される。

次に，g_0，g_1，....，g_{b-1}をすべて加算すると，ａｌｌ‘1’行ベクトルが得られる。従って，もとのH行列の行ベクトルh_jから適当な行を選んで加算すれば，ａｌｌ‘1’行ベクトルが得られるはずである。よって，２ビット誤り検出機能が保証される。

次に，グループ化行列Ａをシンドロームに乗じた結果得られる列ベクトルを考える。この乗算の結果得られるｂビットはｂ分割パリティチェック結果に等しく，単一バイト誤りの誤りパターンに等しくなる。さらに，複数ビットの誤りが，単一のバイト誤りでなく，

ｂビット以下のバースト誤りであっても，誤りビット数とｂ分割パリティチェック結果の重みは一致する。従って，訂正可能な１ビット誤りと，訂正不能な２ビット以上のバースト誤りとは，このバースト誤りのパターンから識別出来るから，符号は単一（ｂビット）バースト誤り検出ＳＥＣ－ＤＥＤ機能を持つ。

以上の論議から，ｂ分割パリティチェック可能なＳＥＣ符号は単一（ｂビット）バースト誤り検出ＳＥＣ－ＤＥＤ符号である。次に，最大符号ビット長について考察する。

まず，パリティチェックの単位である，１グループ内の最大列ベクトル数を調べる。今，グループｉに所属する列ベクトルを（x_0，x_1，....，x_{b-1}）^{T}（但し，^{T}は転置を表す。）と表す。列ベクトルの要素ｘはガロア体ＧＦ（２）の元である。本列ベクトルは次式を満たさねばならない。　ただし，左辺は上からｉ番目のみが‘１’で，他の要素は‘０’の列ベクトルである（最上部の要素のみが‘１’の時を０番目とする）。

$$i)\quad \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_b = \begin{bmatrix} \\ A \\ \\ \end{bmatrix}_{b\times r} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{r-1} \end{bmatrix} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (3\text{-}6)$$

定義（3-2）で述べたように，Ａの階数はｂである。従って，良く知られているように，１次方程式（3-6）の解は，（ｒ－ｂ）個の解の線形結合として得られる。従って列ベクトル（x_0，x_1，....，x_{b-1}）^{T}は2^{r-b}個ある。

同様にして，他グループでの列ベクトル数が得られ，これから，符号の最大符号ビット長はｂ・2^{r-b}となる。但し，式（3-6）において，Ａの階数がｂより小さいと，式（3-6）の解の個数が少なくなる。従って，グループ化行列Ａの階数はｂでなければならない。

（Ｑ．Ｅ．Ｄ．）

上記定理3-2 に従って，具体的に符号の構成例を示す。ｒ＝８，ｂ＝４とし，グループ化行列Ａとしては，次式を選ぶ（行列中の空白は０）。

$$A = \left[\begin{array}{cccc:cccc} 1 & 1 & 1 & 1 & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{array}\right] \quad \cdots\cdots\cdots\cdots\cdots\cdots (3\text{-}7)$$

式（3-7 ）を式（3-6 ）に代入して，グループｉ＝０～３に対する列ベクトルを計算すると，図3-3 の様になる。図中で重みと書かれたものは列ベクトルの重みである。

次に，各グループから1 列ずつ列ベクトルを取り出し，１バイトあたりのＨ行列の列ベクトルを構成する。グループから列を取り出す順序は任意である。しかし，Ｈ行列の重みは符号化・復号化回路のハードウェア量にほぼ比例するから，重みの小さな列ベクトルから取り出すことにする。図3-4 はこの様にして構成した符号であり，データビット長３２ビットを持つ。

更に，表3-1 には，単一バースト誤り検出ＳＥＣ－ＤＥＤ符号の符号ビット長を示す。後述する様に，本符号ビット長は，Bossen，Reddy ，FujiwaraによるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の符号ビット長に等しい。

グループ0

	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1
	0	0	0	1	0	0	1	1	1	0	0	0	1	1	1	1
	0	0	1	0	0	1	0	1	1	0	1	1	0	0	1	1
	0	1	0	0	0	1	1	0	1	1	0	1	0	1	0	1
列＝	1	0	0	0	0	1	1	1	0	1	1	0	1	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
重み＝	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3	5

グループ1

	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1
	0	0	0	0	1	1	1	0	0	0	1	1	0	1	1	1
	0	0	1	1	0	0	1	0	0	1	0	1	1	0	1	1
	0	1	0	1	0	1	0	0	1	0	0	1	1	1	0	1
列＝	0	1	1	0	1	0	0	1	0	0	0	1	1	1	1	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
重み＝	1	3	3	3	3	3	3	3	3	3	3	5	5	5	5	5

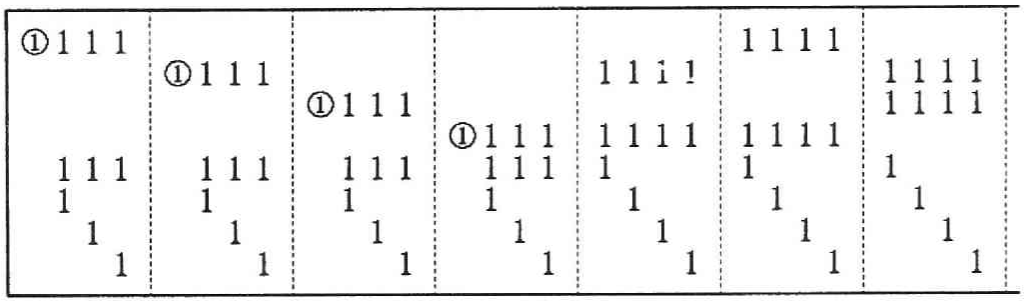
グループ2

	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1
	0	0	0	0	1	1	1	0	0	0	1	1	0	1	1	1
	0	0	1	1	0	0	1	0	0	1	0	1	1	0	1	1
	0	1	0	1	0	1	0	0	1	0	0	1	1	1	0	1
列＝	0	1	1	0	1	0	0	1	0	0	0	1	1	1	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
重み＝	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	5

グループ3

	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1
	0	0	0	0	1	1	1	0	0	0	1	1	0	1	1	1
	0	0	1	1	0	0	1	0	0	1	0	1	1	0	1	1
	0	1	0	1	0	1	0	0	1	0	0	1	1	1	0	1
列＝	0	1	1	0	1	0	0	1	0	0	0	0	1	1	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
重み＝	1	3	3	3	3	3	3	3	3	3	3	5	5	5	5	5

図3-3　グループ0～3に対する列ベクトル



〔b＝4〕

(注) 行列中の空白は「0」,
①はチェックビットを示す。

図3-4 (40,32) 4ビットバースト誤り検出ＳＥＣ－ＤＥＤ符号のH行列

表3-1　単一バースト誤り検出ＳＥＣ－ＤＥＤ符号の符号ビット長

		b：　バイト長（ビット）							
		3	4	6	8	9	12	16	32
r：チェックビット長［ビット］	b＋2	12	16	24	32	36	48	64	128
	b＋3	24	32	48	64	72	96	128	256
	b＋4	48	64	96	128	144	192	256	512
	b＋5	96	128	192	256	288	384	512	1024
	b＋6	192	256	384	512	576	768	1024	2048
	b＋7	384	512	768	1024	1152	1536	2048	4096
	b＋8	768	1024	1536	2048	2304	3072	4096	8192
	b＋9	1536	2048	3072	4096	4608	6144	8192	16384

３．３．２　既存の符号と本符号との関係

ここで，図3-4 の符号はFujiwaraによるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号に等しい。言い換えると，式（3-7 ）のグループ化行列ＡはFujiwara[31]によるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を生成するグループ化行列である。Fujiwaraは，上記図3-4 の符号が，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号であるとしているが，上記の定理により，この符号は単一ｂビットのバースト誤り検出能力を持つＳＥＣ－ＤＥＤ符号である。

ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号はこの他に，Bossen[29]，Reddy [30]等の提案がある。これら既存のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号のなかで，上記の単一バースト誤り検出ＳＥＣ－ＤＥＤの特殊解となる符号は以下の通りである。

(1)BossenによるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号

(2)Reddy によるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号（ただし，ｂ＜５）

(3)FujiwaraによるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号

各符号を生成するグループ化行列Ａを付録3-3 にまとめておく。前述した，Bossen，Reddy （ｂ＜５），Fujiwara，のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号は，Ｈ行列の行間演算により，互いに等価な符号であることを容易に示すことができる。しかし，定理3-2 から，これらＳＥＣ－ＤＥＤ－ＳｂＥＤ符号においては，Ｈ行列の行間演算に対して符号機能が変化しないとの符号一般の自由度以外に，同一グループ内であれば列ベクトルを自由に交換可能であることがあらたに主張される。

この列ベクトルを自由に交換可能な性質によって，符号化・復号化回路のＬＳＩ化に適したモジュラな性質を有する符号を構成したり，符号化・復号化回路の段数を削減するために，Ｈ行列の行方向の重みを平均化することができる（詳細後述）。

グループ化行列Ａには，次の様な性質がある。

〔性質１〕

グループ化行列Ａの各列ベクトルの重みがすべて奇数であれば，このグループ化行列Ａから生成される単一バースト誤り検出ＳＥＣ－ＤＥＤ符号は，奇数重み列の条件を満たす。

（証明）グループ０について考える。グループ０に属する列ベクトルを（x_0，x_1 ..... x_{b-1}）^{T} とする。次式が成立する。

$$
\begin{bmatrix} 1 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \\ \\ 0 \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \\ A \\ \\ \\ \\ \\ \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_i \\ \cdot \\ \\ \cdot \\ x_{b-1} \end{bmatrix} \quad \cdots\cdots (3\text{-}8)
$$

式（3-8）において，Ａの各列ベクトルの中で，x_iが１であるビットに対応した列のベクトルを加算したものが，左辺の列ベクトルである。この時，式（3-8）の右辺のＡの各列ベクトルも奇数重みである。列ベクトル　$(x_0, x_1, \ldots. x_{b-1})^T$ のx_i中，偶数個が１であると仮定する。この場合は，奇数重みのベクトルを偶数個加算した結果が奇数重みとなり，不合理である。x_i中，奇数個が１であるとすると，奇数重みのベクトルを奇数個加算した結果が奇数重みとなり不合理は生じない。　　　　（Q．E．D．）

〔性質２〕

グループ化行列Ａの各列ベクトルの重みがすべて１であれば，このグループ化行列Ａから生成される単一バースト誤り検出ＳＥＣ－ＤＥＤ符号は，Ｈ行列中の列ベクトルの重みが１であるチェック列をｒ列含む。

〔性質３〕

グループ化行列Ａの各列ベクトルの重みがすべて１であり，かつ，Ａの行方向の重みがすべて等しい時には，このグループ化行列Ａから生成される単一バースト誤り検出ＳＥＣ－ＤＥＤ符号は，チェック列を特定のｒ／ｂバイトに集中させ得る。

性質２，性質３の証明は省略する。

グループ化行列Aを用いた誤り検出回路の構成

定理3-2 の符号では，その証明で述べたように，シンドロームにグループ化行列Aを乗ずると単一バースト誤りのパターンとなる。従って，シンドロームからバースト誤り（バイト誤り）を検出する誤り検出回路の構成は容易である。シンドロームにAを乗じたのち，その結果が重み1であれば，訂正可能な1ビット誤り，重みが2以上であれば，訂正不可能なバースト誤りとなる。しかも，同一グループに属する列ベクトルの位置を入れ替えても，誤り検出回路に影響はない。この性質を用いると，誤り検出回路を少ないゲート数で構成できる。

符号化・復号化回路の最適化

定理3-2 より，単一バースト誤り検出ＳＥＣ－ＤＥＤ符号を構成してゆく場合，グループ化行列Aの一般性，及びグループ内での列ベクトルの交換可能性を利用して，b分割パリティチェック可能な符号のなかでの，符号化・復号化回路遅延の最小化，符号化・復号化回路のハードウェア量の最小化が可能となる。

符号化遅延削減のため，実用上の観点からは，H行列中に重み1のチェック列がr個あることが望ましい。従って，〔性質2〕から，グループ化行列Aの列方向の重みはすべて1であるとする。列方向の重みが1のグループ化行列Aの種類は，整数rを空でないb個の部分に分割する場合の数に等しい。このすべてのグループ化行列Aについて，各グループ化行列A毎にH行列の重みが最小となる符号をつくり，これら符号のなかで最も重みの小さな符号を選択すれば，単一バースト誤り検出ＳＥＣ－ＤＥＤ符号として，最小重みの符号が得られる。

符号ビット長４０ビット，データビット長３２ビットの単一4ビットバースト誤り検出ＳＥＣ－ＤＥＤ符号について，最小重み符号の生成結果を表3-2 に示す。表3-2 の各符号は重みが最小となるだけではなく，H行列の行方向の重みも平均化する様配慮してある。最小重み符号は，分割（5，1，1，1）の符号であり，図3-4 の符号と一致する。しかし，表3-2 から，符号化遅延が最小となる符号は，上記の分割（5，1，1，1）の符号ではなく，分割（2，2，2，2）の符号であることがわかる。分割（2，2，2，2）の符号は，次章で提案する巡回性単一バースト誤り検出ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号に他ならない。

表3-2　グループ化行列Aと遅延

グループ化行列Aの行方向の重み	最小重みとなる符号		備考
	H行列の重み	符号化遅延＊	
(2，2，2，2)	120	4	図3-7
(5，1，1，1)	104	5	図3-4
(4，2，1，1)	112	5	
(3，2，2，1)	112	5	
(3，3，1，1)	116	5	

＊）2入力EXOR換算

３．３．３　巡回性単一バースト誤り検出ＳＥＣ－ＤＥＤ符号の構成法

既存のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号では，符号化・復号化回路のＬＳＩ化に適したモジュラな性質を符号に賦与することは出来なかった。本節では，前述の単一バースト誤り検出ＳＥＣ－ＤＥＤ符号に，符号化・復号化回路のＬＳＩ化に適した，モジュラな構成を与える。

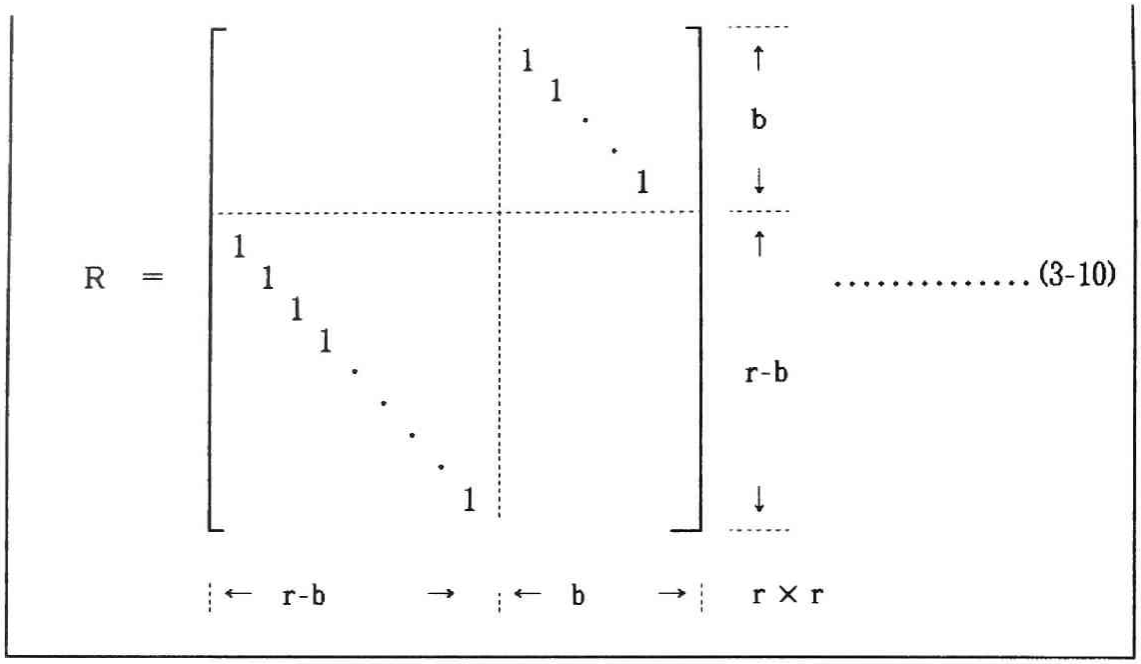
巡回性符号の定義を以下に示す。ただし，巡回オペレーターの巡回ビット数はｂビットに限定した。これは，記憶素子の出力数がｂビットであるため，ｂビット以下の細かい単位で巡回させてモジュラな構成としても，素子の物理的な単位との整合性が悪いためである。

〔定義3-3〕巡回性符号[35]

符号のＨ行列が，生成行列Ｈ$_0$を，列方向に巡回置換した行列から構成されている時，これを巡回性符号と言う。巡回度をＤとすると，Ｈ行列は次式で表される。Ｄはｒ／ｂに等しい。

$$H = [H_0 \mid R \cdot H_0 \mid R^2 \cdot H_0 \mid \cdots \mid R^{D-1} \cdot H_0] \quad \cdots\cdots\cdots\cdots\cdots (3\text{-}9)$$

但し，Ｒはｂビットの巡回置換を行うオペレーターであり，次式で与えられる（行列中の空白は０）。



この巡回性符号では，H_0 に対するハードウェアをそのまま，他の部分の行列R^j H_0 に入出力を差し替えて使用でき，符号化・復号化回路のＬＳＩ化に都合が良い。

次の定理3-3 が成立する。

〔定理3-3 〕巡回性単一バースト誤り検出ＳＥＣ－ＤＥＤ符号

グループ化行列Ａが，単位行列$I_{b\times b}$を行方向にｒ／ｂ個ならべた構成である時

$$A = [\, I_{b\times b} \mid I_{b\times b} \mid \cdots \mid I_{b\times b} \,] \quad \cdots\cdots (3\text{-}11)$$

上記Ａから生成される符号は，Ｄ＝ｒ／ｂ回の巡回性を有する単一バースト誤り検出ＳＥＣ－ＤＥＤ符号であり，その符号ビット長は

$$n = b \sum_{\substack{d \mid (r/b) \\ d:\text{odd}}} \mu(d) \cdot 2^{(r/d-b)} \quad \cdots\cdots (3\text{-}12)$$

である。　ただし，Σは（ｒ/ｂ）の奇約数に対する和を示し，μ（ｄ）はメビウス関数である。

（証明）グループ化行列Aが式（3-11）で表されているから，ある列ベクトル（x_0，x_1，....，x_{r-1}）^{T} が次式を満たす時

$$i)\quad \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \\ 0 \\ \cdot \\ \cdot \\ \\ 0 \end{bmatrix}_b = \begin{bmatrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_{r-1} \end{bmatrix} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (3\text{-}13)$$

列ベクトル（x_0，x_1，....，x_{r-1}）^{T} をbビット巡回置換した列ベクトル（x_b，x_{b+1}，....，x_{r-1}，x_0，x_1，....，x_{b-1}）^{T} は，必ず式（3-13）を満たす。従って，各グループは，bビット毎の巡回置換によって同形となる列ベクトルの群（これを巡回同値類と呼ぶ）から成っている。

bビットを単位としてd回巡回置換すると始めて線形従属となる列ベクトルの数をN（d，b）とする。式（3-11）の対称性から，明らかに，dはDの約数であり，各列ベクトルはd個の列ベクトルを含む巡回同値類の元に成っている。いま，D／dが偶数であるとすると，この列ベクトルは同一の部分列ベクトルを偶数個ならべた構成であることになり，奇数重み列符号の条件に反する。従って，D／dは奇数である。

以上から，符号ビット長は次式で与えられる。

$$b \cdot 2^{b(D-1)} = \sum_{(D/d)\ :\mathrm{odd}} d \cdot N(d,\ b) \quad \cdots\cdots\cdots\cdots\cdots\cdots (3\text{-}14)$$

これに，修正メビウス反転公式[43]を適用して，次式を得る。

$$D \cdot N(D,\ b) = \sum_{d\ :\mathrm{odd}} \mu(d) \cdot b \cdot 2^{(D/d-1)} \quad \cdots\cdots\cdots\cdots\cdots\cdots (3\text{-}15)$$

ただし，メビウス関数は，

$$\mu(d) = \begin{cases} 1 & : d = 1 \\ (-1)^{L} & : d \text{が異なるＬ個の素数の和} \\ 0 & : \text{その他} \end{cases}$$

で定義され，ｄ＝１１までの，μ（ｄ）の値は，表3-3 で与えられる。式（3-15）から，容易に式（3-12）を得る。

表3-3　メビウス関数の値

d	1	2	3	4	5	6	7	8	9	10	11
μ（d）	1	−1	−1	0	−1	1	−1	0	0	1	−1

（Q．E．D．）

式（3-5 ）及び式（3-12）から与えられる符号ビット長を表3-4 に示す。ここで，ｂをバイト長，ｒをチェックビット長として，巡回度Ｄはｒ／ｂで与えられる。巡回化による符号ビット長の短縮は極めて小さい。

図3-5 は，ｒ＝9，ｂ＝3の場合について，各グループ毎の列ベクトルをしめしている。列ベクトルは，３ビットの巡回置換によって同形となる列ベクトルを同一グループ内に必ず含んでいる。但し，「＊」を附した列ベクトルは，３ビットの巡回置換により自分自身に一致するため巡回性単一３ビットバースト誤り検出ＳＥＣ－ＤＥＤ符号の列ベクトルとして使用できない。図3-5 のグループから列を選んで構成したデータビット長３６ビットの単一３ビットバースト誤り検出ＳＥＣ－ＤＥＤ符号のＨ行列を図3-6 に示す。この符号は，重みを最小化しただけではなく，Ｈ行列の行方向の重みも平均化してある。

更に，ｂ＝４，ｒ＝８の巡回性符号の例を図3-7 に示す。巡回度は２であり，グループ化行列Ａは図3-7 中にしめしておく。前節でも述べたように，図3-7 の符号は，行方向の重みが平均化され，図3-4 の従来の符号よりも，１段少ない段数（２入力ＥＸＯＲ換算）で符号化できる。しかも，巡回性符号であるから符号化・復号化回路を繰り返し度２でＬＳＩ化できる。

表3-4　　単一バースト誤り検出ＳＥＣ－ＤＥＤと巡回性単一バースト誤り検出ＳＥＣ－ＤＥＤ符号との符号ビット長比較

r：チェックビット長［ビット］		b：バイト長（ビット）					
		1	2	3	4	5	6
	2 b	2 2	8 8	24 24	64 64	160 160	384 384
	3 b	3 4	30 32	189 192	1020 1024	5115 5120	24570 24576
	4 b	8 8	128 128	1536 1536	16384 16384	163840 163840	1572864 1572864
	5 b	15 16	512 512	12285 12288	…… ……	…… ……	…… ……

（注１）ｂ＝１では，単一バースト誤り検出ＳＥＣ－ＤＥＤ符号は，ＳＥＣ－ＤＥＤ符号と一致する。

（注２）各欄の上段（分子）は巡回性単一バースト誤り検出ＳＥＣ－ＤＥＤ符号の符号ビット長を示し，下段（分母）は単一バースト誤り検出ＳＥＣ－ＤＥＤ符号の符号ビット長を示す。

グループ0

```
                                                                                                                    *
        00101000100100110000100100101001000100101011110011110011111111111
        00001100011001100000001101101110110111011001101100011011001110 10
        00000011000000010110101110111001110111010100010111001111010101 10
        01010001001001000101001001010010001001010011100111100111111111111
列  =   00011000010111000000011011011001101110110111011000010110111001 10
        00000010100000001101111001110111001110101100001110111010101111 00
        10000110010010001010010010000100110010000111101011101011111111111
        00010100001110100000010110110111011001101110110100001101110111 00
        00000001100000011011010111001110111001111000011001110101111010 10
```

重み＝ 11133333333333333333355555555555555555555555555555555557777777773

グループ1

```
                                                                                                                    *
        00000001100000011011001101101100001101101110110101111011001110 10
        00101001000110000101000101010011110000111101000101001011111111111
        00001100011010100000010111001111010111000001110110111011010101 10
        00000011000000010110111011011000011011011001101111010110111001 10
列＝    01010010001000101010001010000111100101011110001010010011111111111
        00011000010101100000001110111010101110100011001101110110101111 00
        00000010100000001101110110110100010110110111011010101101110111 00
        10000100110001010000110000101011101010011100110000100111111111111
        00010100001111000000011001110101111001100010111011001101111010 10
```

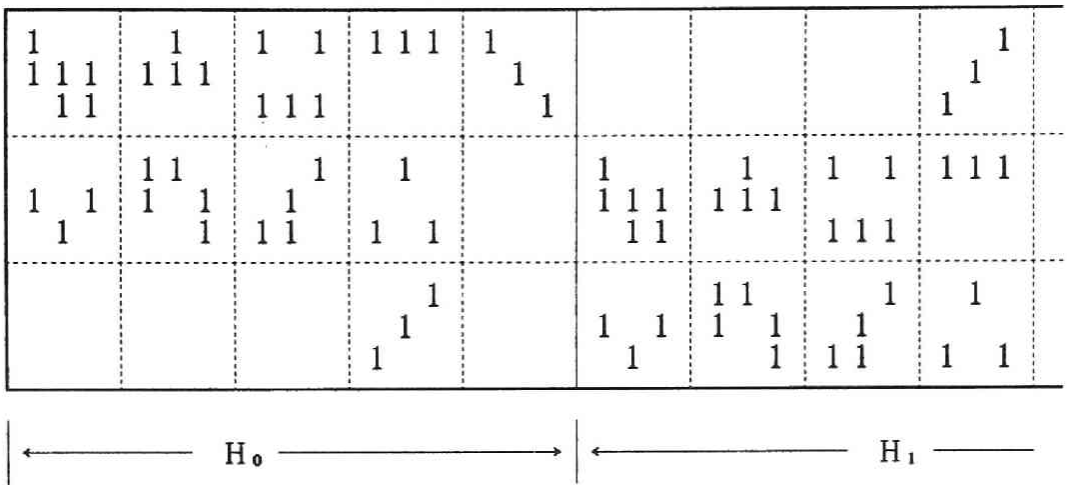
重み＝ 11133333333333333333355555555555555555555555555555555557777777773

グループ2

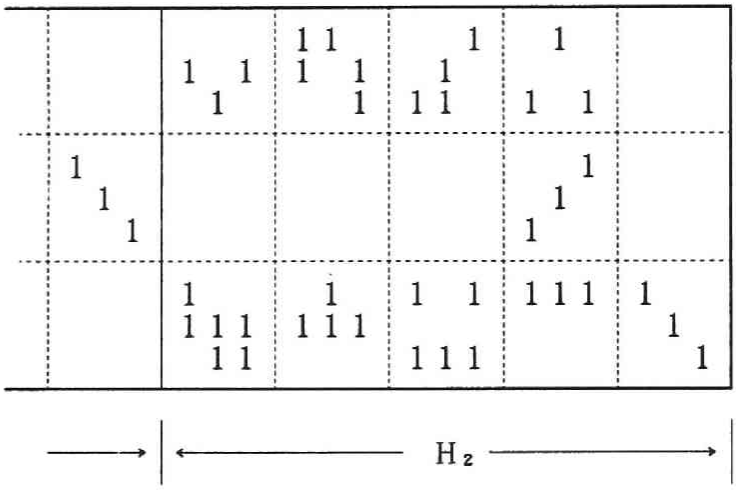
```
                                                                                                                    *
        00000001100000000111001100001101101101111010110110110111001111 00
        00001100011010100000001101110111001100001101110111011001111011 00110
        00101010000110001001000111101010010011101010000101000111111111111
        00000011000000011010111000011011011011010101101101101110111010 10
列＝    00011000010101100000011011001110111000011011001110111010110111 00
        01010000101000110010001011110000100111110000101010001011111111111
        00000010100000010101110100010110110110101111011011011001110101 10
        00010100001111000000010110111001110100010110111001110101101110 10
        10000101010001000100110011100101001011100101010000110011111111111
```

重み＝ 11133333333333333333355555555555555555555555555555555557777777773

図3-5　r＝9，b＝3の場合の各グループ毎の列ベクトル



H_0 H_1

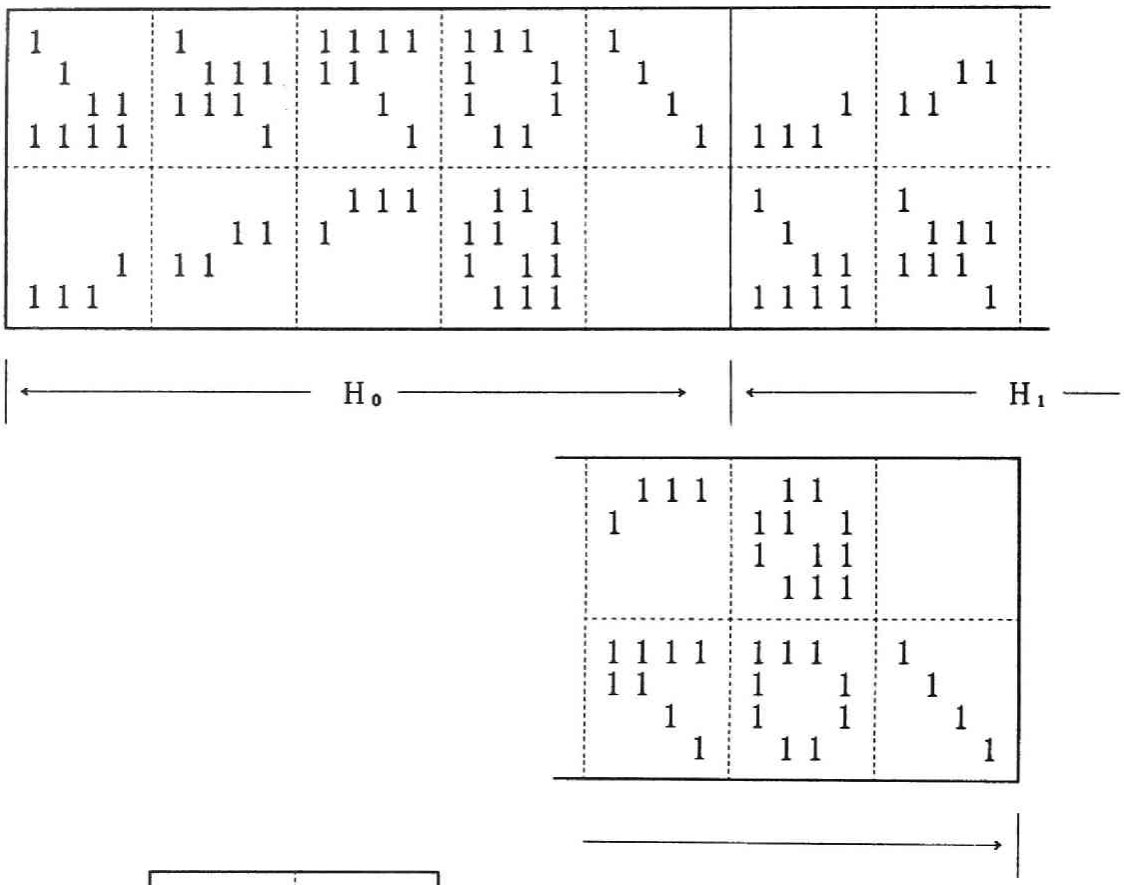


H_2

（注）行列中の空白は「０」を示す。

$A =$

図3-6　巡回性（45,36）３ビットバースト誤り検出ＳＥＣ－ＤＥＤ符号のＨ行列



（注）行列中の空白は「0」を示す。

図3-7　巡回性（40,32）4ビットバースト誤り検出ＳＥＣ－ＤＥＤ符号のH行列

3.4 奇数重み列ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号[42]-[43] ・・・（構成Ｂ）

本節では，新しい奇数重み列ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の構成法について述べ，次に，この符号から，巡回性奇数重み列ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を構成する。本構成法による符号は，前述の単一バースト誤り検出ＳＥＣ－ＤＥＤ符号とほぼ同等の最大符号ビット長を持っている。特に，チェックビット長ｒが大きな範囲（バイト長ｂの３倍以上）では，単一バースト誤り検出ＳＥＣ－ＤＥＤ符号と比較して長い符号ビット長を持つ。

3．4．1 奇数重み列ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の構成法[42] [43]

本構成法では，まず，チェックビット長ｒはバイト長ｂの整数倍であると仮定して議論を進める。ｒが任意の整数の時の構成法については，後述する。

符号のH行列は，ｂ行ｂ列の２進行列（Ｉｑ，Ｍｕ）をもとにして構成する。まず，この２種類のｂ×ｂ正方行列（Ｉｑ，Ｍｕ）を定義する。

〔定義3-3〕正則行列Ｉｑ

ｂ行ｂ列の２進単位行列をＩとして，次の様に表す。

$$I = [a_0, a_1, \ldots\ldots, a_i, \ldots, a_{b-1}] \quad \ldots\ldots\ldots\ldots\ldots\ldots (3\text{-}16)$$

ここで，a_i はｂ個の｛0，1｝の元を有する列ベクトルであり，以下の様になる。

$$a_0 = (1, 0, \ldots, 0, 0, 0, 0)^T$$
$$a_1 = (0, 1, \ldots, 0, 0, 0, 0)^T$$
$$a_2 = (0, 0, \ldots, 0, 0, 0, 0)^T$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$a_{b-1} = (0, 0, \ldots, 0, 0, 0, 1)^T \qquad T：転置$$

この単位行列を，行方向にｑ（ｍｏｄ．ｂ）ビット巡回置換して得られる正則行列をＩｑとする。Ｉｑは次の式で表される。

$$Iq = [a_{b-q}, a_{b-q+1}, \ldots a_{b-q+i}, \ldots a_{b-q+b-1}] \quad \ldots\ldots\ldots (3\text{-}17)$$

$b-q+i$ ：ｍｏｄ．ｂ

一般に，正則行列に関して以下の性質が成立する。Ｉｑは正則であるから，下記の性質を満足する* 。

〔性質〕a_i，a_jを正則行列（ｂ行ｂ列）の列ベクトルとする。

$i, j \in \{0, 1, \ldots, b-1\}$

1）a_iまたは，a_jについて，すべて０の元から成る列ベクトルは存在しない。

2）$i \neq j \mod b$に対して$a_i \neq a_j$となる。

3）正則行列の列ベクトルから，重複を許さずに，任意の個数の列ベクトルを取り出し，これらのｍｏｄ．２加算を行っても，その加算結果は，すべて０の元からなる列ベクトルとはならない。

さらに，以下の行列Ｍｕを定義する。

〔定義3-5 〕正方行列Ｍｕ

整数ｕ（$0 \leq u \leq 2^b - 1$）の２進表示を（$c_0, c_1, \ldots, c_i, \ldots, c_{b-1}$）とし，（ただし，$c_{b-1}$をＭＳＢとする。），その列ベクトルを$U = (c_0, c_1, \ldots, c_i, \ldots, c_{b-1})^T$とする。この列ベクトルＵから，次のｂ次の正方行列を定義する。ただし，ＵとしてＩｑの列ベクトルa_i（$i = 0, 1, \ldots, b-1$）と一致するものは，除外する。

$$Mu = [U, U, \ldots, U] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3\text{-}18)$$

$$U = (c_0, c_1, \ldots, c_i, \ldots, c_{b-1})^T$$

Ｉｑ自体の定義から，Ｕとして，重み１の列ベクトルは存在しないことは明らかである。

以上の定義のもとに，新しいＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を構成することができる。ただし，Ｉｑ及びMu は，それぞれｂ個，（$2^b - b$）個存在する。

＊）ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を構成するためには，行列Ｉｑは，正則で，かつ，列方向の重みの奇偶性が一定であれば良く，必ずしも，単位行列である必要はない。本論文では，説明を簡単化するために行列Ｉｑを単位行列に限定する。なお，符号化・復号化回路のゲート量の削減，符号化・復号化回路の構成の簡明さの観点からは，行列Ｉｑを単位行列とすることが有利である。

〔定理3-4〕H行列の列ベクトル*)をR個のMu，Iqを使用して構成するとき，この列ベクトル中には，少なくとも1個のIqを含み，しかも，列の重み**)は奇数重みである様に構成する。この条件のもとに，相異なる列ベクトルをならべてH行列を構成する時，このH行列で与えられる符号は，SEC－DED－SbED符号であり，符号ビット長は次式で与えられる(42)-(43)。

$$n = \frac{1}{2}\left(2^{Rb} - \left(2^{b} - b\right)^{R} + b^{R}\right) \quad \cdots\cdots\cdots\cdots\cdots\cdots (3\text{-}19)$$

但し，チェックビット長rはRbで与えられる。

*) 正確には，b行b列の行列を構成要素とする列ベクトル。

**) H行列をガロア体GF（2）で表現した場合の列方向の重み。

（証明）

符号のH行列は次式の様に与えられる。

$$H = \begin{bmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,j} & \cdots & h_{0,n/b-1} \\ h_{1,0} & h_{1,1} & \cdots & h_{1,j} & \cdots & h_{1,n/b-1} \\ \vdots & \vdots & & \vdots & & \vdots \\ h_{i,0} & h_{i,1} & \cdots & h_{i,j} & \cdots & h_{i,n/b-1} \\ \vdots & \vdots & & \vdots & & \vdots \\ h_{R-1,0} & h_{R-1,1} & \cdots & h_{R-1,J} & \cdots & h_{R-1,n/b-1} \end{bmatrix} \quad \cdots\cdots (3\text{-}20)$$

$$h_{i,j} \in \{Iq, Mu\}$$

$$q = 0, 1, 2, \ldots, b-1$$

$$u = 0, 3, 5, \ldots, 2^{b}-1$$

（但し，uは1及び2のべき乗ではない。）

正則行列Iqの性質1）2）から，列ベクトルには，all '0' ベクトルはなく，しかも各列ベクトルは相異なることから，SEC機能を有する。

次に，ＳｂＥＤの機能について考える。バイトｊ（＝０，１，２，....，ｎ－１）に誤りパターンｅｊが生じたとする。このバイト誤りのシンドロームは次式で表される。

$$\begin{bmatrix} h_{0,j} \\ h_{1,j} \\ \cdot \\ \cdot \\ h_{i,j} \\ \cdot \\ h_{R-1,j} \end{bmatrix} \cdot e_j = \begin{bmatrix} S_0 \\ S_1 \\ \cdot \\ \cdot \\ S_i \\ \cdot \\ S_{R-1} \end{bmatrix} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (3\text{-}21)$$

この時，符号の構成条件から，$h_{0,j} \ldots h_{R-1,j}$ 中には，少なくとも１個のＩｑを含む。ここで，$h_{i,j}$ ＝Ｉｑとしても，一般性は失われない。ｊバイト目の誤りパターンＥｊに対するシンドロームの関係を表3-5 に示す。表中，ｗ（ｅｊ）は誤りパターンの重み，即ち誤りビット数を示す。これから，バイト誤りのシンドロームが１ビット誤りのシンドロームに一致することはなく，バイト誤りは完全に検出できる。さらに，この符号は奇数重み列の条件を満足する符号であるから，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号となる。次に，この符号の最大符号ビット長を求める。

この符号構成においては，Ｉｑの列ベクトルは常に重み１であり，Ｍｕの列ベクトルの重みは奇数，偶数とも取りうる。これらの要素Ｉｑ，ＭｕをＲ個並べ，しかも，少なくとも１個はＩｑを含む条件を満たして，かつＨ行列が奇数重み列の条件を満足させる必要がある。従って、Ｒ個のなかに，奇数個，重み奇数の要素（Ｉｑ，Ｍｕで，列ベクトルの重みが奇数のもの）が含まれなければならない。このような場合の数（α）は次式で与えられる。

表3—5　jバイト目の誤りパターンe_jに対するシンドロームの関係

誤り シンドローム	w（e_j）＝1，（1ビット誤り）	w（e_j）$\geq$2，（2ビット誤り）
$h_{i,j}$＝Iqであるiに対するS_i	$S_i \in \{a_0, a_1, .. a_{b-1}\} \neq 0$	$S_i \in \{a_0, a_1, .. a_{b-1}\} \neq 0$
$h_{k,j}$＝Muであるkに対するS_k	S_k＝U∈{a_0，a_1，.... a_{b-1}}	S_k＝0*　または ＝U∈{a_0，a_1，..... a_{b-1}}

i，k　∈{0，1，．．．，R−1}　　0*　：零列ベクトル

$$\alpha = \sum_{L=1}^{m} {}_RC_L \quad \text{.....ただし，Lは奇数.....} \quad (3\text{-}22)$$

C：組合せの場合の数

$$m = 2\left\lceil \frac{R+1}{2} \right\rceil - 1, \quad \text{但し} \left\lceil x \right\rceil \text{はxを越えない整数}$$

例えば，R＝3の時には，次の4種類がある。

$$\begin{bmatrix} 奇 \\ 奇 \\ 奇 \end{bmatrix} \quad \begin{bmatrix} 奇 \\ 偶 \\ 偶 \end{bmatrix} \quad \begin{bmatrix} 偶 \\ 奇 \\ 偶 \end{bmatrix} \quad \begin{bmatrix} 偶 \\ 偶 \\ 奇 \end{bmatrix}$$

この時，線形独立性に注意して，「奇」の部分には，Iq，または，Mu中uの2進表示が奇数重みのものを入れ，「偶」の部分には，Mu中のuの2進表示が偶数重みのものを入れる。ただし，「奇」が1個の場合には必ずIqを入れ，複数の「奇」が存在する場合でも，少なくとも一つはIqを入れる必要がある。

Muの中で，偶数重み列を有するものは2^{b-1}個，奇数重み列を有するものは（$2^{b-1}-b$）個存在する。列ベクトル中のIqの個数をPとする時，「奇」がL個存在する列ベクトルの数δは次式で与えられる。

$$\delta_L = (2^{b-1})^{R-L} \cdot \sum_{P=1}^{L} {}_tC_P \cdot b^{P-1} \cdot (2^{b-1}-b)^{L-P}$$

$$= \frac{1}{b}\left((2^{b-1})^R - (2^{b-1})^{R-L} \cdot (2^{b-1}-b)^L\right) \quad \text{...} \quad (3\text{-}23)$$

(3 —22) 式を考慮して，全体の列ベクトルの数ｎ／ｂは次式で与えられる。

$$n/b = \frac{1}{2b}\left(2^{bR} + b^{R} - (2^{b} - b)^{R}\right) \qquad \cdots\cdots\cdots\cdots\cdots\cdots (3\text{-}24)$$

（Q．E．D）

以下の表3-6 に，この符号ビット長を示す。この符号ビット長は，チェックビット長ｒがｂの２倍の時には，ｂ分割パリティチェック可能な単一バースト誤り検出ＳＥＣ－ＤＥＤ符号と同一の符号ビット長となり，チェックビット長ｒがｂの３倍の時には，ｂ分割パリティチェック可能な符号よりも，大きな符号ビット長を持つ。ただし，このままでは，チェックビット長がｂの整数倍に限定される。

ここで，具体的な構成例をｂ＝３，４について，示す。

ｂ＝３の時

Ｉｑは以下の３通りある。

$$I_0 = \begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} \quad I_1 = \begin{matrix}0&1&0\\0&0&1\\1&0&0\end{matrix} \quad I_2 = \begin{matrix}0&0&1\\1&0&0\\0&1&0\end{matrix}$$

Ｍｕは以下の５通りとなる。

$$M_0 = \begin{matrix}0&0&0\\0&0&0\\0&0&0\end{matrix} \quad M_3 = \begin{matrix}0&0&0\\1&1&1\\1&1&1\end{matrix} \quad M_5 = \begin{matrix}1&1&1\\0&0&0\\1&1&1\end{matrix} \quad M_6 = \begin{matrix}1&1&1\\1&1&1\\0&0&0\end{matrix} \quad M_7 = \begin{matrix}1&1&1\\1&1&1\\1&1&1\end{matrix}$$

図3-8 には，チェックビット長ｒ＝９ビットのＳＥＣ－ＤＥＤ－Ｓ３ＥＤ符号の構成例を示す。

ｂ＝４の時

Ｉｑは以下の４通りある。

$$I_0 = \begin{matrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{matrix} \quad I_1 = \begin{matrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\1&0&0&0\end{matrix} \quad I_2 = \begin{matrix}0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0\end{matrix} \quad I_3 = \begin{matrix}0&0&0&1\\1&0&0&0\\0&1&0&0\\0&0&1&0\end{matrix}$$

Ｍｕは以下の様になる。

$$M_0 = \begin{matrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{matrix} \quad M_3 = \begin{matrix}0&0&0&0\\0&0&0&0\\1&1&1&1\\1&1&1&1\end{matrix} \quad M_5 = \begin{matrix}0&0&0&0\\1&1&1&1\\0&0&0&0\\1&1&1&1\end{matrix} \quad M_6 = \begin{matrix}0&0&0&0\\1&1&1&1\\1&1&1&1\\0&0&0&0\end{matrix} \cdots\cdots$$

表3-6 奇数重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の符号ビット長

		b ： バイト長（ビット）							
		1＊	2＊	3	4	5	6	7	8
r：チェックビット長［ビット］	2 b	2	8	24	64	160	384	896	2048
	3 b	4	32	207	1216	6605	33624	162967	762368
	4 b	8	128	1776	22528	258880	.....	.....	.....
	5 b	16	512	14943	400384	.....	.....	.....	.....
	6 b	32	2048	123624	.....	.....	.....	.....	.....
	7 b	64	8192	.....	.....	.....	.....	.....	.....
	8 b	128	.....	.....	.....	.....	.....	.....	.....

注１）＊：SEC-DED 符号に一致する。
注２）....部は未計算

I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0
M_3	M_5	M_6	M_0	M_0	M_0	M_3	M_3	M_3	M_5	M_5	M_5	M_6
M_0	M_0	M_0	M_3	M_5	M_6	M_3	M_5	M_6	M_3	M_5	M_6	M_3

I_0	I_0	M_0	M_0	M_0	M_3	M_5	M_6	M_3	M_5	M_6	M_3	M_5
M_6	M_6	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0
M_5	M_6	M_3	M_5	M_6	M_0	M_0	M_0	M_3	M_3	M_3	M_5	M_5

M_6	M_3	M_5	M_6	M_3	M_5	M_6	M_0	M_0	M_0	M_3	M_3	M_3
I_0	I_0	I_0	I_0	M_0	M_0	M_0	M_3	M_5	M_6	M_3	M_5	M_6
M_5	M_6	M_6	M_6	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0

M_5	M_5	M_5	M_6	M_6	M_6	I_0	M_7	M_7	I_0	I_0	I_0	M_7
M_3	M_5	M_6	M_3	M_5	M_6	M_7	I_0	M_7	I_0	I_1	I_2	I_0
I_0	I_0	I_0	I_0	I_0	I_0	M_7	M_7	I_0	M_7	M_7	M_7	I_0

M_7	M_7	I_0	I_1	I_2	I_0	I_0	I_1	I_0	I_0	I_2	I_0	I_0
I_0	I_0	M_7	M_7	M_7	I_0	I_0	I_0	I_1	I_0	I_0	I_2	I_1
I_1	I_2	I_0	I_0	I_0	I_0	I_1	I_0	I_0	I_2	I_0	I_0	I_2

I_0	I_0	M_0	M_0
I_2	M_0	I_0	M_0
I_1	M_0	M_0	I_0

〔b＝3〕

図3-8　奇数重み列（207,198）ＳＥＣ－ＤＥＤ－Ｓ３ＥＤ符号のH行列

図3-9 には，チェックビット長ｒ＝８ビットのＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号の構成例を示す。この符号は，ｂ分割パリティチェック可能なＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号と同一の最大符号ビット長を持つ。

I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0
M_0	M_3	M_5	M_6	M_9	M_{10}	M_{12}	M_{14}

M_0	M_3	M_5	M_6	M_9	M_{10}	M_{12}	M_{14}
I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0

〔ｂ＝４〕

図3-9 （64,56 ）ＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号のＨ行列

任意のチェックビット長を持つ符号の構成法

以上のべた符号の構成では，チェックビット長はバイト長ｂの整数倍に限定される。しかし，定理（3-4 ）の「チェックビット長= ｂの倍数の符号 」にチェックビットを付加することにより，任意のチェックビット長を持つＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を構成できる（詳細は文献43参照）。チェックビット長をＲｂ＋Ｌとすると，その符号ビット長は

$$n = 2^{L-1} \cdot \{ (2^b)^R + b^R - (2^b - b)^R \}$$

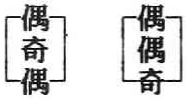
で与えられる。ただし，符号の巡回化は考慮していない。

３．４．２　巡回性奇数重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の構成法

前述したＳＥＣ－ＤＥＤ－ＳｂＥＤ符号は，その構成上，容易に符号のＨ行列に巡回性を与えることができる。

まず，ｂ行ｂ列の正方行列の列重みの奇偶性に注目する。例えば，

である列ベクトルについては，必ずその列をｂビットずつ列方向（縦方向）に巡回置換した列ベクトル



が，かならず，符号のＨ行列中に含まれるからである。

しかしながら，たとえば，

と言った重みを持つ列ベクトルは巡回すると自分自身に一致する。この場合には，巡回後の列ベクトルがもとの列ベクトルと線形独立である必要があり，例えば，

ベクトル　　（１０００｜１０００｜１０００）^{T}

は巡回置換すると自分自身に一致する。一方，

ベクトル　　（１０００｜０１００｜１０００）^{T}

は，巡回置換すると，元の列ベクトルとは線形独立となる。

これらから，以下の定理3-5 が成立する（Fujiwara）。

〔定理3-5〕

巡回性ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の最大符号ビット長ｎは次式で与えられる。

$$n = \frac{1}{2} \cdot \sum_{\substack{d \mid R \\ d\ ;\mathrm{odd}}} \mu(d) \left\{ (2^b)^{R/d} + b^{R/d} - (2^b - b)^{R/d} \right\} \mathrm{G.C.D.}(d, b) \quad \cdots\cdots (3\text{-}25)$$

図3-10には，（１９８，１８９）ＳＥＣ－ＤＥＤ－Ｓ３ＥＤ符号のＨ行列を示す。また，表3-7 には，巡回化した奇数重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の符号ビット長を巡回する前の符号と比較して示す。

表3-7 奇数重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の巡回化による符号ビット長の短縮

		b：バイト長 （ビット）					
		1＊	2＊	3	4	6	8
r：チェックビット長（ビット）	2 b	2 2	8 8	24 24	64 64	384 384	2048 2048
	3 b	3 4	30 32	198 207	1212 1216	33606 33624	762360 762368
	4 b	8 8	128 128	1776 1776	22528 22528	・・・	・・・
	5 b	15 16	510 512	14940 22528	400380 400384	・・・	・・・
	6 b	30 32	2040 2048	123552 400384	・・・	・・・	・・・
	7 b	63 64	8190 8192	・・・	・・・	・・・	・・・

（注１） ｂ＝１，２はＳＥＣ－ＤＥＤ符号に一致する。
（注２） 各欄の上段（分子）は巡回化した時の符号ビット長，下段（分母）は巡回化しない時の符号ビット長。

I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0
M_3	M_5	M_6	M_0	M_0	M_0	M_3	M_3	M_3	M_5	M_5	M_5	M_6
M_0	M_0	M_0	M_3	M_5	M_6	M_3	M_5	M_6	M_3	M_5	M_6	M_3

I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	M_0	M_0	M_0	M_3
M_6	M_6	M_7	I_0	I_1	I_2	I_0	I_0	M_0	I_0	I_0	I_0	I_0
M_5	M_6	M_7	M_7	M_7	M_7	I_1	I_2	M_0	M_3	M_5	M_6	M_0

M_5	M_6	M_3	M_5	M_6	M_3	M_5	M_6	M_3	M_5	M_6	M_7	M_7
I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0
M_0	M_0	M_3	M_3	M_3	M_5	M_5	M_5	M_6	M_6	M_6	M_7	I_0

M_7	M_7	I_1	I_2	M_0	M_3	M_5	M_6	M_0	M_0	M_0	M_3	M_3
I_0	I_0	I_0	I_0	I_0	M_0	M_0	M_0	M_3	M_5	M_6	M_3	M_5
I_1	I_2	I_0	I_0	M_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0

M_3	M_5	M_5	M_5	M_6	M_6	M_6	M_7	I_0	I_1	I_2	I_0	I_0
M_6	M_3	M_5	M_6	M_3	M_5	M_6	M_7	M_7	M_7	M_7	I_1	I_2
I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0	I_0

M_0
M_0
I_0

〔b = 3〕

図3-10. 奇数重み列巡回性（198,189）ＳＥＣ－ＤＥＤ－Ｓ３ＥＤ符号のH行列

3.5 最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の構成[45)-48)] ・・（構成Ｃ）

本章では，現在知られているＳＥＣ－ＤＥＤ－ＳｂＥＤ符号として，最長の符号ビット長を持つＳＥＣ－ＤＥＤ－ＳｂＥＤ符号[45)-48)]を提案する。符号構成は，以下の２種類である。

(1)チェックビット長ｒ＝ｂ＋２の巡回性最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号

(2)バイト長ｂ＝４に対する奇数重みＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号

３．５．１　既存のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の問題点

（ｉ）チェックビット長の削減

汎用計算機システムでは，情報ビット長ｋとして６４ビットを用いることが多い。ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号は，Reddy[30]，Fujiwara[31]，著者[40)43)]により構成法が改良されてきた。しかし，バイト長ｂ＝４ビット，情報ビット長ｋ＝６４ビットのＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号のチェックビット長ｒは，上記のどの構成法を用いても，同一情報ビット長のＳＥＣ－ＤＥＤ符号に比べて１ビット増加する。従って，下記の様な観点から，このチェックビット長の削減が望まれる。

（イ）チェックビット長増加により，記憶素子数が増加する。

（ロ）上記の符号長は７３ビットとなる。７３は素数であり，データパス系のレピータビリティを確保しにくい。

（ハ）チェックビット／シンドロームは，９ビットとなり，８ビットを単位として作られる事の多い高集積論理ＬＳＩとの整合性が悪い。

（ii）符号化・復号化回路の最適化

符号化・復号化回路のゲート量は符号のＨ行列の１の数（重み）にほぼ比例する。符号化・復号化回路のゲート数・遅延の最小化の観点から，Ｈ行列の重みが最小で，しかも行方向の重みが平均化した符号が望まれる。また，符号化・復号化回路のＬＳＩ化を容易と

する為には，H行列にモジュラな繰り返し性[35]を持つ事が望ましい。

この様な観点から，著者[45]-[48]，Chen[37]，Varanasi[38]らにより，新しいＳＥＣ－ＤＥＤ－ＳｂＥＤ符号が提案された。本章で述べる符号は，ほぼ同時に提案されたChen, VaranasiによるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号に比べ，以下の点ですぐれている。

(1)実用的な諸元である，チェックビット長ｒ＝ｂ＋２またはバイト長ｂ＝４において，最長の符号ビット長をもつ。

(2)チェックビット長ｒ＝ｂ＋２において，巡回性最小重み符号である。従って，最小の金物量で高速な符号化・復号化が達成でき，かつ，符号化・復号化回路のＬＳＩ化に適する。

３．５．２　最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の構成法

本節では最初に，チェックビット長ｒ＝ｂ＋２に対するＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の新しい構成法（定理3-6，定理3-7）を示す。符号の構成法はｂが奇数，偶数で異なるが，符号ビット長はいずれもｂ（ｂ＋２）ビットであり，符号化復号化回路・遅延が最小となる最小重み符号である。また，本符号は，H行列にモジュラな構成を有し，符号化・復号化回路のＬＳＩ化にも適している。

次に，このチェックビット長ｒ＝ｂ＋２の符号をもとにして，任意のチェックビット長を持つＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を構成する（定理3-8）。また，ｂ＝４に対して，更に符号長を延ばすことのできる別の構成法（定理3-9）を示す。

チェックビット長ｒ＝ｂ＋２の場合

巡回性符号の定義（定義3-3）に注目し，本節では，巡回オペレーターとして，１ビットの巡回置換を実行するものを考える。従って，巡回オペレーターＲは以下の様になる。

〔定義3-6 〕１ビット単位の巡回オペレーターＲ

巡回性符号の定義3-3 において，１ビットずつの巡回置換を行うときの，巡回オペレーターＲは次式で与えられる。

$$
R = \left[\begin{array}{cccccccc|c}
 & & & & & & & & 1 \\ \hline
1 & & & & & & & & \\
 & 1 & & & & & & & \\
 & & 1 & & & & & & \\
 & & & 1 & & & & & \\
 & & & & 1 & & & & \\
 & & & & & 1 & & & \\
 & & & & & & \ddots & & \\
 & & & & & & & 1 & \\
 & & & & & & & & 1
\end{array}\right] \quad \cdots\cdots\cdots (3\text{-}26)
$$

(注) 行列中の空白は「０」

更に，最小重み符号の定義を示す。

〔定義3-7 〕最小重み符号

符号のＨ行列の重みが，与えられた機能を持つ符号として最小である時，この符号を最小重み符号とよぶ。

ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の場合には，Varanasiにより，最小のチェックビット長がｂ＋２であることが証明されている(38)。従って，たとえば，チェックビット長がｂ＋２で，しかもそのＨ行列の重みが同等の符号ビット長，チェックビット長を持つ最小重みＳＥＣ－ＤＥＤ符号と同一であれば，そのＳＥＣ－ＤＥＤ－ＳｂＥＤ符号は，最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号である。

〔定理3-6 〕最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号（ｂが奇数の時）

ｂが３以上の奇数の時，巡回性符号の生成行列Ｈ。を式（3-27）に示す（ｂ＋２）行ｂ列２進行列とする。Ｈ。，及びこのＨ。を１ビットずつ列方向に巡回置換した行列から作成される式（3-28）の（ｂ＋２）行（ｂ＋２）ｂ列２進行列によりそのＨ行列が与えられる符号はＳＥＣ－ＤＥＤ－ＳｂＥＤ符号である。

（証明は付録3-1 に示す。）

$$H_0 = \begin{bmatrix} & & & & & & & & \kappa & \lambda \\ 1 & 1 & 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & & 0 & 0 & 0 \\ \cdots & & & & & & & \cdots & & \\ \cdots & & & & & & & \cdots & & \\ 0 & 0 & 0 & 0 & 0 & 0 & & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (3\text{-}27)$$

$$H = [H_0,\ RH_0,\ R^2 \cdot H_0,\ ,\ ,\ ,\ R^{b+1} \cdot H_0] \quad \cdots\cdots\cdots\cdots (3\text{-}28)$$

ただし，上式のκ，λは列を指定するために付加した記号であり，行列の構成要素ではない。また，巡回オペレーターＲは式（3-26）に示したものである。

ここで，ｂ＝３及び５について，Ｈ。をしめしておく。

$$H_0 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad H_0 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

バイト長ｂ＝３，チェックビット長ｒ＝５のＳＥＣ－ＤＥＤ－Ｓ３ＥＤ符号のＨ行列を図3-11に示す。本符号は同一のｂ，ｒを持つBossen,Reddy,Fujiwara らによる既存の符号に比べて，約１．３倍の符号ビット長を持つ。また、ｂ＝５，チェックビット長ｒ＝７のＳＥＣ－ＤＥＤ－Ｓ５ＥＤ符号のＨ行列を図3-12に示す。

$$H = \left[\begin{array}{ccccc|ccccc|ccccc|ccccc|ccccc} 1&1&1&0&1&0&0&1&0&0&1&0&0&1&0&0&0&1&0&0 \\ 0&1&0&1&1&1&0&1&0&0&1&0&0&1&0&0&1&1&1&0 \\ 1&0&0&0&1&0&1&1&1&0&0&1&0&1&0&0&1&0&0&0 \\ 1&0&0&1&0&0&0&1&0&0&1&1&1&0&1&0&0&1&0&0 \\ 0&1&0&1&0&0&1&0&0&0&1&0&1&0&1&0&1&1&1&1 \end{array}\right]$$

H_0 H_1 H_2 H_3 H_4

〔b＝3〕

図3-11　最小重み巡回性（15,12）ＳＥＣ－ＤＥＤ－Ｓ３ＥＤ符号のH行列

$$\left[\begin{array}{ccccc|ccccc|ccccc|ccccc|ccccc} 1&1&1&1&1&0&0&0&1&0&0&0&1&1&0&1&1&0&0&0 \\ 0&0&0&0&1&1&1&1&1&1&0&0&0&1&0&0&0&1&1&0 \\ 1&0&0&0&0&0&0&0&1&0&1&1&1&1&1&0&0&0&1&0 \\ 1&1&0&0&0&1&0&0&1&1&0&0&0&0&1&1&1&1&1&1 \\ 0&1&1&0&0&1&0&0&0&1&1&0&0&0&1&0&0&0&1&0 \\ 0&0&1&1&0&1&1&0&0&0&1&1&0&0&0&1&0&0&0&1 \\ 0&0&0&1&1&0&1&1&0&0&0&1&1&0&0&0&1&0&0&0 \end{array}\right]$$

H_0 H_1 H_2 H_3 H_4

$$\left[\begin{array}{ccccc|ccccc} 1&0&0&0&0&0&0&0&1&0 \\ 1&1&0&0&0&1&0&0&0&0 \\ 0&1&1&0&0&1&1&0&0&0 \\ 0&0&1&1&0&0&1&1&0&0 \\ 0&0&0&1&1&0&0&1&1&0 \\ 1&1&1&1&1&0&0&0&1&1 \\ 0&0&0&0&1&1&1&1&1&0 \end{array}\right]$$

H_5 H_6

〔b＝5〕

（注）行列中の空白は「０」

図3-12　最小重み巡回性（35,28）ＳＥＣ－ＤＥＤ－Ｓ５ＥＤ符号のH行列

〔定理3-7 〕最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号（ｂが偶数の時）

ｂが２以上の偶数の時，巡回性符号の生成行列Ｈ₀を式（3-29）に示す（ｂ＋２）行２ｂ列２進行列とする。Ｈ₀，及びこのＨ₀を２ビットずつ列方向に巡回置換した行列から作成される式（3-30）の（ｂ＋２）行（ｂ＋２）ｂ列２進行列でそのＨ行列が与えられる符号はＳＥＣ－ＤＥＤ－ＳｂＥＤ符号である。

（証明は付録3-1 に示す）

$$H_0 = \left[\begin{array}{cccccccc:cccccccc}
1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&0&1&0&1&0\\
1&1&1&1&1&1&1&1&1&1&1&1&0&1&0&1&0&1&0&1\\
1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0\\
\hdots&&&&&&&&\hdots&&&&&&&&\hdots&&&&&&&&\hdots\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0
\end{array}\right] \quad \cdots\cdots\cdots\cdots\cdots\cdots (3\text{-}29)$$

$$H = [H_0,\ R^{2} H_0,\ R^{4} \cdot H_0,\ ,\ ,\ R^{b} \cdot H_0] \quad \cdots\cdots\cdots\cdots\cdots\cdots (3\text{-}30)$$

ここで，ｂ＝４，８について，Ｈ₀の構成をしめしておく。行列中の空白は「０」である。

$$H_0 = \left[\begin{array}{cccc:cccc}
1&1&1&1&1&&1&\\
1&1&1&1&&1&&1\\
1&&&&&1&&1\\
&1&&&1&&&\\
&&1&&&1&&\\
&&&1&1&&&
\end{array}\right] \qquad H_0 = \left[\begin{array}{cccccccc:cccccccc}
1&1&1&1&1&1&1&1&1&&1&&1&&1&\\
1&1&1&1&1&1&1&1&&1&&1&&1&&1\\
1&&&&&&&&&1&&1&&1&&\\
&1&&&&&&&1&&1&&1&&\\
&&1&&&&&&&1&&&&&\\
&&&1&&&&&1&&&&&&\\
&&&&1&&&&&&1&&&&\\
&&&&&1&&&&1&&&&&\\
&&&&&&1&&&&&&1&&\\
&&&&&&&1&&&&1&&&
\end{array}\right]$$

ｂ＝４　の時　　ｂ＝８の時

図3-13には，バイト長ｂ＝４，チェックビット長ｒ＝６のＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号の構成例を示す。本符号の符号ビット長は，同一のｂ，ｒに対するBossenらによるＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号の１．５倍である。また，図3-14には，ｂ＝８，チェックビット長ｒ＝１０のＳＥＣ－ＤＥＤ－Ｓ８ＥＤ符号のＨ行列を示す。

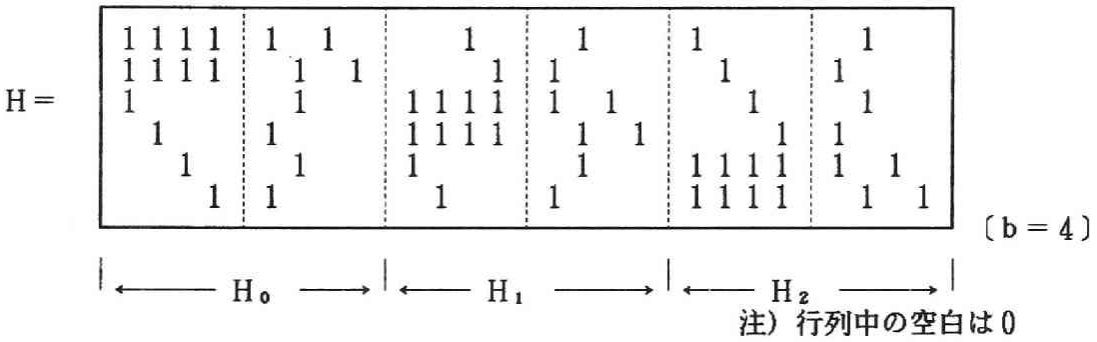
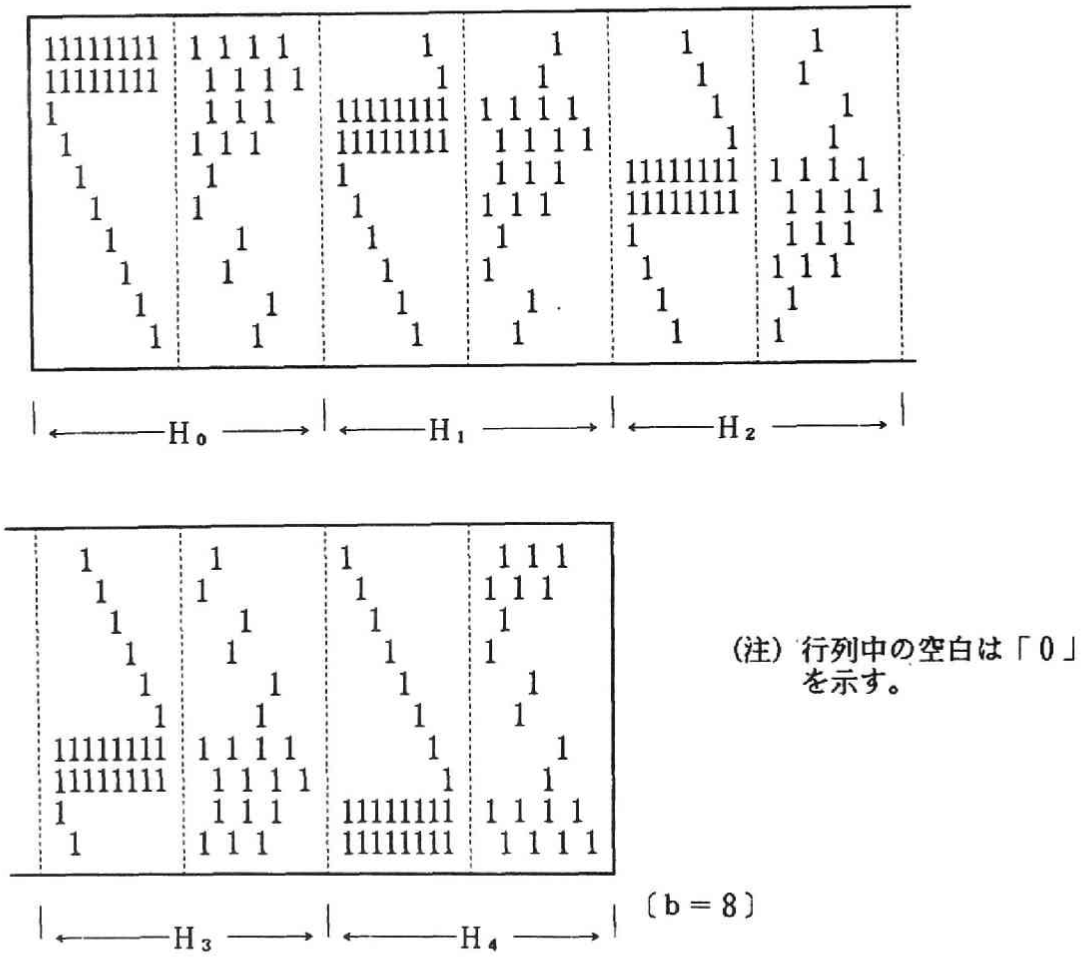


図3-13. 最小重み巡回性（24，18）ＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号のＨ行列



（注）行列中の空白は「0」を示す。

図3 -14　最小重み巡回性（80,70）ＳＥＣ－ＤＥＤ－Ｓ８ＥＤ符号のＨ行列

任意のチェックビット長ｒの場合

次の定理3-8 により，前出の定理3-6 ,3-7の符号から，任意のｒ，ｂに対するＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を構成出来る。

〔定理3-8 〕任意のチェックビット長を持つＳＥＣ－ＤＥＤ－ＳｂＥＤ符号

定理3-6,3-7 で与えられるH行列をK_0とする。更に，ｍを任意の正整数とし，列ベクトルG_i を整数ｉの２進表現（$0 \leq i < 2^m$）とする。次式でそのH行列が与えられる符号はＳＥＣ－ＤＥＤ－ＳｂＥＤ符号であり，チェックビット長はK_0のチェックビット長とｍとの和，符号ビット長は（ｂ＋２）ｂ2^m ビットとなる。

$$H = \begin{bmatrix} K_0, K_0, K_0, K_0, \ldots\ldots, K_0 \\ G_0, G_1, G_2, G_3, \ldots\ldots, G_{2^m-1} \end{bmatrix} \quad \ldots\ldots\ldots (3\text{-}31)$$

証明は省略する。図3-15はｂ＝４，ｒ＝７のＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号の構成例である。

1111	1010	0010	0100	1000	0100	1111	1010	0010	0100	1000	0100
1111	0101	0001	1000	0100	1000	1111	0101	0001	1000	0100	1000
1000	0100	1111	1010	0010	0100	1000	0100	1111	1010	0010	0100
0100	1000	1111	0101	0001	1000	0100	1000	1111	0101	0001	1000
0010	0100	1000	0100	1111	1010	0010	0100	1000	0100	1111	1010
0001	1000	0100	1000	1111	0101	0001	1000	0100	1000	1111	0101
0000	0000	0000	0000	0000	0000	1111	1111	1111	1111	1111	1111

〔ｂ＝４〕

図3-15　(48,41) ＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号のH行列

ｂ＝４でかつｒが偶数の時の他の構成法

次の定理3-9 により，ｂ＝４，ｒが偶数の時に限り，より符号ビット長の長いＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号を構成できる。

〔定理3-9 〕ＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号

チェックビット長ｒを４以上の任意の偶数とする。以下のStepⅠ～Ⅳにより与えられる符号はＳＥＣ－ＤＥＤ－ＳｂＥＤ符号であり，符号ビット長は$2^{r-1}-2^{r/2}$ビットである。（証明は付録3-1 に示す）

（ＳｔｅｐⅠ）ｒ／２ビット要素の列ベクトルｇを１個決める。ｇは任意であるが，最終的に生成される符号のＨ行列の重みを小さくするためには，ｇ＝ａｌｌ‘1’が望ましい。

（ＳｔｅｐⅡ）ｒ／２ビット要素の列ベクトルｆｑを作る。但し，ｇとｆの重みの総和は奇数とする。ｆｑは全部で$2^{r/2-1}$通りある（ｑ＝０，１，２..，$2^{r/2-1}-1$）。

（ＳｔｅｐⅢ）ｆｑ　の中から任意に２個を取り出し，f_i，f_jとする。定義から$0 \leq i < 2^{r/2}$，$0 \leq j < 2^{r/2}$，$i \neq j$である。このf_i，f_jから，ｒ行４列の行列$h_{i,j}$を次の様に作る。ただし，「＋」はＧＦ（２）上のベクトル和を示す。

$$h_{i,j} = \left[\begin{array}{c|c|c|c} g+f_i+f_j & g+f_i+f_j & f_i & f_j \\ f_i & f_j & g+f_i+f_j & g+f_i+f_j \end{array}\right] \quad \cdots\cdots(3\text{-}32)$$

（ＳｔｅｐⅣ）異なる$h_{i,j}$を並べることにより，次式の様にＨ行列を構成する。

$$H=[h_{0,1},\ h_{0,2},\ h_{0,3},\ h_{0,4},\ \ldots\ h_{0,p},\ h_{1,2},\ h_{1,3},\ h_{1,4},\ \ldots\ h_{1,p},\ h_{2,3},\ h_{2,4},\ h_{2,5},\ \ldots\ h_{2,p},\ h_{3,4},\ \ldots\ h_{p-1,p}] \quad \cdots\cdots(3\text{-}33)$$

ただし，$P = 2^{r/2} - 1$とする。

なお，一般のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号では，１ビット誤りと，バイト内３ビット誤りはいずれもシンドロームが奇数重みとなり，弁別が困難となり易い。これに対して，本符号では，付録で述べるように，重み３のバイト誤りのシンドローム中の，上からｒ／２ビット，または下からｒ／２ビットは必ずｇに一致する。従って多入力ＡＮＤゲート２個の簡明な構成で，バイト内３ビット誤りを弁別できる。

上記の定理の構成例を，ｂ＝４，ｒ＝８の場合について示す。ｇとしてａｌｌ‘1’ベクトルを選ぶ。ｆｑ は奇数重みでなければならない。ｆｑ は以下の８通りある。

$$fq = \begin{matrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1, & 0, & 0, & 0, & 0, & 1, & 1, & 1. \end{matrix}$$

このｆｑ から，２列を取り出して$h_{i,j}$を作る。$h_{i,j}$は２８個あり，例えば，$h_{0,1}$は次の様になる。

$$h_{0,1} = \begin{bmatrix} 1&1&0&0 \\ 1&1&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \\ 0&0&1&1 \\ 0&0&1&1 \\ 0&1&0&0 \\ 1&0&0&0 \end{bmatrix}$$

図3-16にはこうして作られたｂ＝４，ｒ＝８のＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号のＨ行列を示す。本符号はBossen,Reddy,Fujiwara による既存の符号の１．７５倍の符号ビット長をもつ。尚，本符号では，現状のＳＥＣ－ＤＥＤ符号と同等のゲート量及び遅延で符号化・復号化できる。

11	11	1	1	1	1	11	11	1	1	1	1
11	1	11	1	1	11	1	1	11	1	1	11
1	11	11	1	11	1	1	1	1	1111	1	1111
1	1	1	1	1111	1111	1111	11	11	11	1	1
11	11	1	1	1	1	11	11	1	1	1	1
11	1	11	1	1	11	1	1	11	1	1	11
1	11	11	1	11	1	1	1	1	1111	1	1111
1	1	1	1	1111	1111	1111	11	11	11	1	111

11	1	1	1	1	11	1111	1111	1111	1	1111	1111
1	1	1111	1111	1	1111	1	1	11	1	1111	1
1111	11	1	11	1	1	1	11	1	1	1	1111
1	11	11	1	1	1	11	1	1	1	1	1
11	1	1	1	1	11	1111	1111	1111	1	1111	1111
1	1	1111	1111	1	1111	1	1	11	1	1111	1
1111	11	1	11	1	1	1	11	1	1	1	1111
1	11	11	1	1	1	11	1	1	1	1	1

1	1111	1	1
1111	1	1111	1
1111	1	1	1111
1	1111	1111	1111
1	1111	1	1
1111	1	1111	1
1111	1	1	1111
1	1111	1111	1111

(注) 行列中の空白は「0」

〔b = 4〕

図3-16　(112,104) ＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号のH行列

図3-16の（１１２，１０４）ＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号は，特定のバイトにチェックビットが集中していない。特定のバイトにチェックビットを集中させるには，H行列の行間での演算を行う必要があり，このようにして，チェックビットを特定のバイトに集中させたＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の例を図3-17に示す。H行列には，２の繰り返し性があり，符号化・復号化回路のＬＳＩ化にも適している。

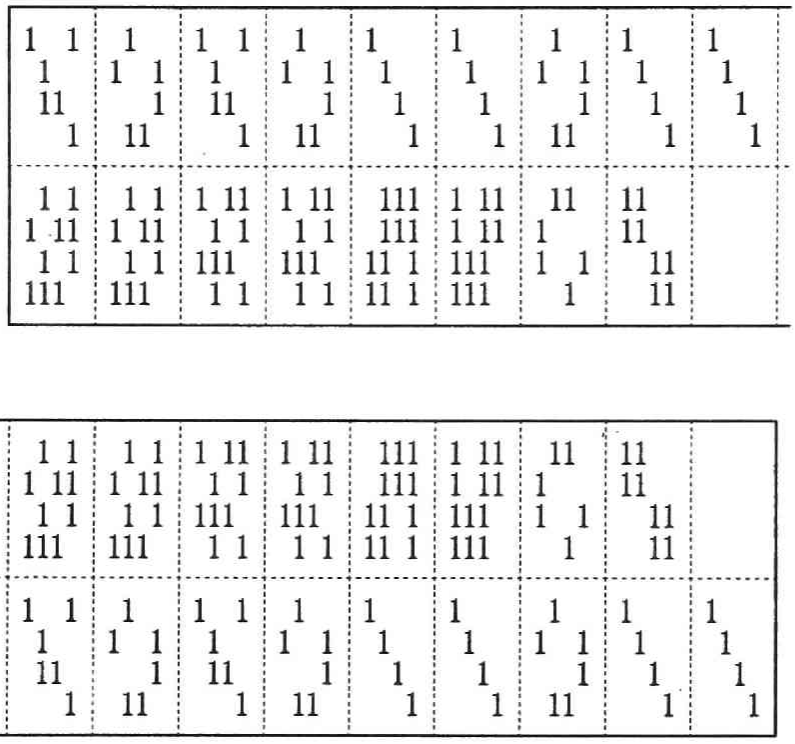


図3-17.　(72,64) ＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号

符号ビット長

本節で示したＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の符号ビット長を表3-8 にしめしておく。次節で述べる様に，本符号は，実用的な緒元である，ｒ＝ｂ＋２または，ｂ＝４で，現在知られているＳＥＣ－ＤＥＤ－ＳｂＥＤ符号のなかで，最長の符号ビット長を持つ。

表3-8　ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の符号ビット長（定理3-6 ～3-9 ）

		b：バイト長（ビット）							
		3	4	6	8	9	１２	１６	３２
r：チェックビット長「ビット」	b＋2	15	24	48	80	99	168	288	1088
	b＋3	30	48	96	160	198	336	576	2196
	b＋4	60	112	192	320	396	672	1152	4392
	b＋5	120	224	384	768	792	1344	2304	8784
	b＋6	240	480	768	1536	1584	2688	4608	17568
	b＋7	480	960	1536	3072	3168	5376	9216	35136
	b＋8	960	1984	3072	6144	6336	10752	18432	70272
	b＋9	1920	3968	6144	12288	12672	21504	36864	140544

3.6 符号ビット長の比較

本節では，本論文で提案したＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の符号ビット長を既存のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号と比較する。以下に，本論文で提案した符号を含めて，現在提案されているＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を列挙しておく。順序は，提案された年代順である。下線を付した符号は，本論文の符号である。

第１グループ（構成はシンプルだが，第２グループにくらべると符号ビット長が小さい）

（符号Ⅰ）Bossen等によるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号[29].....(符号Ⅳの特殊解)

（符号Ⅱ）Reddy によるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号[30]..(一部は符号Ⅳの特殊解)

（符号Ⅲ）FujiwaraによるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号[31].....(符号Ⅳの特殊解)

（符号Ⅳ）著者等によるモジュラな構成を容易に与え得るＳＥＣ－ＤＥＤ－ＳｂＥＤ符号[42]-[43] ・・・構成Ｂ

（符号Ⅴ）著者による単一バースト誤り検出ＳＥＣ－ＤＥＤ符号[39]-[40] ・・構成Ａ

第２グループ（符号化効率の良い符号）

（符号Ⅵ）ChenによるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号[37]

（符号Ⅶ）Varanasi等によるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号[38]

（符号Ⅷ）著者による最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号[45]-[48] ・・・構成Ｃ

ただし，符号Ⅳ，Ⅴ，Ⅷでは，モジュラな性質をＨ行列に与えることができるが，ここではモジュラな構成は考慮しない。

符号ビット長の比較

（ａ）第１グループ

符号Ⅰ，符号Ⅱ（ｂ＜５の時のみ），符号Ⅲは，符号Ⅴに包含される。従って，符号Ⅰ，Ⅱ（ｂ＜５），Ⅲ，Ⅴの符号ビット長は相等しい。ただし，ｂ≧５にたいしては，符号Ⅱは，符号Ⅰ，Ⅲ，Ⅴよりも，大きな符号ビット長を持つ。

一方，符号Ⅳは，チェックビット長ｒが３ｂ未満では，符号Ⅰ，Ⅱ（ｂ＜５），Ⅲ，Ⅴと，ほぼ同等の符号ビット長であり，チェックビット長が３ｂ以上では，符号Ⅰ，Ⅱ（ｂ＜５），Ⅲ，Ⅴよりも大きな符号ビット長をもつ。但し，ｂ≧５では，かなりチェックビット長を大きくしない限り，符号Ⅳの符号ビット長は符号Ⅱの符号ビット長を越えることが出来ない。

（ｂ）第２グループ

これらの符号は，上記の符号Ⅰ～Ⅴにくらべて，大きな符号ビット長を持つ符号であるが，それぞれ，現在知られる最大の符号ビット長を与えるｒ，ｂの範囲が異なる。符号Ⅵは，ｂ＝３，５，６，７，かつｒ＞ｂ＋３では，最長の符号ビット長を与える。一方，符号Ⅶは，ｂ＞７，かつ，ｒ＞ｂ＋２では優れた構成法であるが，ｂの値が小さくなると，効率が悪くなる。著者の符号Ⅷは，ｂ＝４，およびｒ＝ｂ＋２という極めて実用性の高い範囲で，最大の符号ビット長をもっている。

表3-9 には，最大符号ビット長を与える構成法の提案者をまとめておく。ｒ＝ｂ＋２，ｂ＋３かつｂ＜１３では，著者の構成とChenの構成法が同一の符号ビット長を与える。しかし，Chenの符号は，最小重みでも，巡回性符号でもない。従って著者の符号が，より高い実用性を持つ。

一方，符号Ⅳの著者によるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号は，チェックビット長ｒを大きくしてゆくと，その符号ビット長が，同一チェックビット長を持つＳＥＣ－ＤＥＤ符号の符号ビット長に近づく* 。これに対して，表3-9 の金田，Varanasi,Chen の符号の符号ビット長は，（ｂ≧５の範囲では）ｒを大きくしていっても，ある一定値（同一チェックビット長を持つＳＥＣ－ＤＥＤ符号の符号ビット長に，ある一定値を乗じた値）に漸近し，ＳＥＣ－ＤＥＤ符号の符号ビット長には漸近しない。従って，極端にチェックビット長が長い範囲では，符号Ⅳは，現在得られている最大の符号ビット長を与える。このことから，ｂ＞４の範囲では，現在得られている構成法よりも優れた符号が存在すると予想できるが，構成法は未知であり，Varanasi,Chen とも，「Open Problem」として残している。

---

*）ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の符号ビット長の自明な上界は，ＳＥＣ－ＤＥＤ符号の符号ビット長である。チェックビット長が十分長い範囲では，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の符号ビット長の上限は，ＳＥＣ－ＤＥＤ符号の符号ビット長に漸近すると推定される。

表3-9 最大符号ビット長を与える構成法

		b：バイト長（ビット）							
		3	4	6	8	9	12	16	32
r：チェックビット長［ビット］	b＋2	金田 Chen	金田 Chen	金田 Chen	金田 Chen	金田 Chen	金田 Chen	金田	金田
	b＋3	金田 Chen	金田 Chen	金田 Chen	Varana si	Varana si	Varana si	Varana si	Varana si
	b＋4	Chen	金田	Chen	Varana si	Varana si	Varana si	Varana si	Varana si
	b＋5	Chen	金田	Chen	Varana si	Varana si	Varana si	Varana si	Varana si
	b＋6	Chen	金田	Chen	Varana si	Varana si	Varana si	Varana si	Varana si
	b＋7	Chen	金田	Chen	Varana si	Varana si	Varana si	Varana si	Varana si

（注）金田，Chen,Varanasi はそれぞれ符号の提案者を示す。

３．７　結語

本章で示した主な成果は以下の通りである。

（１）単一バースト誤り検出ＳＥＣ－ＤＥＤ符号の構成法（構成Ａ）を提案した。本符号は，従来,Bossen,Reddy,Fujiwaraにより提案されていた１ビット誤り訂正・２ビット誤り検出・単一バイト誤り検出（ＳＥＣ－ＤＥＤ－ＳｂＥＤ）符号を包含し，その符号ビット長は次式であらわされる。ただし，ｒはチェックビット長，ｂはバイト長である。

$$n = b \cdot 2^{r-b}$$

（２）ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の構成法を２種類提案した。各構成法による符号ビット長は以下のとおりである。

（構成Ｂ）奇数重み列ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号。

$$n = \frac{1}{2}\left(2^{bR} + b^{R} - \left(2^{b} - b\right)^{R}\right)$$

ただし，チェックビット長$r = bR$とする。

（構成Ｃ）最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号。

$$n = b\,(b+2)\,2^{r-b-2}$$

ｂ＝４，ｒ＝偶数に対して，さらに優れた構成法があり，その符号ビット長は

$$n = 2^{r-1} - 2^{r/2}$$ となる。

これらの符号のなかで，構成Ｂは次項の巡回性ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を構成するのに適しており，構成Ｃは，実用的な諸元にたいして，現在知られている最大符号ビット長を与える符号である。

（3）符号化・復号化回路のＬＳＩ化に適した，巡回性ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を提案した。符号は，上記の構成Ａ，Ｂ，Ｃの符号をもとにして構成できる。巡回化による符号ビット長の短縮は小さい。

（付録３－１） 定理 （3-6 ） ， （3-7 ） ，（3-9 ） の証明

1． 定理3-6 の証明

式 (3-28) は重み１の列ｒ列と重み３の列のみから構成され，奇数重み列の条件を満たす。また，列ベクトルの線型独立性は，容易に確かめられる。従って，少なくともＳＥＣ－ＤＥＤ機能を持つ。

次に単一バイト誤り検出能力（ＳｂＥＤ機能）について調べる。バイト誤りは，式 (3-28) の各バイトに生じる。しかし，式 (3-28) の対称性からＨ。，即ち式 (3-27) ，のバイト誤りに対する検出機能を証明すれば十分である。単一バイト誤りのパターンをｅ，その重みをｗ（ｅ）とする。

（ｉ）単一バイト誤りｅの重みｗ（ｅ）が偶数（≥ 2）の時。

式 (3-27) のrank＝ｂは容易に証明でき，バイト誤りｅとシンドロームは１対１対応である。ｗ（ｅ）＝０に対するシンドロームは全零であるから，ｗ（ｅ）≥ 2の単一バイト誤りのシンドロームは非零偶数重みとなり検出できる。

（ii）単一バイト誤りｅの重みｗ（ｅ）が奇数（≥ 3）の時。

ｗ（ｅ）≥ 3の奇数重みバイト誤りが検出できない（誤訂正される）のは，シンドロームの重みが３以下の時である。以下の４通りに分けて考える。

〔イ〕式 (3-27) の右端の２列（κ，λ）には誤りの無い時。

式 (3-27) の形から，シンドロームの重み≤ 3となるのは，バイト誤りパターン中の全ての１が互いに隣接している時のみである。しかし，この場合のシンドロームでは，‘1’と‘1’が隣接する事はない。従ってバイト誤りは検出出来る。

〔ロ〕列λに誤りが生じ，列κには誤りの生じていない時。

シンドロームの重みが３となるのは，列λ，κ以外の列に生じた誤りが互いに隣接して

いる時のみである。しかし，この場合のシンドロームでは，‘1’と‘1’が隣接する事はなく，バイト誤りは検出出来る。

〔ハ〕列κには誤りが生じ，列λには誤りの生じていない時。明らかにシンドロームの重みは5以上であり，検出出来る。

〔ニ〕列κと列λに誤りの生じている時。

明らかにシンドロームの重みは5以上であり，検出出来る。尚，式（3-28）の符号の符号ビット長が（ｂ＋2）ｂである事は，容易にわかる。

（Q．E．D．）

2．　定理3-7 の証明

式（3-30）は，重み1の列ベクトルｒ個，及び，重み3の列ベクトルのみから構成され，奇数重み列の条件を満たす。式（3-30）の各列ベクトルの線型独立性は，容易に確かめられる。従って，少なくともＳＥＣ－ＤＥＤ符号である。

単一バイト誤り検出能力（ＳｂＥＤ機能）について調べる。式（3-30）の対称性から，最左端の2バイト，即ち式（3-29）の2バイトについて，検出機能を証明すれば十分である。

<u>式（3-29）左側のバイトに重み2以上のバイト誤りが生じた時</u>

この場合には，バイト誤りの重みが偶数であると，シンドロームの重みはバイト誤りの重みに等しく，検出できる。また，重み3以上の奇数重みバイト誤りに対しては，シンドロームの重みが5以上になり，検出できる。

<u>式（3-29）右側のバイトに重み2以上のバイト誤りｅが生じた時</u>

この場合には，以下の様に分けて考える

（ｉ）バイト誤りｅの重みｗ（ｅ）が偶数（≥ 2）の時。

式（3-29）の右側のバイトについて，rank＝ｂは自明である。従って，バイト誤りｅと

シンドロームは1対1対応である。w（e）＝0に対するシンドロームは全零であるから，バイト誤りの重みw（e）が2以上の単一バイト誤りのシンドロームは非零の偶数重みとなり検出できる。

（ii）単一バイト誤りeの重みw（e）が奇数（≥ 3）の時。

まず，証明を容易にする為，式（3-29）のH$_0$の各列ベクトルを上から2ビットずつ区切って考える。2ビットずつ区切られた列ベクトルのパターンをここでは部分パターンと呼ぶ。部分パターンには，$\binom{0}{0}$，$\binom{0}{1}$，$\binom{1}{0}$，$\binom{1}{1}$の4通りがある。

一例として，b＝8に対するある列ベクトル

$$V = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

は，以下の部分パターンに分解できる。

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

上記の様にして，式（3-29）のH$_0$の各列ベクトルは，以下の様にグループ分けできる。

(typeⅠ) $\binom{0}{1}$が1個，$\binom{1}{1}$が1個，他は$\binom{0}{0}$．

(typeⅡ) $\binom{1}{0}$が1個，$\binom{1}{1}$が1個，他は$\binom{0}{0}$．

以上はH$_0$の左半分の列ベクトル。

(typeⅢ) $\binom{1}{0}$が1個，$\binom{0}{1}$が2個，他は$\binom{0}{0}$．但し，部分パターン$\binom{1}{0}$と部分パターン$\binom{0}{1}$は隣接し，$\binom{1}{0}$が$\binom{0}{1}$の直上にある。

(typeⅣ) $\binom{0}{1}$が1個，$\binom{1}{0}$が2個，他は$\binom{0}{0}$．但し，部分パターン$\binom{1}{0}$と部分パターン$\binom{0}{1}$は隣接し，$\binom{0}{1}$が$\binom{1}{0}$の直上にある。

(typeⅤ) $\binom{1}{0}$が1個，他は$\binom{0}{0}$．

(typeⅥ) $\binom{0}{1}$が1個，他は$\binom{0}{0}$．

以上はH$_0$の右半分の列ベクトル。

式（3-29）のH$_0$を2ビットずつ巡回置換して作られる他の列ベクトルも，typeⅠ～typeⅥのいずれかに属する。ただし，最上端の部分パターンの直上には最下端の部分パターンがあると見なす。従って，訂正可能な1ビット誤りのシンドロームはいずれかのtypeに属する。

一方，式（3-29）の右側のバイトの列ベクトルを，奇数列目と偶数列目に分けて，次式の様に並び換える。

$$M=\begin{array}{c} \kappa \qquad\qquad \lambda \\ \left[\begin{array}{cc:cc} 1\,1\,1\,..\,1\,1 & & 0\,0\,0\,..\,0\,0 & \\ 0\,0\,0\quad 0\,0 & & 1\,1\,1\quad 1\,1 & \\ 0\,0\,0\quad 0\,0 & & 1\,1\,1\quad 1\,0 & \\ 1\,1\,1\quad 1\,0 & & 0\,0\,0\quad 0\,0 & \\ 0\,0\,0\quad 0\,0 & & 1\,0\,0\quad 0\,0 & \\ 1\,0\,0\quad 0\,0 & & 0\,0\,0\quad 0\,0 & \\ 0\,0\,0\quad 0\,0 & & 0\,1\,0\quad 0\,0 & \\ 0\,1\,0\quad 0\,0 & & 0\,0\,0\quad 0\,0 & \\ \cdots\quad\cdots & & \cdots\quad\cdots & \\ 0\,0\,1\quad 0\,0 & & 0\,0\,0\quad 0\,0 & \\ 0\,0\,0\quad 0\,0 & & 0\,0\,0\quad 0\,0 & \\ 0\,0\,0\quad 0\,0 & & 0\,0\,0\quad 0\,0 & \\ 0\,0\,0\quad 0\,0 & & 0\,0\,0\quad 1\,0 & \\ 0\,0\,0\,..\,1\,0 & & 0\,0\,0\,..\,0\,0 & \end{array}\right] \\ \quad M0 \qquad\qquad M1 \end{array}$$

ここで，κ，λは，式（3-29）の右半分のバイトの右端から2列目と1列目の列ベクトルをそれぞれ示している。

上式の左半分，右半分をそれぞれM0，M1と表す。バイト誤りeがこの行列Mに生じた時のシンドロームは，M0の部分から生じた部分シンドロームとM1の部分から生じた部分シンドロームの単なる論理和（OR）に等しい。

奇数重みのバイト誤りを以下の4通りに分ける。

（a）M0は誤りがなく，M1に3以上の奇数個の誤りがある時

シンドロームの重みが3になるのは，誤りの重みが3で，かつ，列λに誤りがある時である。この時，シンドロームは1個の$\binom{0}{1}$と2個の$\binom{1}{0}$を含み，他は$\binom{0}{0}$である。

しかし，（$\binom{0}{1}$）の直下には，（$\binom{0}{0}$）が入り，（$\binom{1}{0}$）ではない。従って，シンドロームは上記の何れのtypeにも属さず，検出できる。

（b）Ｍ１は誤りがなく，Ｍ０に３以上の奇数個の誤りがある時

上記（a）と同様の議論で検出できる。

（c）Ｍ０には２以上の偶数個の誤りがあり，Ｍ１には１以上の奇数個の誤りがある時。

上述した様に，バイト誤りｅが生じた時のシンドロームは，Ｍ０部分から生じた部分シンドロームとＭ１部分から生じた部分シンドロームの単なるＯＲに等しい。従って，シンドロームの重みが３となるのは，Ｍ０から生じるシンドロームの重みが２でＭ１から生じるシンドロームの重みが１の時のみである。　しかるに，Ｍ１から生じるシンドロームの重みが１である為には，Ｍ１の側の誤りは１個で，列λに生じていなければならない。一方，Ｍ０から生じるシンドロームの重みが２であるためには，Ｍ０側には２個の誤りがなければならず，列κと他の１列に誤りの生じている時，及び，列κ以外の２列に誤りの生じている時，である。しかし，このいずれの場合でも，シンドロームは３個の（$\binom{0}{1}$）と（$\binom{0}{0}$）のみから構成され，検出できる。

（d）Ｍ１には２以上の偶数個の誤りがあり，Ｍ０には１以上の奇数個の誤りがある時。

上記（c）と同様の議論で検出できる。

以上の（a）～（d）から，単一バイト誤りは検出できる。式（3-30）の符号ビット長が（ｂ＋２）・ｂであることは容易に証明出来る。

（Q．E．D．）

３．　定理3-9 の証明

本符号は明らかに奇数重み列の条件を満たす。符号の構成上，列ベクトルの線型独立性を示すためには，式（3-32）の左端の列に一致する列ベクトルが他に無いことを示せば十分である。式（3-32）の左端の列に一致する恐れのあるのは，列ベクトルの下ｒ／２ビットに同一のf_iを持つ列である。しかし，同一のf_iを持つ他の列ベクトルの上ｒ／２ビ

ットが，式（3-32）の最左端の列と一致したとすると，　同一のf_i，f_j　から2個の$h_{i,j}$を作ったことになり不合理である。従って，式（3-33）の列ベクトルは線型独立である。

また，式（3-32）中のどの3列を加算しても，加算結果の上r／2ビットか下r／2ビットは必ずgとなる。従ってバイト内3ビット誤りのシンドロームは1ビット誤りとは区別できる。

また，式（3-32）の4列を加算すると，i≠jの条件から，all '0'とはならない。従って，バイト内4ビット誤りは検出できる。

以上から，式（3-33）はSEC－DED－SbED符号である。

また，fqは$2^{r/2-1}$個ある。従って，$h_{i,j}$の個数は$2^{r/2-1}$から2個選ぶ組合せの数に等しい。これから，容易に符号ビット長$2^{r-1}-2^{r/2}$を導くことができる。

（Q．E．D．）

（付録３－２） 計算機実験によるＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の構成

本付録では，発見的手法を用いた，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の構成手法について述べる。ここで述べる手法によれば，最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を任意のｂ，ｒに対して構成できる。ただし，発見的手法であるため，必ずＳＥＣ－ＤＥＤ－ＳｂＥＤ符号が得られる保証はない。

本手法は，最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の定理の発見に，大きく寄与した。

ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の発見的構成法を以下にのべる。

アルゴリズム

以下のステップにより，最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を構成できる。

（ＳｔｅｐⅠ）必要なデータビット長を持つ，最小重み巡回性ＳＥＣ－ＤＥＤ符号を構成する。ただし，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の構成を目的とするから，構成されたＳＥＣ－ＤＥＤ符号の符号ビット長は，ｂの倍数とする。

（ＳｔｅｐⅡ）乱数を用いて，上記ＳｔｅｐⅠで構成されたＳＥＣ－ＤＥＤ符号のＨ行列の列ベクトルの順序を，ランダムにならびかえる。ＳＥＣ－ＤＥＤ機能は変化しない。

（ＳＴＥＰⅢ）上記ＳｔｅｐⅡで得られた符号のＨ行列の列ベクトル位置をｉ，ｊで表すことにする。

Ｈ行列の列ベクトルｉ，ｊを入れ替えた場合にたいして，以下の評価関数を計算する。計算は，ｉとｊは等しくなく，しかも，同一のバイト内にｉ，ｊが含まれない，総てのｉ，ｊの組合せに対して実行する。計算には，計算機シミュレーションを使用する。

$$f(i, j) = \kappa f_0 - f_1$$

ただし，ここで，κは非常に大きい定数であり，f_0，$f_1 << \kappa$とする。また，f_0，f_1は以下の意味を持つものとする。

f_0 ： 単一バイト誤りのなかで，検出できない誤りの数。

f_1 ： バイト内1ビットの誤りと1～bビットのバイト誤りにより生じる，二重バイト誤りのなかで，検出出来ない誤りの数。

（ＳｔｅｐⅣ）上記のｉ，ｊの総ての組合せのなかで，もっとも小さいｆ（ｉ，ｊ）を与えるｉ，ｊに相当する列ベクトルを入れかえる。この時$f_0 = 0$であれば，得られた符号はＳＥＣ－ＤＥＤ－ＳｂＥＤ符号であるから，ＳｔｅｐⅥへ飛ぶ

（ＳｔｅｐⅤ）ＳｔｅｐⅢに飛ぶ。

（ＳｔｅｐⅥ）ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の構成の終了。

この手法は，「山登り法」とＡＩ研究の分野でよばれる手法である。山登り法の欠点は，本来の正しい解（$f_0 = 0$を与える解）が別のところに存在するにもかかわらず，より大きなf_0（$f_0 > 0$）を与える「山」に登り，正しい解に到達しない現象があることである。この，大きなf_0を与える「山」に登ると，上記のアルゴリズムは停止せず，永久にループする。したがって，上記のアルゴリズムがループして，その前のステップで入れ替えた列ベクトルのペア（ｉ，ｊ）をふただび入れ替えてもとにもどす動作が連続する様になれば，ＳｔｅｐⅡに飛び，別の乱数を用いてＨ行列をランダマイズしなおして，処理を始めからやりなおす必要がある。

いうまでもなく，ＳｔｅｐⅠ～Ⅴを何回やっても，アルゴリズムが終了しないことがある。これは，そのチェックビット長では，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号が構成出来ないことを意味している可能性が高い。従って，この場合には，処理を打ち切り，より大きなチェックビット長を持つ最小重み巡回性ＳＥＣ－ＤＥＤ符号を構成し，ＳｔｅｐⅠからやり直す必要がある。

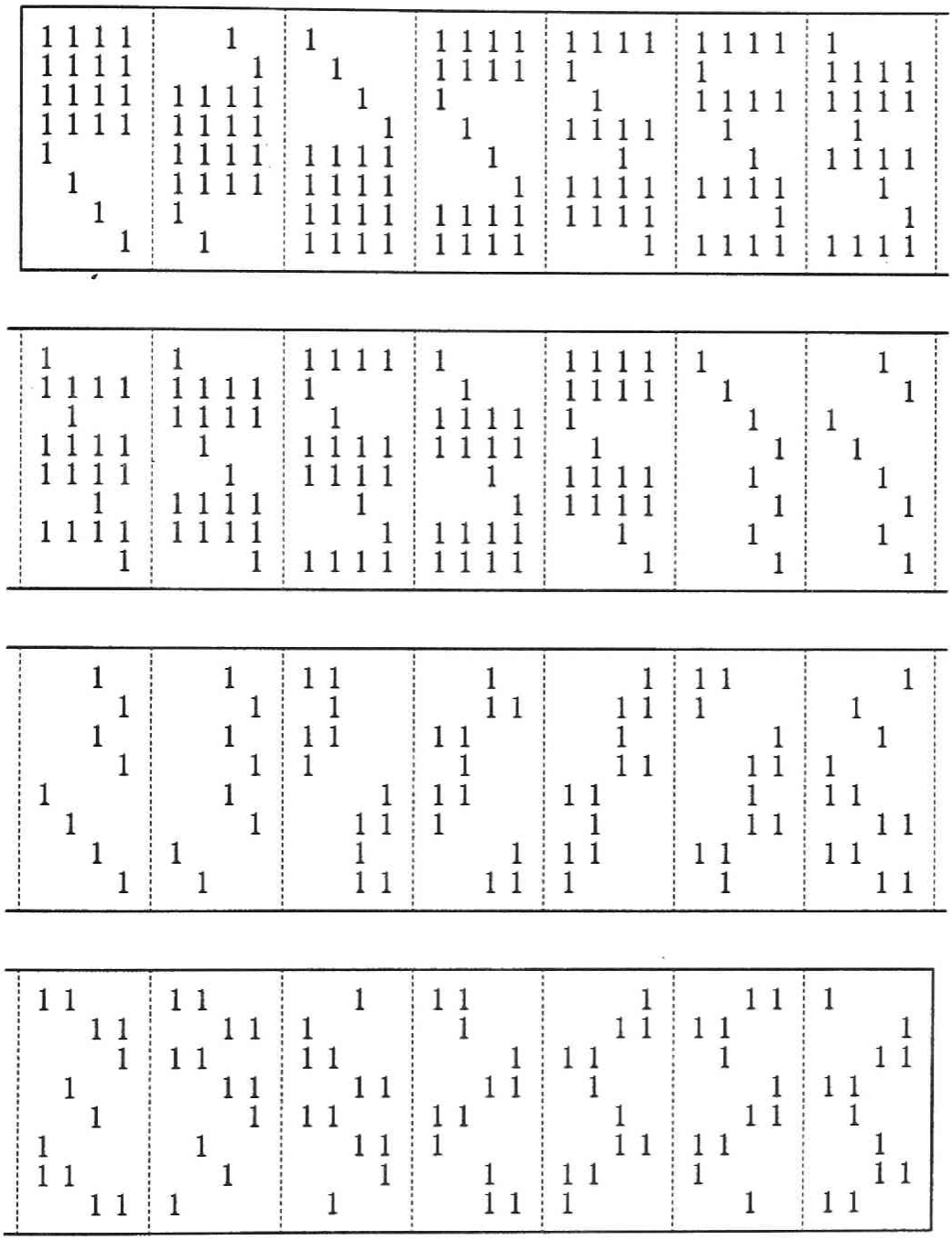
なお，ＳｔｅｐⅣで，同一のｆ（ｉ，ｊ）を与えるペア（ｉ，ｊ）が，複数個存在する場合があり，この時には，なんらかのルールにより，特定の１ペアを選択する必要がある。実験に使用したインプリメンテーションでは，ｉが小さいものを優先し，ｉが同一の時に

は，ｊの小さいものを優先した。しかし，このルールは，符号構成上の意味はなく，単に，インプリメンテーションが容易なために採用したルールである。

具体的な構成例。

このアルゴリズムを使用すると，半導体記憶装置として要求される諸元のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を，現実的な時間で構成できる。

付図3-2-1 には，計算機で構成された，（１１２，１０４）最小重み奇数重み列ＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号のH行列をしめしておく。この符号は，本文の定理3-9 では構成できない。



〔b＝4〕

（注）行列中の空白は「0」を示す。

付図3-2-1　最小重み奇数重み列（112,104）ＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号のH行列

（付録３－３）　グループ化行列Ａの例

BossenのＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を構成するためのグループ化行列は，次式で与えられる。行列中の空白は「０」を示す。

$$A = \left[\begin{array}{c|c|c} 1 & & 1\ 1\ 1\cdots\ 1 \\ \hline & & \begin{array}{l} 1 \\ \ \ 1 \\ \ \ \ 1 \\ \ \ \ \ \cdot \\ \ \ \ \ \ \cdot \\ \ \ \ \ \ \ 1 \\ \ \ \ \ \ \ \ 1 \\ \ \ \ \ \ \ \ \ 1 \end{array} \end{array}\right] \begin{array}{l} \uparrow \\ \\ \\ \\ b \\ \\ \\ \downarrow \end{array}$$

$\leftarrow\ b-1\ \longrightarrow$　　$b \times r$

Reddy のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号のグループ化行列は，次式で与えられる。

$$A = \left[\begin{array}{c|c|cc} & & 1 & 1 \\ \hline 1 & & 1 & 1 \\ 1 & & 1 & 1 \\ \hline 1 & & 1 & 1\ 1 \end{array}\right] \quad : \quad b = 4$$

$4 \times r$

$$A = \left[\begin{array}{c|c|cc} 1 & & & 1 \\ 1 & & 1 & \\ \hline 1 & & 1\ 1 & \end{array}\right] \quad : \quad b = 3$$

$3 \times r$

ただし，ｂ＞４に対するReddy のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号は，ｂ分割パリティチエックできない。

第４章　ＳｂＥＣ－ＤｂＥＤ符号の構成[49]-[52]

4.1　緒言

本章では，単一のバイト誤りを訂正し，かつ２重のバイト誤りを検出する，単一バイト誤り訂正・２重バイト誤り検出符号（ＳｂＥＣ－ＤｂＥＤ符号）の新しい構成法を提案する[49]-[52]。本提案のＳｂＥＣ－ＤｂＥＤ符号は，既存のReed&Solomon [28]による構成法とは異なり，任意の符号ビット長に対して符号を構成できる。更に，符号化・復号化回路のＬＳＩ化に適する巡回性ＳｂＥＣ－ＤｂＥＤ符号を提案する。

4.2　既存のＳｂＥＣ－ＤｂＥＤ符号の問題点

ＳｂＥＣ－ＤｂＥＤ符号は，単一のバイト誤りを訂正し，二重のバイト誤りを検出する符号である。複数ビット出力素子を用いた半導体記憶装置にＳｂＥＣ－ＤｂＥＤ符号を適用すると，単一の素子故障を救済し，二重の素子故障を検出できる。一方，現状の１ビット出力記憶素子とＳＥＣ－ＤＥＤ符号を用いた記憶装置では，単一の素子故障を救済し，かつ，二重の素子故障を検出できる。従って，ＳｂＥＣ－ＤｂＥＤ符号の複数ビット出力記憶素子への適用は，現状の記憶装置仕様の自然な継承と考えることができる。

本節では，既存のＳｂＥＣ－ＤｂＥＤ符号[28][26]の問題点について述べる。

〔問題点１・・・最大符号ビット長が限定されている。〕

Bossen[26]は，半導体記憶装置へＳｂＥＣ－ＤｂＥＤ符号を適用することを提案した。Bossenの符号は，Reed-Solomonによるガロア体ＧＦ（2^b）上のハミング距離ｄ＝４を持つ符号[28]に相当する。この既存のＳｂＥＣ－ＤｂＥＤ符号の最大符号ビット長は，ｂ（2^b-1）に等しい。また，Wolf[53]は，Reed-SolomonのＳｂＥＣ－ＤｂＥＤ符号のH行列に，３個のチェックシンボルが付加できることを示している。本論文では，この３個のチェックシンボルを付加したＳｂＥＣ－ＤｂＥＤ符号をReed-Solomon型のＳｂＥＣ－ＤｂＥＤ符号と呼ぶ。Reed-Solomon型のＳｂＥＣ－ＤｂＥＤ符号の最大符号ビット長は，ｂ・

（2^b ＋2）となる。

Reed-Solomon型のＳｂＥＣ－ＤｂＥＤ符号の具体的なデータビット長は，以下の値となる。

バイト長ｂ＝３　データビット長　　２１ビット
バイト長ｂ＝４　データビット長　　６０ビット
バイト長ｂ＝８　データビット長　２０４０ビット

半導体記憶装置のデータビット長は，６４ビット，１２８ビット等が主流である。従って，上記の値からもわかる様に，Reed-Solomon型のＳｂＥＣ－ＤｂＥＤ符号は，バイト長ｂ＝３，４等の比較的小さなバイト長に対して，不十分な符号ビット長しか持っていない。特に，バイト長ｂ＝４は，実用的な緒元であり，このｂに対して符号を構成出来ない事は，大きな問題である。

〔問題点２・・・符号化・復号化回路のモジュラな構成に適した巡回性符号ではない〕

また，従来のReed-Solomon型のＳｂＥＣ－ＤｂＥＤ符号は，Ｈ行列に繰り返し性を持たないため，その符号化・復号化回路に，ＬＳＩ化に適したモジュラな構成を与えることはできない。

本章では，上記の既存のＳｂＥＣ－ＤｂＥＤ符号の問題点を解決し，任意の符号ビット長に対して構成できるＳｂＥＣ－ＤｂＥＤ符号を提案する。さらに，符号化・復号化回路のＬＳＩ化に適したモジュラな構成を持つＳｂＥＣ－ＤｂＥＤ符号を提案し，符号化・復号化回路の遅延削減，回路量削減の観点から，Ｈ行列の重みを最小化する。

４．３　ＳｂＥＣ－ＤｂＥＤ符号

４．３．１　Reed & Solomon によるＳｂＥＣ－ＤｂＥＤ符号

本節では，既存のReed-Solomon型ＳｂＥＣ－ＤｂＥＤ符号の構成法を詳説する。このReed-Solomon型ＳｂＥＣ－ＤｂＥＤ符号は，本論文で提案する，新しいＳｂＥＣ－ＤｂＥＤ

符号を構成する際に使用する。

ガロア体ＧＦ（2^b）の定義

まず，ガロア体ＧＦ（2^b）を定義する。ｇ（ｘ）をｂ次の原始生成多項式とする。

$$g(x) = \sum_{i=0}^{b} g_i \cdot x^i \quad \cdots\cdots (4\text{-}1)$$

ここで，Σはモジュロ２の加算を表す。多項式ｇ（ｘ）の同伴行列（Companion Matrix）Ｔは以下の正則２進行列で定義できる。

$$T = \begin{bmatrix} 0, & 0, & 0, & \cdots\cdots & 0, & 0, & g_0 \\ 1, & 0, & 0, & \cdots\cdots & 0, & 0, & g_1 \\ 0, & 1, & 0, & \cdots\cdots & 0, & 0, & g_2 \\ \cdot & & & & & \cdot & \\ \cdot & & & & & \cdot & \\ \cdot & & & & & \cdot & \\ \cdot & & & & & \cdot & \\ \cdot & & & & & \cdot & \\ \cdot & & & & & \cdot & \\ \cdot & & & & & \cdot & \\ 0, & 0, & 0, & \cdots\cdots & 1, & 0, & g_{b-2} \\ 0, & 0, & 0, & \cdots\cdots & 0, & 1, & g_{b-1} \end{bmatrix} \quad \cdots\cdots (4\text{-}2)$$

Ｔのべき乗，及び，ｂ行ｂ列の零行列「０」から構成される，以下の集合は，ガロア体ＧＦ（2^b）を形成する。

$$\{0,\ I,\ T^1,\ \ldots\ T^{2^b-2}\}$$

ここで，「０」はｂ行ｂ列の零行列，「Ｉ」はｂ行ｂ列の単位行列である。なお，本論文を通じて，ｂ＝２，４に対する生成多項式は，以下の原始既約多項式を使用する。

ｂ＝２　：　$x^2 + x + 1 = 0$

ｂ＝４　：　$x^4 + x + 1 = 0$

これらの生成多項式に対するＴのべき乗の一覧を付録4-1 に示しておく。

Reed & Solomon はガロア体ＧＦ（2^b）上でハミング距離ｄを持つ符号を提案した。

特に，ｄ＝４の時が，ＳｂＥＣ－ＤｂＥＤ符号であり，更に，このReed & Solomonによる ＳｂＥＣ－ＤｂＥＤ符号には，３バイトのチェックバイトを付加することができる[53]。このチェックバイトを付加されたReed-Solomon型ＳｂＥＣ－ＤｂＥＤ符号のＨ行列を以下に示す。

$$H = \left[\begin{array}{ccccccccc|c} I & I & I & I & \cdot\cdot & I & \cdot\cdot\cdot & I & & I \\ I & T^{1} & T^{2} & T^{3} & \cdot\cdot & T^{i} & \cdot\cdot\cdot & T^{2^{b}-2} & & I \\ I & T^{2} & T^{4} & T^{6} & \cdot\cdot & T^{2i} & \cdot\cdot\cdot & T^{2(2^{b}-2)} & & I \end{array}\right]$$

（空白は０要素を示す）

.....(4-3)

この符号の最大符号ビット長は以下の式で与えられる。

$$n = b(2^{b}+2) \qquad \text{.....(4-4)}$$

このReed-Solomon型のＳｂＥＣ－ＤｂＥＤ符号はチェックビット長が３ｂビットに固定され，任意の符号ビット長をもつＳｂＥＣ－ＤｂＥＤ符号を構成することはできない。

４．３．２　新しいＳｂＥＣ－ＤｂＥＤ符号の構成法[49]-[52]

次に，任意の符号ビット長に対して構成出来る新しいＳｂＥＣ－ＤｂＥＤ符号の構成法を提案する。

まず最初に，ガロア体ＧＦ（2^{b}）上でハミング距離ｄ＝４を持つ符号（ＳｂＥＣ－ＤｂＥＤ符号）を２個選び，そのＨ行列をH_v，H_wとする。２個のＳｂＥＣ－ＤｂＥＤ符号は任意に選ぶことができ，同一の符号であっても良い。さらに，各符号のガロア体ＧＦ（2^{b}）上の符号長を，それぞれ，ｎ（Hv の符号長），ｍ（H_w の符号長）とする。次に，以下の列ベクトルV_i，W_jを考える。列ベクトルV_i，W_jはガロア体ＧＦ（2^{b}）上の列ベクトルである。

V_i

V_iは，Ｈ行列H_vの列ベクトルであり，ｉ＝０，１，２，.....，ｎ－１となる。

W_j

W_j は，以下に述べる変形により，H_w から得られる行列H_w ''の列ベクトルであり，$j=0, 1, 2, \ldots\ldots, m-1$ となる。以下に，H_w ''の変形過程を述べる。

まず，H行列H_w を，そのハミング距離を保ったまま（即ち，ＳｂＥＣ－ＤｂＥＤ符号としての機能を失うことなく）行列H_w ' に変換する。ただし，ここで，H_w ' の最上段の1行（ガロア体（2^b）上で）は，ａｌｌ‘Ｉ’でなければならない。任意のＳｂＥＣ－ＤｂＥＤ機能を持つ符号について，H行列の行間演算のみでａｌｌ‘Ｉ’を生成できるとは言えない。しかし，少なくとも，Reed-Solomon型ＳｂＥＣ－ＤｂＥＤ符号（式4-3）では，この様な変形は容易である。（具体的な変形の過程は，のちほど例示する。）

次に，このａｌｌ‘Ｉ’行をH_w ' から取り去り，結果として得られる行列をH_w ''とする。このH_w ''の列ベクトルがW_j である。

以上の変形過程を図4-1 に示す。

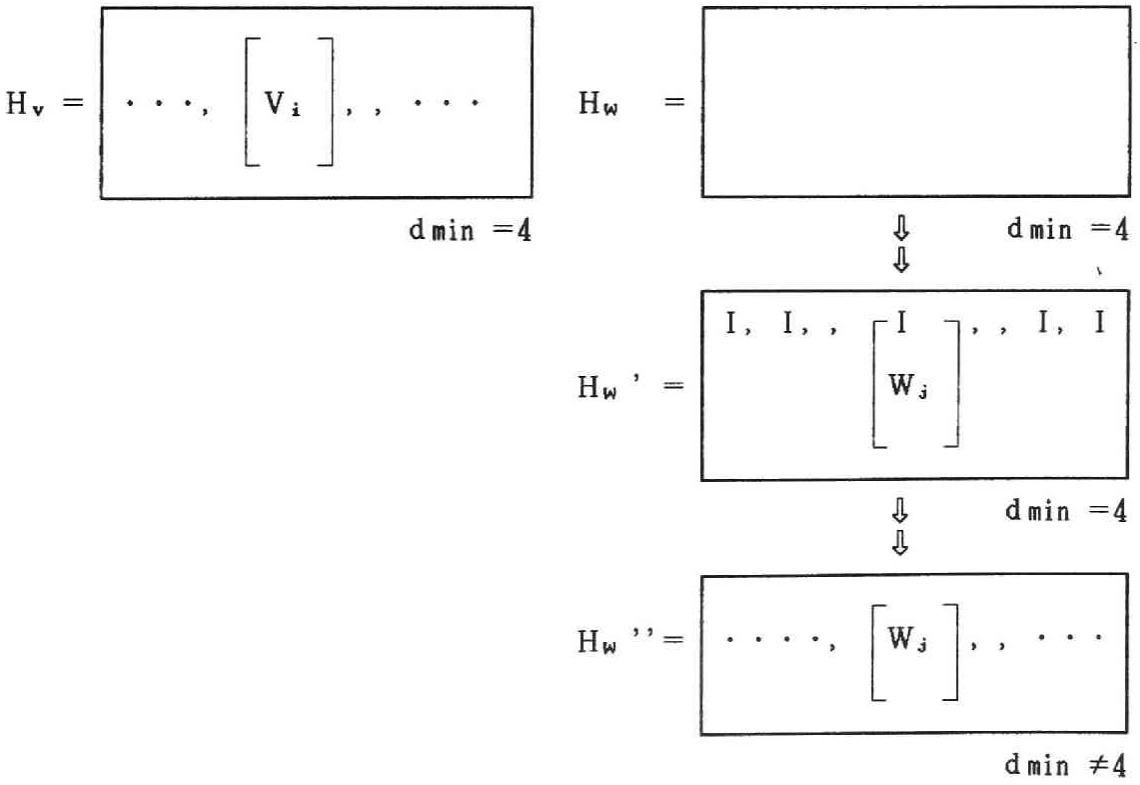


図4-1　H_v, H_w, H_w ', H_w ''

以上の準備のもとに，以下の新しいＳｂＥＣ－ＤｂＥＤ符号の構成法を提案する。

〔定理4-1〕ＳｂＥＣ－ＤｂＥＤ符号の構成法[49] [52]

以下の式（4-5）で定義される列ベクトルを$C_{i,j}$とする。そのH行列が$C_{i,j}$（$i=0, 1, 2, 3, \cdots, n-1$．$j=0, 1, 2, \cdots, m-1$）から構成される符号は，ガロア体（$2^b$）上でハミング距離$d=4$を持ち，ＳｂＥＣ－ＤｂＥＤ符号となる。ただし，同一の$C_{i,j}$を２度以上使用することはないものとする。

$$C_{i,j} = \begin{bmatrix} V_i \\ W_j \end{bmatrix} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (4\text{-}5)$$

（証明）

ハミング距離$d=4$を持つ必要かつ十分な条件は，H行列中から任意の３列以下の列ベクトルを取り出した時に，その取り出された列ベクトルが線形独立であることである。

相異なる任意の３個の列ベクトル（$C_{i0,j0}$，$C_{i1,j1}$，$C_{i2,j2}$）をH行列から取り出す。仮に，$e_0=e_1=e_2=0$が以下の式（4-6）の唯一の解であるならば，列ベクトル式(4-5) から構成される符号はハミング距離$d=4$を持ち，ＳｂＥＣ－ＤｂＥＤ符号となる。ここで，e_0，e_1，e_2はガロア体ＧＦ（2^b）上の３個の誤りパターンである。

$$e_0 \begin{bmatrix} V_{i0} \\ W_{j0} \end{bmatrix} + e_1 \begin{bmatrix} V_{i1} \\ W_{j1} \end{bmatrix} + e_2 \begin{bmatrix} V_{i2} \\ W_{j2} \end{bmatrix} = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \quad \cdots\cdots\cdots (4\text{-}6)$$

上記のＨｖの定義から，$e_0=e_1=e_2=0$が以下の式（4-7）の唯一の解であることは明らかである。

$$e_0 \begin{bmatrix} Vi' \end{bmatrix} + e_1 \begin{bmatrix} Vi'' \end{bmatrix} + e_2 \begin{bmatrix} Vi''' \end{bmatrix} = \begin{bmatrix} 0 \\ \cdot \\ 0 \end{bmatrix} \quad \cdots\cdots\cdots (4\text{-}7)$$

i', i'' , i'''は互いに相異なる。 i', i'' , i'''∈ {0, 1, ..., n－1}

また，Hwの定義から，$e_0 = e_1 = e_2 = 0$が以下の式（4-8）の唯一の解であることは明らかである。

$$e_0 \begin{bmatrix} I \\ Wj' \end{bmatrix} + e_1 \begin{bmatrix} I \\ Wj'' \end{bmatrix} + e_2 \begin{bmatrix} I \\ Wj''' \end{bmatrix} = \begin{bmatrix} 0 \\ \cdot \\ 0 \end{bmatrix} \quad \cdots\cdots\cdots (4\text{-}8)$$

j', j'' , j'''は互いに相異なる。 j', j'' , j'''∈ {0, 1, ..., m－1}

式（4-6）において，以下の3通りの場合を考える必要がある。

1） i0 ＝i1 ＝i2 ＝0の時

式（4-6）から，W_{j0}, W_{j1}, W_{j2} は互いに相異なる。上記，式（4-8）を考慮すれば，$e_0 = e_1 = e_2 = 0$が（4-6）式の唯一の解であることは明らかである。

2） i0 , i1 , i2 中の2個が相等しく，他の1個がそれらとは異なる時。

i0 ＝i1 , i1 ≠i2 としても，その一般性は失われないであろう。この仮定から，式（4-6）において，V_{i1}とV_{i2}は等しくは無い。従って，以下の式が導出される。

$$(e_0 + e_1) \cdot \begin{bmatrix} V_{i1} \end{bmatrix} + e_2 \cdot \begin{bmatrix} V_{i2} \end{bmatrix} = \begin{bmatrix} 0 \\ \cdot \\ 0 \end{bmatrix} \quad \cdots\cdots\cdots\cdots (4\text{-}9)$$

これから，以下の条件を得る。

$$(e_0 + e_1) = 0$$
$$e_2 = 0$$

従って，式（4-6）は以下の様に簡略化できる。

$$e_0 \begin{bmatrix} V_{i0} \\ W_{j0} \end{bmatrix} + e_1 \begin{bmatrix} V_{i1} \\ W_{j1} \end{bmatrix} = \begin{bmatrix} 0 \\ \cdot \\ 0 \end{bmatrix} \quad \cdots\cdots\cdots\cdots\cdots\cdots (4\text{-}10)$$

ここで，上記の左辺の２個の列ベクトルは線形独立である。従って，$e_0 = e_1 = e_2 = 0$が式（4-6）の唯一の解であることは明らかである。

2）i0，i1，i2が互いに相異なる時。

式（4-7）を考慮すれば，$e_0 = e_1 = e_2 = 0$が式（4-6）の唯一の解であることは明らかである。

以上の議論に依って，$e_0 = e_1 = e_2 = 0$が（4-6）式の唯一の解であることは明らかであり，そのH行列のハミング距離は４となる。符号長は，ガロア体ＧＦ（2^b）上で，ｍ×ｎであることは，自明である。　　（Q．E．D．）

上記の定理4-1 から，新しいＳｂＥＣ－ＤｂＥＤ符号のクラスを導くことができる。以下に具体的な構成を示す。まず，H_w を式（4-3）のReed-Solomon型ＳｂＥＣ－ＤｂＥＤ符号であるとする。H_w は，そのH行列の１行がａ１１‘I’となる様に変形を加える必要がある。この変形は以下の様に実行できる。

まず，式（4-3）の２番目の行ベクトルに任意のガロア体ＧＦ（2^b）の元T^a を乗じる。そして，その乗じた結果を式（4-3）の最初の行ベクトルに加算する。もし，この加算結果が零元（「０」）を含まなければ，各列ベクトル毎に，第１行目の要素が単位元（「Ｉ」）となる様に特定の元を乗じる変形が可能となる。第一列目に零元（「０」）を含まない様な元T^a の個数は2^{b-1} であることを容易に導出できる。

具体的に，バイト長ｂ＝２の時について，符号を構成してみよう。H_v，H_w としては，Reed-Solomon型Ｓ２ＥＣ－Ｄ２ＥＤ符号，式（4-12），を選ぶ。

$$H_w = \left[\begin{array}{ccc:ccc} I & I & I & I & 0 & 0 \\ I & T^1 & T^2 & 0 & I & 0 \\ I & T^2 & T^1 & 0 & 0 & I \end{array}\right] \quad \cdots\cdots\cdots\cdots\cdots\cdots (4\text{-}12)$$

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad T^1 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, \quad T^2 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad 0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

式（4-12）は，元T^aとして，T^1を選択することにより，以下の様に変形できる*)。

$$H_w' = \left[\begin{array}{ccc:ccc} I & I & I & I & I & I \\ \hdashline I & T^2 & T^2 & 0 & I & 0 \\ T^2 & I & T^2 & 0 & 0 & I \end{array}\right] \quad \cdots\cdots\cdots\cdots\cdots\cdots (4\text{-}13a)$$

$$H_w'' = \left[\begin{array}{ccc:ccc} I & T^2 & T^2 & 0 & I & 0 \\ T^2 & I & T^2 & 0 & 0 & I \end{array}\right] \quad \cdots\cdots\cdots\cdots\cdots\cdots (4\text{-}13b)$$

式（4-13b ）は，m＝6の符号長を持つH_w''である。

さらに，ここで，H_vとして，式（4-12）に示したReed-Solomon型Ｓ２ＥＣ－Ｄ２ＥＤ符号を使用することにする。定理4-1 に従い，H_vとH_wから，一つのＳ２ＥＣ－Ｄ２ＥＤ符号を構成できる。この符号を図4-2 に示す。図4-2 の符号は，ｎ＝７２ビットの符号ビット長を持ち，ｋ＝６２ビットのデータビット長を有する。

＊）変形過程を具体的に以下に示す。

⑴まず，式（4-12）の第２行に，T^1を乗じる。

$$H_w = \left[\begin{array}{ccc:ccc} I & I & I & I & 0 & 0 \\ T^1 & T^2 & I & 0 & T^1 & 0 \\ I & T^2 & T^1 & 0 & 0 & I \end{array}\right]$$

⑵次に，上式の第２，第３行を第１行に加算する。

$$H_w = \left[\begin{array}{ccc:ccc} T^1 & I & T^1 & I & T^1 & I \\ T^1 & T^2 & I & 0 & T^1 & 0 \\ I & T^2 & T^1 & 0 & 0 & I \end{array}\right]$$

⑶次，各列の第一要素で，各列を割る。

$$H_w = \left[\begin{array}{ccc:ccc} I & I & I & I & I & I \\ I & T^2 & T^2 & 0 & I & 0 \\ T^2 & T^2 & I & 0 & 0 & I \end{array}\right]$$

これから，列ベクトル位置を入れ換えて，上式の式（4-13a)を得る。

I	I	I	I			I	I	I	I		
I	T^1	T^2		I		I	T^1	T^2		I	
I	T^2	T^1			I	I	T^2	T^1			I
I	I	I	I	I	I	T^2	T^2	T^2	T^2	T^2	T^2
T^2	T^2	T^2	T^2	T^2	T^2	I	I	I	I	I	I

I	I	I	I			I	I	I	I		
I	T^1	T^2		I		I	T^1	T^2		I	
I	T^2	T^1			I	I	T^2	T^1			I
T^2	T^2	T^2	T^2	T^2	T^2						
T^2	T^2	T^2	T^2	T^2	T^2						

I	I	I	I			I	I	I	I		
I	T^1	T^2		I		I	T^1	T^2		I	
I	T^2	T^1			I	I	T^2	T^1			I
I	I	I	I	I	I						
						I	I	I	I	I	I

（注）行列中の空白は「0」である。

図4-2　(72,62) S2EC－D2ED符号のH行列

同様にして，そのH行列がガロア体上で7行の行ベクトルを持つＳ２ＥＣ－Ｄ２ＥＤ符号は，２個のＳ２ＥＣ－Ｄ２ＥＤ符号（5行の行ベクトルを持つＳ２ＥＣ－Ｄ２ＥＤ符号と3行の行ベクトルを持つＳ２ＥＣ－Ｄ２ＥＤ符号）から合成できる。一般的に，この様にして，奇数の行ベクトル数，即ち奇数バイト数のチェックバイトを持つＳｂＥＣ－ＤｂＥＤ符号が上記の定理4-1 から構成できる。

偶数個の行ベクトルを持つＳｂＥＣ－ＤｂＥＤ符号は，以下の行列をＨｗ'として使用し，定理4-1 を適用することにより構成できる。

$$Hw' = \begin{bmatrix} I & I \\ 0 & I \end{bmatrix} \qquad \cdots\cdots\cdots\cdots (4\text{-}14)$$

ガロア体上で6行をそのH行列に持つＳ２ＥＣ－Ｄ２ＥＤ符号の構成を図4-3aに示す。この図4-3aの符号では，チェックバイト（列ベクトルの各要素のなかで，１個のみが「Ｉ」であり，他はすべて「０」である列に相当するバイト）の位置が陽には示されていない。しかし，チェックバイトを生成することは容易であり，例えば，図4-3aの符号の上から４，５，６番目の行ベクトルを１番目の行に加算すると，図4-3bの様に，チェックバイトが陽に示された符号を構成できる。

以上の構成法により，定理4-1 から構成されるＳｂＥＣ－ＤｂＥＤ符号の符号ビット長は以下の式で表される。ただし，Ｒはチェックバイト長であり，チェックビット長ｒ＝Ｒｂとなる。

$$n = b\,(2^{b} + 2)^{(R-1)/2} \quad \ldots .\ R : \mathrm{Odd} \quad (\geq 3) \quad \cdots\cdots (4\text{-}15a)$$

$$n = 2b\,(2^{b} + 2)^{(R-2)/2} \quad \ldots\ R : \mathrm{Even} \quad (\geq 4) \quad \cdots\cdots (4\text{-}15b)$$

表4-1 には，上記の符号ビット長を示す。

本論文で述べた構成法により，実用的な諸元を持つＳｂＥＣ－ＤｂＥＤ符号を構成可能となる。ただし，上記の式(4-15a,15b) は限界符号ビット長ではないと思われ，著者によるＳｂＥＣ－ＤｂＥＤ符号の研究の後，理論的興味から，更に長い符号ビット長を持つＳｂＥＣ－ＤｂＥＤ符号の研究[(55)-(56)] が続けられている。

I	I	I	I			I	I	I	I			I	I	I	I		
I	T^1	T^2		I		I	T^1	T^2		I		I	T^1	T^2		I	
I	T^2	T^1			I	I	T^2	T^1			I	I	T^2	T^1			I
						I	I	I	I	I	I						
												I	I	I	I	I	I

I	I	I	I			I	I	I	I			I	I	I	I		
I	T^1	T^2		I		I	T^1	T^2		I		I	T^1	T^2		I	
I	T^2	T^1			I	I	T^2	T^1			I	I	T^2	T^1			I
I	I	I	I	I	I	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2
T^2	T^2	T^2	T^2	T^2	T^2	I	I	I	I	I	I	T^2	T^2	T^2	T^2	T^2	T^2

I	I	I	I			I	I	I	I			I	I	I	I		
I	T^1	T^2		I		I	T^1	T^2		I		I	T^1	T^2		I	
I	T^2	T^1			I	I	T^2	T^1			I	I	T^2	T^1			I
						I	I	I	I	I	I						
												I	I	I	I	I	I
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I

I	I	I	I			I	I	I	I			I	I	I	I		
I	T^1	T^2		I		I	T^1	T^2		I		I	T^1	T^2		I	
I	T^2	T^1			I	I	T^2	T^1			I	I	T^2	T^1			I
I	I	I	I	I	I	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2
T^2	T^2	T^2	T^2	T^2	T^2	I	I	I	I	I	I	T^2	T^2	T^2	T^2	T^2	T^2
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I

(注) 行列中の空白は「0」である。

図4-3a (144,132) S2EC-D2ED符号のH行列

I	I	I	I							I	I					I	I
I	T^1	T^2		I		I	T^1	T^2		I		I	T^1	T^2		I	
I	T^2	T^1			I	I	T^2	T^1			I	I	T^2	T^1			I
						I	I	I	I	I	I						
												I	I	I	I	I	I

T^2	T^2	T^2	T^2	T^1	T^1	T^2	T^2	T^2	T^2	T^1	T^1	I	I	I	I		
I	T^1	T^2		I		I	T^1	T^2		I		I	T^1	T^2		I	
I	T^2	T^1			I	I	T^2	T^1			I	I	T^2	T^1			I
I	I	I	I	I	I	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2
T^2	T^2	T^2	T^2	T^2	T^2	I	I	I	I	I	I	T^2	T^2	T^2	T^2	T^2	T^2

				I	I	I	I	I	I			I	I	I	I		
I	T^1	T^2		I		I	T^1	T^2		I		I	T^1	T^2		I	
I	T^2	T^1			I	I	T^2	T^1			I	I	T^2	T^1			I
						I	I	I	I	I	I						
												I	I	I	I	I	I
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I

T^1	T^1	T^1	T^1	T^2	T^2	T^1	T^1	T^1	T^1	T^2	T^2					I	I
I	T^1	T^2		I		I	T^1	T^2		I		I	T^1	T^2		I	
I	T^2	T^1			I	I	T^2	T^1			I	I	T^2	T^1			I
I	I	I	I	I	I	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2
T^2	T^2	T^2	T^2	T^2	T^2	I	I	I	I	I	I	T^2	T^2	T^2	T^2	T^2	T^2
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I

(注) 行列中の空白は「0」である。

図4-3b (144,132) S2EC-D2ED符号のH行列

表4-1 ＳｂＥＣ－ＤｂＥＤ符号の符号ビット長

		b ：バイト長（ビット）				
		1 ＊＊）	2	3	4	5
R：チェックバイト長	3	4	12	30	72	170
	4	8	24	60	144	340
	5	16	72	300	1296	5780
	6	32	144	600	2592	11560
	7	64	432	3000	23328	196520

＊）チェックビット長　ｒ＝Ｒｂ　で与えられる。
＊＊）ｂ＝１はＳＥＣ－ＤＥＤ符号に等しい。

4．3．3　巡回性ＳｂＥＣ－ＤｂＥＤ符号の構成法[49]-[52]

本節では，符号化・復号化回路のＬＳＩ化に適したモジュラな構成を持つＳｂＥＣ－ＤｂＥＤ符号を提案する。まず，定理4-2 として，２の繰り返し度を持つＳｂＥＣ－ＤｂＥＤ符号を提案する。次に，さらに繰り返し度の大きな符号の構成法を提案する。繰り返し度の高いＳｂＥＣ－ＤｂＥＤ符号の理論的な構成は極めて困難であるため，実用的な緒元を選び，計算機を使用して，符号の生成と，符号化・復号化回路量の削減のための最適化を行う。

繰り返し度２を持つＳｂＥＣ－ＤｂＥＤ符号

その符号化・復号化回路が２個の同一回路から構成されるＳｂＥＣ－ＤｂＥＤ符号を，ここでは，繰り返し度２を持つＳｂＥＣ－ＤｂＥＤ符号と呼ぶ。ここで提案する符号は，必ずしも巡回性ＳｂＥＣ－ＤｂＥＤ符号ではない。

〔定理4-2a〕繰り返し度２のＳｂＥＣ－ＤｂＥＤ符号の構成（１）

以下の式（4-16）でそのＨ行列が与えられる符号は，ＳｂＥＣ－ＤｂＥＤ符号である。

$$
H=\left[\begin{array}{cccccc|ccccc}
I & I & I & \cdots & I & & I & I & I & \cdots & \\
T^{1} & T^{2} & T^{3} & & T^{P} & & T^{-P} & T^{-P+1} & T^{-P+2} & & \\
T^{-1} & T^{-2} & T^{-3} & \cdots & T^{-P} & & T^{P} & T^{P-1} & T^{P-2} & \cdots &
\end{array}\right]
$$

Module 0　　Module 1

$$
\left[\begin{array}{ccc|ccc}
\cdots I & & & I & 0 & 0 \\
\cdots T^{-1} & & & 0 & I & 0 \\
\cdots T^{1} & & & 0 & 0 & I
\end{array}\right] \qquad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (4\text{-}16)
$$

ここで，$P=2^{b-1}-1$ とする。

（証明）式（4-3）に示されたReed-Solomon型ＳｂＥＣ－ＤｂＥＤ符号を用いて，その２番目の列ベクトルの各要素をT^1で除算し，３番目の列ベクトルの各要素をT^2で除算する。以下，同様にして，ｉ番目の列ベクトルの各要素をT^{i-1}で除する。さらに，最初の列ベクトル（ａｌｌ「Ｉ」ベクトル）を取り除く。次に，得られたＨ行列の各行ベクトルを巡回的に，１行ずつ上方に回転する。これにより，上記の式（4-16）を得る。

（Ｑ．Ｅ．Ｄ）

上記のＳｂＥＣ－ＤｂＥＤ符号は，Module 0とModule 1の２つの部分に区分できる。Module 0の上から２行目の行は，Module 1の上から３行目の行に等しく，Module 0の上から３行目の行は，Module 1の上から２行目の行に等しい。従って，Module 0のために作成した回路をそのままModule 1に適用できる。この性質は，符号化・復号化回路のＬＳＩ化に適する。

上記の（4-16）式で，チェックバイト（右端の３列）を除くと，総ての列で，２行目の要素と３行目の要素の積は一定値（この場合は，Ｉ）になっている。この積の値は，Ｉのみでなく，ガロア体ＧＦ（2^b）の任意の元とすることができる。式（4-17）には，ｂ＝４で，かつ，積の値＝T^{14}の場合のＨ行列を示す。

$$H=\left[\begin{array}{cccc|cccc|c}
I & I & I & \cdots I & I & I & \cdots I & I \\
I & T^{1} & T^{2} & \cdots T^{6} & T^{14} & T^{13} & \cdots T^{8} & \\
T^{14} & T^{13} & T^{12} & \cdots T^{8} & I & T^{1} & \cdots T^{6} & I
\end{array}\right] \qquad \cdots\cdots\cdots\cdots (4-17)$$

以上の定理（4-2a）は，そのチェックバイト長が３ｂビットに限定される。次に，任意のチェックバイト長が選択できるＳｂＥＣ－ＤｂＥＤ符号の構成法を示す。

まず，準備として，以下の生成行列Ｈ。を考察する。

H_0 の定義

以下の式（4-18）は，ガロア体ＧＦ（2^b）上で表現されたＳｂＥＣ－ＤｂＥＤ符号のH行列であると定義する。ただし，H_0 の最上部の行は，ａ１１「Ｉ」である。

$$H = \left[\begin{array}{c|c} \begin{bmatrix} (I,\ I,\ ,\ ,\ I) \\ H_0 \end{bmatrix} & \begin{matrix} I & & & \\ & I & & \\ & & I & \\ & & & \cdot \\ & & & & I \end{matrix} \end{array}\right] \quad \cdots\cdots\cdots\cdots\cdots\cdots (4\text{-}18)$$

このH_0を使用して，以下の様に２の繰り返し度を持つＳｂＥＣ－ＤｂＥＤ符号を構成できる。

〔定理4-2b〕繰り返し度２のＳｂＥＣ－ＤｂＥＤ符号の構成（２）

以下の式（4-19）でそのH行列が与えられる符号は，ＳｂＥＣ－ＤｂＥＤ符号である

ただし，$^{\circ}H$ は，H_0 行列の行ベクトルの順序を上下反転した行列である。

$$H = \left[\begin{array}{c|c|c} \begin{bmatrix} \\ H_0 \\ \\ 0,\ 0,\ 0,\ \cdots,\ 0 \end{bmatrix} & \begin{bmatrix} 0,\ 0,\ 0,\ \cdots,\ 0 \\ {}^{\circ}H \end{bmatrix} & \begin{matrix} I\,0\,0\,0\,..\,0\,0 \\ 0\,I\,0\,0\,..\,0\,0 \\ 0\,0\,I\,0\,..\,0\,0 \\ 0\,0\,0\,I\,..\,0\,0 \\ \cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots I\,0 \\ 0\,0\,0\,0\,..\,0\,I \end{matrix} \end{array}\right]$$

← Module 0 → ← Module 1 →

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (4\text{-}19)$$

（証明）式（4-19）に注目する。まず，ＳｂＥＣ機能を証明する。式（4-18）の定義から，Module 0中では，各列ベクトルは線形独立である。同様に，Module 1中では，各列ベクトルは線形独立である。ここで，Module 0の最上部の行はａ１１「Ｉ」であり，最も下部の行はａ１１「０」であることに注意する。Module 1の最上部の行はａ１１「０」であり，最下部の行はａ１１「Ｉ」であるから，シンドロームの最上部及び最下部の要素を調べれば，単一バイト誤りがModule 0，Module 1のどちらに生じたか判別できる。さらに，式（

4-19）の右方のチェックバイトを除くと，Module 0，Module 1には，チェック列（列ベクトル中の１要素のみがＩで，他は０の列）は存在しない。従って，ＳｂＥＣ機能が証明された。

次に，ＤｂＥＤ機能について考える。二重のバイト誤りがいずれもModule 0中に生じた時のＤｂＥＤ機能，いずれもModule 1中に生じた時のＤｂＥＤ機能，および，二重のバイト誤りがいずれもチェックバイトに生じた時のＤｂＥＤ機能は，自明である。さらに，二重バイト誤りの内，一方がチェックバイトに生じ，のこり１バイトの誤りがModule 0またはModule 1に生じた時には，式（4-18）の定義から，ＤｂＥＤ機能は保証される。二重のバイト誤りのなかで，一方がModule 0に生じ，他方がModule 1に生じた時には，シンドロームの最上部及び最下部の要素がいずれも「非零元」となって，二重バイト誤りの発生を検出できる。したがって，ＤｂＥＤ機能が証明された。　　（Q．E．D．）

上記の定理（4-2b）による構成例を以下にしめす。図4-4aは，データビット長６４ビット，バイト長４ビットの構成例である。ここでは，H。として，Reed-Solomon型ＳｂＥＣ－ＤｂＥＤ符号をそのまま使用している。この例は，ｂ＝４の場合であるが，任意のｂについて，同種のＳｂＥＣ－ＤｂＥＤ符号を構成できることは明らかである。さらに，図4-4bには，行ベクトルの順序を入れ替えてしめしておく。

図4-5 は，ｂ＝２，データビット長６４ビットの構成例である。ここでは，H。は，図4-2 の符号をもとにして構成してある。（詳細な構成過程は付録4-2 に示す。）

$$H = \left[\begin{array}{ccccc:ccccc:c} I & I & I & \cdots & I & 0 & 0 & 0 & \cdots & 0 & I & \\ I & T^{1} & T^{2} & \cdots & T^{7} & I & T^{2} & T^{4} & \cdots & T^{14} & & I \\ I & T^{2} & T^{4} & \cdots & T^{14} & I & T^{1} & T^{2} & \cdots & T^{7} & & \\ 0 & 0 & 0 & \cdots & 0 & I & I & I & \cdots & I & & I \end{array}\right]$$

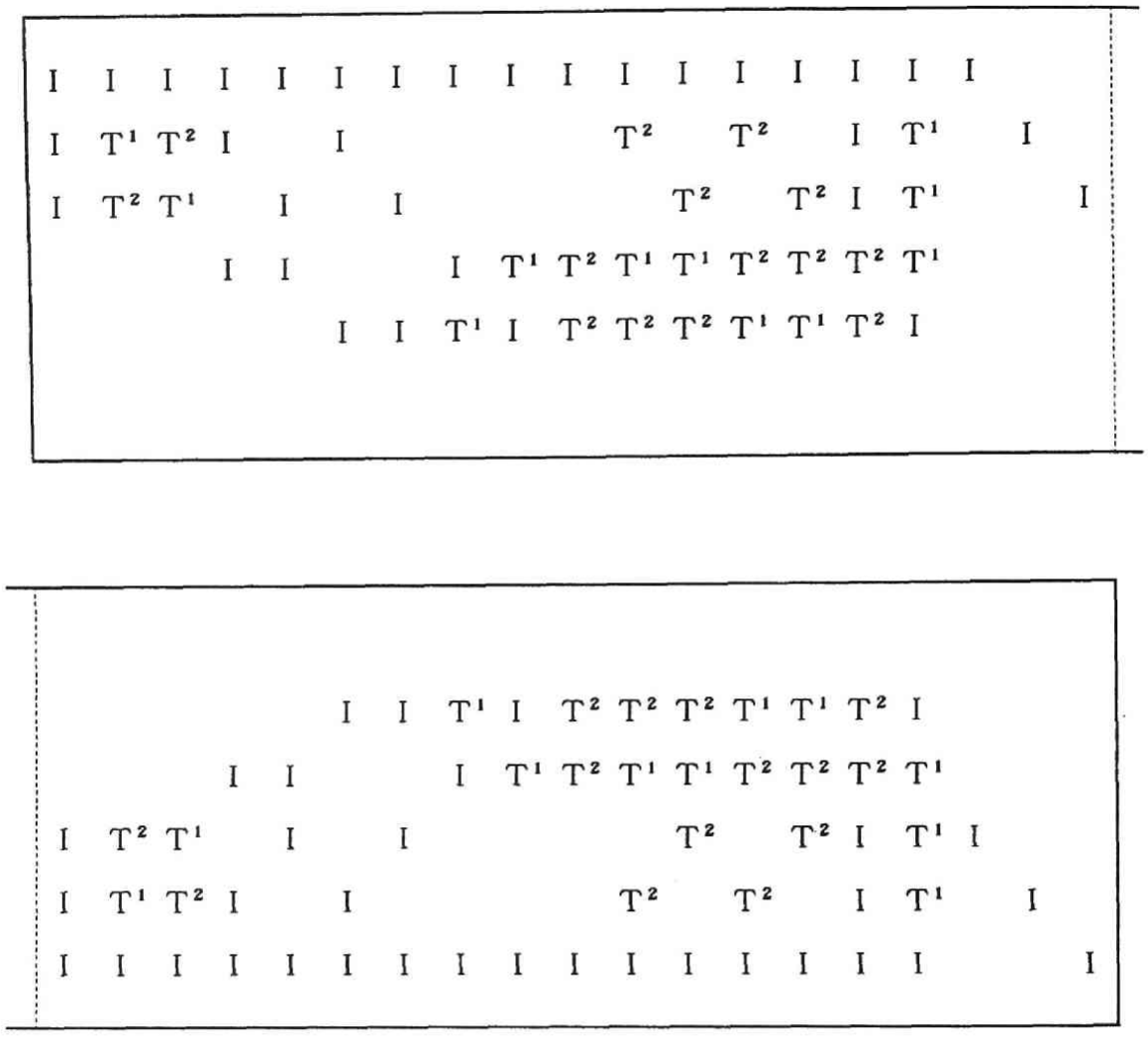
（注）行列中の空白は「0」である。

図4-4a　(80,64）S4EC－D4ED符号のH行列

$$H = \left[\begin{array}{ccccc:ccccc:c} I & I & I & \cdots & I & 0 & 0 & 0 & \cdots & 0 & I & \\ 0 & 0 & 0 & \cdots & 0 & I & I & I & \cdots & I & & I \\ I & T^{1} & T^{2} & \cdots & T^{7} & I & T^{2} & T^{4} & \cdots & T^{14} & & I \\ I & T^{2} & T^{4} & \cdots & T^{14} & I & T^{1} & T^{2} & \cdots & T^{7} & & I \end{array}\right]$$

（注）行列中の空白は「0」である。

図4-4b　(80,64）S4EC－D4ED符号のH行列



（注）行列中の空白は「０」である。

図4-5　(74,64) Ｓ２ＥＣ－Ｄ２ＥＤ符号のＨ行列

繰り返し度4以上を持つ巡回性ＳｂＥＣ－ＤｂＥＤ符号の構成

以上のべたＳｂＥＣ－ＤｂＥＤ符号は，繰り返し度が2に限定される。次に，4以上の繰り返し度を持つ巡回性ＳｂＥＣ－ＤｂＥＤ符号の構成法について考察する。ＳｂＥＣ－ＤｂＥＤ符号は，ガロア体ＧＦ（2^b）上の符号であるから，巡回化を考慮する時は，ｂビット毎の巡回を考える必要がある。従って，巡回性符号のH行列としては，定義（3-3）を使用できる。

巡回性ＳｂＥＣ－ＤｂＥＤ符号の一般的構成法は提案されておらず，構成は非常に困難と考えられてきた。その理由を以下に述べる。

(1)ＳＥＣ－ＤＥＤ符号では，ＳＥＣ符号のH行列にａｌｌ”１”行を付加するだけで構成できる。しかし，ＳｂＥＣ符号のH行列にａｌｌ”Ｉ”行を付加するだけでは，ＳｂＥＣ－ＤｂＥＤ機能を実現できない。

(2)このため，ＳｂＥＣ符号の列ベクトルから，なんらかの条件を満たす列ベクトルを抜き出してＳｂＥＣ－ＤｂＥＤ符号を構成する必要がある。ここで，ＳｂＥＣ符号の列ベクトルから，既に，いくつかの列ベクトルが抜き出され，ＳｂＥＣ－ＤｂＥＤ符号のH行列が部分的に作成されていると仮定する。この状況のもとで，更に，ＳｂＥＣ符号の列ベクトルから，ＳｂＥＣ－ＤｂＥＤ符号のH行列に付加すべき列ベクトルを抜き出し，ＳｂＥＣ－ＤｂＥＤ符号のH行列を大きくすることを考える。明らかに，この時，ＳｂＥＣ符号の列ベクトルから抜き出す列ベクトルは，その列ベクトル単独では判断できず，ＳｂＥＣ－ＤｂＥＤ符号のH行列に存在する列ベクトルに依存する。

(3)更に，巡回性ＳｂＥＣ－ＤｂＥＤ符号を構成する事を考えると，生成部分行列H_0にとりこむべき列ベクトルを決めるときには，生成部分行列H_0に既に存在している列ベクトルとの関係を考慮するだけではなく，その生成部分行列H_0が巡回されて構成される他の部分行列の列ベクトルとの関係を考慮する必要がある。

(4)この様な事情により，任意の巡回性を持ち，しかも冗長度の小さいＳｂＥＣ－ＤｂＥＤ符号を，理論的に構成することは，極めて困難である。

以上の様な理由から，本論文では，生成部分行列H_0については，ＳｂＥＣ－ＤｂＥＤ

機能を理論的に保証し，この生成部分行列H。から巡回化して生成されるH行列のＳｂＥＣ－ＤｂＥＤ機能は，計算機により確認する手法を採用する。特に，符号の緒元としては，実用性が高いバイト長ｂ＝４，情報ビット長ｋ＝６４，１２８を選ぶ。

まず，バイト長ｂ＝４，データビット長ｋ＝１２８ビットの符号を考える。この場合，巡回度４の（１４４，１２８）Ｓ４ＥＣ－Ｄ４ＥＤ符号の生成部分行列Ｈ。は，検査バイトを除いて，８列の列ベクトルを持つ。この８列をReed-Solomon型ＳｂＥＣ－ＤｂＥＤ符号から選択すると，

$$C_{15\ 8} = 6435 \text{ 通り}$$

の組合せを生じ，現実的な処理時間で符号を構成することはできない。従って，生成部分行列Ｈ。を構成するもととなるＳｂＥＣ－ＤｂＥＤ符号として，定理（4-2a）のＳｂＥＣ－ＤｂＥＤ符号を使用し，生成部分行列Ｈ。自体にも２の繰り返し性を与え，場合の数の削減をはかる。

〔巡回性Ｓ４ＥＣ－Ｄ４ＥＤ符号の構成手法〕

以下に巡回性Ｓ４ＥＣ－Ｄ４ＥＤ符号の構成手法を示す。

（Ｓｔｅｐ　Ⅰ）定理（4-2a）のＳ４ＥＣ－Ｄ４ＥＤ符号を構成する。

（Ｓｔｅｐ　Ⅱ）上記のＳ４ＥＣ－Ｄ４ＥＤ符号の（チェックバイトを除く）１４列の列ベクトルから，８列を取り出す。ただし，この８列自体が，２の繰り返し性を持つように，列ベクトルを選択する。

（Ｓｔｅｐ　Ⅲ）上記の８列の列ベクトルから構成される行列の下に，ａ１１「０」行を付加し，４行の行列とし，さらにこれに，１列のチェックバイトを付加して生成部分行列Ｈ。を作成する。

（Ｓｔｅｐ　Ⅳ）上記の生成部分行列Ｈ。から巡回度４の（１４４，１２８）符号を構成し，計算機シミュレーションにより，Ｓ４ＥＣ－Ｄ４ＥＤ機能を検査する。

以上のステップを具体的にしめす。

データビット長１２８ビットの巡回性Ｓ４ＥＣ－Ｄ４ＥＤ符号

まず，定理（4-2a）のＳ４ＥＣ－Ｄ４ＥＤ符号を構成する。ここでは，積の値＝T^{14}の場合を示す。

$$H=\left[\begin{array}{cccc:ccc:c} I & I & I & \cdots I & I & \cdots I & I \\ I & T^{1} & T^{2} & \cdots T^{6} & T^{14} & T^{13}\cdots T^{8} & \quad I \\ T^{14} & T^{13} & T^{12}\cdots T^{8} & & I & T^{1}\cdots T^{6} & \quad I \end{array}\right]$$

次に，このＳ４ＥＣ－Ｄ４ＥＤ符号のチェックバイトを除く部分から，２の繰り返しを保って，８列の列ベクトルをとりだす。取り出し方は，３５通りある。一例として，次の４列を取り出したとする。

$$\begin{array}{cccc:cccc} I & I & I & I & I & I & I & I \\ T^{3} & T^{2} & T^{1} & I & T^{14} & T^{13} & T^{12} & T^{11} \\ T^{11} & T^{12} & T^{13} & T^{14} & I & T^{1} & T^{2} & T^{3} \end{array}$$

この８列から構成される行列に，さらに，ａｌｌ〝０〟行と，チェックバイトを付加して，生成部部分行列H_0構成する。

$$H_0=\left[\begin{array}{cccc:cccc:c} I & I & I & I & I & I & I & I & I \\ T^{3} & T^{2} & T^{1} & I & T^{14} & T^{13} & T^{12} & T^{11} & 0 \\ T^{11} & T^{12} & T^{13} & T^{14} & I & T^{1} & T^{2} & T^{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

このH_0を巡回させて（１４４，１２８）符号を構成する。しかし，ここまでては，H_0内のＳ４ＥＣ－Ｄ４ＥＤ機能，および，符号全体のＳ４ＥＣ機能は保証されていても，符号全体のＳ４ＥＣ－Ｄ４ＥＤ機能は保証されない。従って，計算機シミュレーションにより，Ｓ４ＥＣ－Ｄ４ＥＤ機能を確認する。

上記のH_0から構成された符号は，実際には，巡回性（１４４，１２８）Ｓ４ＥＣ－Ｄ

４ＥＤ符号である。構成された巡回性（１４４，１２８）Ｓ４ＥＣ－Ｄ４ＥＤ符号を図4-6 に示す。Ｈ。を構成するときの，列ベクトルの２番目の要素と３番目の要素との積の値は，１５通りあり，その各々に対して，３５通りのＨ。の構成法がある。これらのなかで，実際にＳ４ＥＣ－Ｄ４ＥＤ符号となる場合の数を表4-2 に示す。図4-6 のＳ４ＥＣ－Ｄ４ＥＤ符号は，表4-2 に示した符号のなかで，ガロア体ＧＦ（２）で考えたＨ行列の重みが最小となる符号である。Ｈ行列の重みは，符号化・復号化回路の回路量・遅延と関係があり，図4-6 の符号は，符号化・復号化回路量・遅延の観点から，最適化された符号である。（なお，上記では，生成部分行列の列ベクトル数を９列としているが，１１列としても，Ｓ４ＥＣ－Ｄ４ＥＤ符号を構成でき，この時の最大符号ビット長は，１７６ビットとなる。）

以上の計算機によるＳｂＥＣ－ＤｂＥＤ符号の構成では，生成部分行列Ｈ。の列ベクトル（チェックバイトを除く）の第２番目の要素と第３番目の要素の積は一定とした。しかし，符号の機能を実現するためには，，かならずしもＨ。の列ベクトルの第２番目の要素と第３番目の積が一定である必要はない。このような観点から，生成部分行列Ｈ。に２の繰り返し性を与え，しかも，生成部分行列Ｈ。の第１行目の要素は，「Ｉ」で，第４行目の要素が「０」となる巡回性（１４４，１２８）Ｓ４ＥＣ－Ｄ４ＥＤ符号を計算機により構成できる。符号のＨ行列は多数得られるが，そのうち，最小の重みを持つ符号を生成する生成部分行列を図4-7 に示す。

図4-7 から構成される（１４４，１２８）ＳｂＥＣ－ＤｂＥＤ符号のＨ行列の重みは５６８であり，図4-6 の符号のＨ行列の重みは５９２である。従って，約４％ほど，図4-7 の符号の方が，符号化・復号化回路量が少ない。

I	I	I	I	I	I	I	I	I
T^{11}	T^{12}	T^{13}	T^{14}	I	T^{1}	T^{2}	T^{3}	0
T^{3}	T^{2}	T^{1}	I	T^{14}	T^{13}	T^{12}	T^{11}	0
0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0
I	I	I	I	I	I	I	I	I
T^{11}	T^{12}	T^{13}	T^{14}	I	T^{1}	T^{2}	T^{3}	0
T^{3}	T^{2}	T^{1}	I	T^{14}	T^{13}	T^{12}	T^{11}	0

T^{3}	T^{2}	T^{1}	I	T^{14}	T^{13}	T^{12}	T^{11}	0
0	0	0	0	0	0	0	0	0
I	I	I	I	I	I	I	I	I
T^{11}	T^{12}	T^{13}	T^{14}	I	T^{1}	T^{2}	T^{3}	0

T^{11}	T^{12}	T^{13}	T^{14}	I	T^{1}	T^{2}	T^{3}	0
T^{3}	T^{2}	T^{1}	I	T^{14}	T^{13}	T^{12}	T^{11}	0
0	0	0	0	0	0	0	0	0
I	I	I	I	I	I	I	I	I

図4-6 巡回性 (144,128) S 4 E C − D 4 E D符号のH行列

表4-2　（１４４，１２８）Ｓ４ＥＣ－Ｄ４ＥＤ符号の個数

積の値	生成個数	積の値	生成個数
I	0	T^{7}	１１
T^{1}	4	T^{8}	4
T^{2}	4	T^{9}	１１
T^{3}	１１	T^{10}	0
T^{4}	4	T^{11}	１１
T^{5}	0	T^{12}	１１
T^{6}	１１	T^{13}	１１
		T^{14}	１１

$$H_0 = \left[\begin{array}{ccccccccc:} I & I & I & I & I & I & I & I & I \\ I & T^{1} & T^{3} & T^{14} & T^{2} & T^{13} & T^{4} & T^{7} & 0 \\ T^{2} & T^{13} & T^{4} & T^{7} & I & T^{1} & T^{3} & T^{14} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$H_0 = \left[\begin{array}{ccccccccc:} I & I & I & I & I & I & I & I & I \\ I & T^{1} & T^{2} & T^{14} & T^{4} & T^{13} & T^{3} & T^{12} & 0 \\ T^{4} & T^{13} & T^{3} & T^{12} & I & T^{1} & T^{2} & T^{14} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$H_0 = \left[\begin{array}{ccccccccc:} I & I & I & I & I & I & I & I & I \\ I & T^{1} & T^{2} & T^{14} & T^{6} & T^{13} & T^{3} & T^{12} & 0 \\ T^{6} & T^{13} & T^{3} & T^{12} & I & T^{1} & T^{2} & T^{14} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$H_0 = \left[\begin{array}{ccccccccc:} I & I & I & I & I & I & I & I & I \\ I & T^{1} & T^{2} & T^{5} & T^{12} & T^{3} & T^{13} & T^{14} & 0 \\ T^{12} & T^{3} & T^{13} & T^{14} & I & T^{1} & T^{2} & T^{5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

図4-7 部分生成行列H_0のなかで，最小重みの行列

データビット長６４ビットの巡回性Ｓ４ＥＣ－Ｄ４ＥＤ符号

データビット長ｋ＝６４ビットの巡回性（８０，６４）Ｓ４ＥＣ－Ｄ４ＥＤ符号の構成結果を図4-8 に示す。この符号は，上記の巡回性（１４４，１２８）Ｓ４ＥＣ－Ｄ４ＥＤ符号と同様の手法で生成された符号であるが，使用しているガロア体ＧＦ（2^b）の元から，最小重み符号であることが容易に判る。

I	I	I	I	I	0	0	0	0	0
T^{14}	I	T^1	T^2	0	I	I	I	I	I
T^2	T^1	I	T^{14}	0	T^{14}	I	T^1	T^2	0
0	0	0	0	0	T^2	T^1	I	T^{14}	0

T^2	T^1	I	T^{14}	0	T^{14}	I	T^1	T^2	0
0	0	0	0	0	T^2	T^1	I	T^{14}	0
I	I	I	I	I	0	0	0	0	0
T^{14}	I	T^1	T^2	0	I	I	I	I	I

図4-8　巡回性 (80,64) Ｓ４ＥＣ－Ｄ４ＥＤ符号のＨ行列

4.4 結語

本章で示した主な成果は以下の通りである。

（1）新しいＳｂＥＣ－ＤｂＥＤ符号の構成法を提案した。本提案の符号は，従来のReed-SolomonのＳｂＥＣ－ＤｂＥＤ符号とは異なり，任意の符号ビット長に対して構成できる。このため，バイト長ｂ＝２，３，４等の小さな値にたいしても，半導体記憶装置用符号として必要な符号緒元を持つＳｂＥＣ－ＤｂＥＤ符号を構成できる。符号ビット長は，以下の式で与えられる。ただし，Ｒはチェックバイト長であり，チェックビット長ｒ＝Ｒｂとなる。

$$n = b\,(2^b + 2)^{(R-1)/2} \quad \ldots R:\mathrm{Odd} \quad (\geq 3)$$

$$n = 2b\,(2^b + 2)^{(R-2)/2} \quad \ldots R:\mathrm{Even} \quad (\geq 4)$$

（2）符号化・復号化回路のＬＳＩ化に適したモジュラな構成を持つＳｂＥＣ－ＤｂＥＤ符号を提案した。符号の構成手法は以下の２通り提案した。

(a) ２の繰り返し度を持つＳｂＥＣ－ＤｂＥＤ符号の理論的構成法

符号ビット長は，ｂ（2^b ＋１）ビットである。

(b) ４以上の繰り返し度を持つＳｂＥＣ－ＤｂＥＤ符号

４以上の繰り返し度を持つＳｂＥＣ－ＤｂＥＤ符号を理論的に構成することは，困難であるため，符号緒元として実用性が最も高いと考えられる，バイト長ｂ＝４ビット，データビット長ｋ＝６４，１２８ビットを選び，計算機を用いて，符号の生成と符号化・復号化回路のゲート量・遅延の最小化を行った。

（付録４－１） 同伴行列Ｔとそのべき乗

本付録では，ｂ＝２とｂ＝４の時について，同伴行列Ｔとそのべき乗をしめす。生成多項式は以下の通りである。

$$b=2 \quad : \quad x^2 + x + 1 = 0$$

$$b=4 \quad : \quad x^4 + x + 1 = 0$$

ｂ＝２の時

$$0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$T^1 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \qquad T^2 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

ｂ＝４の時

$$0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T^1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$T^2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad T^3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \qquad T^4 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$T^5 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad T^6 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \quad T^7 = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

$$T^8 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad T^9 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad T^{10} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

$$T^{11} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad T^{12} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \quad T^{13} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

$$T^{14} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

（付録４－２） 図4-5 の符号の構成

本付録では，図4-5 の(74,64) Ｓ２ＥＣ－Ｄ２ＥＤ符号の構成過程をしめす。まず，もとにする(72,62) Ｓ２ＥＣ－Ｄ２ＥＤ符号の列ベクトルを以下に示す。本符号は，（１２，６）Ｓ２ＥＣ－Ｄ２ＥＤ符号と定理 (4-1)により構成された符号に対して，チェックバイトを構成するための行間演算を加えて構成した。

I	I	I	I	0	0	0	0	0	0	I	I	0	0	0	0	I	I
I	T^1	T^2	0	I	0	I	T^1	T^2	0	I	0	I	T^1	T^2	0	I	0
I	T^2	T^1	0	0	I	I	T^2	T^1	0	0	I	I	T^2	T^1	0	0	I
0	0	0	0	0	0	I	I	I	I	I	I	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	I	I	I	I	I	I

T^2	T^2	T^2	T^2	T^1	T^1	T^2	T^2	T^2	T^2	T^1	T^1	I	I	I	I	0	0
I	T^1	T^2	0	I	0	I	T^1	T^2	0	I	0	I	T^1	T^2	0	I	0
I	T^2	T^1	0	0	I	I	T^2	T^1	0	0	I	I	T^2	T^1	0	0	I
I	I	I	I	I	I	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2	T^2
T^2	T^2	T^2	T^2	T^2	T^2	I	I	I	I	I	I	T^2	T^2	T^2	T^2	T^2	T^2

次に，第１行目の要素（各列ベクトルの最上部の要素）が「０」である列ベクトルを取り除く。結果は，以下の様になる。ただし，チェックバイトは右端に移動した。

I	I	I	I	I	I	I	T^2	T^1	T^1	T^2	T^2	T^2	T^2	T^1	T^1	T^2	T^2	T^2
I	T^1	T^2	I	0	I	0	0	I	0	I	T^1	T^2	0	I	0	I	T^1	T^2
I	T^2	T^1	0	I	0	I	0	0	I	I	T^2	T^1	0	0	I	I	T^2	T^1
0	0	0	I	I	0	0	I	I	I	I	I	I	T^2	T^2	T^2	T^2	T^2	T^2
0	0	0	0	0	I	I	T^2	T^2	T^2	T^2	T^2	T^2	I	I	I	I	I	I

$$
\begin{array}{cccccccc}
I & I & I & I & 0 & 0 & 0 & 0 \\
I & T^1 & T^2 & 0 & I & 0 & 0 & 0 \\
I & T^2 & T^1 & 0 & 0 & I & 0 & 0 \\
I & T^2 & T^1 & 0 & 0 & 0 & I & 0 \\
T^2 & T^2 & T^2 & 0 & 0 & 0 & 0 & I
\end{array}
$$

各列ベクトルの第一要素は，「Ｉ」でなければならない。このため，各列ベクトルの第一要素が「Ｉ」となる様に，各列ベクトルに特定の元を乗算する。

$$
\begin{array}{cccccccccccccccccc}
I & I & I & I & I & I & I & I & I & I & I & I & I & I & I & I & I & I \\
I & T^1 & T^2 & I & 0 & I & 0 & 0 & T^2 & 0 & T^1 & T^2 & I & 0 & T^2 & 0 & T^1 & T^2 & I \\
I & T^2 & T^1 & 0 & I & 0 & I & 0 & 0 & T^2 & T^1 & I & T^2 & 0 & 0 & T^2 & T^1 & I & T^2 \\
0 & 0 & 0 & I & I & 0 & 0 & T^1 & T^2 & T^2 & T^1 & T^1 & T^1 & I & T^1 & T^1 & I & I & I \\
0 & 0 & 0 & 0 & 0 & I & I & I & T^1 & T^1 & I & I & I & T^1 & T^2 & T^2 & T^1 & T^1 & T^1
\end{array}
$$

$$
\begin{array}{ccccccccc}
I & I & I & & I & 0 & 0 & 0 & 0 \\
I & T^1 & T^2 & & 0 & I & 0 & 0 & 0 \\
I & T^2 & T^1 & & 0 & 0 & I & 0 & 0 \\
I & T^2 & T^1 & & 0 & 0 & 0 & I & 0 \\
T^2 & T^2 & T^2 & & 0 & 0 & 0 & 0 & I
\end{array}
$$

上記の列ベクトルから構成される行列の下に，全「０」の行ベクトルを付加すれば，１繰り返し単位の行列として使用できることになる。ただし，ここでは，データビット長64ビットなので，生成された符号の符号化・復号化回路のゲート量の最小化の観点から，（チェックバイトを除いて）列ベクトルを重みの小さいものから，１６列選び，図4-5 の(76,64) Ｓ２ＥＣ－Ｄ２ＥＤ符号を構成した。

第５章 符号化・復号化回路の構成

５．１ 緒言

本章では，バイト誤り検出・訂正符号の符号化・復号化回路の構成法について述べ，バイト誤り検出・訂正符号の符号化・復号化回路が，現状のＳＥＣ－ＤＥＤ符号の場合と，ほぼ同等の遅延・ゲート量で実現できることを明らかにする。

符号化・復号化回路のブロック構成を図5-1 に示す。図5-1 の中で，チェックビット生成回路は，受信した被符号化情報からチェックビットを生成する回路であり，シンドローム生成回路は，記憶素子等から読み出された情報から，シンドロームビットを生成する回路である。チェックビット生成・シンドローム生成は，各ビット（チェックビットまたはシンドロームビット）ごとに，排他的ＯＲ回路ツリーにより，並行して実行する。誤り検出回路は，シンドロームのパターンから，誤りの発生，訂正の可否を判断する回路である。誤り訂正回路は，２入力排他的ＯＲ回路をデータ幅だけ並べた構成を持つ。シンドロームデコード回路は，シンドロームのパターンから，誤りの生じているビット位置を指摘する信号を生成する回路である。シンドロームのデコードには，最も高速に復号化できる，完全並列復号法を使用する必要がある。

以下，⑴単一バースト誤り検出ＳＥＣ－ＤＥＤ符号，⑵ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号，⑶ＳｂＥＣ－ＤｂＥＤ符号について，符号化・復号化回路の構成を述べる。そして，各種の符号について，符号化・復号化のための回路量・遅延を比較する。

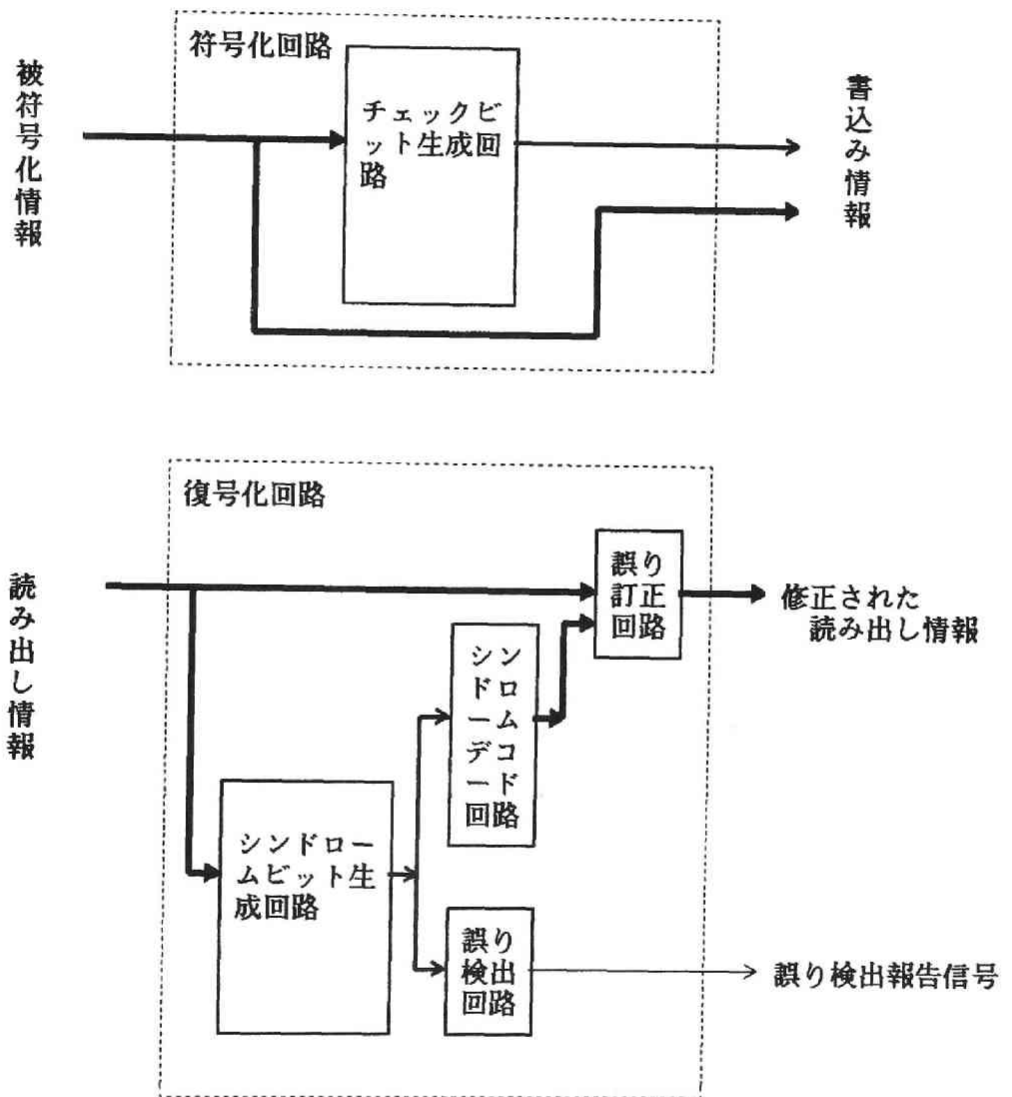


図5-1　符号化・復号化回路のブロック構成

5.2 単一バースト誤り検出ＳＥＣ－ＤＥＤ符号

本節では，単一バースト誤り検出ＳＥＣ－ＤＥＤ符号の符号化・復号化回路の構成について，特徴的な項目を述べる。

5．2．1 誤り検出回路

本符号は，ｂ分割してパリティチェックできるため，ｂ分割パリティチェックした結果の重みが１であれば訂正可能な誤り，２以上であれば訂正できない誤りが検出できる。この性質を使用して，誤り検出回路を容易に構成できる。図5-2 には，データビット長ｋ＝３２ビット，バイト長ｂ＝４ビットの巡回性単一バースト誤り検出ＳＥＣ－ＤＥＤ符号（図3-7)に対する，誤り検出回路を示す。

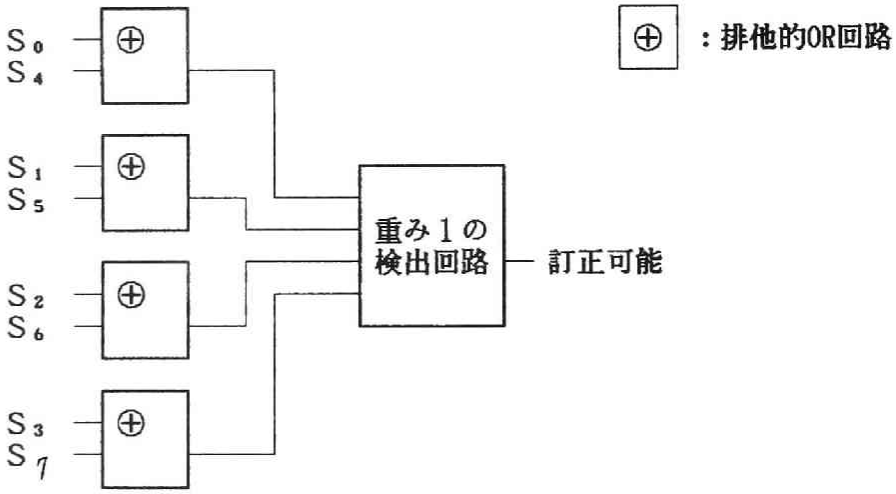


図5-2　単一バースト誤り検出ＳＥＣ－ＤＥＤ符号の誤り検出回路例

5．2．2 チェックビット生成回路・シンドローム生成回路

これらの回路は，符号のＨ行列から直ちに構成できる。図3-7 の単一バースト誤り検出ＳＥＣ－ＤＥＤ符号のＨ行列と，対応するチェックビット生成回路，シンドローム生成回路（部分）を図5-3 に示しておく。ただし，「‘」は誤りを含む可能性のある読み出し情報を示す。

	dddd 0123	dddd 4567	dddd 8911 01	dddd 1111 2345	CCCC 0123	dddd 1111 6789	dddd 2222 0123	dddd 2222 4567	dddd 2233 8901	CCCC 4567	
H =	1 1 11 1111	1 111 111 1	1111 11 1 1	111 1 1 1 1 11	1 1 1 1	 1 111	 11 11 	111 1 	11 11 1 1 11 111	 	S_0 S_1 S_2 S_3
	 1 111	 11 11 	111 1 	11 11 1 1 11 111	 	1 1 11 1111	1 111 111 1	1111 11 1 1	111 1 1 1 1 11	1 1 1 1	S_4 S_5 S_6 S_7

(注) H行列の上部に示した，d12等は，信号名称を示す。

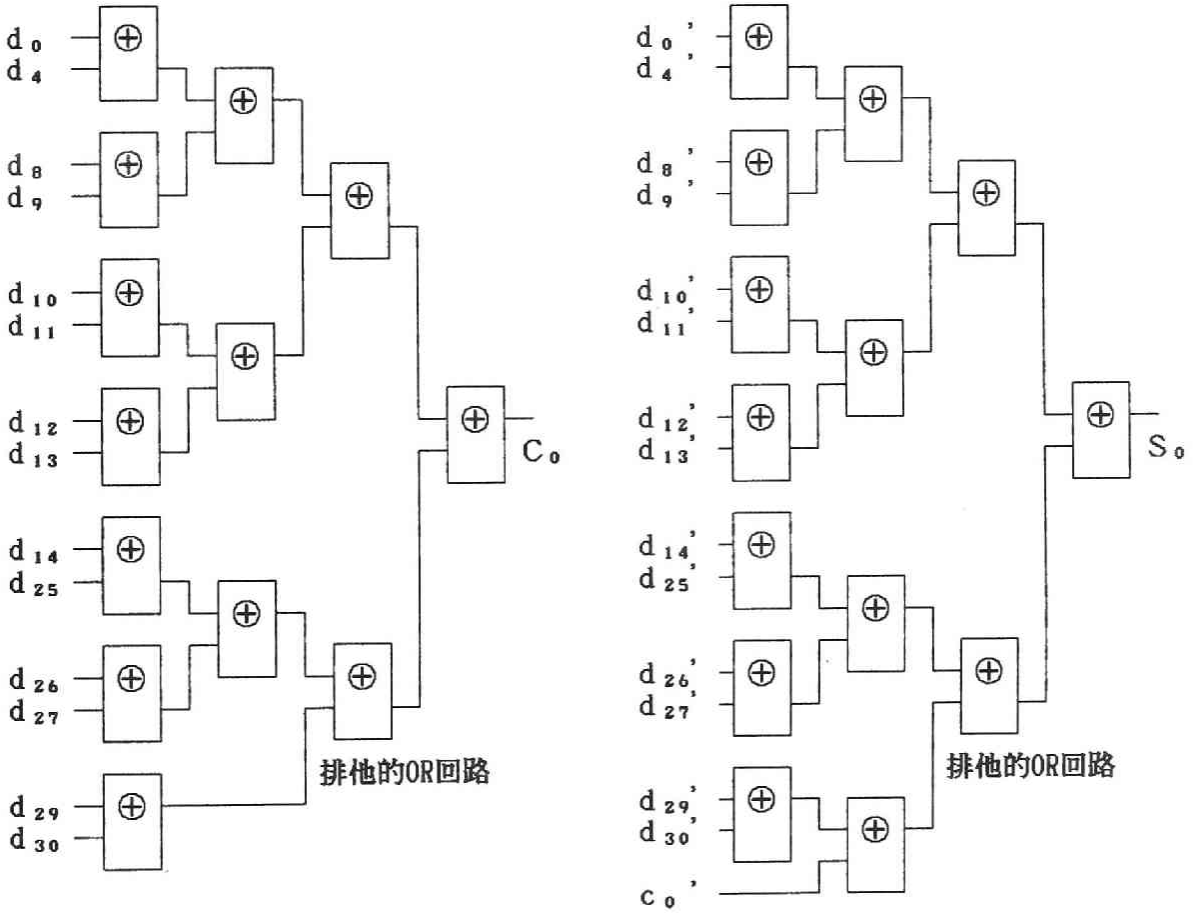
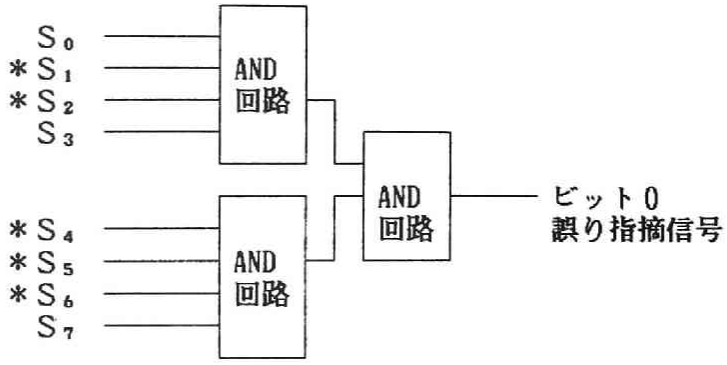


図5-3　チェックビット生成・シンドロームビット生成回路（部分）

5．2．3　シンドロームデコード回路

シンドロームデコード回路は，シンドロームから誤りビット位置を指摘する信号を生成する回路である。従って，当該ビット位置での誤り発生を報告するシンドロームパターンが到着すると「1」を出力し，それ以外の時には「0」を出力するパターンマッチング回路を，各ビットごとに設けておくことにより構成できる。図5-4 には，図3-7 の符号に対

するシンドロームデコード回路（部分）を示しておく。



＊S_i はS_i の否定信号を示す。

図5-4 シンドロームデコード回路（部分）

5．2．4　全体構成

以上の各回路ブロックから構成される，符号化・復号化回路の全体を図5-5 に示す。チェックビット生成回路は同一の単位回路２個から構成され，２個の単位回路の間には，部分的なパリティ和を転送する信号線を必要とする。シンドローム生成回路も同様の構成を持つ。シンドロームデコード回路は，チェックビット生成回路と，同様に２の繰り返し度を持ち，入力となるシンドロームを差し替えることにより，符号全体に対するシンドロームのデコードを可能としている。

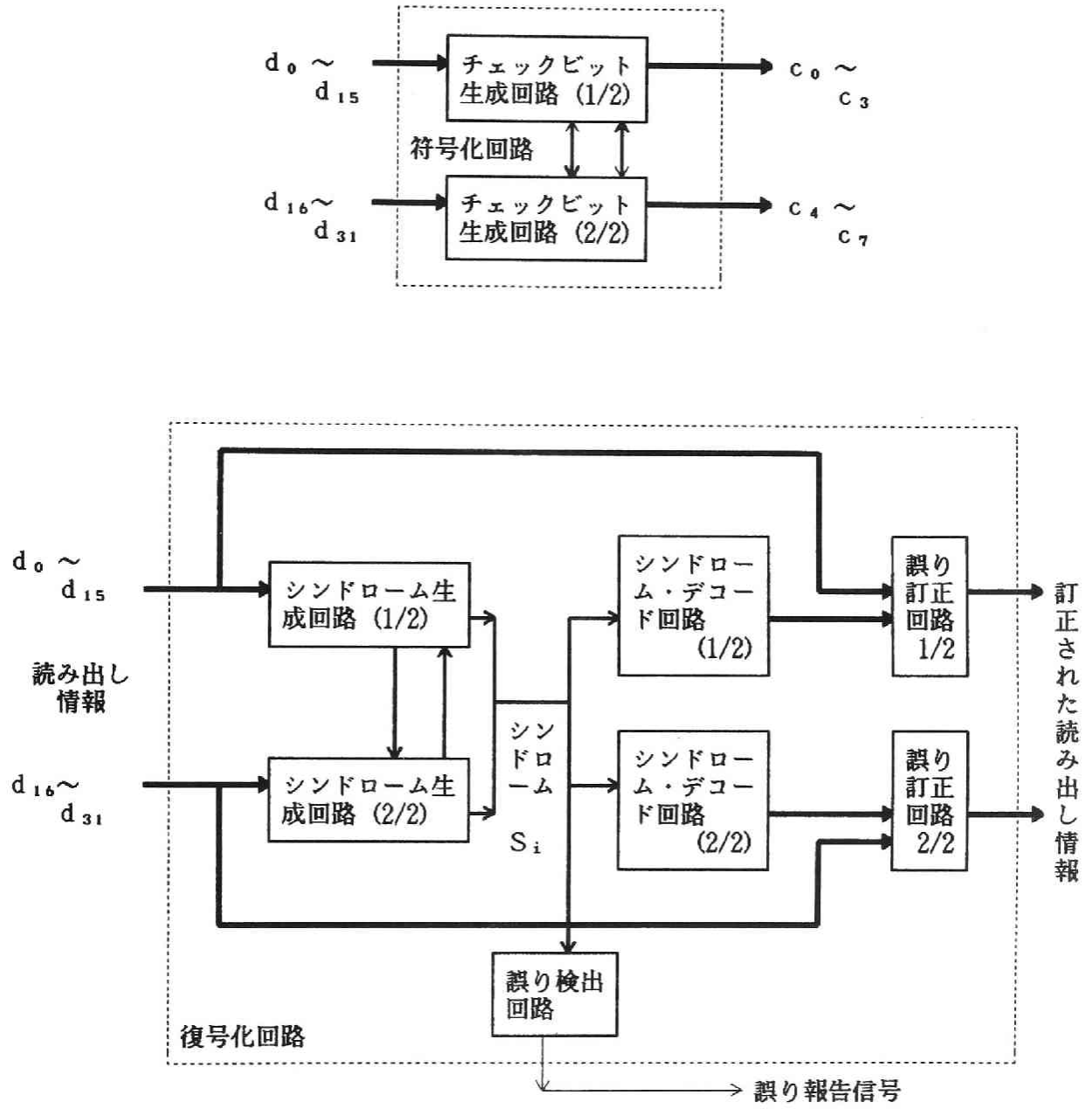


図5-5　くり返し２を持つ符号化・復号化回路の全体構成

5．3　ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の場合

ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の符号化・復号化回路は，前節の単一バースト誤り検出ＳＥＣ－ＤＥＤ符号と同様にして，構成できる。ただし，誤り検出回路は，符号構成法に依存して，特有の構成を持つ。以下，本論文で提案した各種ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号に対する誤り検出回路について述べる。

5．3．1　新しい奇数重み列ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号

この符号では，「シンドロームが奇数重みで，かつ，シンドロームをｂビット毎のブロックに分割して考えた時に，少なくとも一つのブロックに重み１のパターンが存在する」時に限り，１ビット誤りとして訂正でき，それ以外では，訂正できないシンドロームパターンの検出として扱う必要がある。この性質を利用すると，誤り検出回路を容易に構成できる。

図5-6 には，図3-9 の (64,56)ＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号に対する誤り検出回路の構成列を示す。

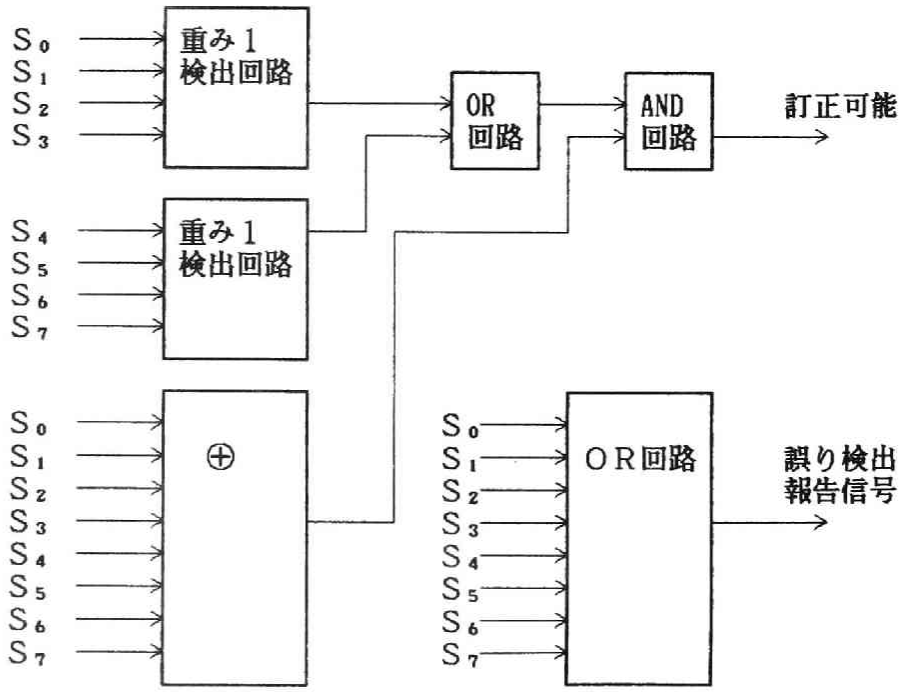


図5-6　誤り検出回路の構成列（そのⅡ）

5．3．2　最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号

符号化・復号化回路の基本的な構成は，前述の奇数重み列ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号と同一である。特に，ｂ＝４の時の構成法では，バイト内の３ビット誤りのシンドロームの上半分または下半分がｇに一致する特徴があり，この性質を使用すると，誤り検出回路を簡明に構成できる。図3-16の符号に対する誤り検出回路の構成を図5-7 に示す。

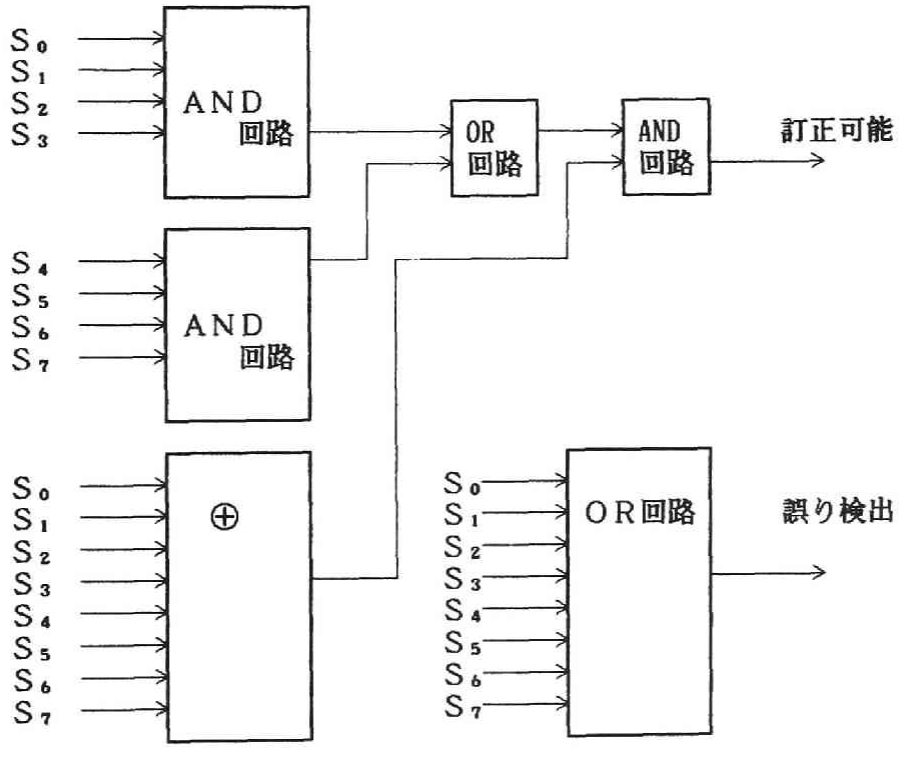


図5-7　最小重みＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号に対する誤り検出回路構成例

5.4 ＳｂＥＣ－ＤｂＥＤ符号

ＳｂＥＣ－ＤｂＥＤ符号においても，パリティチェック回路ツリーにより，チェックビット生成・シンドローム生成を実行できる。一方，シンドロームデコード回路，誤り検出回路の構成は，誤り訂正が１ビットに限定される前述の符号とは，大きく異なる。以下，ＳｂＥＣ－ＤｂＥＤ符号のシンドロームデコード回路，誤り検出回路について，概略を述べる。

5．4．1　シンドロームデコード回路

ＳｂＥＣ－ＤｂＥＤ符号，ＳｂＥＣ符号などの単一バイト誤り訂正符号では，シンドロームから，

(1)誤りの生じているバイト位置，

(2)バイト誤りパターン

を特定する必要がある。以下に示すReed-Solomon型ＳｂＥＣ－ＤｂＥＤ符号の場合を例として，シンドロームから誤りビット位置指摘信号を作成する過程をのべる。

この例では，シンドロームは，S_0，S_1，S_2の３バイトある。誤りバイト位置をｉ，誤りパターンをｅとする時，ｉとｅは，次の様に計算できる。

$$H=\left[\begin{array}{cccccccccc|ccc} I & I & I & \cdots & I & I & I & \cdots & I & I & I & 0 & 0 \\ I & T^{1} & T^{2} & \cdots & T^{i-1} & T^{i} & T^{i+1} & \cdots & T^{2^{b}-3} & T^{2^{b}-2} & 0 & I & 0 \\ I & T^{2} & T^{4} & \cdots & T^{2i-2} & T^{2i} & T^{2i+2} & \cdots & T^{2(2^{b}-3)} & T^{2(2^{b}-2)} & 0 & 0 & I \end{array}\right]$$

<u>誤りがチェックバイト以外にある時</u>

誤りパターン：

$$e=S_0$$

誤りバイト位置：

$$(T^i S_0 = S_1) \quad \text{and} \quad (T^i S_1 = S_2) \text{が成立するi}$$

誤りがチェックバイトにある時

S_0，S_1，S_2 の中で，2個が「0」であり，他の1個は「0」でない時である。この場合には，「0」でないS_i が誤りパターンであり，i＝0，1，2に応じて，チェックバイト位置が決まる。

5．4．2　誤り検出回路

誤りの発生は，シンドロームがａｌｌ‘0’か否かを検出することにより判定できる。しかし，ＳｂＥＣ－ＤｂＥＤ符号では，誤り訂正の可否をシンドロームパターンから直ちに判定することは難しい。従って，シンドロームデコード回路のデコード結果を集計し，どこかの1バイトで訂正が実行されている時に，訂正可能な誤りの検出とする。シンドロームはａｌｌ‘0’ではなく，かつ，誤りバイト位置を特定出来ない時は，二重バイト誤り検出となる。

5．4．3　シンドロームデコード回路構成例

シンドロームデコード回路の例として，図4-8 の巡回性 (80,64)Ｓ４ＥＣ－Ｄ４ＥＤ符号に対する，シンドロームデコード回路を図5-8 に示す。この図5-8 のLSI を4個使用することにより，シンドロームデコード回路全体を構成できる。チェックビット生成回路を例として，4個のＬＳＩから構成される回路の全体を図5-9 に示す。

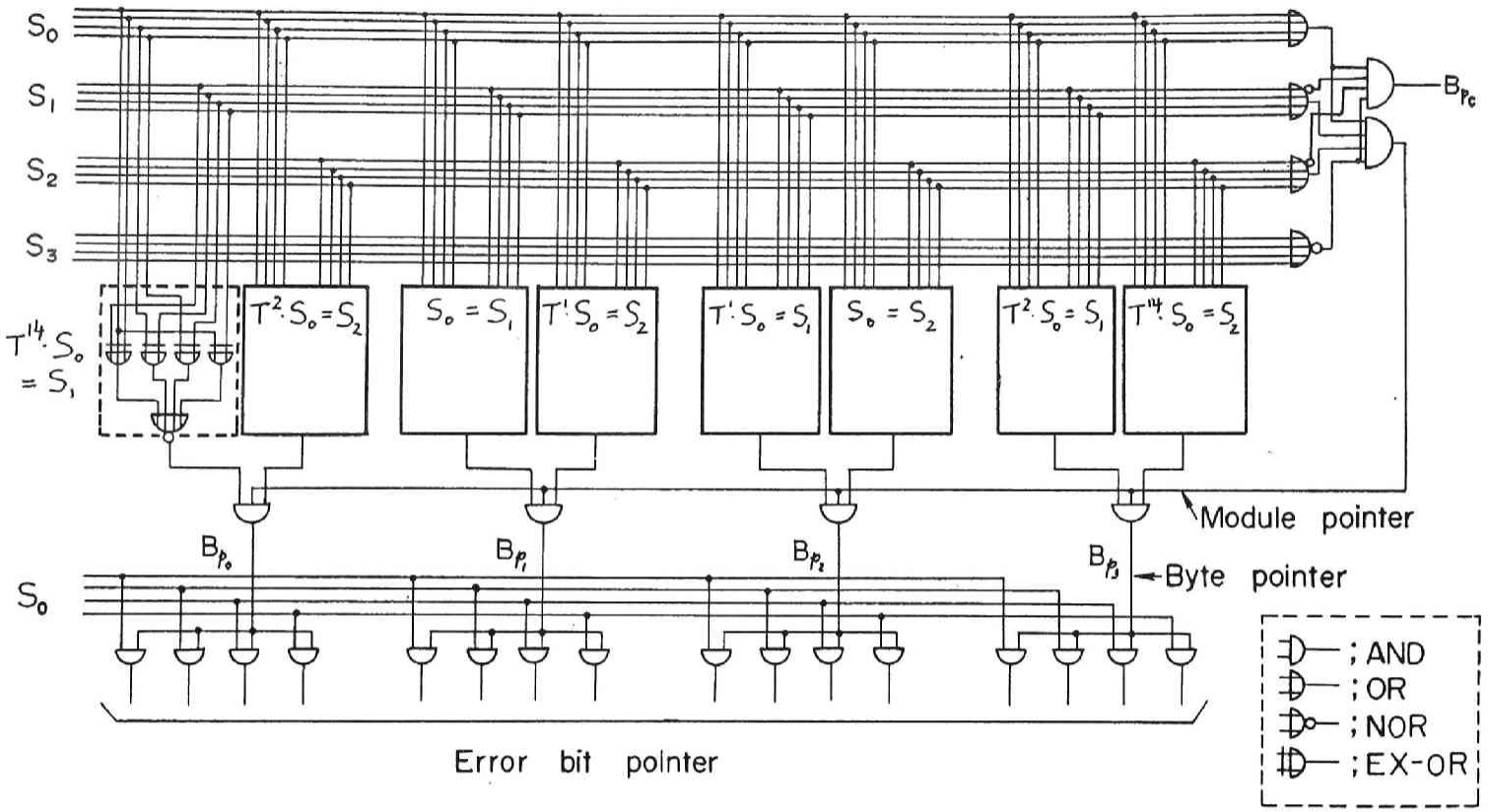


図5-8　　シンドロームデコード回路構成例

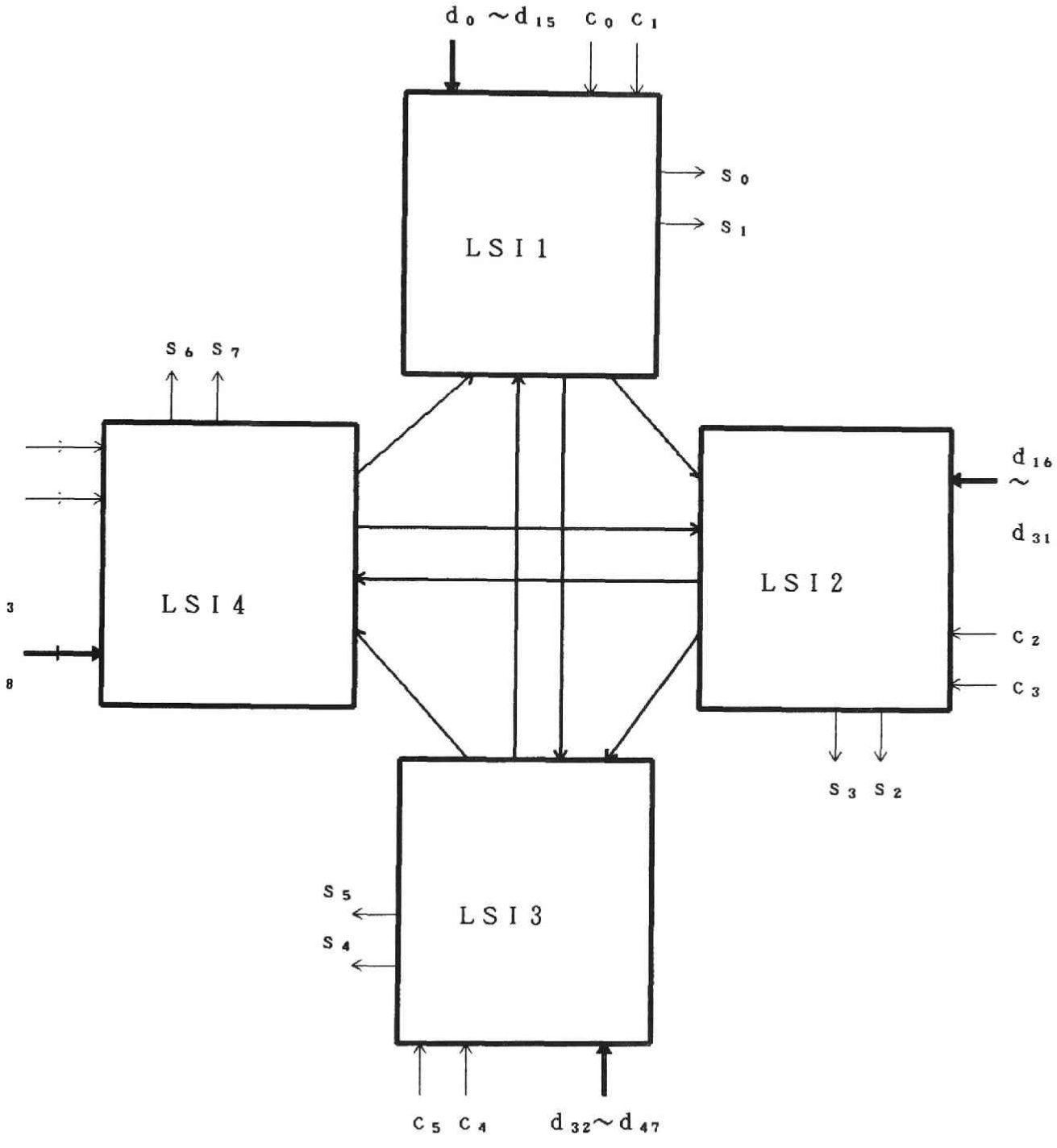


図5-9　シンドローム生成回路の構成

5.5 符号化・復号化回路のゲート量比較

本節では，各種のバイト誤り検出・訂正符号について，符号化・復号化に必要な，ゲート量，遅延を比較する。表5-1 は，モジュラな性質を用いてＬＳＩ化した時の，ＬＳＩ規模を示す。ＬＳＩのゲート規模，ピン数等は，実用的にも妥当な範囲にある。さらに，図5-10には，主要なバイト誤り検出・訂正符号に対する符号化・復号化ゲート量と遅延を比較して示す。この図から，データビット長ｋ＝６４ビットでは，符号化・復号化ゲート量・遅延に大きな差はないものの，データビット長ｋ＝１２８ビットでは，符号の機能により，符号化・復号化ゲート量・遅延が若干変化することがわかる。しかし，いずれの場合にも，現状のＳＥＣ－ＤＥＤ符号との差は大きなものではなく，符号化・復号化回路のＬＳＩ化を前提とすれば，実用上，問題となる遅延・ゲート量の増加は見られない。

表5-1　LSI化に適した符号におけるＬＳＩ化規模

符号種別	回路の繰り返し度	チェックビット生成回路・シンドローム生成回路			シンドロームデコード回路			参照
		ゲート数＊	ピン数	遅延＊＊	ゲート数＊	ピン数	遅延＊＊	
最小重み(39,32) SEC-DED 符号	2	９９	２７	5	４４	２３	2	(16)
(40,32)SEC-DED-S4ED 符号	2	７６	２８	5	３０	２４	3	図3-9
最小重み(72,64) SEC-DED 符号	8	５３	２３	6	２０	１８	2	(16)
(72,64)SEC-DED-S4ED 符号　号	2	１６０	４８	6	１１５	４０	3	図3-17
(80,64)S4EC-D4ED 符号	4	１１６	４０	6	１１０	３２	5	図4-8
最小重み(137,128)SEC-DED符号	8	１１９	３２	7	３５	２７	2	(16)
(144,128)S4EC-D4ED符号	4	１４０	５６	7	３８４	４８	5	図4-6

＊）４入力ＯＲ－ＮＯＲ換算　　＊＊）論理段数

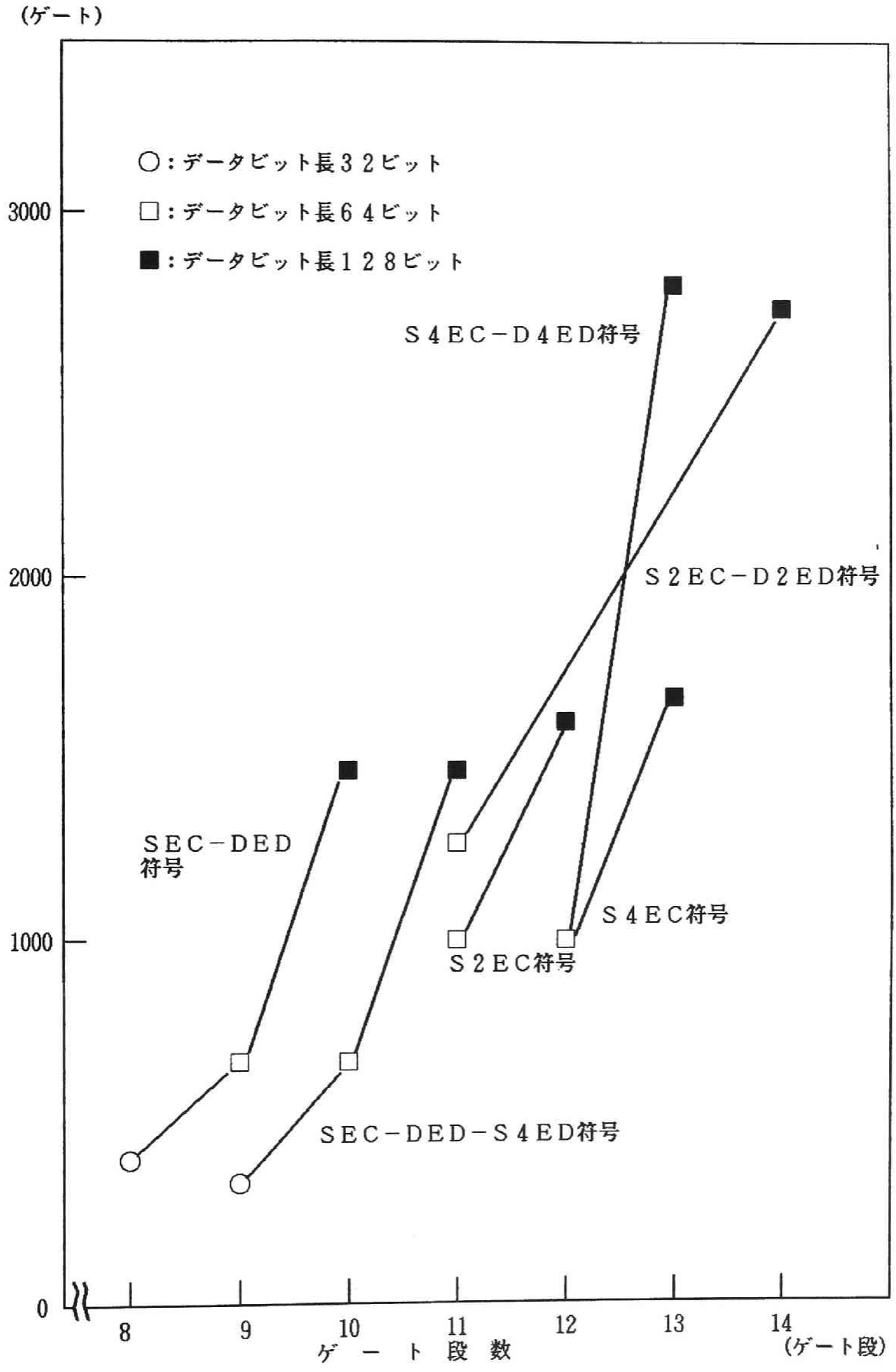


図5-10　符号化・復号化回路ゲート量と遅延の相対比較

５．５　結語

本章では，各種のバイト誤り検出・訂正符号について，その符号化・復号化回路の構成，ゲート量，遅延をあきらかにした。その結果，以下の結論を得た。

⑴本論文で提案した，単一バースト誤り検出ＳＥＣ－ＤＥＤ符号，奇数重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号，最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の誤り検出回路は，簡明かつ系統的に構成できる。

⑵本論文で示した各種バイト誤り検出・訂正符号について，データビット長ｋ＝６４ビットでは，符号の機能にかかわわらず，符号化・復号化ゲート量・遅延は，ほぼ同等である。一方，データビット長ｋ＝１２８ビットでは，符号の機能により，ゲート量が変化するが，遅延は，高々２～３割の変化にとどまる。いずれの場合にも，現状のＳＥＣ－ＤＥＤ符号との差は僅少であり，符号化・復号化回路のＬＳＩ化を前提とすれば，実用上，無視し得る。

以上から，本論文で提案した各種のバイト誤り検出・訂正符号は，符号化・復号化回路の回路化の観点からも，半導体記憶装置用符号として，十分な実用性を有すると判断できる。

第６章　高信頼化効果[58][62][63][77]

６．１　緒言

半導体記憶装置に誤り検出・訂正符号を適用するためには，その信頼度向上効果を定量的に評価する必要がある。本章では，本論文で提案した各種バイト誤り検出・訂正符号の高信頼化効果について論ずる[58]。また，α線等によるソフトエラーを考慮した，半導体記憶装置記憶部の信頼度近似算出手法を提案する[62][63][77]。

６．２　複数ビット出力記憶素子に対するバイト誤り検出・訂正符号の効果[58]

本節では，複数ビット出力記憶素子に対する，(1)ＳＥＣ－ＤＥＤ符号，(2)ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号，(3)ＳｂＥＣ－ＤｂＥＤ符号，(4)ＳｂＥＣ符号の高信頼化効果を定量的に明らかにし，各符号の適用範囲について論ずる。

６．２．１　信頼度算出モデル

算出モデル

一般に，半導体記憶装置は，

(1)記憶部――――記憶素子とその駆動回路を搭載したメモリーカード。

(2)制御論理部―――対論理装置インターフェース制御，記憶部制御等のための制御回路（装置の電源も含む。）

の２つの部分から構成される。このなかで，制御論理部は，誤り検出・訂正符号（ＥＣＣ）の効果が及ばない部分であり，使用されている半導体部品の故障率積み上げにより，制御論理部故障率を算出できる。これに対して，記憶部は，誤り検出・訂正符号による高信頼化効果が及ぶ。従って，本論文では，記憶部の故障率算出手法について明らかにする。記憶装置全体の故障率が必要な場合には，本論文で述べた手法により算出した記憶部故障率に，制御論理部の故障率を加算することにより，ただちに求めることができる。

記憶部（メモリーカード）は，記憶素子を主要な構成部品とし，その他に，アドレスバッファ回路，制御信号バッファ回路，入出力データバッファ回路等を含む。これらの回路のなかで，入出力データバッファ回路はビット対応であり，ＥＣＣによる故障救済対象となる。しかし，アドレスバッファ回路・制御信号バッファ回路は，故障の影響が複数の記憶素子におよぶため，ＥＣＣによる故障救済対象外となる。

上述の直接周辺回路のＦＩＴ数は小さく，６４Ｋビット素子の場合を例にとると，（回路構成によって変化するが），１ＭＢあたり数千ＦＩＴ程度までと考えられる。従って，本論文では，図6-1 に示す様に，記憶素子のみを信頼度解析の対象とし，以下の条件のもとに解析を進める。

(1)故障の生起はランダム生起。

(2)故障率算出対象は，記憶素子のみに限定。

(3)ＣＰＵからの書込みは無いものとする。（ワースト・ケースで評価する。）

(4)１個の記憶素子が故障した時には，βの割合で，出力ビットの２ビット以上に誤りが生じるものとし，$1-\beta$の割合で１ビットの誤りが生じるものとする。

(5)出力ビットに誤りがある時は，記憶素子内の全アドレスの当該ビットに誤りを生じるものとする。

(6)定期保守を考慮し，定期保守時には，全ての故障素子が修復されるものとする。

以下に記号をまとめておく。

λ_m	：記憶素子の故障率（ソフトエラーは考慮しない。）
n	：符号ビット長（従って，ｎ／ｂ個の記憶素子が１符号化単位となる。）
m	：アドレス方向の記憶素子数
τ	：定期保守間隔
β	：１個の記憶素子が故障した時に，２ビット以上の複数ビット誤りを生じる割合

半導体記憶装置では，故障が生じた際に，この故障を確実に検出できることが要求される。このため，故障率中にしめる「検出できない故障率」（以下，非検出故障）の大きさ

が問題となる。従って，本節では，故障率を，全故障率と，検出できない故障率にわけて考えることにする。

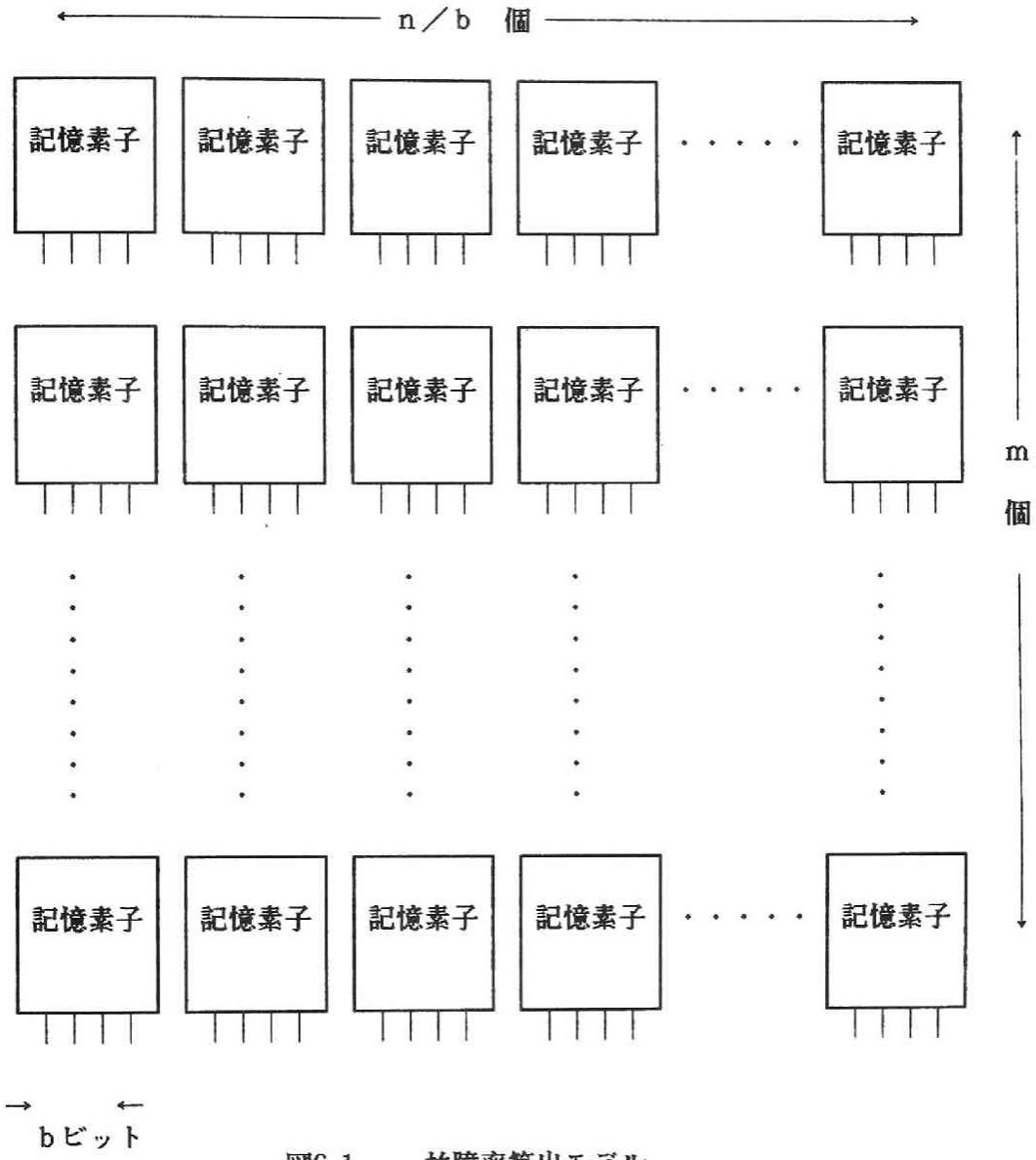


図6-1　　故障率算出モデル

6．2．2　記憶部故障率の導出

故障の生起をランダムとしているから，時間T＝0からT＝tの間に，1個の記憶素子が故障する確率は次式で与えられる。

$$P_m(t)=1-e^{\lambda_m t} \quad \cdots\cdots(6\text{-}1)$$

記憶素子は，1符号化単位にn／b個あり，n／b個の記憶素子中，1個，2個，3個以上故障する確率は，それぞれ以下の式で与えられる。ただし，Cは組合せを表す。

$$P_1(t)={}_{(n/b)}C_1 \cdot P_m(t)\cdot(1-P_m(t))^{(n/b)-1} \quad \cdots\cdots(6\text{-}2)$$

$$P_2(t)={}_{(n/b)}C_2 \cdot P_m(t)^2 \cdot(1-P_m(t))^{(n/b)-2} \quad \cdots\cdots(6\text{-}3)$$

$$P_3(t)=1\quad P_0(t)-P_1(t)-P_2(t) \quad \cdots\cdots(6\text{-}4)$$

ここで，$P_0(t)$は記憶素子n／b個中に故障が無い確率であり，$e^{-\lambda_m nt/b}$となる。

各$P_1(t)$，$P_2(t)$，$P_3(t)$の故障確率に対する検出能力は，符号により異なる。したがって，各符号ごとに，n／b個の記憶素子に対する故障確率（$P_4(t)$とする），及び，非検出故障確率（$P_5(t)$とする）を求める必要がある。これらの$P_4(t)$，$P_5(t)$を用いて，全記憶素子に対する故障率・非検出故障率を（定期保守を考慮して）以下の様に導出できる[59]。

$$\text{全故障率}\quad Q(\tau)=\frac{1-\{1-P_4(\tau)\}^m}{\int_0^\tau \{1-P_4(t)\}^m dt} \quad \cdots\cdots(6\text{-}5)$$

$$\text{非検出故障率}\quad Q_n(\tau)=\frac{P_5(\tau)\times m}{\int_0^\tau \{1-P_5(t)\times m\}\, dt} \quad \cdots\cdots(6\text{-}6)$$

以下に，各符号ごとの故障確率$P_4(t)$，非検出故障確率$P_5(t)$の算出式を示す。

SEC－DED符号

SEC－DED符号では，訂正できるのは，1ビット誤りのみであり，2ビット以上の誤りの訂正はできない。従って，n／b個の記憶素子に対する故障確率（$P_4(t)$），及び，非検出故障確率（$P_5(t)$）は，以下の様になる。

$$P_4(t) = \beta P_1(t) + P_2(t) + P_3(t) \quad \cdots\cdots (6\text{-}7)$$

$$P_5(t) = k_0 \beta P_1(t) + k_1 P_2(t) + k_2 P_3(t) \quad \cdots\cdots (6\text{-}8)$$

ここで，　k_0：2ビット以上の単一バイト誤りを見逃す割合，

k_1：二重バイト誤りを見逃す割合，

k_2：三重以上のバイト誤りを見逃す割合，であり，

k_0, k_1, k_2は計算機シミュレーションにより求める必要がある。

SEC－DED－SbED符号

SEC－DED－SbED符号では，訂正できるのは，1ビット誤りのみであり，2ビッ誤りの訂正はできない。従って，n／b個の記憶素子に対する故障確率（$P_4(t)$），及び，非検出故障確率（$P_5(t)$）は，以下の様になる。

$$P_4(t) = \beta P_1(t) + P_2(t) + P_3(t) \quad \cdots\cdots (6\text{-}9)$$

$$P_5(t) = k_1 P_2(t) + k_2 P_3(t) \quad \cdots\cdots (6\text{-}10)$$

ここで，　k_1：二重バイト誤りを見逃す割合，

k_2：三重以上のバイト誤りを見逃す割合，であり，

k_1, k_2は計算機シミュレーションにより求める必要がある。

ここで，式(6-9)は，SEC－DED符号の故障確率を表す式(6.7)と同一であり，式(6-10)は，後述のSbEC符号の非検出故障確率を表す式(6.12)と同一である。このことから，概略的に，SEC－DED－SbED符号は，「SEC－DED符号と同一の全故障確率を持ち，SbEC符号と同一の非検出故障確率を持つ」と言える。

ＳｂＥＣ符号

ＳｂＥＣ符号では，単一のバイト誤りを完全に訂正できる。しかし，二重バイト誤りの検出は不完全である。従って，ｎ／ｂ個の記憶素子に対する故障確率（$P_4(t)$），及び，非検出故障確率（$P_5(t)$）は，以下の様になる。

$$P_4(t) = P_2(t) + P_3(t) \quad \cdots\cdots (6\text{-}11)$$

$$P_5(t) = k_1 P_2(t) + k_2 P_3(t) \quad \cdots\cdots (6\text{-}12)$$

ここで，　k_1 ：二重バイト誤りを見逃す割合，

k_2 ：三重以上のバイト誤りを見逃す割合，であり，

k_1, k_2 は計算機シミュレーションにより求める必要がある。

ＳｂＥＣ－ＤｂＥＤ符号

ＳｂＥＣ－ＤｂＥＤ符号では，単一のバイト誤りを完全に訂正でき，しかも，二重バイト誤りを完全に検出できる。従って，ｎ／ｂ個の記憶素子に対する故障確率（$P_4(t)$），及び，非検出故障確率（$P_5(t)$）は，以下の様になる。

$$P_4(t) = P_2(t) + P_3(t) \quad \cdots\cdots (6\text{-}13)$$

$$P_5(t) = k_2 P_3(t) \quad \cdots\cdots (6\text{-}14)$$

ここで，　k_2 ：三重以上のバイト誤りを見逃す割合，であり，

k_2 は計算機シミュレーションにより求める必要がある。

６．２．３　結果の比較

代表的な諸元であるデータビット長ｋ＝６４ビット，バイト長ｂ＝４ビットの場合について，各符号の高信頼化効果を具体的に導出した結果を図6-2 に示す。使用した各符号の多数ビット誤りに対する計算機シミュレーション結果は，付録6.1 に示す。

図6-2 及び他の符号諸元を持つ符号に対する評価結果から，以下の結論を得る。

⑴ＳＥＣ－ＤＥＤ符号またはＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を使用した時の故障率は，誤り訂正を実施しない時の全記憶素子の故障率にβ（１個の記憶素子が故障した時に，２ビット以上の多数ビット誤りを生じる割合）を乗じた値となり，しかも，定期保守間隔の影響を受けない。従って，これらの符号は，βの値が小さい時，あるいは，記憶素子数が小さく，あまり大きな高信頼化効果を要求されない装置への応用に適する。

⑵ＳＥＣ－ＤＥＤ符号の２ビット以上の単一バイト誤りに対する検出能力は低い。従って，ＳＥＣ－ＤＥＤ符号の適用は，βが数%以下と小さい時に限ることが望ましい。

⑶ＳｂＥＣ符号またはＳｂＥＣ－ＤｂＥＤ符号を適用することにより，記憶素子部の故障率を１～２桁改善できる。しかし，非検出故障率には，大きな差があり，ＳｂＥＣ－ＤｂＥＤ符号の非検出故障率は無視し得るのに対して，ｂ＝４では，ＳｂＥＣ符号の非検出故障率は，無視できない。従って，ＳｂＥＣ符号の適用は，バイト長ｂが８などの大きな値の時に限ることが望ましい。

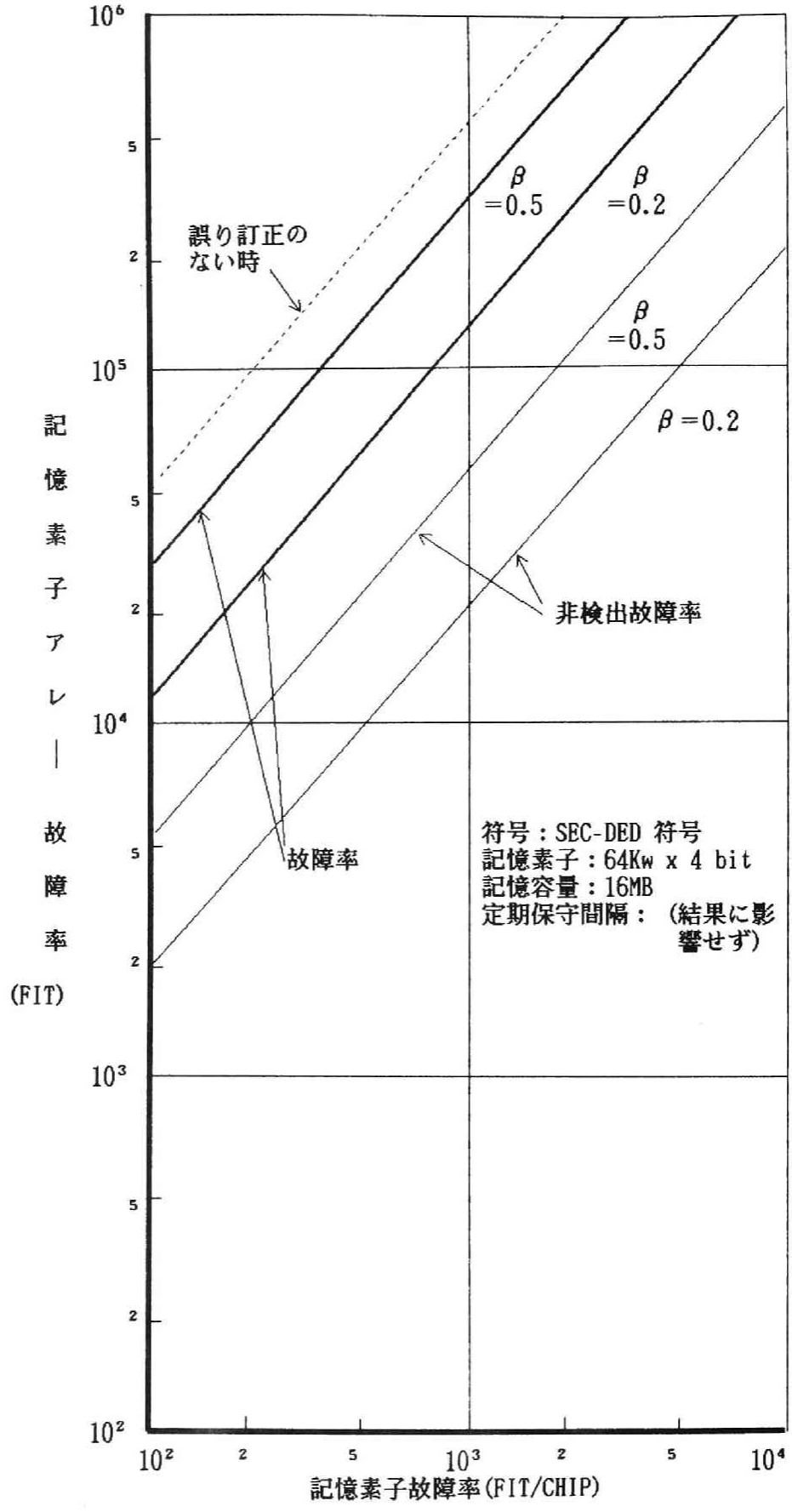


図6-2　複数ビット出力記憶素子に対するバイト誤り検出・訂正符号の効果(1/4)

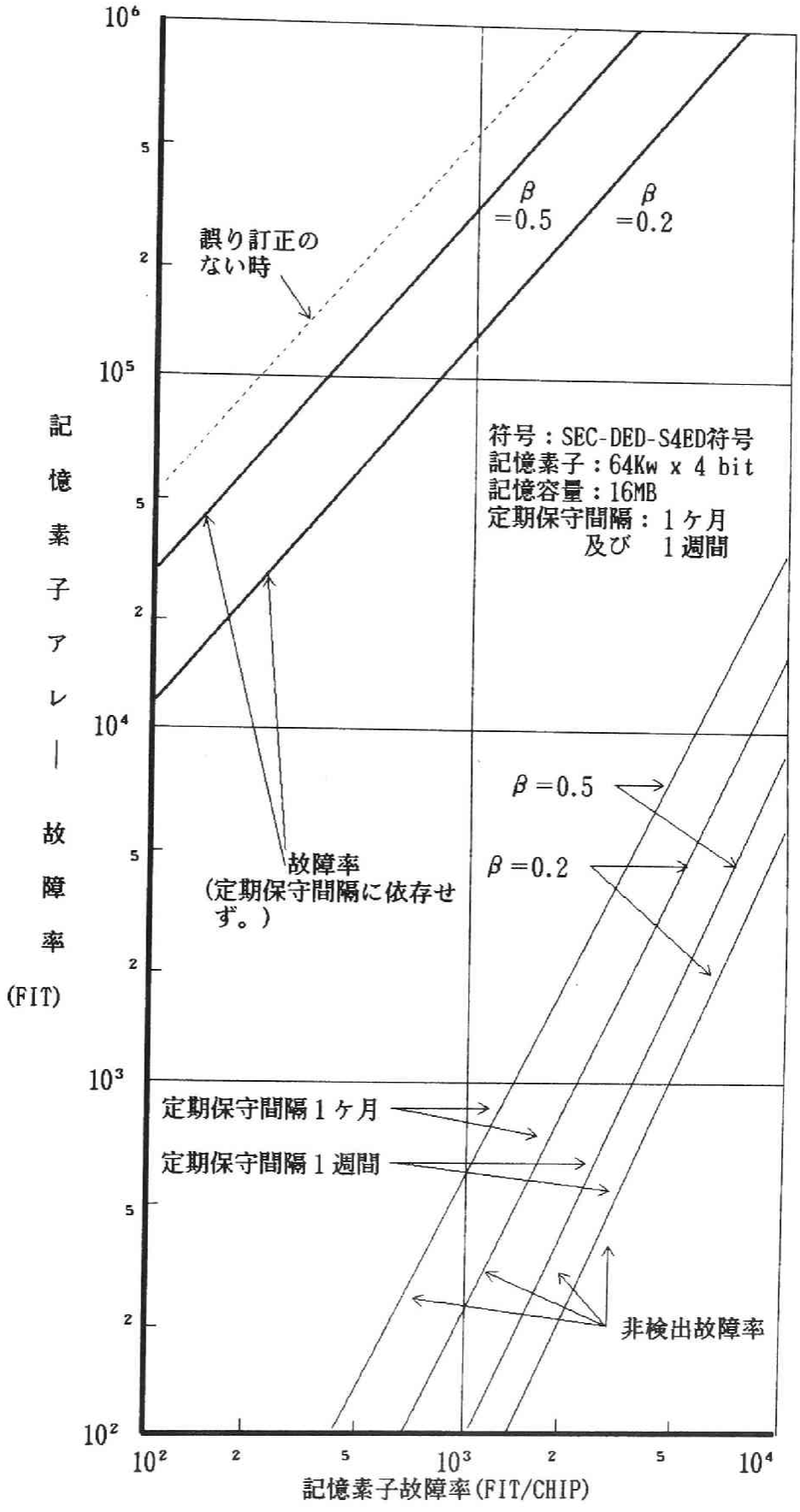


図6-2　複数ビット出力記憶素子に対するバイト誤り検出・訂正符号の効果(2/4)

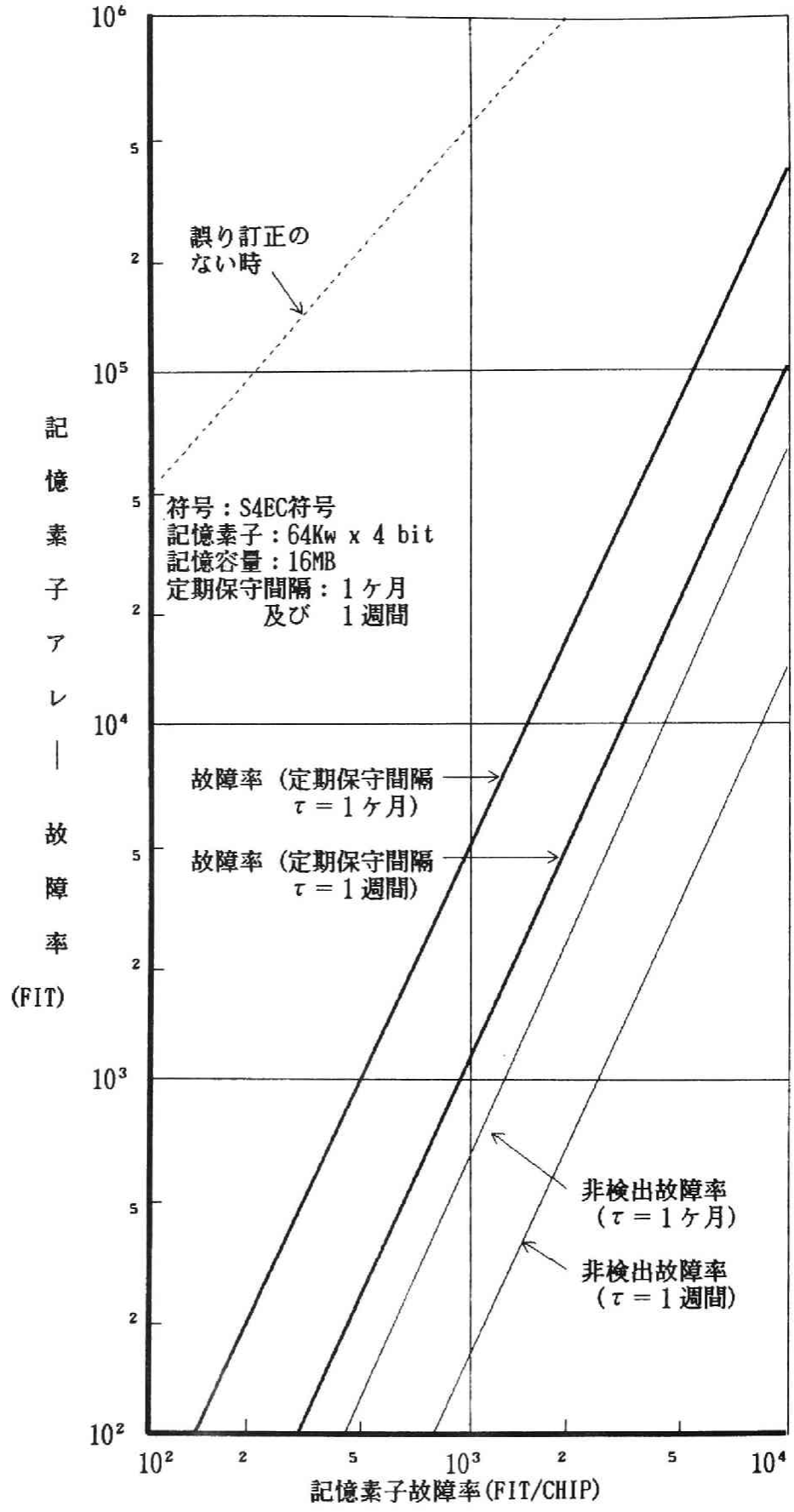


図6-2　複数ビット出力記憶素子に対するバイト誤り検出・訂正符号の効果(3/4)

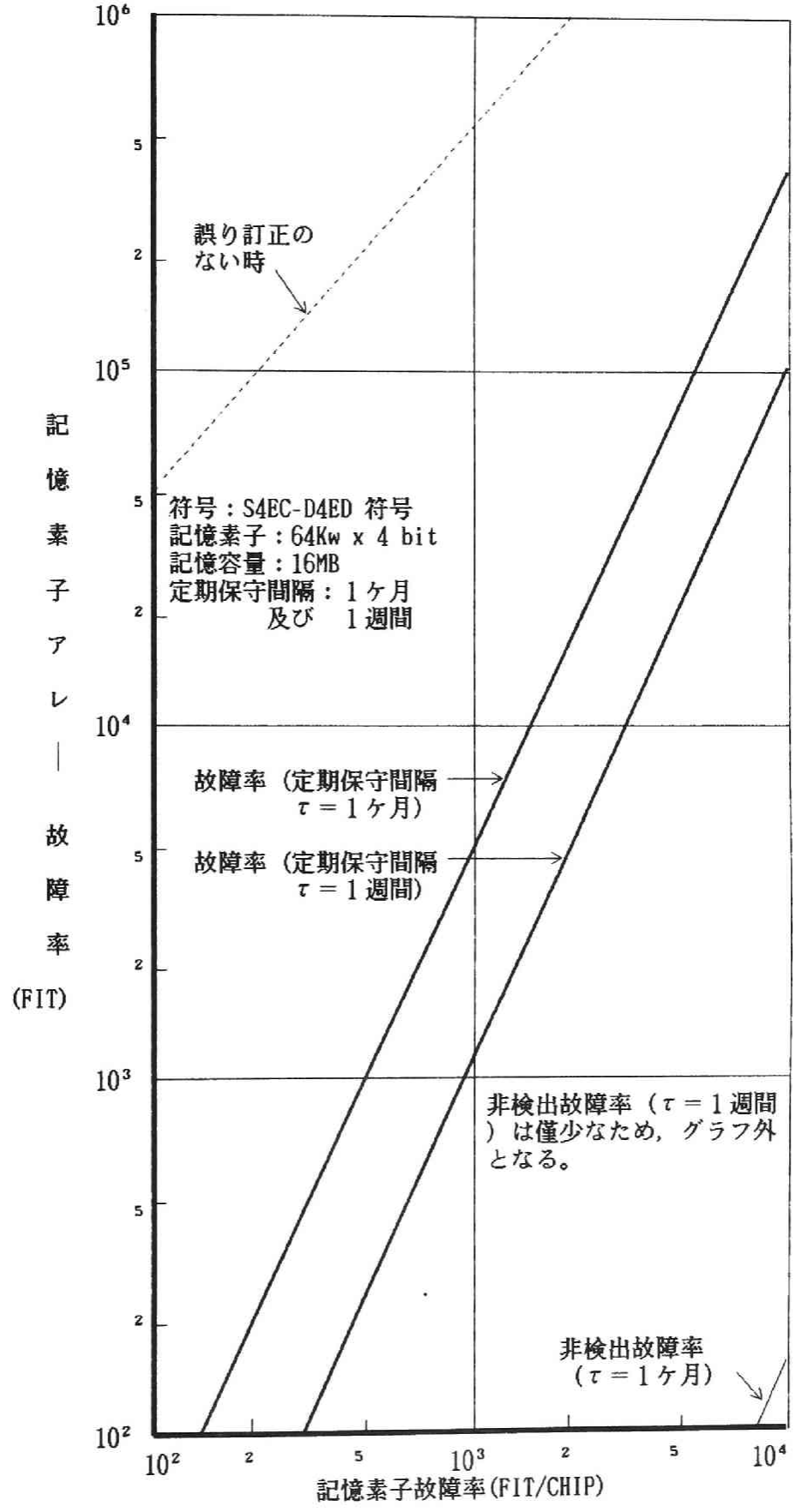


図6-2　複数ビット出力記憶素子に対するバイト誤り検出・訂正符号の効果(4/4)

6.3 ソフトエラーを考慮した半導体記憶装置の信頼度設計[62]-[63]

半導体記憶素子の高集積化に伴い，記憶素子に生じるソフトエラー[60]が問題となっている。現行の大型計算機用半導体記憶装置は，ＳＥＣ－ＤＥＤ符号による誤り訂正機能を備えているため，ソフトエラーの発生頻度が小さければ，この誤り訂正機能により対処できる。しかし，発生頻度が一定の限界を越えて大きくなると，現状の誤り訂正機能では高信頼化効果が不十分となる。この様な観点から，種々の装置構成上のソフトエラー対策手法（パトロール走査方式，チップ交替方式等）が提案され実用に供されている。

パトロール走査方式，チップ交替方式等を適用した場合の装置故障率を算出するため，従来，複雑な解析[64]やシミュレーション[67][68][69]を用いる手法が開発されて来た。しかし，これらの手法では，主要な故障モードを抽出したり，装置設計上で最適な装置構成パラメータを見通し良く設計することは困難である。迅速かつ見通しの良い装置設計を行うために，簡便な故障率算出法が望まれる。

本節では，実用性の高い，４種類のソフトエラー対策手法を選び，それぞれの手法を適用した場合の装置故障率を求める解析的な式を導出し，この結果から，見通し良く故障率を算出できる近似式を提案する。また，各手法の効果を比較し，パトロール走査方式では，ＳＥＣ－ＤＥＤ符号のみによる場合と比較してソフトエラーを２倍程度許容できること，これに対して，Erasure 訂正方式，チップ交替方式では１０倍以上許容できることを明らかにする。

以下，最初に，評価対象となる実用的な高信頼化手法を示し，次に，厳密な故障率算出式を示す。更に，近似式を示して，厳密式との誤差について論ずる。

6．3．1　ソフトエラーに対する装置側の高信頼化対策

ソフトエラーにたいする装置構成上の対策として，種々の手法が提案されている。これら手法のなかで，

⑴高信頼化のために必要となるハードウェア量が実用的に許容できる範囲であること，

⑵誤訂正の生じる危険性が低いこと，

等の観点から実用性の高い手法を表6-1 に示す。但し，記憶素子としては，1ビットデータ出力素子を前提とする。また，現状の誤り検出・訂正符号（ＥＣＣ）のみによる手法も，ソフトエラー対策の一つとして，表6-1 に含める。

表6-1　ソフトエラーにたいする高信頼化手法

手法	概要	文献
ＳＥＣ－ＤＥＤ符号のみ	1ビット誤り訂正・2ビット誤り検出符号による。ＣＰＵからの読みだし時に1ビット誤りを訂正する。	16
ＳＥＣ－ＤＥＤ符号 ＋ パトロール走査方式	短い周期（数分以下）で，半導体記憶装置記憶全アドレスを読みだし，誤りが生じていれば訂正して再書込みする。	61
ＳＥＣ－ＤＥＤ符号 ＋ パトロール走査方式 ＋ Erasure 訂正方式	固定故障1ビットとソフトエラー1ビットによって2ビットの誤りが生じると，テストパターンを書込み，再読みだしすることによって，固定故障1ビットを特定し，Erasure 訂正手法によって，残る1ビットを訂正する。	65
ＳＥＣ－ＤＥＤ符号 ＋ パトロール走査方式 ＋ チップ交替方式	予め予備記憶素子を設けておき，固定故障を生じた記憶素子を，この予備記憶素子と切り替える。切り替え契機として (1)あらゆる固定故障 (2)全セル固定故障 の2通りがある。予備素子への正しいデータの回復は，予備素子を含む読みだし単位の全アドレスに対する，読みだし⇒1ビット誤り訂正　⇒再書込みにより行う。	64

注）適用手法の「＋」は，各手法を併用することを示す。組合せとしては，上記以外にもたとえば，チップ交替方式でパトロール走査方式を併用しない方法等が考えられるが，これら除外した組合せでは高信頼化効果が限定されたものとなるため実用的ではない。

以下に，表6-1 の各手法を簡単に説明しておく。ただし，ECC のみの時については，説明を省略する。

〔パトロール走査方式〕[61]

パトロール走査方式は，短い周期で主記憶部の全アドレスを読みだし，1ビット誤りが生じていれば，符号能力により訂正・再書込みする手法である。1ビット誤りとして読みだされるソフトエラーはこの再書込みにより訂正される。このパトロール走査方式は，一

旦，１ビット記憶素子の全語にまたがる固定故障（全セル固定故障）が生ずると，同一の読み出し単位内に生じたソフトエラーを訂正できない欠点がある。この全セル固定故障とソフトエラーの重畳による訂正不能誤りは（ソフトエラーが固定故障に比べて多い通常の場合には，）主要な故障モードである。

全セル固定故障とソフトエラーとの重畳による訂正不能誤りを防ぐために考案された手法が，以下に述べるErasure 訂正方式，チップ交替方式である。

〔Erasure 訂正方式〕[65]

ＳＥＣ－ＤＥＤ符号では，２ビット誤りのうち１ビットのビット位置が明らかになれば，残る１ビットのビット位置を指摘できる。この性質を使用して，２ビット誤りを訂正する手法が，Erasure 訂正方式である。固定故障１ビットの位置を特定するためには，２～３種のテストパターンを書込み・再読み出しする手法を使用する。

〔チップ交替方式〕[64]

チップ交替方式では，固定故障がいずれかの記憶素子に生じると，この固定故障記憶素子を予備の記憶素子と切り替える。素子切り替えのためには，半導体記憶装置内に，セレクタ回路を設ける必要がある。切り替え後の予備記憶素子には，本来記憶されているべき正しいデータは記憶されていない。従って，切り替え後には，予備記憶素子を含む読みだし単位に対して，「読み出し⇨１ビット誤り訂正⇨再書込み」を行っておかねばならない。

なお，表6-1 に示した手法以外にも，ポインタによる２ビット誤り訂正手法，逆パターンの再書込みによる２ビット誤り訂正を行うマスク訂正手法等が提案されている。しかし，これらの手法は，いずれも，誤訂正を生じる恐れがあり，実用的には，改善の余地を残している。

以下，上記４種類の手法における高信頼化効果を定量的に明らかにしてゆく。

６．３．２　高信頼化効果（厳密式）

算出モデル

本節では，記憶素子の出力ビット数が１ビットとなった他は，図6-2 の記憶素子アレーを故障率算出の対象とする。以下の条件を前提とする。

⑴故障の生起はランダム生起。

⑵故障率算出対象は，記憶素子のみに限定。

⑶ＣＰＵからの書込みは無いものとする。

⑷定期保守を考慮し，定期保守時には，全ての故障素子が修復されるものとする。

⑸記憶素子の故障モードを「全セル固定故障」「単独セル固定故障」「ソフトエラー」の３種類に区分する。（詳細後述）

だだし，ここでは，誤訂正の十分小さな手法のみを採り上げているので，誤訂正の確率に相当する検出出来ない故障率については考慮しない。

故障率のモード区分

近年，記憶素子の集積度の増大とともに，素子固定故障時でも，かならずしも当該素子の全セルがスタックするとは言えなくなってきている。記憶素子チップ内での面積比率を参考にすると，固定故障の故障モードは，以下の様に区分できる。

故障モード区分例（６４Ｋビット素子）

全セル固定故障	２０％
ビット線固定故障	１５％
ワード線固定故障	１５％
セル単独固定故障	５０％

故障率の算出に当たっては，ビット線，ワード線の故障を単独セル固定故障に含めて考えても，大きな誤差を生じないことを確認した（詳細は省略する）。従って，全セル固定故障８０％が上記故障モード区分例と等価となる。なお，単独セル固定故障が固定故障中の

１００％に近いことは，近似式の導出にあたって近似に使用することがある。

以下に，故障率算出のシンボル一覧を示す。

λ_h	：記憶素子１個当たりの固定故障率（１／時間）
λ_s	：記憶素子１個当たりのソフトエラー率（１／時間）
n	：ＥＣＣビット幅（ビット）
m	：語方向の記憶素子数。（個）
τ	：定期保守間隔（時間）
α_a	：固定故障が生じた時に，全セル固定故障となる割合。
α_c	：固定故障が生じた時に，単独セル固定故障となる割合。
A	：記憶素子のワード数

（A）　ＥＣＣのみの時の訂正不能故障率

表6-1 の項番１に相当するＥＣＣのみが適用されている場合の訂正不能故障率を求める。定期保守間隔τの時，訂正不能故障率は次式で与えられる。

$$Q_1(\tau) = \frac{p_1(\tau)}{\int_0^{\tau} \{1 - p_1(t)\}\, dt} \quad \cdots\cdots\cdots\cdots\cdots\cdots (6\text{-}15)$$

$p_1(t)$ は次式で与えられ，t＝0からt＝tまでの間に記憶部に訂正不能誤りが生じる確率である。式の導出については，付録6-2 を参照されたい。

$$p_1(t) = 1 - \left[e^{-(n-1)(\lambda_s + \lambda_h)t} \cdot \left[n\,(1 - e^{-\alpha_a \lambda_h t}) + \left[(1-n)\, e^{-(\lambda_s + \lambda_h)t/A} + n\, e^{-\alpha_a \lambda_h t/A} \right] A \right] \right]^m$$

$$\cdots\cdots\cdots\cdots\cdots\cdots (6\text{-}16)$$

（B）　パトロール走査方式における訂正不能故障率

ＥＣＣが適用され，かつパトロール走査を適用した場合の訂正不能故障率を求める。この場合，全セル固定故障が存在しない間は，ソフトエラーを訂正できる。しかし，全セル固定故障が発生すると，全セル固定故障と同一の読みだし単位中に生じたソフトエラーは訂正できなくなる。従って，図6-3 の様に，区間を分けて考える。まず，記憶素子はｎ個とし，Ｔ＝ｔで全セル固定故障を生じたものとする。

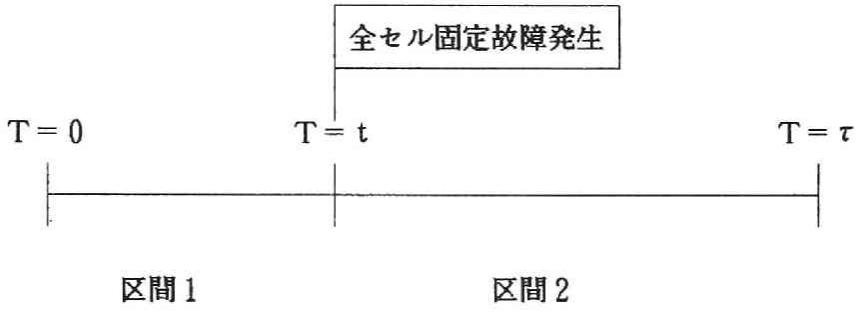


図6-3　区間の区分（1）

区間1：Ｔ＝0～Ｔ＝ｔ

全セル固定故障は無い。単独セル固定故障が生じている可能性がある。ソフトエラーはパトロール走査により修正を受けるので無視できる。

区間2：Ｔ＝ｔ～Ｔ＝τ

区間1で，単独セル固定故障が生じていると，Ｔ＝ｔでの全セル固定故障により，訂正不能となる。区間1で，単独セル固定故障が生じていなければ，この区間2で，なんらかの故障が生じると訂正不能となる。

以上の2区間から，ＥＣＣ訂正不能となる条件をもとめ，Ｔ＝0～τまでの間で積分する。これによって，ｎ個の記憶素子に対して，訂正不能な誤りを生じる確率が次式の様に与えられる。なお，Ｔ＝0～τの間に，全セル固定故障が生じなくても，単独セル固定故障とソフトエラーによる2ビット誤りが生じる事がある。しかし，その確率は小さく無視しうる。式の導出については，付録6-2 を参照されたい。

$$p'(\tau) = (1 - e^{-\alpha_a \lambda_h t}) + n\alpha_a \lambda_h \frac{e^{-(n-1)(\lambda_s + \lambda_h)\tau} - e^{-n\lambda_h \tau}}{(n-1)\lambda_s - \lambda_h} \quad \cdots\cdots (6\text{-}17)$$

全体では，m×n個の記憶素子があるため，定期保守間隔τの時の訂正不能故障率は次式で与えられる。

$$Q_2(\tau) = \frac{p_2(\tau)}{\int_0^\tau \{1 - p_2(\tau)\}\, d\tau} \quad \cdots\cdots (6\text{-}18)$$

ここに，$p_2(\tau)$ は次式で与えられる。

$$p_2(\tau) = 1 - \{1 - p'(\tau)\}^m \quad \cdots\cdots (6\text{-}19)$$

(C)　Erasure 訂正方式における訂正不能故障率

パトロール走査方式を併用したErasure 訂正方式では，ソフトエラーが発生しても，訂正できる。しかし，固定故障のみによって2ビット以上の訂正不能誤りが生じると，訂正できなくなる。（なお，理論的には，固定故障のみによって2ビット誤りが生じても，テストパターンによってその位置が確定でき，訂正できるはずである。しかし，この様な固定故障による2ビット誤りをも訂正可能範囲に含めると，多数ビット誤りにたいする誤訂正の恐れが増大して実用的とは言えなくなる。）

定期保守間隔τの時の訂正不能故障率は次式で与えられる。

$$Q_3(\tau) = \frac{p_3(\tau)}{\int_0^\tau \{1 - p_3(\tau)\}\, d\tau} \quad \cdots\cdots (6\text{-}20)$$

ここに，$p_3(\tau)$ は次式で表される。

$$p_3(\tau) = 1-\left[1-(1-e^{-n\lambda_h \alpha_a t})(1-e^{-(n-1)\lambda_h t})\right]^m \quad \cdots\cdots\cdots\cdots\cdots\cdots (6\text{-}21)$$

ここで，単独セル固定故障のみによって，２ビット以上の誤りが生じる確率は無視した。式の導出は，付録6-2 を参照されたい。

（D）　チップ交替方式における訂正不能故障率

チップ交替方式では，予備記憶素子は１個とし，この予備記憶素子には故障は生じないものとして，故障率を求める。チップ交替の契機としては，「あらゆる固定故障」と「全セル固定故障」の２通りが考えられる。

（D－１）交替契機があらゆる固定故障の時

図6-4 に示す２区間に分けて考える。Ｔ＝ｔで固定故障が生じて交替するものとする。ただし，読出しの単位となるｎ個の記憶素子を本論文では，「ＥＣＣグループ」と呼ぶ。

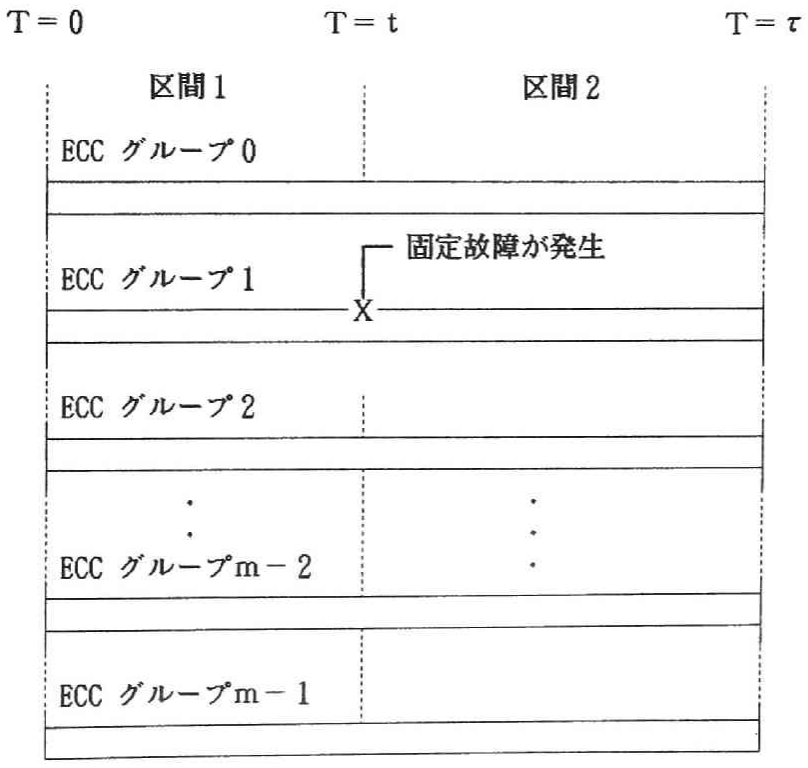


図6-4　区間の区分（２）

区間1：T＝0～T＝t

この区間1では，総てのＥＣＣグループにおいて，固定故障は存在しない。ソフトエラーはパトロール走査により訂正される。

区間2：T＝t～T＝τ

交替の生じたＥＣＣグループについても，他のＥＣＣグループについても，パトロール走査を適用した場合と同様の確率で訂正不能故障を生じる。

以上の条件から，定期保守間隔τの時の訂正不能故障率は，次式で与えられる。

$$Q_{4A}(\tau) = \frac{p_{4A}(\tau)}{\int_0^{\tau} \{1 - p_{4A}(\tau)\}\, d\tau} \quad \cdots\cdots\cdots\cdots\cdots\cdots (6\text{-}22)$$

ここで，$p_{4A}(\tau)$ は次式で与えられ，本式中のp'(t)は式 (6-17) で定義される。式の導出については，付録6-2 を参照されたい。

$$p_{4A}(\tau) = mn\lambda_h \int_0^{\tau} e^{-mn\lambda_h t} \cdot \{1-(1- p'(\tau-t))^{m}\}\, dt \quad \cdots\cdots\cdots (6\text{-}23)$$

上式は被積分関数が積分形式で与えられており，数値積分に多くの計算量を必要とする。また，式から主要な故障モードを抽出したり，一般的見通しをつけることも難しい。

（D－2）交替契機が全セル固定故障の時

図6-5 に示す2区間に分ける。T＝tで全セル固定故障が生じて交替するものとする。訂正不能誤りを生じる条件を表6-2 にまとめて示す。訂正不能故障率は次式で与えられる。

$$Q_{4B}(\tau) = \frac{p_{4B}(\tau)}{\int_0^{\tau} \{1 - p_{4B}(\tau)\}\, d\tau} \quad \cdots\cdots\cdots\cdots\cdots\cdots (6\text{-}24)$$

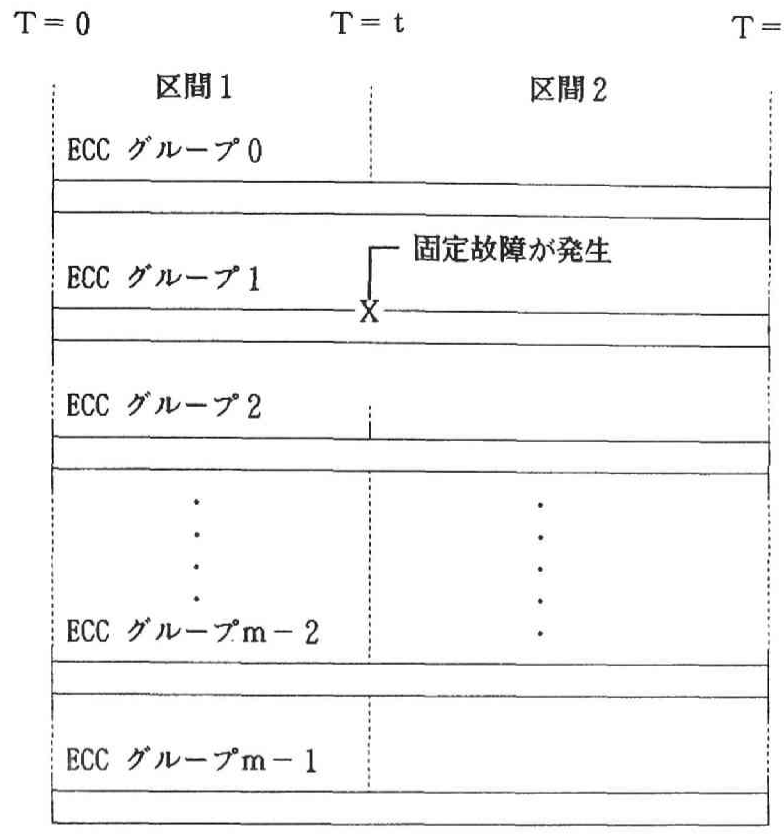


図6-5　区間の区分（3）

ここで，$p_{4B}(\tau)$ は次式で与えられる。本式中の $p'(t)$ は式(6-17)で与えられる。式の導出については，付録6-2 を参照されたい。

$$p_{4B}(\tau) = mn\alpha_a \lambda_h \int_0^{\tau} e^{-\alpha_a mn \lambda_h t} \times \left[\left[(1- e^{-n\alpha_c \lambda_h t}) + e^{-n\alpha_c \lambda_h t} \cdot p'(\tau - t) \right] + \left[1 - \left[1 \quad e^{-n\alpha_c \lambda_h t} \cdot p'(\tau - t) - (1- e^{-n\alpha_c \lambda_h t}) \cdot (1 - e^{-(n-1)\alpha_a \lambda_h (\tau - t)}) \right]^{m-1} \right] \right] dt \quad \cdots\cdots\cdots (6\text{-}25)$$

表6-2　訂正不能となる条件

区間	交替の対象となるＥＣＣグループ		その他のＥＣＣグループ	
区間１	単独セル固定故障無し。	単独セル固定故障あり。	単独セル固定故障無し。	単独セル固定故障あり。
Ｔ＝ｔ	全セル固定故障が発生		— — — — — — — —	
区間２	パトロール走査を適用した，期間τ—ｔの間の訂正不能確率に同じ。	訂正不能	パトロール走査を適用した，期間τ—ｔの間の訂正不能確率に同じ。	全セル固定故障が生じると訂正不能

６．３．３　高信頼化効果の近似的算出手法

前章で示した様に，ソフトエラーを考慮した主記憶装置記憶部の故障率算出式は複雑であり，装置パラメーターの見通しを得るためには適さない。本来，故障率計算は何桁もの有効数値をもとめても無意味である。むしろ，支配項となる故障モードが何であるかを簡潔に読み取れることが望ましい。逆に，主要な故障モードが読み取れることは，計算結果が物理モデルとしても納得のゆくものであることをチェックするために都合が良い。

本章では，前記の各高信頼化手法について，実用上十分な精度を有する近似式を提案する。近似にあたって，以下の公式を用いた。いずれも，期間τの間に故障が生じる確率ｐ（τ）が十分小さい時，良い近似を与える。実用上は，定期保守間隔中にしばしばシステムダウンすることはなく，以下の近似式は実用上，十分な精度を持っている。

近似１：　$$F = \frac{p(\tau)}{\int_0^{\tau} \{1 - p(t)\}\, dt}$$

$$= \frac{-\ln(1 - p(\tau))}{\tau} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (6\text{-}26)$$

近似2：　$\ln(1-x) = -\sum_{n=1}^{\infty} \frac{x^n}{n}$，　$0 < x < 1$　より，

$$F = \frac{-\ln\{1-p(\tau)\}}{\tau}$$

$$= \frac{p(\tau)}{\tau} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \text{(6-27)}$$

近似式の導出に当たっては，厳密式から数値計算に依って，主要故障モードを抽出し，これに，マクローリン級数展開を用いて近似式を求める手法を取った。λ_s / λ_h の値に応じて，主要故障モードが変化するため，近似式は，λ_s / λ_h の値に応じて領域分けをする必要がある。

（A）ＥＣＣのみの時の訂正不能故障率近似式

ＥＣＣのみの時の近似式は，λ_s / λ_h の値に応じて，以下の３領域に分けて考える。

（Ⅰ）λ_s / λ_h が小さく，全セル固定故障１ビットとその他の故障１ビットにより生じる２ビット誤りが故障の支配項となる領域。
（Ⅱ）λ_s / λ_h が比較的大きく，全セル固定故障が発生すると，直後にソフトエラーが生じて訂正不能となる領域。
（Ⅲ）λ_s / λ_h が極めて大きく，ソフトエラーのみによる２ビット誤りが支配項となる領域。

これらの各領域について，厳密式（6-15）の近似式を示す。

<u>領域Ⅰ</u>　：式（6-16）を，exp(x)のマクローリン級数展開の１次の項まで取って近似すると，次式を得る。

$$p_1(t) = mn(n-1)\alpha_a \lambda_h (\lambda_s + \alpha_c \lambda_h) t^2 \cdots\cdots \text{(6-28)}$$

従って，式（6-15）の近似式は，以下の様に与えられる。本式は，$\lambda_s / \lambda_h < 10^2$ 程度の領域で有効である。

$$\underline{Q}_I(\tau)_I = mn^2 \alpha_a \lambda_h (\lambda_h + \lambda_s) \tau \quad \cdots\cdots\cdots\cdots\cdots\cdots (6\text{-}29)$$

なお，$\underline{Q}$はQの近似を示し，添字の「I」は，領域を示す。

式（6-29）から，この領域での主要な故障モードは，（λ_s が λ_h に比べて大きい時は）$\alpha_a \lambda_h$ と λ_s の積であることが判る。即ち，全セル固定故障1ビットとソフトエラー1ビットによる2ビット誤りは主要な故障モードである。例えば，ソフトエラー（λ_s）が固定故障率（λ_h）の100倍あるとすると，$\underline{Q}$の値は，固定故障のみを考えた時の100倍になる。このことから，「ソフトエラーはインターミテントエラーであるから，SEC－DED符号による誤り検出訂正システムを持つ主記憶装置では問題とはならない」と考えることはできない。

領域Ⅱ　：全セル固定故障が生じると，直後に2ビット誤りとなる。従って，近似式は，以下の様になる。本式は，誤り訂正を行わない時の全セル故障のFIT数に等しい。

$$\underline{Q}_I(\tau)_{II} = mn\alpha_a \lambda_h \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (6\text{-}30)$$

領域Ⅲ　：ソフトエラーが極めて多く，固定故障を生じる前に，ソフトエラーのみによる2ビット誤りが生じる。従って，近似式は次式で与えられる。なお，式中の1／Aは，2番目のソフトエラーが同一読みだしアドレスに生じる確率であると考えることができる。

$$\underline{Q}_I(\tau)_{III} = \tfrac{1}{2} mn^2 \lambda_s^2 \tau / A \quad \cdots\cdots\cdots\cdots\cdots\cdots (6\text{-}31)$$

式（6-29）～式（6-31）と，厳密式（6-15）との比較を図6-6 に示す。記憶素子は64Kビット素子であり，n＝72，記憶容量は16MB（m＝32）である。近似は，τ＝1月に較べると，τ＝1週間のほうがよく，この場合には，一部を除いて，10%以下の

誤差に納まっている。各ソフトエラー率λ_sに対して，近似式，$\underline{Q}_1(\tau)_{\mathrm{I}}$，$\underline{Q}_1(\tau)_{\mathrm{II}}$，$\underline{Q}_1(\tau)_{\mathrm{III}}$のうち，どれを使用するかは，以下の条件による。

$\underline{Q}_1(\tau)_{\mathrm{I}} < \underline{Q}_1(\tau)_{\mathrm{II}}$ であれば，$\underline{Q}_1(\tau)_{\mathrm{I}}$を

$\underline{Q}_1(\tau)_{\mathrm{III}} > \underline{Q}_1(\tau)_{\mathrm{II}}$ であれば，$\underline{Q}_1(\tau)_{\mathrm{III}}$を

その他の中間領域であれば，$\underline{Q}_1(\tau)_{\mathrm{II}}$を使用する。

$\underline{Q}_1(\tau)_{\mathrm{I}}$と$\underline{Q}_1(\tau)_{\mathrm{II}}$との区切りは，正確には，$\lambda_s = 1/n\tau \quad \lambda_h$であり，この例では，$\lambda_s / \lambda_h = 10^2$程度となる。

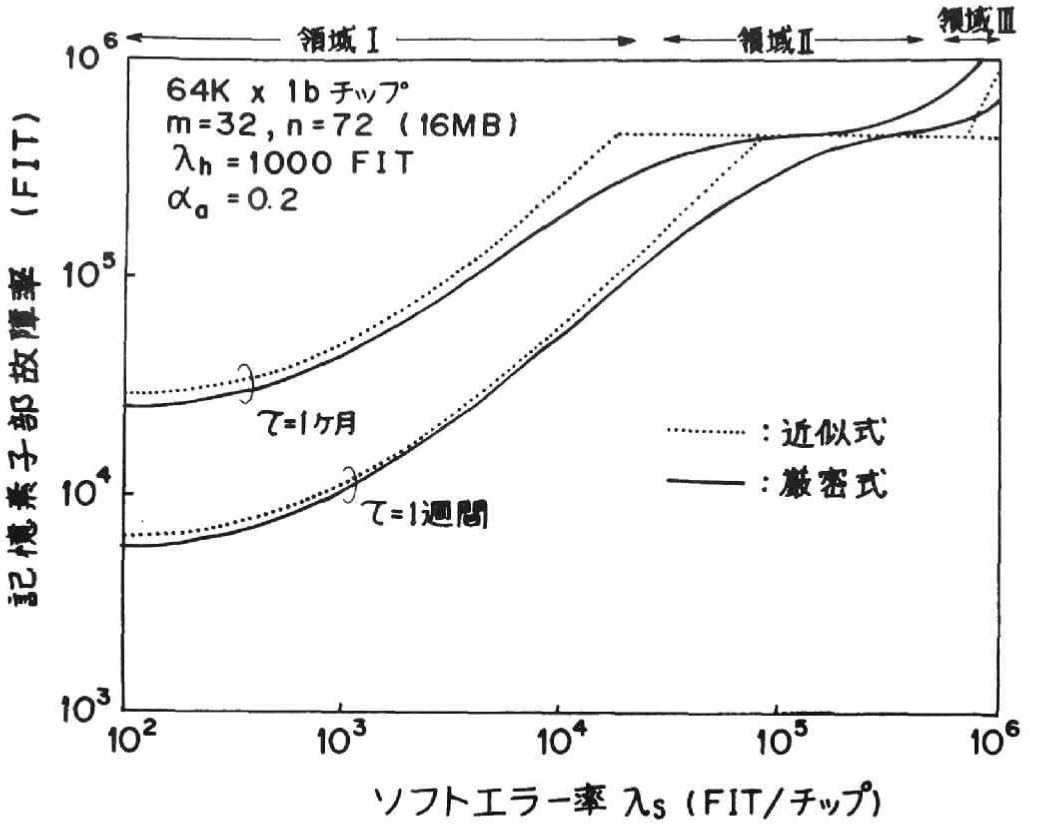


図6-6　故障率の算出(ECC のみの時)

（B）パトロール走査方式における訂正不能故障率近似式

前節と同様に，λ_s / λ_h の値にて，2領域に区分して考える。

（I）$\lambda_s / \lambda_h < 10^2$ の領域

ここでは，全セル固定故障1ビットと他の故障1ビットによる2ビット誤りが支配項となる。

（II）$\lambda_s / \lambda_h > 10^2$ の領域

全セル固定故障が生じると，直後にソフトエラーが生じて，訂正不能となる。

各領域における厳密式（6-18）の近似式は次の様に与えられる。

領域Ⅰ：　式（6-19）をexp（x）のマクローリン級数展開の2次の項まで取って，次式を得る。

$$p_2(\tau)_I = \frac{1}{2} m n^2 \alpha_a \lambda_h \left((1+\alpha_c) \lambda_h + \lambda_s \right) \tau^2$$

$$= m n^2 \alpha_a \lambda_h \left(\lambda_h + \frac{1}{2} \lambda_s \right) \tau^2 \quad \cdots\cdots\cdots\cdots\cdots\cdots \quad (6\text{-}32)$$

従って，式（6-18）は次式で近似できる。パトロール走査を行っても，主要な故障モードは全セル固定故障とソフトエラーによる2ビット誤りである。

$$Q_2(\tau)_I = m n^2 \alpha_a \lambda_h \left(\lambda_h + \frac{1}{2} \lambda_s \right) \tau \quad \cdots\cdots\cdots\cdots\cdots\cdots \quad (6\text{-}33)$$

領域Ⅱ　：全セル固定故障が生じると，直後に訂正不能となる。近似式は次式で与えられる。

$$Q_2(\tau)_{II} = m n \alpha_a \lambda_h \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (6\text{-}34)$$

図6-6 と同一緒元に対するパトロール走査方式の厳密式と近似式の計算結果を図6-7 に示す。

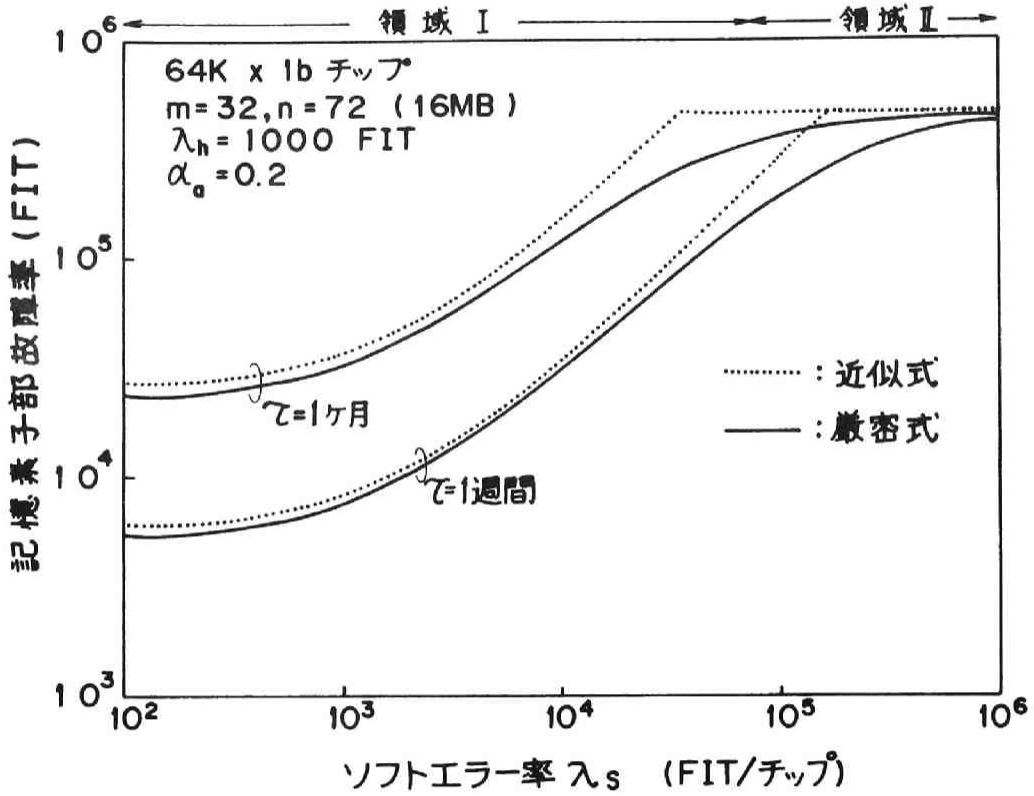


図6-7　故障率の算出(パトロ―ル走査方式)

（C）Erasure 訂正方式における訂正不能故障率近似式

厳密式（6-20）の近似式は次のように与えられる。

$$\underline{Q_3}(\tau) = mn^2 \alpha_a \lambda_h^2 \tau \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (6\text{-}35)$$

図6-8 に，厳密式と近似式の値を示す。記憶素子は６４Ｋビット素子であり，ｎ＝７２，容量は１６ＭＢ（ｍ＝３２）である。近似は極めて良好である。

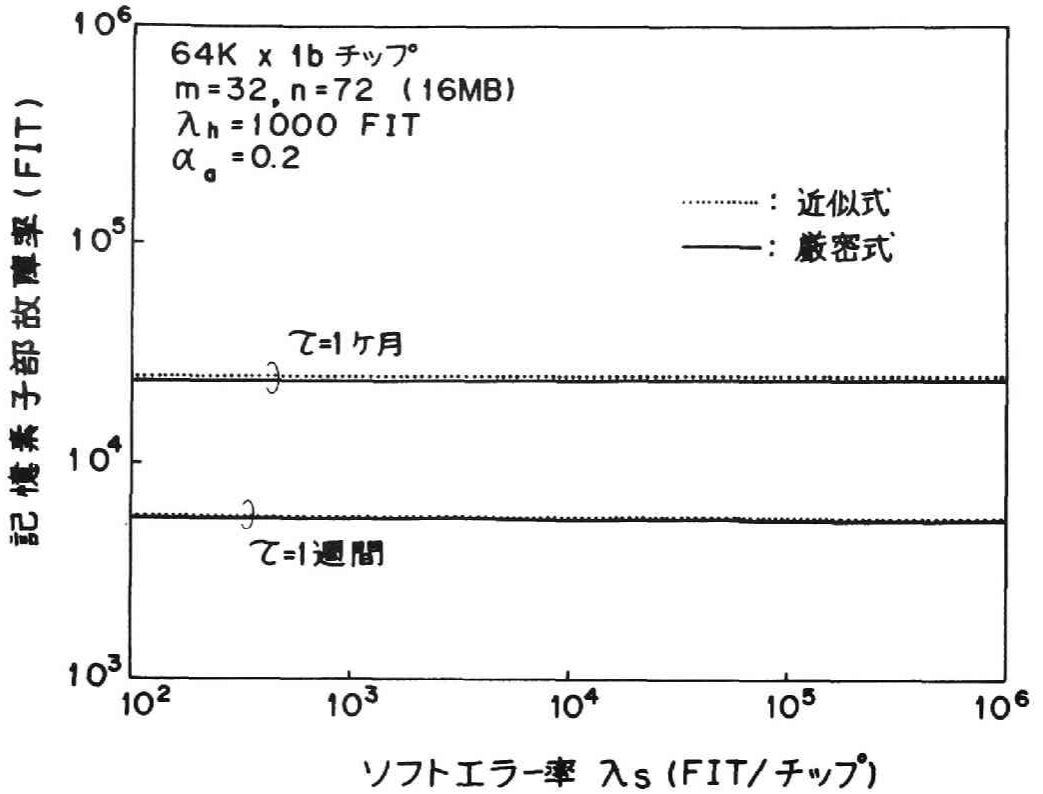


図6-8　故障率の算出(Erasure 訂正方式)

（D）　チップ交替方式における訂正不能故障率近似式

（D－1）交替契機があらゆる固定故障の時

チップ交替方式では，故障確率の算出に式（6-17）を使用する。式（6-17）の近似にあたっては領域を2つに分けた。同様に，本方式においても，以下の2領域を考える。

領域Ⅰ：λ_s / λ_h が小さい（$< 10^2$）とき。式（6-22）の近似式は次式で与えられる。

$$\underline{Q_{4A}}(\tau)_{I} = 1/3\, m^2 n^3 \alpha_a \lambda_h^2 (\lambda_h + \tfrac{1}{2}\lambda_s) \tau^2$$
$$= [mn\lambda_h \tau] \times [mn^2 \alpha_a \lambda_h (\lambda_h + \tfrac{1}{2}\lambda_s) \tau] \times 1/3 \quad \cdots\cdots (6\text{-}36)$$

上式の第1項（$mn\lambda_h \tau$）は，定期保守間隔中に固定故障を生じて交替する確率。第2項（$mn^2 \alpha_a \lambda_h (\lambda_h + \tfrac{1}{2}\lambda_s) \tau$）はパトロール走査方式のみを適用した際の訂正不能故障率である。係数（1／3）は，多重故障の生じる順序による。

領域Ⅱ：　λ_s / λ_h が大きい（$> 10^2$）とき。λ_s が大きいため2個目の全セル固定故障が生じると直後に訂正される。近似式は次式で表される。

$$\underline{Q_{4A}}(\tau)_{II} = \tfrac{1}{2} m^2 n^2 \lambda_h^2 \alpha_a \tau \quad \cdots\cdots (6\text{-}37)$$

図6-9 に，近似式と厳密式による計算例を示す。諸元は図6-8 と同一である。$\tau = 1$ヶ月でやや近似が悪いのは，交替を生じる確率（$1 - \exp(-mn\lambda_h \tau)$）が0．82となり，1に近づくために，級数展開では，近似が不十分であるためであると考えられる。

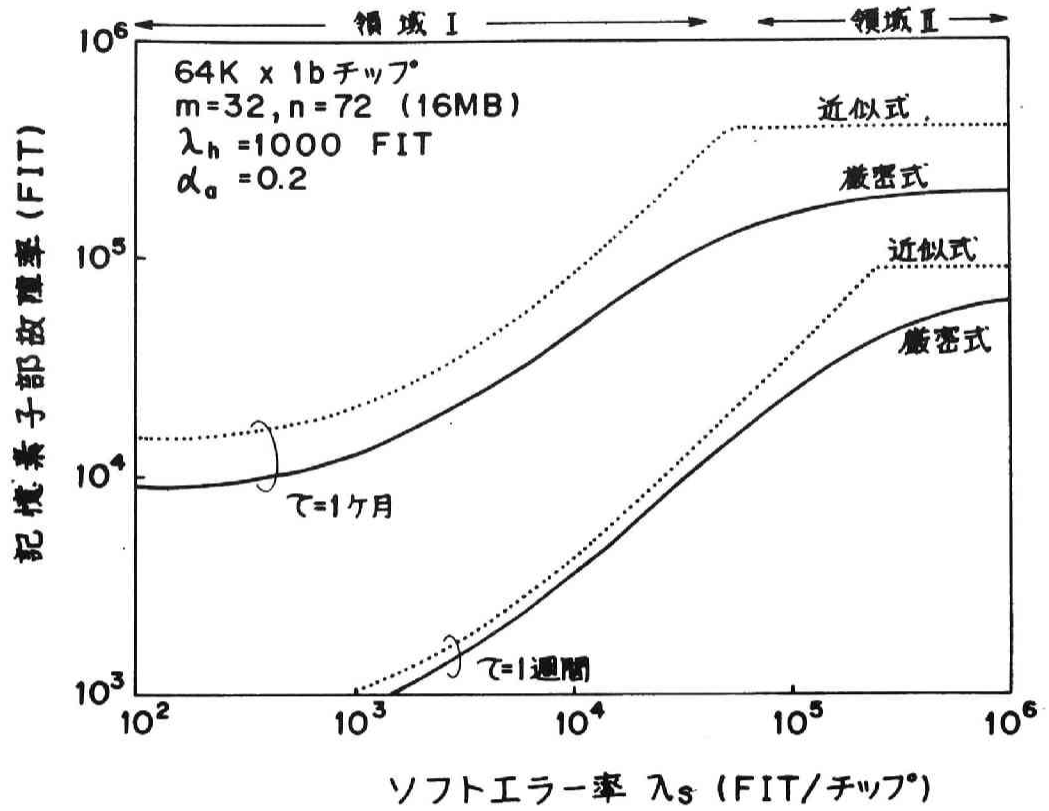


図6-9　故障率の算出(チップ交替方式D1)

（D－2）交替契機が全セル固定故障の時

λ_s / λ_h の値に応じて，以下の２領域に区分して考える。

領域Ⅰ　：λ_s / λ_h が小さい（$< 10^2$）時。

厳密式（6-24）の近似式は次式で与えられる。

$$\underline{Q}_{4B}(\tau)_I = \frac{mn^2 \alpha_c \alpha_a \lambda_h^2 \tau}{2} + \frac{(mn\alpha_a \lambda_h \tau)\{mn^2 \alpha_a \lambda_h (\lambda_h + \lambda_s/2)\tau\}}{3} \quad \cdots\cdots (6\text{-}38)$$

上式第１項（$mn^2 \alpha_c \alpha_a \lambda_h^2 \tau / 2$）は，全セル固定故障によって交替しようとしたが，すでに単独セル固定故障が存在していたために，交替に失敗する確率である。第２項中の第１項（$mn\alpha_a \lambda_h \tau$）は，全セル固定故障が生じて交替する確率，第２項中の第２項（$mn^2 \alpha_a \lambda_h (\lambda_h + \lambda_s/2)\tau\} / 3$）はパトロール走査方式のみが行われている時の故障率である。式（4-37）と式(4-38)を比較すると，交替契機があらゆる固定故障から，全セル固定故障に変わったこと，式（6-38）には，交替失敗の項が加わった点に差がある。

領域Ⅱ　：λ_s / λ_h が大きい（$> 10^2$）時。

λ_s が大きいため２番目の全セル固定故障が生じると，直後に訂正不能となる。従って，近似式は次の様に与えられる。

$$\underline{Q}_{4B}(\tau)_{II} = \frac{1}{2}(mn\alpha_a \lambda_h)^2 \tau \quad \cdots\cdots (6\text{-}39)$$

図6-10には，近似式と厳密式による計算例をしめす。緒元は図6-9 と同一であり，近似は良好である。

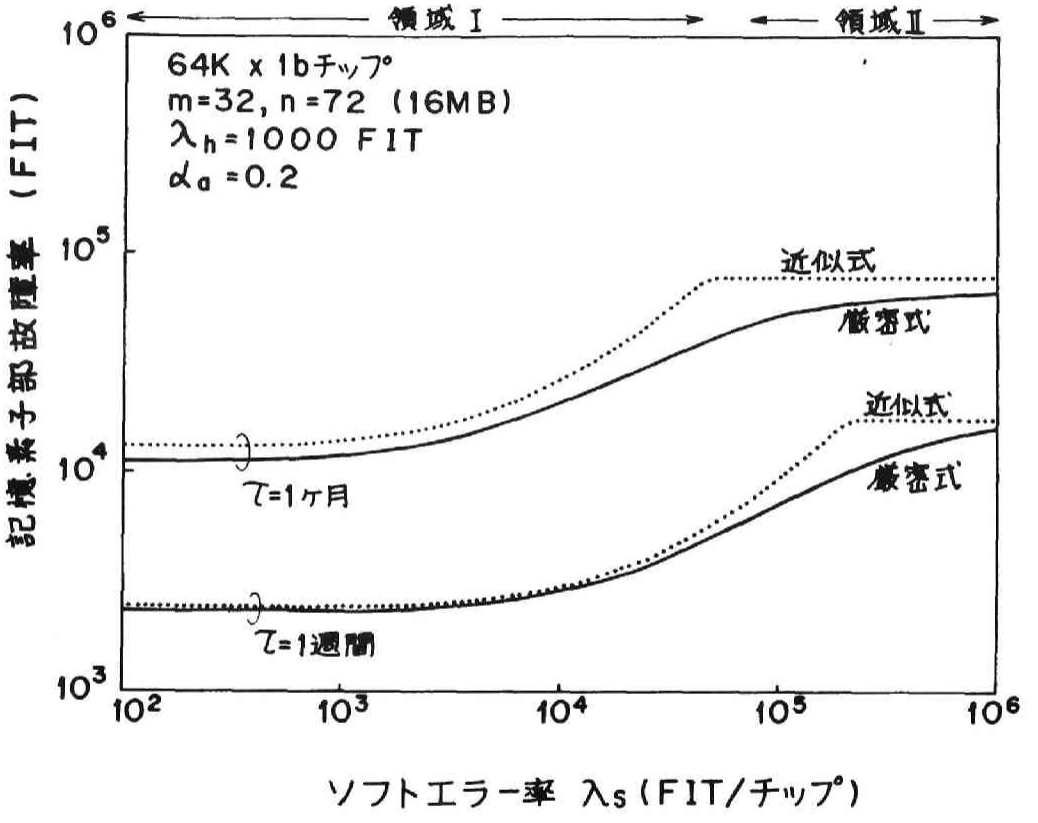


図6-10　故障率の算出(チップ交替方式D2)

６．３．４　　各方式の比較

本章で求めた近似式を表6-3 に示す。また，図6-11には，近似式による各方式の故障率を示す。記憶素子は６４Ｋビット素子，ｎ＝７２，容量は１６ＭＢ，定期保守間隔１週間の場合をしめす。

また，表6-4 には，各方式の誤り検出訂正用のハードウェア量をしめした。パトロール走査方式を適用した際のハードウェア量増加は，ＥＣＣのみの場合に比べて，３０％増にとどまっている。しかし，Erasure 訂正方式，チップ交替方式では，ハードウェア量が倍増している。このハードウェア量は，ｎ（ＥＣＣビット幅，表４ではｎ＝７２ビットである。）にほぼ比例する。従って，他のｎの値に対しても，ハードウェア量に関して同様の結論が得られる。

表6-4　誤り検出訂正のためのハードウェア

項番	手法	ＩＣ数（比）
1	ＥＣＣのみ	1
2	パトロール走査方式	１．３
3	Erasure 訂正方式	２．２
4	チップ交替方式	２．１

注）
汎用ＳＳＩ，ＭＳＩによる構成例。ｎ＝７２ビットとした。タイミング回路，診断用付加回路は含めていない。

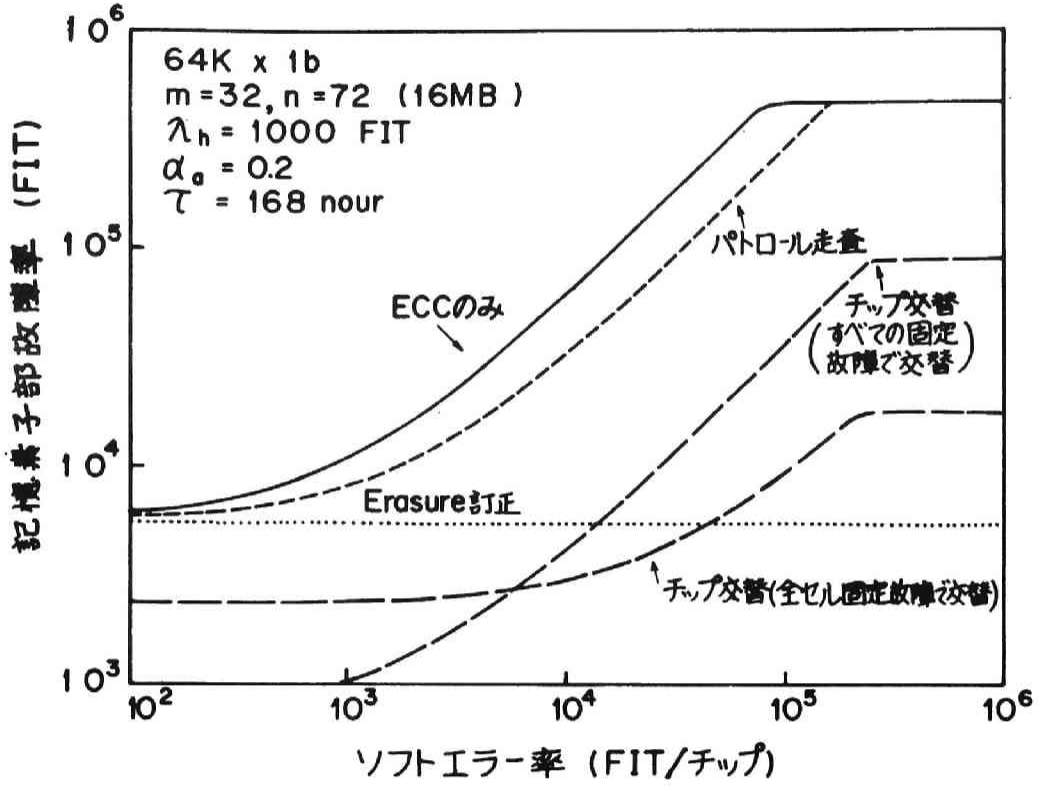


図6-11　各方式による記憶部故障率(近似値)

表６－３　主記憶装置記憶部故障率近似式一覧

項番	手法			近似式	
				λ_s が小さい（$\lambda_s / \lambda_h < 10^2$）とき	λ_s が大きい（$\lambda_s / \lambda_h > 10^2$）とき
1	ＥＣＣのみ			$\underline{Q}_{1,I} = mn^2 \alpha_a \lambda_h (\lambda_h + \lambda_s) \tau$	$\underline{Q}_{1,II} = mn\alpha_a \lambda_h$
2	パトロール走査方式			$\underline{Q}_{2,I} = mn^2 \alpha_a \lambda_h (\lambda_h + \frac{1}{2}\lambda_s) \tau$	$\underline{Q}_{2,II} = mn\alpha_a \lambda_h$
3	Erasure 訂正方式			$\underline{Q}_3 = mn^2 \alpha_a \lambda_h^2 \tau$	
4	チップ交替方式	A	あらゆる固定故障で交替する時	$\underline{Q}_{4A,I} = 1/3 mn \lambda_h \tau \cdot \underline{Q}_{2,I}$	$\underline{Q}_{4A,II} = \frac{1}{2} (mn\lambda_h)^2 \alpha_a \tau$
5		B	全セル固定故障で交替する時	$\underline{Q}_{4B,I} = 1/3 mn\alpha_a \lambda_h \tau \underline{Q}_{2,I} + \frac{1}{2}\alpha_c \underline{Q}_3$	$\underline{Q}_{4B,II} = \frac{1}{2} (mn\lambda_h \alpha_a)^2 \tau$

（注１）ＥＣＣのみの時は，ソフトエラーが更に多くなると，近似式は$\underline{Q}_{1,III} = \frac{1}{2} mn^2 \alpha_a \lambda_s^2 \tau / A$となる。
（注２）使用シンボル一覧　　Ａ：記憶素子のアドレス数

λ_h ：記憶素子の固定故障（１／時間）
λ_s ：記憶素子のソフトエラー率（１／時間）
ｎ　：誤り検出・訂正符号のビット幅（ビット）
ｍ　：語方向のメモリ素子数（ｍ　×　ｎ　で全メモリ素子数）
α_a ：記憶素子の固定故障が生じた時，それが全セル固定故障である割合
α_c ：記憶素子に固定故障が生じた時，それが，単独セル固定故障である割合
τ　：定期保守間隔

図6-11において，ＥＣＣのみのときと，パトロール走査の時を比較すると，パトロール走査方式の適用により，実効的に故障率がほぼ半減する。主記憶装置は，記憶部とこれを制御する制御論理部から構成される。制御論理部の故障率は，現状の主記憶装置では，２０万ＦＩＴ／１６ＭＢ程度と考えられる。従って，ソフトエラーによる主記憶装置故障率の増加を数十％に押さえるには，ＥＣＣのみの時のソフトエラー許容値は10^4 ＦＩＴ／チップ程度となる。このソフトエラー許容値は固定故障の約１０倍である。

チップ交替方式では，パトロール走査方式の故障率に対して，$1/3 \cdot mn\lambda_h \tau$倍（あらゆる固定故障を切り替え契機とする時），$1/3 \cdot mn\alpha_a \lambda_h \tau$倍（全セル固定故障を切り替え契機とする時）に等しい。ただし，全セル固定故障を切り替え契機とする時には，切り替え時にすでに単独セル固定故障が生じていて，切り替え失敗になる場合がある。このため，全セル固定故障を切り替え契機とする時には，λ_sが小さくなると，故障率が$\frac{1}{2}\alpha_c \alpha_a mn^2 \lambda_h^2$に漸近してゆく。チップ交替方式の適用により，ソフトエラー許容値は，ＥＣＣのみの時に比べて，１０～１００倍となる。また，Erasure 訂正方式でも，ソフトエラー許容値が大幅に増加する。

なお，現在の６４ＫビットＭＯＳ記憶素子は，チップコート等の対策により，ソフトエラーを含めて，素子あたりの故障率を１０００ＦＩＴ以下にすることが可能であると言われている。この様なソフトエラー発生率では，高信頼化手法として，ＥＣＣのみで十分であり，Erasure 訂正方式，チップ交替方式は不要であると考えられる。

6．4　結語

1．複数ビット出力記憶素子に対するバイト誤り検出・訂正符号の高信頼化効果を明らかにし，以下の結論を得た。ただし，1個の記憶素子が故障した時に，2ビット以上の複数ビットに誤りが生じる割合をβとする。

(1)ＳＥＣ－ＤＥＤ符号，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を適用した時には，記憶素子部の故障率は，誤り訂正を行わない時のβ倍となり，定期保守間隔にはほとんど依存しない。

(2)上記(1)の場合の非検出故障率は，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号では無視しうる値である。しかし，ＳＥＣ－ＤＥＤ符号では，無視しえない。

(3)ＳｂＥＣ符号，ＳｂＥＣ－ＤｂＥＤ符号を適用した時には，記憶素子部の信頼度が1桁～2桁向上する。しかし，ｂが4程度と小さい時には，ＳｂＥＣ符号は二重バイト誤りに対する検出率が低いため，実用的ではない。

(4)ＳｂＥＣ－ＤｂＥＤ符号では，非検出故障率は無視しうる。

2．ソフトエラー対策として提案されている，(1)ＳＥＣ－ＤＥＤ符号のみによる方式，(2)パトロール走査方式，(3)Erasure 訂正方式，(4)チップ交替方式の4種の高信頼化手法について，主記憶装置記憶部の故障率を求め，これから，故障率の近似計算手法を導出した。ただし，記憶素子は1ビット出力素子とした。近似式の厳密解にたいする誤差は，定期保守間隔1週間の場合で，１０％以下であり，装置信頼度設計上十分な精度を有している。

各手法の信頼度にたいする効果を比較し，次の結論を得た。

(1)ＳＥＣ－ＤＥＤ符号のみを適用した場合の記憶部故障率は，ソフトエラーを無視した故障率のほぼ，$1+\lambda_s/\lambda_h$（λ_sは記憶素子のソフトエラー率，λ_hは記憶素子の固定故障率）倍となる。

⑵ＳＥＣ－ＤＥＤ符号に，パトロール走査方式を併用した場合のソフトエラー許容値は，ＳＥＣ－ＤＥＤ符号のみの場合の２倍程度である。

⑶Erasure 訂正方式，チップ交替方式によれば，ソフトエラー許容値は，ＳＥＣ－ＤＥＤ符号のみの場合にくらべて，１０～１００倍増大する。但し，誤り訂正のためのハードウェア量が倍増する。

なお，本章で示した，信頼度近似設計手法は，６４Ｋビット記憶素子を用いて実用化した，日本電信電話公社仕様の大型計算機システムDIPS-11 5シリーズ主記憶装置の装置設計に使用し，部品側への信頼度目標値算定等に活用した。

（付録　６－１）　符号の多数ビット誤りに対する検出率

本付録では，代表的な半導体記憶装置用誤り検出・訂正符号について，多数ビット誤りに対する検出率を評価した結果を示す。多数ビット誤りに対する検出率を，解析的に求めることは難しい。このため，多数ビット誤りに対する検出率の評価には，シミュレーション・プログラムを作成して使用した。本プログラムでは，該当するすべての誤りパターンに対するシンドロームを作成し，このシンドロームを検出できるか否かを検査する。プログラムの詳細は，付録6-3 にしめす。

評価の対象とした符号を以下に列挙する。

(1)Hsiao による(72,64) ＳＥＣ－ＤＥＤ符号

(2)金田による(72,64) ＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号.....(図3-17)

(3) (80,64) Ｓ４ＥＣ符号

(144,128) Ｓ４ＥＣ符号

(3) (80,64)Ｓ４ＥＣ－Ｄ４ＥＤ符号.....(図4-8)

(144,128) Ｓ４ＥＣ－Ｄ４ＥＤ符号.....(図4-6)

ＳＥＣ－ＤＥＤ符号

Hsiao によるMinimum-Optimum Odd-weight-column ＳＥＣ－ＤＥＤ符号に対する，シミュレーション結果を付表6-1-1 に示す。この例では，データビット長が６４ビットであるが，データビット長が，３２ビット，１２８ビット等，２のべき上である時には，検出率には大差は無い。

付表6-1-1　Hsiao による(72,64) ＳＥＣ－ＤＥＤ符号の検出率

誤りのパターン		非検出パターン数	パターン総数	見逃す率 (%)	
ランダムな３ビット誤り		33568	59640	56.3	
４ビット誤り		8392	1028790	0.8	
単一バイト誤り	バイト内１ビット誤り	１ビット誤りとして訂正		0.	*1
	２ビット誤り	0	108	0.	33.3
	３ビット誤り	40	72	55.6	
	４ビット誤り	8	18	44.4	
二重バイト誤り	１ビット＆　１ビット	0	2448	0.	*2
	１ビット＆　２ビット	4504	7344	61.3	
	１ビット＆　３ビット	8	4896	0.2	
	１ビット＆　４ビット	856	1224	7.0	
	２ビット＆　２ビット	160	5508	2.9	30.0
	２ビット＆　３ビット	4084	7344	55.6	
	２ビット＆　４ビット	2	1836	0.1	
	３ビット＆　３ビット	2	2448	0.1	
	３ビット＆　４ビット	668	1224	54.6	
	４ビット＆　４ビット	28	153	18.3	

*1) バイト内２～４ビット誤りの生じる確率は同一とした時の，バイト誤りに対する誤訂正率。１ビット誤りについては除外する。

*2) 総誤りパターンに対する，検出出来ないパターン数を示す。

ＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号

本論文で提案した，２の繰り返し度を持つ奇数重み（７２，６４）ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号に対する，シミュレーション結果を付表6-1-2 に示す。二重バイト誤りに対する検出率が低いときには，実用上問題となるが，この付表6-1-2 からも，比較的高い検出率を持っていることがわかる。

付表6-1-2　(72,64) ＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号の検出率

誤りのパターン		非検出パターン数	パターン総数	見逃す率 (%)	
ランダムな３ビット誤り		32800	59640	55.0	
４ビット誤り		8200	1028790	0.8	
単一バイト誤り	バイト内１ビット誤り	１ビット誤りとして訂正		0.	*1
	２ビット誤り	0	108	0.	0.
	３ビット誤り	0	72	0.	
	４ビット誤り	0	18	0.	
二重バイト誤り	１ビット＆　１ビット	0	2448	0.	*2
	１ビット＆　２ビット	4168	7344	56.8	
	１ビット＆　３ビット	0	4896	0.	
	１ビット＆　４ビット	624	1224	51.0	
	２ビット＆　２ビット	96	5508	1.7	27.7
	２ビット＆　３ビット	3648	7344	49.7	
	２ビット＆　４ビット	0	1836	0.	
	３ビット＆　３ビット	128	2448	5.2	
	３ビット＆　４ビット	840	1224	68.6	
	４ビット＆　４ビット	16	153	10.4	

*1) バイト内２～４ビット誤りの生じる確率は同一とした時の，バイト誤りに対する誤訂正率。１ビット誤りについては除外する。

*2) 総誤りパターンに対する，検出出来ないパターン数を示す。

Ｓ４ＥＣ符号

ＳｂＥＣ符号の多数ビット誤り誤りに対する検出率を評価した結果を示す。ただし，半導体記憶装置として実用的なデータビット長を持つＳｂＥＣ符号の二重バイト誤りに対する検出率は，ｂ＝８の時には，解析的に求めることができる。従って，ここでは，ｂ＝４の場合について示す。データビット長ｋ＝６４ビットの場合の符号構成を付図6-1-1 に示し，シミュレーション結果を付表6-1-3 に示す。また，データビット長ｋ＝１２８ビットの場合の符号構成を付図6-1-2 に示し，シミュレーション結果を付表6-1-4 に示す。これらのデータビット長では，チェックビット長は１２ビットで良いが，この例では，いずれも１６ビットの冗長を持たせて検出率を高くした。なお，ランダムな３ビット，４ビットの誤りについては，誤りが単一のバイト内に生じて訂正できる場合は，検出できるものとして扱う。このランダムな３ビット，４ビットの誤りに対する検出率は，三重，四重のバイト誤りにたいする検出率のめやすと考えることができる。

ＳｂＥＣ－ＤｂＥＤ符号

ＳｂＥＣ符号の多数ビット誤りに対する検出率を評価した結果を示す。ただし，半導体記憶装置として実用的なデータビット長を持つＳｂＥＣ－ＤｂＥＤ符号の三重バイト誤りに対する検出率は，ｂ＝８の時には，解析的に求めることができる。従って，ここでは，ｂ＝４の場合について示す。データビット長ｋ＝６４ビットの場合の符号構成を付図6-1-3 に示し，シミュレーション結果を付表6-1-5 に示す。また，データビット長ｋ＝１２８ビットの場合の符号構成を付図6-1-4 に示し，シミュレーション結果を付表6-1-6 に示す。ランダムな３ビット，４ビットの誤りについては，誤りが単一のバイト内に生じて訂正できる場合は，検出できるものとして扱う。このランダムな３ビット，４ビットの誤りに対する検出率は，三重，四重のバイト誤りにたいする検出率のめやすと考えることができる。

$$H = \left[\begin{array}{ccccc:ccccc} I & I & I & I & I & 0 & 0 & 0 & 0 & 0 \\ I & T^1 & T^2 & T^3 & 0 & I & I & I & I & I \\ 0 & 0 & 0 & 0 & 0 & I & T^1 & T^2 & T^3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 0 & I & T^1 & T^2 & T^3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ I & I & I & I & I & 0 & 0 & 0 & 0 & 0 \\ I & T^1 & T^2 & T^3 & 0 & I & I & I & I & I \end{array}\right]$$

付図6-1-1　(80,64) S4 EC符号

付表6-1-3　(80,64) S4 EC符号の検出率

誤りのパターン		非検出パターン数	パターン総数	見逃す率 (%)	
ランダムな3ビット誤り　*1		4352	82080	5.30	
4ビット誤り　*1		48332	1581560	3.06	
二重バイト誤り	1ビット& 1ビット	384	3040	12.6	*2
	1ビット& 2ビット	648	9120	7.1	
	1ビット& 3ビット	392	6080	6.4	
	1ビット& 4ビット	112	1520	7.4	
	2ビット& 2ビット	712	6840	10.4	8.42
	2ビット& 3ビット	656	9120	7.2	
	2ビット& 4ビット	152	2280	6.7	
	3ビット& 3ビット	376	3040	12.4	
	3ビット& 4ビット	120	1520	7.9	
	4ビット& 4ビット	48	190	25.3	

*1) ランダムな誤りの中で，訂正可能なバイト誤りとなるものは除外する。
*2) 総誤りパターンに対する，検出できないパターンの割合を示す。

$$
H = \left[\begin{array}{ccccccccc:ccccccccc}
I & I & I & I & I & I & I & I & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
I & T^1 & T^2 & T^3 & T^4 & T^5 & T^6 & T^7 & 0 & I & I & I & I & I & I & I & I & I \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & T^1 & T^2 & T^3 & T^4 & T^5 & T^6 & T^7 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hdashline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & T^1 & T^2 & T^3 & T^4 & T^5 & T^6 & T^7 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
I & I & I & I & I & I & I & I & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
I & T^1 & T^2 & T^3 & T^4 & T^5 & T^6 & T^7 & 0 & I & I & I & I & I & I & I & I & I
\end{array}\right]
$$

付図6-1-2　(144,128) S4 EC符号

付表6-1-4　(144,128) S4 EC符号の検出率

誤りのパターン		非検出パターン数	パターン総数	見逃す率 (%)	
ランダムな3ビット誤り　*1		37256	487200	7.65	
4ビット誤り　*1		666292	17178840	3.88	
二重バイト誤り	1ビット& 1ビット	1760	10080	17.5	*2
	1ビット& 2ビット	4376	30240	14.5	
	1ビット& 3ビット	2880	20160	14.3	
	1ビット& 4ビット	744	5040	14.8	
	2ビット& 2ビット	3728	22680	14.3	15.24
	2ビット& 3ビット	4376	30240	14.5	
	2ビット& 4ビット	1072	7560	14.2	
	3ビット& 3ビット	1760	10080	17.5	
	3ビット& 4ビット	744	5040	14.8	
	4ビット& 4ビット	160	630	25.4	

*1) ランダムな誤りの中で，訂正可能なバイト誤りとなるものは除外する。
*2) 総誤りパターンに対する，検出できないパターンの割合を示す。

$$H=\left[\begin{array}{ccccc:ccccc:ccccc:ccccc}
I & I & I & I & I & 0 & 0 & 0 & 0 & 0 & T^2 & T^1 & I & T^{14} & 0 & T^{14} & I & T^1 & T^2 & 0 \\
T^{14} & I & T^1 & T^2 & 0 & I & I & I & I & I & 0 & 0 & 0 & 0 & 0 & T^2 & T^1 & I & T^{14} & 0 \\
T^2 & T^1 & I & T^{14} & 0 & T^{14} & I & T^1 & T^2 & 0 & I & I & I & I & I & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & T^2 & T^1 & I & T^{14} & 0 & T^{14} & I & T^1 & T^2 & 0 & I & I & I & I & I
\end{array}\right]$$

付図6-1-3　(80,64) S4 EC－D4 ED符号

付表6-1-5　(80,64) SbEC－DbED符号の検出率

誤りのパターン	非検出パターン数	パターン総数	見逃す率 (%)
ランダムな3ビット誤り　*1	932	82080	1.15
4ビット誤り　*1	13230	1581560	0.84

*1) ランダムな誤りの中で，訂正可能なバイト誤りとなるものは除外する。

$$H=\left[\begin{array}{ccccccccc:ccccccccc}
I & I & I & I & I & I & I & I & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
T^{11} & T^{12} & T^{13} & T^{14} & I & T^1 & T^2 & T^3 & 0 & I & I & I & I & I & I & I & I & I \\
T^3 & T^2 & T^1 & I & T^{14} & T^{13} & T^{12} & T^{11} & 0 & T^{11} & T^{12} & T^{13} & T^{14} & I & T^1 & T^2 & T^3 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & T^3 & T^2 & T^1 & I & T^{14} & T^{13} & T^{12} & T^{11} & 0 \\
\hdashline
T^3 & T^2 & T^1 & I & T^{14} & T^{13} & T^{12} & T^{11} & 0 & T^{11} & T^{12} & T^{13} & T^{14} & I & T^1 & T^2 & T^3 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & T^3 & T^2 & T^1 & I & T^{14} & T^{13} & T^{12} & T^{11} & 0 \\
I & I & I & I & I & I & I & I & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
T^{11} & T^{12} & T^{13} & T^{14} & I & T^1 & T^2 & T^3 & 0 & I & I & I & I & I & I & I & I & I
\end{array}\right]$$

付図6-1-4　(144,128) S4 EC－D4 ED符号

付表6-1-6　(144,128) S4 EC－D4 ED符号の検出率

誤りのパターン	非検出パターン数	パターン総数	見逃す率 (%)
ランダムな3ビット誤り　*1	5548	487200	1.14
ランダムな4ビット誤り　*1	175886	17178840	1.03

*1) ランダムな誤りの中で，訂正可能なバイト誤りとなるものは除外する。

（付録　６－２　）信頼度厳密式の導出

本付録では，信頼度厳密式の導出過程の概略をのべる。略号は，本文と同一である。

（A）ＥＣＣのみの時

ｔ＝０からｔ＝ｔ までに全セル固定故障が１記憶素子に生じる確率は次の様になる。

$$q_1(t) = 1 - e^{-\lambda_h t \alpha_a} \quad \cdots\cdots (付6\text{-}2\text{-}1)$$

時間ｔの間に，全セル固定故障がｎ個の記憶素子に生じない（q_2），１個生じる（q_3），２個以上生じる（q_4）確率は，それぞれ以下の式で与えられる。

$$q_2(t) = 1 - e^{-\lambda_h nt\alpha_a} \quad \cdots\cdots (付6\text{-}2\text{-}2)$$

$$q_3(t) = 1 - e^{-\lambda_h (n-1)t\alpha_a} \cdot (1 - e^{-\lambda_h t \alpha_a})\, n \quad \cdots\cdots (付6\text{-}2\text{-}3)$$

$$q_4(t) = 1 - q_2(t) - q_3(t) \quad \cdots\cdots (付6\text{-}2\text{-}4)$$

これらの確率のなかで，$q_4(t)$ の場合には訂正できない。$q_3(t)$ の場合には，他のｎ－１個の記憶素子中にソフトエラーを生じると，訂正不能である。他のｎ－１個の記憶素子に，ソフトエラーを生じる確率$q_5(t)$ は次式で与えられる。

$$q_5(t) = 1 - e^{-(n-1)t\,(\alpha_c \lambda_h + \lambda_s)} \quad \cdots\cdots (付6\text{-}2\text{-}5)$$

$q_2(t)$ の場合には，ｎ個の記憶素子にソフトエラーまたは単独セル固定故障による２ビット誤りを生じると，訂正できない。その訂正できない確率は，次式であたえらえる。

$$q_6(t) = 1 - \left\{ e^{-(\alpha_c \lambda_h + \lambda_s)\, t/A} + n\, e^{-(\alpha_c \lambda_h + \lambda_s)\,(n-1)\, t/A} \left(1 - e^{-(\alpha_c \lambda_h + \lambda_s)\, t/A}\right) \right\}^A \quad \cdots (付6\text{-}2\text{-}6)$$

以上の式から，訂正できない誤りを生じる確率は次式で与えられる。

$$p_1(t) = 1 - \{1 - q_2(t)\, q_6(t) - q_3(t)\, q_5(t) - q_4(t)\}^m \quad \cdots\cdots\cdots\cdots\cdots\cdots (付6\text{-}2\text{-}7)$$

上記の式を変形して，容易に式 (6-16) を得る。

(B) パトロール走査方式の時

T＝0からT＝t までにnの記憶素子には，全セル固定故障はなく，かつ，単独セル固定故障が生じる確率は次の様になる。

$$q_9 = (1 - e^{-\lambda_h nt\alpha_c})\, e^{-\lambda_h nt\alpha_a} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (付6\text{-}2\text{-}8)$$

Δtの間に全セル固定故障の生じる確率は，次式で与えられる。

$$q_{10} = n\alpha_a\, \lambda_h\, \Delta t \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (付6\text{-}2\text{-}9)$$

T＝t＋Δtから T＝τまでに何らかの誤りを生じる確率は，次式で与えられる。

$$q_{11} = 1 - e^{-\{(\lambda_h + \lambda_s)(\tau - t)\}(n-1)} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (付6\text{-}2\text{-}10)$$

上記の条件から，期間τの間に，訂正できない誤りを生じる確率は次式で与えられる。

$$p'(\tau) = \int_0^{\tau} n\alpha_a\, \lambda_h \left[(1 - e^{-\lambda_h nt\alpha_c})\, e^{-\lambda_h nt\alpha_a} + e^{-\lambda_h nt} \left\{ 1 - e^{-\{(\lambda_h + \lambda_s)(\tau - t)\}(n-1)} \right\} \right] dt \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (付6\text{-}2\text{-}11)$$

上記の積分は初等的に積分でき，本文式(6-17)をえる。

(C) Erasure 訂正の時

Erasure 訂正の場合には，パトロール走査を併用するものとする。これにより，ソフト

エラーが蓄積することはなくなる。従って，記憶部の故障は，固定故障のみを考えればよい。まず，時間tの間に，1記憶素子に，全セル固定故障，単独セル固定故障を生じる確率は以下の式で与えられる。

$$q_{30} = 1 - e^{-\lambda_h t \alpha_a} \quad \cdots\cdots (付6\text{-}2\text{-}12a)$$

$$q_{31} = 1 - e^{-\lambda_h t \alpha_c} \quad \cdots\cdots (付6\text{-}2\text{-}12b)$$

訂正できない2ビット誤りの原因となるのは，1）全セル固定故障と全セル固定故障，2）全セル固定故障と単独セル固定故障の重なり，であり，3重以上の故障についても，少なくとも1個，全セル固定故障があれば，訂正不能となる。これから，式（6-21）をえる。

（D）チップ交替方式(D-1)

あらゆる固定故障を交替契機とするから，時間T＝0からT＝tまでのあいだには，固定故障が，すべてのＥＣＣグループに存在しない。その確率は以下の式で与えられる。

$$q_{40} = e^{-mnt\lambda_h} \quad \cdots\cdots (付6\text{-}2\text{-}13)$$

また，Δtの間に，固定故障が生じる確率は，次式で与えられる。

$$q_{41} = n \lambda_h \Delta t \quad \cdots\cdots (付6\text{-}2\text{-}14)$$

従って，ただちに式(6-23)をえる。

チップ交替方式(D-2)

まず，時刻T＝0からT＝tまでの間に全セル固定故障が生じていない確率は次式で与えられる。

$$q_{51}(t) = e^{-mn \alpha_a \lambda_h t} \quad \cdots\cdots (付6\text{-}2\text{-}15)$$

Δtの間に全セル固定故障を生じる確率は，次式で与えられる。

$$q_{51} = n \lambda_h \Delta t \quad \cdots\cdots (付6\text{-}2\text{-}16)$$

T＝t＋Δtから T＝τまでの間にシステムダウンを生じる確率は，それぞれ次式で与えられる。

(1)交替したＥＣＣグループに，時刻T＝0からT＝tまでの間に単独セル固定故障の存在する確率。

$$q_{53} = 1 - e^{-n \lambda_h t \alpha_c} \quad \cdots\cdots (付6\text{-}2\text{-}17)$$

(1)交替したＥＣＣグループに，時刻T＝0からT＝tまでの間に単独セル固定故障の存在しない確率。

$$q_{54} = e^{-n \lambda_h t \alpha_c} \quad \cdots\cdots (付6\text{-}2\text{-}18)$$

（a）上式の条件のもとで，交替したＥＣＣグループに訂正不能な誤りの生じる確率は以下の式で与えられる。ただし，ここで，p'(τ) は式(6-17)で与えられる。

$$q_{55} = p'(\tau - t) \quad \cdots\cdots (付6\text{-}2\text{-}19)$$

（b）上式の条件のもと，交替したＥＣＣグループ以外のＥＣＣグループで，訂正できない誤りを生じる確率は以下の様に与えられる。

（b－1）当該ＥＣＣグループに，T＝0からT＝tまでの間に，単独セル固定故障の生じていた時に，訂正出来ない誤りを生じる確率は以下の式であたえられる。

$$q_{56} = (1 - e^{-n \alpha_c \lambda_h t})(1 - e^{-(n-1) \alpha_c \lambda_h (\tau - t)}) \quad \cdots\cdots (付6\text{-}2\text{-}20)$$

（b－2）当該ＥＣＣグループに，T＝0からT＝tまでの間に，単独セル固定故障の存在しない時に，訂正出来ない誤りを生じる確率は以下の式であたえられる。

$$q_{57} = e^{-n\,\alpha_c\,\lambda_h\,t} \cdot p'(\tau - t) \quad \text{.....(付6-2-21)}$$

以上の式から，式 (6-25) は次の式を変形することにより得られる。

$$p_{4B} = \int_0^{\tau} q_{51}(t)\ q_{52}(t) \left\{ q_{53}(t) + q_{54}(t) \left\{ 1 - \{1 - q_{55}(t)\} \cdot \{1 - q_{56}(t) - q_{57}(t)\}^{m-1} \right\} \right\} dt \quad \text{.....(付6-2-22)}$$

（付録６－３）符号評価プログラムＥＣＣＥＳ

本付録では，バイト誤り検出・訂正符号の機能の確認，多数ビット誤りに対する検出能力を評価することを目的として作成した，符号評価プログラムＥＣＣＥＳについて，その概要を述べる。

訂正能力の確認

符号の訂正能力を確認するためには，⑴訂正可能な誤りパターンに対するシンドロームがすべて相異なり，かつ，⑵どのシンドロームも「ａｌｌ‘0’」ではない，ことを確認する必要がある。誤りシンドロームの個数をｍとすると，ｍが「ａｌｌ‘0’」ではないことを確認する為に，Ｏ（ｍ）の処理が必要であり，互いに相い異なる事を証明するのに，ｍ個のなかから２個ずつ取り出して比較する方法を使用すると，Ｏ（ｍ(ｍ-1)/2）の手間を必要とする。１ビット誤り訂正符号，単一バイト誤り訂正符号では，実用的な諸元の符号にたいしてはこの処理時間が問題となる事はない。ただし，２ビット以上の多数ビット誤りを訂正する符号，あるいは，二重バイト誤り訂正符号などの，高い誤り訂正機能を持つ符号では（後述する）比較回数の小さな Hash 法を使用する必要がある。

多数ビット誤りに対する検出率の評価

この評価項目では，ある多数ビットの誤りが生じた時に，その多数ビット誤りを検出できる確率がどの程度かを評価する。まず，以下のことに注意する必要がある。

⑴ある誤りパターンについて，その誤りに対するシンドロームを検出できるか否かは，一意に決まる。

⑵符号語中のどのビットに誤りが生じるかを予測することはできない。

従って，多数ビット誤りに対する検出率を評価するためには，すべての多数ビット誤りパターンに対してシンドロームを生成し，その各シンドロームを検出できるか否かを評価すれば良い。この際，シンドロームは以下の３種に区分される。

(1)誤訂正：シンドロームが訂正可能な誤りに対するシンドロームと一致するため，誤ったビットを訂正する。

(2)非検出：シンドロームが「ａｌｌ ‘0’」となり，符号能力では，誤りの発生を検知できない。

(3)検出　：シンドロームは，「ａｌｌ ‘0’」ではなく，しかも，訂正可能な誤りに対するシンドロームとも一致していない。

本論文では，上記の「非検出」を「誤訂正」に含めて考える。

以上の評価結果から，誤りの検出確率，誤訂正確率は以下の様に計算できる。

検出確率　$$P_d = \frac{\text{検出可能な誤りパターン数}}{\text{全誤りパターン数}}$$

誤訂正確率　$P_m = 1 - P_d$

$$= \frac{(\text{誤った訂正するパターン数}) + (\text{シンドロームがall'0'となるパターン数})}{\text{全誤りパターン数}}$$

誤りパターンの総数

（A）ランダム誤りの総数

符号ビット長をｎビットとする。この時，ランダム誤りの総数（N_p）は次の式で与えられる。

$$N_p = {}_nC_m \quad \text{………………………………………(付6-3-1)}$$

ただし，ｍは誤りビット数である。

このランダム誤りの総数は，ｍが多いと，莫大な数になる。付表6-3-1 には，代表的な符号ビット長に対するランダム誤りの総数を示す。後述するHash法を使用しても，評価可能なパターン数は，10の６～７乗が限度である。従って，評価可能なランダム誤りは４ビット程度が限度であることがわかる。現実には，５ビット以上のランダム誤りの検出確率が装置信頼度設計上，不要であり，問題とはならない。

付表6-3-1　ランダム誤りの総数

符号ビット長	ランダム2ビット	ランダム3ビット	ランダム4ビット	ランダム5ビット
31	465	4,495	31,465	169,911
32	496	4,960	35,960	201,376
64	2,016	41,664	63,537	7,624,512
72	2,556	59,640	1,028,790	13,991,544
80	3,160	82,160	1,581,580	24,040,016
137	9,316	419,220	14,043,870	373,566,942
144	10,296	487,334	17,178,876	481,008,528

（B）バイト誤りの総数

bビットのバイト内に生じる誤りパターン数は，

$2^b - 1$　通り

である。従って，符号ビット長Nbビットに対する2重のバイト誤りパターン数は，

$N_p = {}_nC_2 \ (2^b - 1)^2$　通り

であり，以下同様に，i重のバイト誤りのパターン数は，

$N_p = {}_nC_i \ (2^b - 1)^i$　通り

となる。計算機処理時間を考慮すると，二重のバイト誤り程度は，評価できるが，3重を越えると困難となる。現実には，三重のバイト誤りの検出率が装置の故障率に大きく寄与することはなく，また，ランダムな3ビット誤りの検出率を，三重のバイト誤りの検出率のめやすと考えることもできる。付表6-3-2,付表6-3-3 には，b＝2，4における二重バイト誤りのパターン数をまとめておく。

付表6-3-2　b＝2に対する2重のバイト誤りパターン数

符号ビット長	誤りパターン		
	1ビット，1ビット	1ビット，2ビット	2ビット，2ビット
72	2.520	2.520	640
76	2.812	2.812	703
140	9.660	9.660	2,415

付表6-3-3　b＝4に対する2重のバイト誤りパターン数

誤りパターン	符号ビット長			
	68	72	80	144
4ビット& 4ビット	136	153	190	630
4ビット& 3ビット	1,088	1,224	1,520	5,040
4ビット& 2ビット	1,632	1,836	2,280	7,560
4ビット& 1ビット	1,088	1,224	1,520	5,040
3ビット& 3ビット	2,176	2,448	3,040	10,080
3ビット& 2ビット	6,528	7,344	9,120	30,240
3ビット& 1ビット	4,352	4,896	6,080	20,160
2ビット& 2ビット	4,896	5,508	6,840	22,680
2ビット& 1ビット	6,528	7,344	9,120	30,240
1ビット& 1ビット	2,176	2,448	3,040	10,080

検索アルゴリズムの高速化

上述した様に，多重バイト誤り，多数ビット誤りの誤りパターン数は，極めて大きい。従って，誤りに対するシンドロームが訂正可能なシンドロームに一致するか否かをしらべるためには，極力高速のアルゴリズムを使用する必要がある。ＥＣＣＥＳでは，当初，シンドロームの検索にリニアサーチ法の採用を試みた。しかし，本手法は，記憶容量が少なくて良いものの，極めて多くの処理時間を必要とし，事実上，ランダム3ビットを越える評価が不可能となることが明らかとなった。このため，最終的には，Hash検索法を用いたシンドローム検索を使用した。

プログラム概要

ＥＣＣＥＳのプログラムの概要を付表6-3-4 にしめす。ＥＣＣＥＳは，DEMOS-E 及びVAX-11/780にPL/IとFORTRAN をもちいてインプリメントされており，そのソースコードステップ数は，約４．７Ｋである。　なお，Hash法を使用すると，DEMOS-E のTSS ユーザー領

域の関係上，チェックビット長を１２ビットまでしか採ることができない。したがって，DEMOS-E 用ＥＣＣＥＳは，ＴＳＳ用のチェックビット長が最大１２ビットに限定されたものと，バッチ用の最大１６ビットまでのチェックビット長を評価可能なものの２通りをもうけた。

付表6-3-4 ＥＣＣＥＳプログラム概要

プログラム名	訂正能力の評価	機能				
		ランダム２ビット誤り評価	単一バイト誤り検出評価	二重バイト誤り検出評価	ランダム３ビット誤り評価	ランダム４ビット誤り評価
ECCES1	１ビット誤り訂正能力	○	—	—	○	○
DECEST	２ビット誤り訂正能力	—	—	—	○	○
ECCES2	b=2 の単一バイト誤り訂正能力	—	—	○	○	○
ECCES4	b=4 の単一バイト誤り訂正能力	—	—	○	○	○
ECCESG	１ビット誤り訂正能力	—	○	○	○	○
D4EDEST*	b=4 の単一バイト誤り訂正能力	—	—	○	○	—

*)DEMOS-E TSS でも，チェックビット長１６ビットまでの評価が可能。ただし，評価速度は，リニアサーチ法を使用しているため，遅くなる。

第７章　バイト誤り検出・訂正符号の実用化

本章では，本論文で提案したバイト誤り検出・訂正符号の実用装置への適用例を述べる。バイト誤り検出・訂正符号の商用半導体記憶装置への適用は，著者の知る限りでは，本章でのべるもの以外には知られていない。

7.1　緒言

本論文では，半導体記憶装置に実用的な符号として，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号，及びＳｂＥＣ－ＤｂＥＤ符号の構成法を追求した。これらの符号は，日本電信電話公社により，符号化・復号化回路のＬＳＩ化が行われ，実用化されている。

本章では，実用化した符号の諸元，実用化結果についてのべる。実用化した符号は，いずれも，バイト長ｂ＝４の符号であり，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号については，データビット長ｋ＝３２ビット，ＳｂＥＣ－ＤｂＥＤ符号については，データビット長ｋ＝６４ビットの符号を実用化した。いずれの場合にも，符号にモジュラな性質を与え，符号化・復号化ＬＳＩのパターンコード数を小さく抑えている。実用化した符号の諸元を表7-1にしめす。

表7-1　商用半導体記憶装置へのバイト誤り検出・訂正符号の適用

符号種別	データビット長	バイト長	チェックビット数	適用装置
ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号*)	３２	４	８	日本電信電話公社商用システム*1)
ＳｂＥＣ－ＤｂＥＤ符号	６４	４	１６	日本電信電話公社商用システム*2)

*1) 記憶素子は１ビット出力素子である。
*2) 富士通株式会社のＭ３８０／３８２も同一の符号を使用している。

7.2 ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号

ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号に関しては，本論文で述べたように，種々の符号構成を開発した。これらの中で，実際にＬＳＩ化を行ったのは，３．４節で述べた奇数重み列ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号であり，４ビット出力の記憶素子に対する適用を狙いとして，バイト長としてｂ＝４を選んだ。本ＬＳＩは，日本電信電話公社の商用システムに適用することを目的として製作したものであり，データビット長は，ｋ＝３２ビットである。符号の選択にあたっては，以下の点を考慮した。

(1)バイト長ｂ＝４ビット，データビット長ｋ＝３２ビットに対しては，チェックビット長ｒの最小値は７ビットである。しかし，記憶素子が４ビット単位であることを考慮すると，チェックビット長ｒ＝８ビットとしても，経済上の差は生じない。従って，チェックビット長ｒ＝８とし，むしろ，符号化・復号化速度の最小化を主眼として，符号構成を選択することとした。

(2)更に，モジュラな性質を持つＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を採用して，符号化・復号化回路のＬＳＩパターン・コード数の削減を図ることとした。ただし，ＬＳＩのゲート規模を考慮すると，高々２程度の繰り返しで十分であり，４以上の大きな繰り返しは不要である。

(3)符号化・復号化速度を主眼として，符号構成を選択する場合には，Ｈ行列の重みだけではなく，復号化回路を構成する場合の，ゲートのファンアウト数を極力小さく抑える必要があることが実際の回路設計から明らかとなった。

以上の様な検討から，図3-9 の（６４，５６）ＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号を短縮化した，（４０，３２）ＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号を採用することとした。符号化・復号化回路は，約３００ゲート規模のシェットキＴＴＬ―ＬＳＩを用いて構成した。符号のＨ行列を図7-1 に示す。また，実際のＬＳＩの外観を図7-2 に示す。

1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1111 1111	1111 1111	1111 1111	1111 1111	
1111 1111	1111 1111	1111 1111	1111 1111		1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1

(注) 行列中の空白は「0」を示す。

図7-1 (40,32)SEC-DED-SbED符号

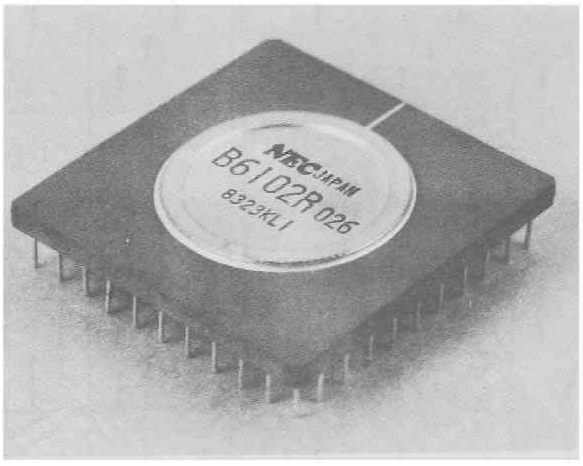


図7-2 (40,32)SEC-DED-SbED符号用符号化・復号化ＬＳＩ

7.3 ＳｂＥＣ－ＤｂＥＤ符号

近年，超大型汎用計算機の中には，３階層の半導体記憶装置を持つ機種が出現している。たとえば，富士通Ｍ－３８０／３８２は，半導体記憶装置として，通常のキャッシュ，主記憶の他に，それらの速度差をうめるための中間階層バッファ記憶を持ち，この中間階層バッファ記憶には，４ビット出力の記憶素子を採用している。中間階層バッファ記憶は，主記憶装置との間で，スワップ方式で，データを転送しており，バッファ記憶装置の高信頼化のため，ＳｂＥＣ－ＤｂＥＤ符号を採用している。

日本電信電話公社商用機においても，一部機種に，４ビットのバイト誤り訂正符号（ＳｂＥＣ－ＤｂＥＤ符号）を採用している。また，ＳｂＥＣ－ＤｂＥＤ符号の採用に当たっては，モジュラな符号を採用して，ＬＳＩのパターン・コード数の削減を図った。データビット長は，システム側から６４ビットが要求されており，記憶素子が４ビット出力のため，（８０，６４）Ｓ４ＥＣ－Ｄ４ＥＤ符号を採用した。また，ＬＳＩのゲート規模を考慮すると，高々２程度の繰り返しで十分であり，４以上の大きな繰り返しは不要である。以上の用な検討から，符号として，図4-4bの（８０，６４）Ｓ４ＥＣ－Ｄ４ＥＤ符号を採用した。符号のＨ行列をＧＦ（２）表現で，図7-3 に示す。符号は，そのＨ行列に２の繰り返し性を持つ。

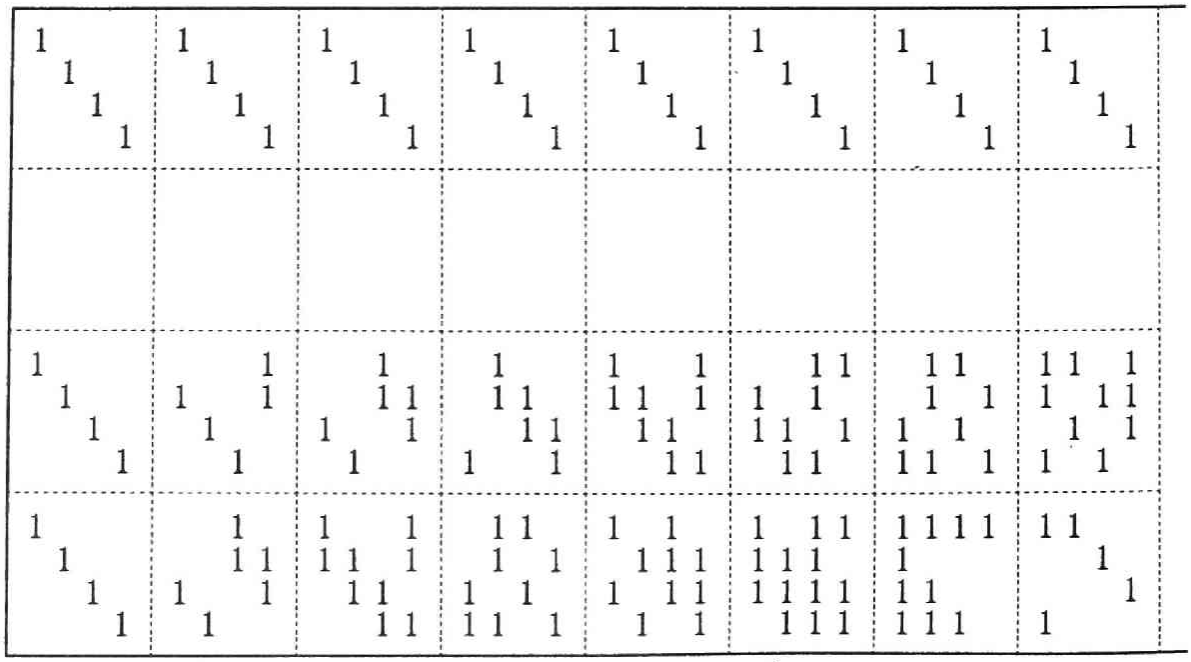
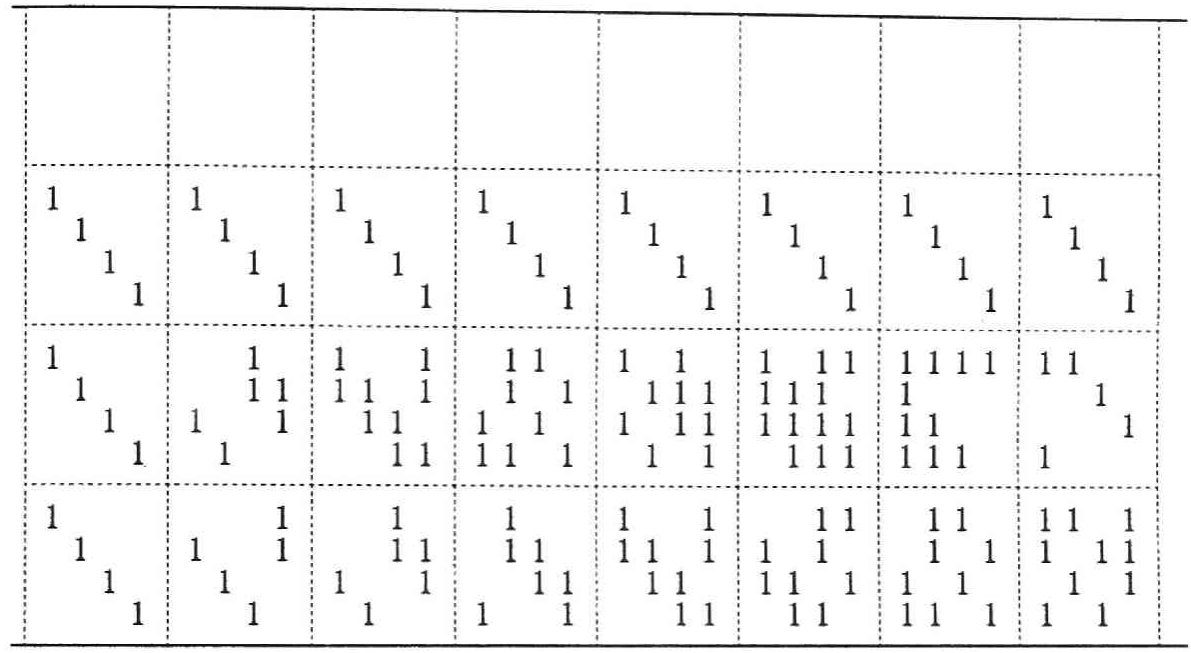
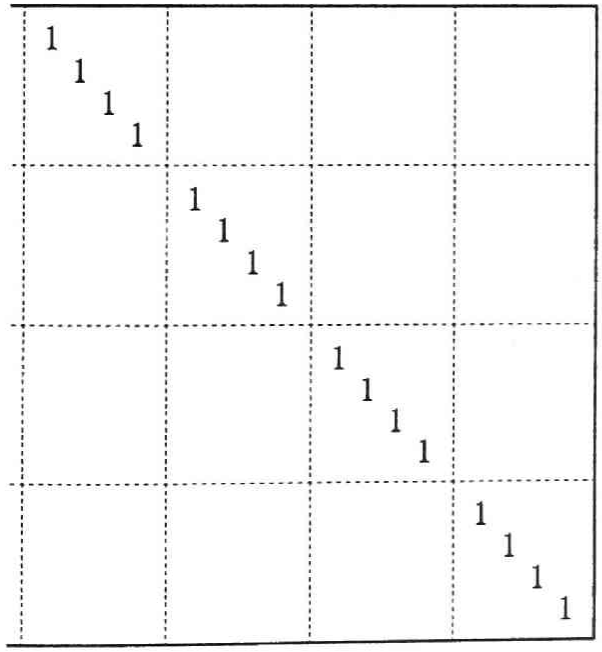


図7-3 (80,64) Ｓ４ＥＣ－Ｄ４ＥＤ符号（１／２）





(注) 行列中の空白は「0」を示す。

図7-3 (80,64) S4EC-D4ED符号 (2／2)

７．４　結語

本章では，バイト誤り検出・訂正符号の実用化結果をのべた。実用化した符号は，以下に示す２種類の符号であり，いずれも本論文で提案した符号である。符号化・復号化回路についてはＬＳＩ化を図り，ＬＳＩパターン・コード数を削減するために，モジュラな構成を持つ符号を採用した。これにより，本論文で提案した符号の実用性をシステム的観点からも実証できたと考えられる。

⑴バイト長ｂ＝４，データビット長ｋ＝３２のＳＥＣ－ＤＥＤ－Ｓ４ＥＤ符号
⑵バイト長ｂ＝４，データビット長ｋ＝６４のＳ４ＥＣ－Ｄ４ＥＤ符号

更に，実用化経験から，以下の点が明らかとなった。
⑴符号化・復号化回路のＬＳＩ化を容易とするモジュラな符号構成は効果的である。ただし，繰り返し度は，今日のＬＳＩゲート規模を考慮すると，あまり大きくとる必要はなく，２～４程度で十分である。
⑵モジュラな構成の符号化・復号化回路では，チェックビット／シンドロームの生成に，他ＬＳＩの出力結果を利用する。この構成は，ＬＳＩの入出力信号用セルの遅延が内部のゲート・セルの遅延に比べて極めて大きいため，速度上の要求条件が厳しい時には問題を生じる。従って，入出力信号用セルの遅延が問題となる場合には，チェックビット／シンドロームの生成のために，全符号ビット長のデータを１個のＬＳＩに入力する等の対策を必要とする。

なお，上記の２種類のＬＳＩとも，昭和５８年～昭和５９年より，日本電信電話公社の商用装置に導入され，順調に稼働している。

第８章　結論

本論文では，今後増大すると予想される複数ビット出力記憶素子を用いた半導体記憶装置の高信頼化に効果的な，バイト誤り検出・訂正符号の構成法に関して研究し，実用情報処理装置への適用に耐えうる，優れた経済性・性能を持つ符号構成を提案した。

本論文で明らかにした主要な結論を以下に示す。

⑴半導体記憶装置に適したバイト誤り検出・訂正符号の機能の明確化。

複数ビット出力記憶素子を用いた半導体記憶装置の高信頼化には，符号の多数ビット誤りにたいする検出能力，半導体記憶装置に対する高信頼化要求等を考慮すると，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号（１ビット誤り訂正・２ビット誤り検出・単一バイト誤り検出符号），ＳｂＥＣ－ＤｂＥＤ符号（単一バイト誤り訂正・二重バイト誤り検出符号）が適している。しかし，既存のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号，ＳｂＥＣ－ＤｂＥＤ符号は，チェックビット長が大きくて経済的に装置に適用出来ない（ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号），符号ビット長を自由に選択出来ない（ＳｂＥＣ－ＤｂＥＤ符号）等の欠点を持ち，特にバイト長ｂが小さい範囲では，半導体記憶装置に適用できない。

⑵ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号に関して，以下の新しい符号構成法を提案した。

（Ａ）単一バースト誤り検出・ＳＥＣ－ＤＥＤ符号

本符号は，既存のＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を特殊解として包含するが，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号よりも高い，単一バースト誤り検出能力を持つ。

（Ｂ）新しい奇数重み列ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号

本構成による符号は，後述する巡回性ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を構成するのに適している。また，チェックビット長が極端に大きな範囲では，ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号として現在知られている最も長い符号ビット長を持つ。

（Ｃ）最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号

符号化・復号化回路のゲート量が最小となる，最小重みＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を提案した。本符号は，実用的な諸元である，バイト長ｂ＝４，または，チェックビ

ット長ｒ＝ｂ＋２において，現在知られている最長の符号ビット長を持つ符号である。

（D）符号化・復号化回路のＬＳＩ化が容易な，巡回性ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号

符号化・復号化回路のＬＳＩ化に適した，巡回性ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を提案した。巡回性符号は，上記の（A）（B）（C）の符号から構成でき，巡回化による符号ビット長の短縮は僅小である。

以上の研究により，特に，バイト長ｂ＝４ビット，データビット長ｋ＝３２，６４，１２８ビット等の実用的な諸元にたいしては，現状のＳＥＣ－ＤＥＤ符号と同一のチェックビット長でＳＥＣ－ＤＥＤ－ＳｂＥＤ符号を適用することが可能となった。

⑶ＳｂＥＣ－ＤｂＥＤ符号に関して，以下の新しい符号構成法を提案した。

（A）任意の符号ビット長を選択可能なＳｂＥＣ－ＤｂＥＤ符号

本符号は，従来のＳｂＥＣ－ＤｂＥＤ符号とは異なり，任意の符号ビット長に対して符号を構成できる特徴を持つ。

（B）符号化・復号化回路のＬＳＩ化が容易な，巡回性ＳｂＥＣ－ＤｂＥＤ符号

符号化・復号化回路のＬＳＩ化に適した，巡回性ＳｂＥＣ－ＤｂＥＤ符号を提案した。

巡回度２のＳｂＥＣ－ＤｂＥＤ符号については，理論的な構成法を示した。４以上の巡回度を持つＳｂＥＣ－ＤｂＥＤ符号については，理論的に構成することは困難であるため，計算機を使用して，符号の生成と，符号化・復号化回路量／遅延最小化の観点から，符号の最適化を行った。

⑷誤り検出・訂正符号を適用した場合の，装置故障率の近似的算出法を明らかにした。

上記のバイト誤り検出・訂正符号を複数ビット出力記憶素子に適用した時の高信頼化効果，ソフトエラーの信頼度に与える影響などを，見通しよく算出する手法を提案した。本手法は，日本電信電話公社DIPS-11 情報処理装置５シリーズ主記憶装置の設計にあたり，装置目標信頼度の算出，部品信頼度要求値の算出に使用した。

なお本論文で提案した符号（ＳＥＣ－ＤＥＤ－ＳｂＥＤ符号，ＳｂＥＣ－ＤｂＥＤ符号）は，符号化・復号化回路がＬＳＩ化され，日本電信電話公社の商用システムに採用されている。

謝辞

本論文をまとめる過程において御指導・御鞭撻をいただいた京都大学工学部情報工学教室・矢島脩三教授，京都大学工学部電子工学教室・池上文夫教授，同，吉田進助教授に厚く感謝致します。また，本研究に関して，御鞭撻をいただいた，京都大学工学部電子工学教室・板谷良平教授に深く感謝いたします。また，本研究を進める過程で，符号理論に関して御指導を頂いた横浜国立大学情報工学教室・今井秀樹助教授に感謝の意を表します。

本論文は，筆者が日本電信電話公社・武蔵野電気通信研究所で行った研究成果をまとめたものであります。

本研究の機会を与えられ，終始暖かい御指導を頂いた，横河ヒューレットパッカード株式会社・寺島涼博士（前，武蔵野電気通信研究所・電子装置研究部長），武蔵野電気通信研究所・山田正計・電子装置研究部長，松下電子工業株式会社・飯田麒一郎氏（前，武蔵野電気通信研究所・電子装置研究室長），武蔵野電気通信研究所・伊東裕夫・電子装置研究室長，ならびに本論文を執筆する機会を与えていただいた武蔵野電気通信研究所・情報通信基礎研究部・畔柳功芳部長，同，吉田裕統括調査役，同，山下紘一・第一研究室長にこころから御礼を申し上げます。また，本研究を行う契機を与えていただき，御指導を頂いた，武蔵野電気通信研究所・藤原英二博士に対して，深い感謝の意を表します。

さらに，本研究成果の実用装置への適用に関して御指導を頂いた，福田晴幸調査役（武蔵野電気通信研究所），小柳津育郎調査役（横須賀電気通信研究所）に，深く感謝いたします。また，青木克彦調査役（厚木電気通信研究所）からも多くの示唆を頂きました。

本研究は，以上の数多くの方々の御指導・御配慮により達成されたものであり，ここに心から感謝の意を表します。

文献

(1) W.W.Peterson and E.J.Weldon,Jr.:"Error Correcting Codes", Second Edition MIT Press, 1971

(2) S.Lin :"An Introduction to Error Correcting Codes" Prentice-Hall, 1970

(3) E.Berlerkamp :"Algebraic Coding Theory" N.Y. McGraw-Hill, 1968

(4) 宮川, 岩垂, 今井:"符号理論" 昭晃堂,1973

(5) 嵩, 岩垂, 都倉, 稲垣:"符号理論" コロナ社,1975

(6) F.F.Sellers,Jr.,M.Y.Hsiao and L.W.Bearnson,"Error Detecting Logic for Degital Computers" McGraw-Hill, 1968

(7) T.R.N.Rao:"Error Coding for Arithmetic Processors" Academic Press, 1974

(8) W.F.Wakerly:"Error Detecting Codes, Self-Checking Circuits and Applications" North-Holland,1978

(9) R.W.Hamming :"Error Detecting and Error Corrrecting Code" BSTJ Vol.29, (1950)

(10)藤原・金田：”誤り検出訂正符号の応用”, 情報処理学会誌１９８２年４月号, 高信頼化技術特集号, 解説論文

(11)藤原・金田：”主記憶装置用バイト誤り検出訂正符号”, 研究実用化報告　第３０巻４号（１９８１）

(12) M.Y.Hsiao and J.T.Tou:"Application of Error-Correcting Codes in Computer Reliability Studies", IEEE Trans. on C., Vol.18,NO.3 (Aug. 1969)

(13)A.M.Patel and S.J.Hong:"Optimal Rectangular Code for High Density Magnetic Tapes" IBM J.Res.Develop. (Nov,1974)

(14)A.M.Patel:"Error Recovery Scheme for the IBM 3850 Mass Storage System" IBM J.Res. and Dev. Vol.24, No.1 (1980)

(15)T.Kameyama,K.Ono,T.Kawada and H.Nakanishi :"Large Capacity Magnetic Drum Storage System" Review of the Electirical Communication Laboratories,N.T.T. Vol.26,NO.1-2 (Jan-Feb. 1978)

(16) M.Y.Hsiao :"A Class of Optimal Minimum Odd-weight-column SEC-DED Code"

IBM J.Res. and Dev. Vol.14, No.7 (July, 1970)

(17)藤原・青木:"メインメモリの信頼性" 情報処理学会誌 Vol.14, NO.4 (1977)

(18) M.Y.Hsiao,D.C.Bossen and R.T.Chien :"Orthogonal Latin Square Codes" ,IBM J. Res. and Dev. Vol.14, No.7(July, 1970)

(19)上林・稲垣・矢島:"LSI メモリにおける永久故障の性質を利用した多重誤り訂正方式" 電子通信学会電子計算機研究会資料 EC-77-66

(20)松沢・当麻：" 主メモリに適した多重誤り訂正の一方式" 電子通信学会論文誌Ｄ 第６０巻１０号（１９７７）

(21)堀口・森田 "並列メモリの二重誤り訂正の1 手法 " 電子通信学会電子計算機研究会資料 EC-75-42

(22)杉村・笠原・滑川：" 復号の容易な能率の良い多重誤り訂正符号の１構成法 " 電子通信学会論文誌Ａ　第６１巻１１号（１９７８）

(23)藤原・川上:"主記憶装置における２重誤り訂正の１手法―ＢＣＨ付による逐次訂正法" 電子通信学会電子計算機研究会資料 EC-76-20

(24)今井・上柳:" ２重誤り訂正ＢＣＨ符号の並列復号器について" 電子通信学会論文誌Ｄ 第６０巻９号（１９７７）

(25)S.J.Hong and A.M.Patel:"Double Error Correcting Method and System" U.S.Pat. NO.3714629

(26) D.C.Bossen :"b-Adjacent Error Correction",IBM J.Res. and Dev. Vol.14, No.7 (July, 1970)

(27) S.J.Hong and A.M.Patel :" A General Class of Maximal Codes for Computer Applications" IEEE Trans. on C., Vol.C-21, NO.12 (Dec. 1972)

(28)I.S.Reed and G.Solomon :"Polynomial Codes over Certain Finite fields" J.Soc.Ind.Appl.Math.,Vol.8 (June, 1960)

(29) D.C.Bossen,L.C.Chang and C.L.Chen :"Measurement and Generation of Error Correcting Codes for Package Failures" IEEE Trans. on C., Vol.C-27, NO.3 (March 1978)

(30) S.M.Reddy:"A Class of Linear Codes for Error Control in Byte-per-Card Organized Digital Systems ",IEEE Trans. on C., Vol.C-27,NO.5 (May, 1978)

(31)E.Fujiwara :"Error Control for Byte-per-Package Organized Memory Systems"

Trans. of IECE of JAPAN, Vol.E63,NO.2 (Feb. 1980)

(32)青木・金田・市毛・小林・川上：”ＤＩＳＰ－１１改良主記憶装置”，研究実用化報告第２９巻3号（１９８０）

(33)金田・福田・青木：”ソフトエラー救済を考慮した，半導体記憶装置構成”研究実用化報告，第３３巻１２号（１９８４）

(34)青木・金田・黒川・佐藤：”非同期式優先順位決定回路の高速化手法”，昭和５４年度電子通信学会総合全国大会１５４８

(35)D.C.Bossen,S.J.Hong,M.Y.Hsiao and A.M.Patel:"Modular Distributed Error Detection and Correction Apparatus and Method",U.S. PAT. NO.3825893

(36)E.Fujiwara:"A Modularized b-Adjacent Error Correction Memory Unit", Trans. of IECE of JAPAN, Vol.E60,NO.2 (Feb. 1977)

(37)C.L.Chen:"Error-Correcting Codes with Byte Error-Detection Capability", IEEE Trans. on C., Vol.C-32,NO.7 (July, 1983)

(38)L.A.Dunning and M.R.Varanasi :"Code Construction for Error Control in Byte Organized Memory Systems" IEEE Trans. on C., Vol.C-32, NO.6 (Jume 1983)

(39)金田：”パリティチェックによりバイト誤りを検出可能なＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の構成”，電子通信学会計算機研究会資料　１９８１年１１月

(40)金田：”グループ分割パリティチェック手法を用いたＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の構成”，電子通信学会論文誌Ｄ　第６６巻6号（１９８３）

(41)金田・藤原・青木・大西：”ポインタを用いた主記憶用ブロック誤り訂正手法”，昭和５４年度電子通信学会情報・システム部門全国大会３６６

(42)金田・藤原：”主記憶装置における巡回性バイト誤り検出符号”昭和５６年度電子通信学会総合全国大会１５２１

(43)E.Fujiwara and S.Kaneda :”Rotational Byte Error Detecting Codes for Memory Systems”,Trans IECE.of Japan, E-64,N.5, (Feb.1981)

(44)W.C.Carter and A.B.Wadia :"Design and Analysis of Codes and their Self-Checking Circuit Implementations for Correction and Detection of Multiple b-adjacent Errors",The 10th International Symposium on Falut-Tolerant Computing,Kyoto(1980)

(45)金田：”半導体記憶装置のためのバイト誤り検出符号”昭和５８年度

電子通信学会総合全国大会１５９６

(46)金田：”半導体記憶装置のためのバイト誤り検出符号”，電子通信学会情報理論研究会資料　１９８２年８月

(47)S.Kaneda　：”A Class of Odd-weight-column SEC-DED-SbED Codes for Memory System Applications”IEEE Trans. on C., (July, 1984)

(48)金田：”半導体記憶装置のためのバイト誤り検出符号”，電子通信学会論文誌Ｄ　第６７巻５号（１９８４）

(49)金田・藤原：”主記憶装置におけるＳｂＥＣ－ＤｂＥＤ符号について”昭和５２年度電子通信学会情報部門全国大会２９３

(50)藤原・金田：”主記憶装置におけるＳｂＥＣ－ＤｂＥＤ符号について”電子通信学会計算機研究会資料　１９７７年４月

(51)S.Kaneda and E.Fujiwara　：”Single Byte Error Correcting　Double Byte Error Detecting Codes for Memory Systems”, The 10th International Symposium on Falut-Tolerant　Computing ,Kyoto,　(Oct., 1980)

(52)S.Kaneda and E.Fujiwara　：”Single Byte Error Correcting　Double Byte Error Detecting Codes for Memory Systems”,IEEE Trans. on C.,C-31,NO.7,　(July,1982)

(53)J.K.Wolf :"Adding Two Information Symbols to Certain Nonbinary BCH Codes and Some Applications", BSTJ (Sept. 1969)

(54)A.K.Bhatt and L.L.Kinney :"A High Speed Parallel Encoder/Decoder for b-Adjacent Error-Checking Codes, 3rd USA-JAPAN Computer Conference, 1978

(55)伊藤・中道：”計算機実験により導出された半導体記憶装置用ＳｂＥＣ－ＤｂＥＤ符号”電子通信学会論文誌Ａ　第６６巻８号（１９８３）

(56)C.L.Chen:"Byte-Oriented Error-Correcting Codes for Semiconductor Memory Systems"　The 14th International Symposium　on　Falut-Tolerant　Computing

(57)W.C.Carter,G.B.Leeman,Jr. and A.B.Wadia:"Practical Length Single-bit Error Correction/Double-bit Error Detection Codes for Small values of b. IBM Tech. Dis. Bull., Vol.17,NO.7 (Dec. 1974)

(58)金田：”複数ビット出力記憶素子に対する誤り訂正符号の効果”昭和５３年度電子通信学会総合全国大会　１４１６

(59)Igor Bazovsky:"Reliability Theory and Practice" Prince-Hall, Englewood,USA.

(60)"Special Issue on Device Reliability" IEEE Trans. on E.D.,Vol.ED-26,NO.1 (Jan., 1979)

(61)ジェラルド・エイ・マレー:"エラー訂正方式" 特公昭51-28484

(62)金田・福田：" ソフトエラーを考慮した主記憶装置記憶部の信頼度設計 ", 昭和５６年度電子通信学会情報・システム部門全国大会　４９９

(63)金田・福田：" ソフトエラーを考慮した主記憶装置記憶部の信頼度設計 ", 電子通信学会計算機研究会資料　１９８１年９月

(64)小池・宮坂・谷口・高村・岡崎・池原:"半導体メモリシステムの信頼度改善の１手法" 昭和５６年度電子通信学会全国大会１５３１

(65)D.C.Bossen and M.Y.Hsiao :"A System Solution to the Memory Soft Error Problem" IBM J. of Res. and Dev. Vol.24,NO.3 (May 1980)

(66)C.W.Sundberg:"Erasure and Error Decoding for Semiconductor Memories" IEEE Trans. on C., Vol.C-27,NO.8 (Aug. 1978)

(67)S.K.Kwon and H.E.Harvey :"A Simulation Approach to the Reliability Analysis of Main Storage Systems" 12th Annual Simulation Symposium, (March 1979)

(68)S.K.Kwon : " Reliability Sensitivity of LSI Memory with Single Error Correction" Electronics to Microelectronics (1980)

(69)A.Rutledge:"Monte Carlo Generation of Order Statistics" IBM Technical Report TR22.1279

(70)W.C.Carter and C.E.McCarthy:"Implementation of an Experimental Fault-Tolerant Memory System",IEEE Trans. on C.,Vol.C-25,NO.6 (June 1976)

(71)小柳津・桑原・速水　他:"ＤＩＰＳ―１１　モデル１５，２５，３５，４５本体系装置の実用化 "　研究実用化報告　第３１巻８号（１９８２）

(72)S.S.Eatom et. al.:"A 100ns 64K Dynamic RAM using Redundancy Techniques" ISSCC 81 (1981)

(73)H.Yoshimura et. al.:"A 64Kbit MOS RAM"　ISSCC 78 (1978)

(74) Inmos Data Card, 松下電器貿易株式会社

(75)S.Suzuki,M.Nakao,T.Takeshima and M.Yoshida,:"A 128K Word x 8bDRAM" ISSCC 84

(1984)

(76)E.Baier et. al.:"A 256K NMOS DRAM",ISSCC 84(1984)

(77)金田，福田:"ソフトエラーを考慮した主記憶装置記憶部の信頼度設計"，

電子通信学会論文誌D　第６７巻９号（１９８４）..... 掲載予定

論文目録

主論文

(1)S.Kaneda and E.Fujiwara：”Single Byte Error Correcting　Double Byte Error Detecting Codes for Memory Systems ”, The 10th International Symposium on Falut-Tolerant Computing ,Kyoto, (Oct., 1980)

(2)E.Fujiwara and S.Kaneda：”Rotational Byte Error Detecting Codes for Memory Systems ”,Trans IECE of Japan, E-64,N.5, (Feb.1981)

(3)S.Kaneda and E.Fujiwara：”Single Byte Error Correcting　Double Byte Error Detecting Codes for Memory Systems ”,IEEE Trans. on C.,C-31,NO.7, (July,1982.)

(4) 金田：”グループ分割パリティチェック手法を用いたＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の構成”, 電子通信学会論文誌Ｄ　第６６巻６号（１９８３）

(5) 金田：”半導体記憶装置のためのバイト誤り検出符号”, 電子通信学会論文誌Ｄ第６７巻５号（１９８４）

(6) S.Kaneda :"A Class of Odd-weight-column SEC-DED-SbED Codes for Memory System Applications", IEEE Trans on C. Vol.C-33,NO.8(Aug. 1984)

(7) 金田, 福田:"ソフトエラーを考慮した主記憶装置記憶部の信頼度設計”, 電子通信学会論文誌Ｄ　第６７巻９号（１９８４）.....印刷中

(8) 青木・金田・市毛・小林・川上：”ＤＩＳＰ－１１改良主記憶装置”, 研究実用化報告　第２９巻３号（１９８０）

(9) 藤原・金田：”主記憶装置用誤り検出・訂正符号”, 研究実用化報告　第３０巻４号（１９８１）

(10)藤原・金田：”誤り検出訂正符号の応用”, 情報処理学会誌１９８２年４月号, 高信頼化技術特集号, 解説論文

(11)金田・福田・青木：”ソフトエラー救済を考慮した半導体記憶装置構成”, 研究実用化報告　第３３巻１２号（１９８４）.....印刷中

研究会資料

(1) 藤原・金田：”主記憶装置におけるＳｂＥＣ－ＤｂＥＤ符号について”
電子通信学会計算機研究会資料１９７７年４月

(2) 金田：”パリティチェックによりバイト誤りを検出可能なＳＥＣ－ＤＥＤ－ＳｂＥＤ符号の構成”，電子通信学会計算機研究会資料　１９８１年１１月

(3) 金田・福田：”ソフトエラーを考慮した主記憶装置記憶部の信頼度設計”，
電子通信学会計算機研究会資料１９８１年９月

(4) 金田：”半導体記憶装置のためのバイト誤り検出符号”
電子通信学会報理論研究会資料１９８２年８月

口頭発表資料

(1) 金田・藤原：”主記憶装置におけるＳｂＥＣ－ＤｂＥＤ符号について”
昭和５２年度電子通信学会情報部門全国大会２９３

(2) 金田：”複数ビット出力記憶素子に対する誤り訂正符号の効果”
昭和５３年度電子通信学会総合全国大会１４１６

(3) 青木・金田・黒川・佐藤：”非同期式優先順位決定回路の高速化手法”
昭和５４年度電子通信学会総合全国大会１５４８

(4) 金田・藤原・青木・大西：”ポインタを用いた主記憶用ブロック誤り訂正手法”，
昭和５４年度電子通信学会情報・システム部門全国大会３６６

(5) 金田・藤原：”主記憶装置における巡回性バイト誤り検出符号”
昭和５６年度電子通信学会総合全国大会１５２１

(6) 金田・福田：”ソフトエラーを考慮した主記憶装置記憶部の信頼度設計”，
昭和５６年度電子通信学会情報・システム部門全国大会４９９

(7) 金田：”半導体記憶装置のためのバイト誤り検出符号”
昭和５８年度電子通信学会総合全国大会１５９６

(8) 金田：”設計知識を用いた論理回路の双方向シミュレーション”，情報処理学会第28会全国大会4G-9（昭和59年3 月）