(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2006 年9 月14 日 (14.09.2006)

PCT

(10) 国際公開番号
WO 2006/095489 A1

(51) 国際特許分類:
G06F 12/16 (2006.01) *H03M 13/15* (2006.01)
G06F 11/10 (2006.01)

(21) 国際出願番号: PCT/JP2005/023545

(22) 国際出願日: 2005 年12 月16 日 (16.12.2005)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願2005-060818 2005 年3 月4 日 (04.03.2005) JP

(71) 出願人 (米国を除く全ての指定国について): 国立大学法人 東京工業大学 **(TOKYO INSTITUTE OF TECHNOLOGY)** [JP/JP]; 〒1528550 東京都目黒区大岡山 2-12-1 Tokyo (JP). 独立行政法人宇宙航空研究開発機構 **(JAPAN AEROSPACE EXPLORATION AGENCY)** [JP/JP]; 〒1828522 東京都調布市深大寺東町7-44-1 Tokyo (JP). 市川 愉 **(ICHIKAWA, Satoshi)** [JP/JP]; 〒3050047 茨城県つくば市千現2-1-1 独立行政法人宇宙航空研究開発機構 筑波宇宙センター内 Ibaraki (JP).

(72) 発明者; および

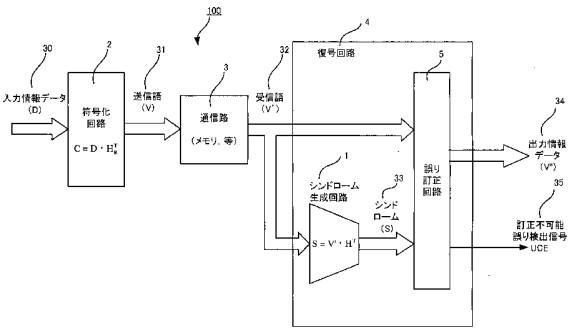
(75) 発明者/出願人 (米国についてのみ): 藤原 英二 **(FUJIWARA, Eiji)** [JP/JP]; 〒1528550 東京都目黒区大岡山2-12-1 国立大学法人東京工業大学内 Tokyo (JP). 鈴木 一克 **(SUZUKI, Kazuyoshi)** [JP/JP]; 〒1528550 東京都目黒区大岡山2-12-1 国立大学法人東京工業大学内 Tokyo (JP). 樫山 俊彦 **(KASHIYAMA, Toshihiko)** [JP/JP]; 〒1528550 東京都目黒区大岡山2-12-1 国立大学法人東京工業大学内 Tokyo (JP).

(74) 代理人: 安形 雄三 **(AGATA, Yuzo)**; 〒1070052 東京都港区赤坂２丁目１３番５号 Tokyo (JP).

[続葉有]

(54) Title: METHOD AND DEVICE FOR CORRECTING/DETECTING IN-BYTE MULTIPLE SPOTTY BYTE ERRORS WITH LIMITED NUMBER OF ERRONEOUS BYTES

(54) 発明の名称: 誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法及び装置



30 INPUT INFORMATION DATA (D)
2 ENCODING CIRCUIT
31 TRANSMISSION WORD (V)
3 COMMUNICATION PATH (MEMORY, ETC.)
32 RECEPTION WORD (V')
4 DECODING CIRCUIT
1 SYNDROME CREATION CIRCUIT
33 SYNDROME
5 ERROR CORRECTION CIRCUIT
34 OUTPUT INFORMATION DATA (V*)
35 UNCORRECTABLE ERROR DETECTION SIGNAL

(57) Abstract: There is provided a device for correcting/detecting in-byte spotty byte errors with limited number of erroneous bytes. The device has a function capable of controlling multiple in-byte spotty byte errors caused in a limited number of bytes. The device includes encoding means for creating a transmission word according to input information data and decoding means for inputting a transmission word which has caused an error in an information transmission path as reception word and correcting or detecting the error. The encoding mean adds inspection information created according to a parity inspection matrix expressing a spotty byte error control code and input information data, to the input information data, thereby creating a transmission word. The decoding means includes syndrome creation means for creating a syndrome of the reception word according to the parity inspection matrix and error correction means for correcting or detecting an error in the reception word according to the syndrome created by the syndrome creation means.

(57) 要約: 限られた数のバイトに生じる多重のバイト内複数スポッティバイト誤りを制御することができる機能を備えた誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置を提供する。 入力情報データを基に送信語を生成する符号化手段と、情報伝送路中で誤りが発生した送信語を受信語として入力して誤りを訂正または検出する復号手段とを備える誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置であって、符号化手段は、スポッティバイト誤り制御符号を表現するパリティ検査行列と、入力情報データとを基に生

[続葉有]

WO 2006/095489 A1

(81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:
— 国際調査報告書

２文字コード及び他の略語については、定期発行される各*PCT*ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

成した検査情報を入力情報データに付加することにより、送信語を生成し、復号手段は、パリティ検査行列を基に受信語のシンドロームを生成するシンドローム生成手段と、シンドローム生成手段により生成されたシンドロームを基に受信語の誤りを訂正または検出する誤り訂正手段とを備える。

明　細　書

誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法及び装置

技術分野

本発明は、スポッティバイト誤り訂正・検出方法及び装置に関し、より詳しくは、複数のビットをバイトとし、その複数のバイトから構成される語（ワード）に対して、１バイト内に複数のスポッティバイト誤りが生じる誤りを『バイト内複数スポッティバイト誤り』と称し、そのバイト内複数スポッティバイト誤りが制限された数のバイトに生じている場合に、その誤り、つまり、『誤りバイト数を制限したバイト内複数スポッティバイト誤り』を検出または訂正するために用いて好適な、誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法及び装置に関する。

背景技術

近年、情報化社会の進展や半導体技術の進歩等により、ディジタルデータの信頼性向上がますます重要なものになってきている。ディジタルデータの伝送または記録を正確に行おうとする場合、送信データが伝送経路中でノイズ等の影響を受けて誤って伝送されたり、素子の故障、あるいは記録媒体等の欠陥等により生じた誤りを検出・訂正する必要がある。この誤りの検出・訂正を実現するために、符号理論を基礎とした誤り訂正・検出符号が従来より種々に開発されてきている。

従来の誤り訂正符号として、例えば、ｂビット（ｂは２以上の整数）

の塊りをバイトと称し、１バイト中の任意の誤りを訂正する符号（S_bＥＣ符号と称する）として優れた Hong-Patel 符号が開示されている（例えば、非特許文献１参照）。

また、高速半導体メモリシステムにおいて、従来最もよく使用された１ビット誤り訂正・２ビット誤り検出符号を包含する特徴を有した奇数重み列バイト誤り訂正符号が開示されている（例えば、非特許文献２参照）。

さらに、１バイト誤りを訂正し、また２バイト誤りを検出する符号（S_bＥＣ－D_bＥＤ符号と称する）は、リードソロモン符号（Reed-Solomon 符号）、及びその改良した効率のよい符号として提案されており、多くの計算機システム等の主記憶装置にすでに採用されている（例えば、非特許文献３や非特許文献４参照）。

また、バイト誤りを検出する符号としては、１ビット誤りを訂正し、かつ２ビット誤りを検出するとともに、１バイトの誤りをも検出する符号（ＳＥＣ－ＤＥＤ－S_bＥＤ符号と称する）が発表され、現在、計算機システムの主記憶装置に多く適用されている（例えば、非特許文献５参照）。

なお、これら１９８０年代後半までの相互に関連した多くの符号研究・開発に関する具体的な内容は、非特許文献６において総括的に述べられている。

その後、新しく開発された高速半導体メモリ用の符号としては、１バイトの誤りを訂正し、かつ２ビット誤りを検出する符号（S_bＥＣ－ＤＥＤ符号と称する）が下記非特許文献７に、また、１バイトの誤りを訂正し、かつ１ビット誤りと１バイト誤りとの同時誤りを検出する符号（S_bＥＣ－（Ｓ＋S_b）ＥＤ符号と称する）が非特許文献８に提案され

ている。さらに、１バイトの誤りを訂正するとともに２ビットの誤りがあれば、これも訂正する符号（$S_bEC-DEC$符号と称する）が非特許文献９に開示されている。

特に、本発明に最も関連する符号として、具体的には、ｂビットからなるバイト内のｔビット（ｔ≦ｂ）までの誤りをｔ／ｂ誤りまたはスポッティバイト誤りと称し、この１スポッティバイト誤りを訂正する$S_{t/b}EC$符号、及び１スポッティバイト誤りを訂正し、かつバイト内ｔビットを越える１バイト誤りを検出する$S_{t/b}EC-S_bED$符号に関する発明が既に開示されている（特許文献１参照）。また、特に$S_{t/b}EC$符号については、非特許文献１０に開示されている。

また、１スポッティバイト誤りを訂正し、かつ２スポッティバイト誤りを検出する$S_{t/b}EC-D_{t/b}ED$符号、及び１スポッティバイト誤りを訂正し、かつ２スポッティバイト誤りを検出し、かつバイト内ｔビットを越える１バイト誤りを検出する$S_{t/b}EC-D_{t/b}ED-S_bED$符号に関する発明が既に出願されている（特許文献２参照）。

さらに、複数個のスポッティバイト誤りを考慮した場合、１バイト内に１スポッティバイト誤りのみが生じる誤りを『バイト内単一スポッティバイト誤り』とし、バイト内単一スポッティバイト誤りに対し、距離ｄを有する一般的な構成法に関する発明が既に出願されている（特許文献３参照）。

また、１バイト内に複数のスポッティバイト誤り（つまり、２以上のスポッティバイト誤り）が生じる誤りを『バイト内複数スポッティバイト誤り』とし、距離ｄを有する一般的な構成法に関する発明が既に出願されている（特許文献４参照）。

１９８０年代半ばまでは、半導体メモリ素子にはデータ１ビットの入

出力を有する素子が主に使用されたことから、１素子の誤りを訂正し２素子までの誤りを検出する１ビット誤り訂正・２ビット誤り検出符号（ＳＥＣ－ＤＥＤ符号と称する）が多く使用されていた。ところが、１９８０年代半ばからメモリ素子の高集積化ニーズに対応して、データ４ビットの入出力を有する素子が主流となりはじめ、バイト幅ｂ＝４ビットである前述のS_4EC-D_4ED符号や$SEC-DED-S_4ED$符号が主に使用されるようになってきた。さらに、１９９０年代半ばからはデータ８ビット、１６ビットの入出力を有する半導体メモリ素子が主流になりはじめた。

しかしながら、従来のS_bEC-D_bED符号のバイト幅ｂにｂ＝８またはｂ＝１６を適用する場合、全体の符号長に対する検査ビット数の占める割合がおよそ３０％から４０％と多大となって、符号化率が低下し、実用的に非常に大きな問題となっていた。

ところで、これらの半導体メモリ素子（ＤＲＡＭ素子、以下、単に素子とも称する）を使用したメモリ装置では、ノイズやアルファ粒子等によって一時故障が発生したり、ＤＲＡＭ素子自体が劣化して動作しなくなる固定故障が発生したりする。最近のＤＲＡＭ素子を用いた装置の８０％以上は一時故障であると言われており、特に、８ビット以上の多ビットデータ入出力を有するＤＲＡＭ素子ではバイト中の１、２、３ビット程度の比較的小さい数のビット誤りが大半であるとされている。

中でも比較的エネルギーレベルの低い電磁ノイズやアルファ粒子による一時誤り、及びメモリセルの固定故障を原因とする１ビット誤りが最も多く、最近では、前記ＤＲＡＭ素子を搭載したモバイル機器が多用されていることから、電磁環境が悪い中での使用も考慮しなければならなくなっている。また、高い高度を飛行する航空機や軍用機で用いられる

電子機器中のＤＲＡＭ素子では、宇宙線に起因するエネルギーレベルの高い中性子粒子等の衝突により、２ビット、３ビット程度の誤りを有する一時故障が発生する可能性が高くなっている。さらに、衛星通信や宇宙通信で用いられる宇宙機器に搭載するＤＲＡＭ素子では、エネルギーレベルの高い粒子との衝突によって大きなダメージを受けてしまうことを考慮する必要があり、この場合２ビット以上の誤りを考慮しなければならないことが知られている。

以上のようなビット誤りの発生状況が多岐にわたっていることを踏まえれば、パラメータｔ及びｂに任意の値を与えて構成できる前記$S_{t/b}EC-S_bED$符号は、８ビット以上のＤＲＡＭ素子を使用したメモリ装置に用いる場合に、非常に実用的な符号方式である。つまり、前記$S_{t/b}EC-S_bED$符号は、バイト幅ｂより小さなｔビット（このｔの値は、誤りの傾向を調査して設計者が任意に決めることが可能である）までの１素子の誤り（スポッティバイト誤り）を訂正することが可能な機能、及び発生確率としては小さいもののｔビットを超える１バイト誤りを検出することが可能な機能を備えることにより、従来のバイト単位で行うリードソロモン符号（ＲＳ符号）等の誤り制御符号と比較して、大幅に少ない数の検査ビットで誤りを訂正・検出することができる。

また、任意の２素子（隣接した２素子に限らない）にまたがる誤りに対して、前記$S_{t/b}EC-D_{t/b}ED-S_bED$符号は、８ビット以上のＤＲＡＭ素子を使用したメモリ装置に用いる場合に非常に実用的な符号化方式である。つまり、前記$S_{t/b}EC-D_{t/b}ED-S_bED$符号は、バイト幅ｂより小さなｔビットまでの１素子の誤りを訂正することが可能な機能（１スポッティバイト誤り訂正、$S_{t/b}EC$）、２素子にまたがるスポッティバイト誤りを検出することが可能な機能（２スポッ

ティバイト誤り検出、$D_{t/b}ED$）、及び発生確率としては小さいもののｔビットを超える１バイト誤りを検出することが可能な機能（１バイト誤り検出、S_bED）を備えた符号であり、従来のバイト単位で行う符号方式と比較して、大幅に少ない数の検査ビットでビット誤りを訂正・検出することができる。

さらに、２素子以上にまたがる誤りに対しては、前記バイト内単一スポッティバイト誤り及び前記バイト内複数スポッティバイト誤りに対し、理論上、一般にＧＦ（2^b）上で距離ｄを有するスポッティバイト誤り制御符号を構成することにより、その誤りを訂正・検出することができるようになっている。

しかしながら、誤りが一度に多くの素子にまたがり生じることは少なく、通常、限られた数の素子、つまり、制限された数のバイトに生じる。また、誤りが生じる素子の数は、高々２、３素子であると考えられる。

本発明は、上述のような事情に鑑みてなされたものであり、本発明の目的は、限られた数のバイトに生じた多重のバイト内複数スポッティバイト誤りを制御することができる機能を備えた、誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法及び装置を提供することにある。

また、本発明のもう１つの目的は、ｔ＝１としたとき、限られた数のバイトに生じたランダムビット誤りを制御することができる機能を備えた、誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法及び装置を提供することにある。

更に、本発明のもう１つの目的は、発生確率が小さいもののバイト内複数スポッティバイト誤りの訂正能力を超える単一バイト誤りを検出す

る機能を備えた、誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法及び装置を提供することにある。

発明の開示

本発明は、入力情報データを基に送信語を生成する符号化手段と、情報伝送路中で誤りが発生した前記送信語を受信語として入力して前記誤りを訂正または検出する復号手段とを備える誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置に関し、本発明の上述目的は、前記符号化手段は、スポッティバイト誤り制御符号を表現するパリティ検査行列と、前記入力情報データとを基に生成した検査情報を前記入力情報データに付加することにより、前記送信語を生成し、前記復号手段は、前記パリティ検査行列を基に前記受信語のシンドロームを生成するシンドローム生成手段と、前記シンドローム生成手段により生成されたシンドロームを基に前記受信語の誤りを訂正または検出する誤り訂正手段とを備えることにより、或いは、前記入力情報データが、b（bは2以上の整数）ビットを1バイトとし、複数のバイトから構成される場合に、ここで、バイト内のtビット（$1 \leqq t \leqq b$）までの誤りをスポッティバイト誤りと称し、1バイト内に複数の前記スポッティバイト誤りが生じる誤りをバイト内複数スポッティバイト誤りと称することを前提とし、前記パリティ検査行列で表現される前記スポッティバイト誤り制御符号は、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正し、かつp_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出することにより、或いは、前記t値、b値、p_1値、p_2値、μ_1値、μ_2値を任意に設定し、ただし、$p_1 \leqq p_2$及び$\mu_1 \leqq \mu_2$が成立することにより、或いは、前記情報伝送路は情報通信システムであるこ

とにより、或いは、前記情報伝送路はメモリシステムであることにより、或いは、前記情報伝送路はバス線回路であることによって効果的に達成される。

また、本発明の上述目的は、前記パリティ検査行列は、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正し、かつp_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出する機能を有する、誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号（$[\mu_1 t/b\mathrm{EC}]_{p_1}-[\mu_2 t/b\mathrm{ED}]_{p_2}$符号）に用いる、R行N列を有する次の行列Hであり、

$$H=\begin{bmatrix} H' & H' & \cdots & H' & \cdots & H' \\ \gamma^0 H'' & \gamma^1 H'' & \cdots & \gamma^i H'' & \cdots & \gamma^{n-1} H'' \\ \gamma^0 H'' & \gamma^2 H'' & \cdots & \gamma^{2i} H'' & \cdots & \gamma^{2(n-1)} H'' \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma^0 H'' & \gamma^{p_1+p_2-1} H'' & \cdots & \gamma^{(p_1+p_2-1)i} H'' & \cdots & \gamma^{(p_1+p_2-1)(n-1)} H'' \end{bmatrix}$$

ただし、$R=q+(p_1+p_2-1)r$、$N=bn$、$0 \leqq i \leqq n-1$、$n=2^r-1$であり、行列H'は、$H'=[h_0', h_1', \cdots, h_{b-1}']$に示すq次の列ベクトル$h_j'$（$0 \leqq j \leqq b-1$）から構成されるq行b列（$q \leqq b$）の2元行列であり、$(\mu_1+\mu_2)t$またはbのうちいずれか小さい値に等しいか大きい列が線形独立であり、$(\mu_1+\mu_2)t \geqq b$のとき、前記行列H'は単位行列を含む正則行列であり、$(\mu_1+\mu_2)t<b$のとき、前記行列H'は最小ハミング距離$(\mu_1+\mu_2)t+1$を有する符号の検査行列、すなわち、$(\mu_1+\mu_2)t$ビット誤り検出機能を有する（b、b－q）符号のパリティ検査行列に等しくなり、また、行列H"は、$H''=[h_0'', h_1'', \cdots, h_{b-1}'']$に示すr次の列ベクトル$h_j''$（$0 \leqq j \leqq b-1$）から構成されるr行

ｂ列（ｒ≦ｂ）の２元行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立であり、ここで、$\lfloor x\rfloor$はｘを超えない最大の整数を表し、また、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ≧ｂのとき、前記行列Ｈ”は単位行列を含む正則行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＜ｂのとき、前記行列Ｈ”は最小ハミング距離$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＋１を有する符号の検査行列、すなわち、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔビット誤り検出機能を有する（ｂ、ｂ－ｒ）符号のパリティ検査行列に等しくなり、γはＧＦ（２）のｒ次の拡大体ＧＦ（2^r）の原始元であって、$\gamma^i H''=[\gamma^i h_0'', \gamma^i h_1'', \cdots, \gamma^i h_{b-1}'']$が成立するように構成されることによって効果的に達成される。

また、本発明の上述目的は、前記パリティ検査行列は、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正し、かつp_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出する機能を有する、誤りバイト数を制限した伸長バイト内複数スポッティバイト誤り制御符号（伸長$[\mu_1 t/b\mathrm{EC}]_{p_1}-[\mu_2 t/b\mathrm{ED}]_{p_2}$符号）に用いる、Ｒ行Ｎ列を有する次の行列Ｈであり、

$$H=\begin{bmatrix} H' & H' & \cdots & H' & \cdots & H' & H' & O_{q\times r'} \\ \gamma^0 H'' & \gamma^1 H'' & \cdots & \gamma^i H'' & \cdots & \gamma^{n-1} H'' & O_{r'\times b} & O_{r'\times r'} \\ \gamma^0 H'' & \gamma^2 H'' & \cdots & \gamma^{2i} H'' & \cdots & \gamma^{2(n-1)} H'' & O_{r'\times b} & O_{r'\times r'} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \gamma^0 H'' & \gamma^{p_1+p_2-1} H'' & \cdots & \gamma^{(p_1+p_2-1)i} H'' & \cdots & \gamma^{(p_1+p_2-1)(n-1)} H'' & O_{r'\times b} & I_{r'} \end{bmatrix}$$

ただし、ｒ’≧ｒであり、Ｒ＝ｑ＋（p_1+p_2-1）ｒ’、Ｎ＝ｂ（ｎ＋１）＋ｒ’、０≦ｉ≦ｎ－１、ｎ＝$2^{r'}-1$であり、$I_{r'}$はｒ’次の単位行列で、Oは零行列であり、行列Ｈ’は、Ｈ’＝［h_0'、h_1'、・・・、h_{b-1}'］に示すｑ次の列ベクトルh_j'（０≦ｊ≦ｂ－１

）から構成されるｑ行ｂ列（ｑ≦ｂ）の２元行列であり、（$\mu_1+\mu_2$）ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立であり、（$\mu_1+\mu_2$）ｔ≧ｂのとき、前記行列Ｈ’は単位行列を含む正則行列であり、（$\mu_1+\mu_2$）ｔ＜ｂのとき、前記行列Ｈ’は最小ハミング距離（$\mu_1+\mu_2$）ｔ＋１を有する符号の検査行列、すなわち、（$\mu_1+\mu_2$）ｔビット誤り検出機能を有する（ｂ、ｂ－ｑ）符号のパリティ検査行列に等しくなり、また、行列Ｈ”は、Ｈ”＝［h_0”、h_1”、・・・、h_{b-1}”］に示すｒ次の列ベクトルh_j”（０≦ｊ≦ｂ－１）から構成されるｒ行ｂ列（ｒ≦ｂ）の２元行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立であり、ここで、$\lfloor x\rfloor$はｘを超えない最大の整数を表し、また、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ≧ｂのとき、前記行列Ｈ”は単位行列を含む正則行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＜ｂのとき、前記行列Ｈ”は最小ハミング距離$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＋１を有する符号の検査行列、すなわち、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔビット誤り検出機能を有する（ｂ、ｂ－ｒ）符号のパリティ検査行列に等しくなり、γはＧＦ（２）のｒ’次の拡大体ＧＦ（$2^{r'}$）の原始元であって、γ^iＨ”＝［$\gamma^i\Phi$（h_0”）、$\gamma^i\Phi$（h_1”）、・・・、$\gamma^i\Phi$（h_{b-1}”）］が成立するように構成され、ここで、Φは加法のもとでＧＦ（2^r）からＧＦ（$2^{r'}$）への単射準同型写像（injective homomorphism）であり、すなわち、Φ：ＧＦ（2^r）→ＧＦ（$2^{r'}$）であることによって効果的に達成される。

また、本発明の上述目的は、前記パリティ検査行列は、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正するとともに、p_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出し、かつ訂正能力を超えた１バイト誤りを検出する機能を有する、誤りバイト数を制限

したバイト内複数スポッティバイト誤り制御符号（$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}-S_bED$符号）に用いる、Ｒ行Ｎ列を有する次の行列Ｈであり、

$$H=\begin{bmatrix} I_b & I_b & \cdots & I_b & \cdots & I_b \\ \gamma^0 H'' & \gamma^1 H'' & \cdots & \gamma^i H'' & \cdots & \gamma^{n-1} H'' \\ \gamma^0 H'' & \gamma^2 H'' & \cdots & \gamma^{2i} H'' & \cdots & \gamma^{2(n-1)} H'' \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma^0 H'' & \gamma^{p_1+p_2-1} H'' & \cdots & \gamma^{(p_1+p_2-1)i} H'' & \cdots & \gamma^{(p_1+p_2-1)(n-1)} H'' \end{bmatrix}$$

ただし、Ｒ＝ｂ＋（ｐ$_1$＋ｐ$_2$－１）ｒ、Ｎ＝ｂｎ、０≦ｉ≦ｎ－１、ｎ＝２^r－１、Ｉ$_b$はｂ次の単位行列であり、行列Ｈ”は、Ｈ”＝［ｈ$_0$”、ｈ$_1$”、・・・、ｈ$_{b-1}$”］に示すｒ次の列ベクトルｈ$_j$”（０≦ｊ≦ｂ－１）から構成されるｒ行ｂ列（ｒ≦ｂ）の２元行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立であり、ここで、$\lfloor x\rfloor$はｘを超えない最大の整数を表し、ここで、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ≧ｂのとき、前記行列Ｈ”は単位行列を含む正則行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＜ｂのとき、前記行列Ｈ”は最小ハミング距離$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＋１を有する符号の検査行列、すなわち、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔビット誤り検出機能を有する（ｂ、ｂ－ｒ）符号のパリティ検査行列に等しくなり、γはＧＦ（２）のｒ次の拡大体ＧＦ（２^r）の原始元であって、γ^{i}Ｈ”＝［γ^{i}ｈ$_0$”、γ^{i}ｈ$_1$”、…、γ^{i}ｈ$_{b-1}$”］が成立するように構成されることによって効果的に達成される。

また、本発明の上述目的は、前記パリティ検査行列は、ｐ$_1$個のバイトに生じるμ$_1$個のスポッティバイト誤りを訂正するとともに、ｐ$_2$個のバイトに生じるμ$_2$個のスポッティバイト誤りを検出し、かつ訂正能力を超えた１バイト誤りを検出する機能を有する、誤りバイト数を制限

した伸長バイト内複数スポッティバイト誤り制御符号（伸長 $[\mu_1 t/b\mathrm{EC}]_{p_1}-[\mu_2 t/b\mathrm{ED}]_{p_2}-\mathrm{S}_b\mathrm{ED}$ 符号）に用いる、Ｒ行Ｎ列を有する次の行列Ｈであり、

$$H=\begin{bmatrix} I_b & I_b & \cdots & I_b & \cdots & I_b & I_b & O_{b\times r'} \\ \gamma^0 H'' & \gamma^1 H'' & \cdots & \gamma^i H'' & \cdots & \gamma^{n-1} H'' & O_{r'\times b} & O_{r'\times r'} \\ \gamma^0 H'' & \gamma^2 H'' & \cdots & \gamma^{2i} H'' & \cdots & \gamma^{2(n-1)} H'' & O_{r'\times b} & O_{r'\times r'} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \gamma^0 H'' & \gamma^{p_1+p_2-1} H'' & \cdots & \gamma^{(p_1+p_2-1)i} H'' & \cdots & \gamma^{(p_1+p_2-1)(n-1)} H'' & O_{r'\times b} & I_{r'} \end{bmatrix}$$

ただし、ｒ’≧ｒであり、Ｒ＝ｂ＋（p_1＋p_2－１）ｒ’、Ｎ＝ｂ（ｎ＋１）＋ｒ’、０≦ｉ≦ｎ－１、ｎ＝$2^{r'}$－１、I_bはｂ次の単位行列であり、$I_{r'}$はｒ’次の単位行列で、Oは零行列であり、行列Ｈ”は、Ｈ”＝［h_0”、h_1”、・・・、h_{b-1}”］に示すｒ次の列ベクトルh_j”（０≦ｊ≦ｂ－１）から構成されるｒ行ｂ列（ｒ≦ｂ）の２元行列であり、ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立であり、ここで、$\lfloor x \rfloor$はｘを超えない最大の整数を表し、ここで、$\lfloor(\mu_1+\mu_2)/2\rfloor t \geqq b$のとき、前記行列Ｈ”は単位行列を含む正則行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor t < b$のとき、前記行列Ｈ”は最小ハミング距離$\lfloor(\mu_1+\mu_2)/2\rfloor t+1$を有する符号の検査行列、すなわち、$\lfloor(\mu_1+\mu_2)/2\rfloor t$ビット誤り検出機能を有する（ｂ、ｂ－ｒ）符号のパリティ検査行列に等しくなり、γはＧＦ（２）のｒ’次の拡大体ＧＦ（$2^{r'}$）の原始元であって、$\gamma^i H''=[\gamma^i \Phi(h_0'')、\gamma^i \Phi(h_1'')、\cdots、\gamma^i \Phi(h_{b-1}'')]$が成立するように構成され、ここで、Φは加法のもとでＧＦ（2^r）からＧＦ（$2^{r'}$）への単射準同型写像（injective homomorphism）であり、すなわち、Φ：ＧＦ（2^r）→ＧＦ（$2^{r'}$）であることによって効果的に達成される。

また、本発明の上述目的は、前記誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号（$[\mu_1 t/b\mathrm{EC}]_{p_1}-[\mu_2 t/b\mathrm{ED}]_{p_2}$符号）は、前記ｔ値を１に設定した場合（ｔ＝１）に、ｐ$_1$個のバイトに生じるμ_1個のランダムビット誤りを訂正し、かつｐ$_2$個のバイトに生じるμ_2個のランダムビット誤りを検出する機能を有する、誤りバイト数を制限したランダムビット誤り制御符号となることにより、或いは、前記誤りバイト数を制限した伸長バイト内複数スポッティバイト誤り制御符号（伸長$[\mu_1 t/b\mathrm{EC}]_{p_1}-[\mu_2 t/b\mathrm{ED}]_{p_2}$符号）は、前記ｔ値を１に設定した場合（ｔ＝１）に、ｐ$_1$個のバイトに生じるμ_1個のランダムビット誤りを訂正し、かつｐ$_2$個のバイトに生じるμ_2個のランダムビット誤りを検出する機能を有する、誤りバイト数を制限したランダムビット誤り制御符号となることにより、或いは、前記誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号（$[\mu_1 t/b\mathrm{EC}]_{p_1}-[\mu_2 t/b\mathrm{ED}]_{p_2}-\mathrm{S}_b\mathrm{ED}$符号）は、前記ｔ値を１に設定した場合（ｔ＝１）に、ｐ$_1$個のバイトに生じるμ_1個のランダムビット誤りを訂正するとともに、ｐ$_2$個のバイトに生じるμ_2個のランダムビット誤りを検出し、かつ訂正能力を超えた１バイト誤りを検出する機能を有する、誤りバイト数を制限したランダムビット誤り制御符号となることにより、或いは、前記誤りバイト数を制限した伸長バイト内複数スポッティバイト誤り制御符号（伸長$[\mu_1 t/b\mathrm{EC}]_{p_1}-[\mu_2 t/b\mathrm{ED}]_{p_2}-\mathrm{S}_b\mathrm{ED}$符号）は、前記ｔ値を１に設定した場合（ｔ＝１）に、ｐ$_1$個のバイトに生じるμ_1個のランダムビット誤りを訂正するとともに、ｐ$_2$個のバイトに生じるμ_2個のランダムビット誤りを検出し、かつ訂正能力を超えた１バイト誤りを検出する機能を有する、誤りバイト数を制限したランダムビット誤り制御符号となることによって効果的に達成される。

また、本発明の上述目的は、前記パリティ検査行列（H）は、t≧b／$\lfloor(\mu_1+\mu_2)/2\rfloor$とした場合に、前記行列H'及び前記行列H"は任意のb列が線形独立な行列となり、p_1バイト誤りを訂正するとともにp_2バイト誤りを検出する機能を有するRS符号のパリティ検査行列と一致することにより、或いは、前記パリティ検査行列（H）は、t≧b／$\lfloor(\mu_1+\mu_2)/2\rfloor$とした場合に、前記行列H'及び前記行列H"は任意のb列が線形独立な行列となり、p_1バイト誤りを訂正するとともにp_2バイト誤りを検出する機能を有する2次伸長RS符号のパリティ検査行列と一致することにより、或いは、前記パリティ検査行列（H）は、t≧b／$\lfloor(\mu_1+\mu_2)/2\rfloor$とした場合に、前記行列H"は任意のb列が線形独立な行列となり、p_1バイト誤りを訂正するとともにp_2バイト誤りを検出する機能を有するRS符号のパリティ検査行列と一致することにより、或いは、前記パリティ検査行列（H）は、t≧b／$\lfloor(\mu_1+\mu_2)/2\rfloor$とした場合に、前記行列H"は任意のb列が線形独立な行列となり、p_1バイト誤りを訂正するとともにp_2バイト誤りを検出する機能を有する2次伸長RS符号のパリティ検査行列と一致することによって効果的に達成される。

また、本発明の上述目的は、前記誤り訂正手段は、前記シンドロームを基に前記受信語の何れのビットが誤っているかを検出するビット誤りポインタを生成し、生成された前記ビット誤りポインタを基に、誤りビットに該当する前記受信語のビット値を反転させることにより、前記受信語の誤りを訂正し、また、前記受信語のビット誤りを訂正できないことを検出した場合に、訂正不可能誤り検出信号を出力することにより、或いは、前記誤り訂正手段は、H'復号手段と、H"乗算手段と、GF（2^r）上の並列復号手段と、GF（2^b）上の誤り生成手段と、反転

手段とから構成されることにより、或いは、前記H'復号手段は、前記シンドローム生成手段で生成された前記シンドロームの上位qビットを基にして、バイトごとの誤りの和（e_A）を生成し、前記H"乗算手段は、前記H'復号手段で生成された前記バイトごとの誤りの和（e_A）を基にして、前記バイトごとの誤りの和（e_A）とH"の転置行列の積を生成し、前記GF（2^r）上の並列復号手段は、前記H"乗算手段で生成された前記バイトごとの誤りの和（e_A）とH"の転置行列の積、及び前記シンドローム生成手段で生成された前記シンドロームの上位qビットを除く残りの下位ビットを基にして、GF（2^r）上のビット誤りポインタ、及びGF（2^r）上の並列復号において誤り訂正が不可能とみなされたときに出力される第1の訂正不可能誤り検出信号を生成し、前記GF（2^b）上の誤り生成手段は、前記H'復号手段で生成された前記バイトごとの誤りの和（e_A）、及び、前記GF（2^r）上の並列復号手段で生成された前記GF（2^r）上のビット誤りポインタを基にして、GF（2^b）上のビット誤りポインタ、及び誤り訂正が不可能とみなされたときに出力される第2の訂正不可能誤り検出信号を生成し、前記反転手段は、前記GF（2^b）上の誤り生成手段で生成された前記GF（2^b）上のビット誤りポインタを基にして、前記GF（2^b）上のビット誤りポインタに対応した前記受信語のビット値を反転させることにより、前記受信語の誤りを訂正することにより、或いは、前記誤り訂正手段では、前記GF（2^r）上の並列復号手段で生成された前記第1の訂正不可能誤り検出信号と、前記GF（2^b）上の誤り生成手段で生成された前記第2の訂正不可能誤り検出信号の論理和をとることにより、訂正能力を超えた誤りを検出したことを示す第3の訂正不可能誤り検出信号を出力することによって効果的に達成される。

また、本発明の上述目的は、前記ＧＦ（２^{b}）上の誤り生成手段は、前記ＧＦ（２^{r}）上のビット誤りポインタ（ｅ’）の１バイトから、復号結果ビット誤りポインタ（$\hat{e}$）の１バイト及びｉバイトにおける検出信号を出力するｔビット誤り訂正復号手段と、入力された複数の前記ｉバイトにおける検出信号の重みの数だけを出力の上位ビットから１を連続して出力する多入力ソート手段と、誤訂正判定を行い、前記ＧＦ（２^{b}）上のビット誤りポインタ（ｅ）の１バイトを出力するｉバイトにおける誤訂正判定手段とを備えることにより、或いは、前記ＧＦ（２^{b}）上の誤り生成手段は、ｔビット誤り訂正復号手段と、多入力ソート手段と、ｉバイトにおける誤訂正判定手段とを備えており、前記ｔビット誤り訂正復号手段は、前記ＧＦ（２^{r}）上のビット誤りポインタ（ｅ’）の所定の１バイトを入力とし、復号結果ビット誤りポインタ（$\hat{e}$）の前記所定の１バイトに対応する１バイトを前記ｉバイトにおける誤訂正判定手段に出力するとともに、訂正能力をこえた誤りを検出する信号であるｉバイトにおける検出信号を前記多入力ソート手段と前記ｉバイトにおける誤訂正判定手段に出力する手段であり、前記多入力ソート手段は、入力される複数の前記ｉバイトにおける検出信号の重み（１の数）を数え、その重みの数だけを出力の上位ビットから１を連続して出力する手段であり、前記ｉバイトにおける誤訂正判定手段は、すべてのバイトの前記ｔビット誤り訂正復号手段から出力される前記復号結果ビット誤りポインタ（$\hat{e}$）、前記ｉバイトにおける検出信号、及び前記バイトごとの誤りの和（e_A）を入力とし、誤訂正判定を行い、前記ＧＦ（２^{b}）上のビット誤りポインタ（ｅ）の１バイトを出力する手段であることにより、或いは、前記ＧＦ（２^{b}）上の誤り生成手段では、前記多入力ソート手段から出力された信号において、その上位２ビット目の信号を

前記第２の訂正不可能誤り検出信号として出力し、また、前記ｉバイトにおける誤訂正判定手段から出力された信号を集め、符号ビット長のビット数を有する前記ＧＦ（２^{b}）上のビット誤りポインタを出力することによって効果的に達成される。

また、本発明の上述目的は、前記ｉバイトにおける誤訂正判定手段では、前記誤訂正判定が次の誤訂正判定数式によって行われ、

$$w_{\mathrm{M}}(\hat{e}_{\mathrm{x}}+e_{\mathrm{A}})\leq \mu_{1}-w_{\mathrm{M}}(\hat{e}_{\mathrm{x}})$$

ただし、$w_{\mathrm{M}}(e)$は$w_{\mathrm{M}}(e)=\sum_{i=0}^{n-1}\left\lceil \frac{w_{\mathrm{H}}(e_{i})}{t}\right\rceil$で定義され、スポッティバイト誤りの個数を表し、$w_{\mathrm{H}}(e_{i})$はベクトルｅ$_{i}$におけるガロア体ＧＦ（２）上のハミング重みであり、前記誤訂正数式が成立している場合に、ｅ$_{x}$＝$\hat{e}_{x}$となり誤訂正はされていないと判定され、前記誤訂正数式が成立していない場合に、ｅ$_{x}$≠$\hat{e}_{x}$となり誤訂正はされていると判定されるようにすることにより、或いは、前記$[\mu_{1}t/b\mathrm{EC}]_{p_{1}}-[\mu_{2}t/b\mathrm{ED}]_{p_{2}}-\mathrm{S}_{b}\mathrm{ED}$符号、及び前記伸長$[\mu_{1}t/b\mathrm{EC}]_{p_{1}}-[\mu_{2}t/b\mathrm{ED}]_{p_{2}}-\mathrm{S}_{b}\mathrm{ED}$符号に対しては、前記Ｈ”乗算手段にＳ$_{I}$以外のシンドロームを新たに入力し、Ｓ$_{I}$≠０かつＳ$_{I}$以外のシンドロームが０の場合に、検出信号を出力し、また、前記ＧＦ（２^{b}）上の誤り生成手段にＳ$_{I}$を新たに入力し、前記ＧＦ（２^{b}）上の誤り生成手段において、ＧＦ（２^{r}）上の誤りｅ’において誤りバイト数が１かどうかを調べ、前記多入力ソート手段を用いてＳ$_{I}$のハミング重みがμ_{1}ｔより大きいかどうかを調べることにより、前記復号手段を構成することによって効果的に達成される。

更に、本発明は、入力情報データを基に送信語を生成する符号化処理ステップと、情報伝送路中で誤りが発生した前記送信語を受信語として入力して前記誤りを訂正または検出する復号処理ステップとを有する誤

りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法に関し、本発明の上述目的は、前記符号化処理ステップは、スポッティバイト誤り制御符号を表現するパリティ検査行列と、前記入力情報データとを基に生成した検査情報を前記入力情報データに付加することにより、前記送信語を生成し、前記復号処理ステップは、前記パリティ検査行列を基に前記受信語のシンドロームを生成するシンドローム生成処理ステップと、前記シンドローム生成処理ステップにより生成されたシンドロームを基に前記受信語の誤りを訂正または検出する誤り訂正処理ステップとを有し、前記入力情報データが、ｂ（ｂは２以上の整数）ビットを１バイトとし、複数のバイトから構成される場合に、ここで、バイト内のｔビット（１≦ｔ≦ｂ）までの誤りをスポッティバイト誤りと称し、１バイト内に複数の前記スポッティバイト誤りが生じる誤りをバイト内複数スポッティバイト誤りと称することを前提とし、前記パリティ検査行列で表現される前記スポッティバイト誤り制御符号は、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正し、かつp_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出し、前記ｔ値、ｂ値、p_1値、p_2値、μ_1値、μ_2値を任意に設定し、ただし、$p_1 \leqq p_2$及び$\mu_1 \leqq \mu_2$が成立し、前記情報伝送路は、情報通信システム、メモリシステム、又はバス線回路の何れかであることによって効果的に達成される。

また、本発明の上述目的は、前記誤り訂正処理ステップは、前記シンドロームを基に前記受信語の何れのビットが誤っているかを検出するビット誤りポインタを生成し、生成された前記ビット誤りポインタを基に、誤りビットに該当する前記受信語のビット値を反転させることにより、前記受信語の誤りを訂正し、また、前記受信語のビット誤りを訂正で

きないことを検出した場合に、訂正不可能誤り検出信号を出力することにより、或いは、前記誤り訂正処理ステップは、H’復号処理ステップと、H”乗算処理ステップと、ＧＦ（2^r）上の並列復号処理ステップと、ＧＦ（2^b）上の誤り生成処理ステップと、反転処理ステップとを有することにより、或いは、前記H’復号処理ステップは、前記シンドローム生成処理ステップで生成された前記シンドロームの上位ｑビットを基にして、バイトごとの誤りの和（e_A）を生成し、前記H”乗算処理ステップは、前記H’復号処理ステップで生成された前記バイトごとの誤りの和（e_A）を基にして、前記バイトごとの誤りの和（e_A）とH”の転置行列の積を生成し、前記ＧＦ（2^r）上の並列復号処理ステップは、前記H”乗算処理ステップで生成された前記バイトごとの誤りの和（e_A）とH”の転置行列の積、及び前記シンドローム生成処理ステップで生成された前記シンドロームの上位ｑビットを除く残りの下位ビットを基にして、ＧＦ（2^r）上のビット誤りポインタ、及びＧＦ（2^r）上の並列復号において誤り訂正が不可能とみなされたときに出力される第１の訂正不可能誤り検出信号を生成し、前記ＧＦ（2^b）上の誤り生成処理ステップは、前記H’復号処理ステップで生成された前記バイトごとの誤りの和（e_A）、及び、前記ＧＦ（2^r）上の並列復号処理ステップで生成された前記ＧＦ（2^r）上のビット誤りポインタを基にして、ＧＦ（2^b）上のビット誤りポインタ、及び誤り訂正が不可能とみなされたときに出力される第２の訂正不可能誤り検出信号を生成し、前記反転処理ステップは、前記ＧＦ（2^b）上の誤り生成処理ステップで生成された前記ＧＦ（2^b）上のビット誤りポインタを基にして、前記ＧＦ（2^b）上のビット誤りポインタに対応した前記受信語のビット値を反転させることにより、前記受信語の誤りを訂正することによ

り、或いは、前記誤り訂正処理ステップでは、前記ＧＦ（2^r）上の並列復号処理ステップで生成された前記第１の訂正不可能誤り検出信号と、前記ＧＦ（2^b）上の誤り生成処理ステップで生成された前記第２の訂正不可能誤り検出信号の論理和をとることにより、訂正能力を超えた誤りを検出したことを示す第３の訂正不可能誤り検出信号を出力することによって効果的に達成される。

また、本発明の上述目的は、前記ＧＦ（2^b）上の誤り生成処理ステップは、前記ＧＦ（2^r）上のビット誤りポインタ（ｅ’）の１バイトから、復号結果ビット誤りポインタ（$\hat{e}$）の１バイト及びｉバイトにおける検出信号を出力するｔビット誤り訂正復号処理ステップと、入力された複数の前記ｉバイトにおける検出信号の重みの数だけを出力の上位ビットから１を連続して出力する多入力ソート処理ステップと、誤訂正判定を行い、前記ＧＦ（2^b）上のビット誤りポインタ（ｅ）の１バイトを出力するｉバイトにおける誤訂正判定処理ステップとを有することにより、或いは、前記ＧＦ（2^b）上の誤り生成処理ステップは、ｔビット誤り訂正復号処理ステップと、多入力ソート処理ステップと、ｉバイトにおける誤訂正判定処理ステップとを有し、前記ｔビット誤り訂正復号処理ステップは、前記ＧＦ（2^r）上のビット誤りポインタ（ｅ’）の所定の１バイトを入力とし、復号結果ビット誤りポインタ（$\hat{e}$）の前記所定の１バイトに対応する１バイトを前記ｉバイトにおける誤訂正判定処理ステップに出力するとともに、訂正能力をこえた誤りを検出する信号であるｉバイトにおける検出信号を前記多入力ソート処理ステップと前記ｉバイトにおける誤訂正判定処理ステップに出力し、前記多入力ソート処理ステップは、入力される複数の前記ｉバイトにおける検出信号の重み（１の数）を数え、その重みの数だけを出力の上位ビットか

ら１を連続して出力し、前記ｉバイトにおける誤訂正判定処理ステップは、すべてのバイトの前記ｔビット誤り訂正復号処理ステップから出力される前記復号結果ビット誤りポインタ（$\hat{e}$）、前記ｉバイトにおける検出信号、及び前記バイトごとの誤りの和（e_A）を入力とし、誤訂正判定を行い、前記ＧＦ（2^b）上のビット誤りポインタ（ｅ）の１バイトを出力することにより、或いは、前記ＧＦ（2^b）上の誤り生成処理ステップでは、前記多入力ソート処理ステップから出力された信号において、その上位２ビット目の信号を前記第２の訂正不可能誤り検出信号として出力し、また、前記ｉバイトにおける誤訂正判定処理ステップから出力された信号を集め、符号ビット長のビット数を有する前記ＧＦ（2^b）上のビット誤りポインタを出力することによって効果的に達成される。

また、本発明の上述目的は、前記ｉバイトにおける誤訂正判定処理ステップでは、前記誤訂正判定が次の誤訂正判定数式によって行われ、

$$w_M(\hat{e}_x + e_A) \le \mu_1 - w_M(\hat{e}_x)$$

ただし、$w_M(e)$は$w_M(e) = \sum_{i=0}^{n-1} \left\lceil \frac{w_H(e_i)}{t} \right\rceil$で定義され、スポッティバイト誤りの個数を表し、$w_H(e_i)$はベクトル$e_i$におけるガロア体ＧＦ（２）上のハミング重みであり、前記誤訂正数式が成立している場合に、$e_x = \hat{e}_x$となり誤訂正はされていないと判定され、前記誤訂正数式が成立していない場合に、$e_x \neq \hat{e}_x$となり誤訂正はされていると判定されるようにすることにより、或いは、前記$[\mu_1 t/bEC]_{p_1} - [\mu_2 t/bED]_{p_2} - S_b ED$符号、及び前記伸長$[\mu_1 t/bEC]_{p_1} - [\mu_2 t/bED]_{p_2} - S_b ED$符号に対しては、前記Ｈ”乗算処理ステップに$S_{II}$、$S_{III}$、$S_{IV}$を新たに入力し、$S_I \neq 0$かつ$S_{II} = S_{III} = S_{IV} = 0$の場合に、検出信号を出力し、また、前記ＧＦ

（2^{b}）上の誤り生成処理ステップにS_Iを新たに入力し、前記ＧＦ（2^{b}）上の誤り生成処理ステップにおいて、ＧＦ（2^{r}）上の誤りｅ’において誤りバイト数が１かどうかを調べ、前記多入力ソート処理ステップでS_Iのハミング重みが$\mu_1 t$より大きいかどうかを調べることにより、復号処理を行うことによって効果的に達成される。

図面の簡単な説明

第１図は、本発明の一実施形態である、誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置の概略構成を示すブロック図である。

第２図は、第１図に示された本発明の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置における全体動作の処理手順を示すフローチャートである。

第３図は、本発明によるパリティ検査行列の一例であって、符号長Ｎ＝９０ビット、情報長Ｋ＝６４ビット、検査長Ｒ＝２６ビット、バイト長ｂ＝８ビット、バイト内訂正ビット数ｔ＝１ビット、訂正数μ_1＝検出数μ_2＝４、誤りバイト制限数$p_1 = p_2 = 2$の符号パラメータを有する短縮（９０、６４）［$4_{1/8}$ＥＣ］$_2$符号のパリティ検査行列を示す図である。

第４図は、第３図に示すパリティ検査行列を基本行変形して得られる符号化用（９０、６４）［$4_{1/8}$ＥＣ］$_2$組織符号のパリティ検査行列を示す図である。

第５図は、第４図に示す符号化用（９０、６４）［$4_{1/8}$ＥＣ］$_2$組織符号をもとに構成した符号化回路（上位１４ビット）の具体例であり、Ｋ＝６４ビットの入力情報データからパリティ検査行列により２６ビ

ットの検査情報を生成する回路の概念図である。

第６図は、第４図に示す符号化用（９０、６４）［$4_{1/8}$ＥＣ］$_2$組織符号をもとに構成した符号化回路（下位１２ビット）の具体例であり、Ｋ＝６４ビットの入力情報データからパリティ検査行列により２６ビットの検査情報を生成する回路の概念図である。

第７図は、本発明で開示された機能を有する符号に対する復号回路の概略構成を示すブロック図である。

第８図は、第３図に示す（９０、６４）［$4_{1/8}$ＥＣ］$_2$符号をもとに具体的に構成したシンドローム生成回路（上位１４ビット）を説明するための概念図である。

第９図は、第３図に示す（９０、６４）［$4_{1/8}$ＥＣ］$_2$符号をもとに具体的に構成したシンドローム生成回路（下位１２ビット）を説明するための概念図である。

第１０図は、第３図に示す（９０、６４）［$4_{1/8}$ＥＣ］$_2$符号をもとに具体的に構成したＨ’復号回路の構成図である。

第１１図は、第３図に示す（９０、６４）［$4_{1/8}$ＥＣ］$_2$符号をもとに具体的に構成したＨ”乗算回路の構成図である。

第１２図は、第３図に示す（９０、６４）［$4_{1/8}$ＥＣ］$_2$符号に対するＧＦ（２^r）上の並列復号回路の概略構成を示すブロック図である。

第１３図は、第１２図に示すＧＦ（２^r）上の並列復号回路における（ｉ、ｊ）に対する誤り生成回路の一例の回路構成図である。

第１４図は、第１２図に示すＧＦ（２^r）上の並列復号回路におけるＧＦ（２^r）上の誤り算出回路の構成図である。

第１５図は、第３図に示す（９０、６４）［$4_{1/8}$ＥＣ］$_2$符号に対

するＧＦ（２^{b}）上の誤り生成回路の概略構成を示すブロック図である。

第１６図は、第１５図に示すＧＦ（２^{b}）上の誤り生成回路におけるｔビット誤り訂正復号回路の例であり、１バイト目に対してＧＦ（２^{r}）上のビット誤りポインタから復号結果ビット誤りポインタ（$\hat{e}$）７１の上位４ビットを生成する回路の構成図である。

第１７図は、第１５図に示すＧＦ（２^{b}）上の誤り生成回路におけるｔビット誤り訂正復号回路の例であり、１バイト目に対してＧＦ（２^{r}）上のビット誤りポインタを復号結果ビット誤りポインタ（$\hat{e}$）７１の下位４ビットを生成する回路の構成図である。

第１８図は、第１５図に示すＧＦ（２^{b}）上の誤り生成回路におけるｔビット誤り訂正復号回路の例であり、１バイト目に対して訂正能力をこえた誤りを検出する回路の構成図である。

第１９図は、第１５図に示すＧＦ（２^{b}）上の誤り生成回路における１２入力ソート回路の構成図である。

第２０図は、第１５図に示すＧＦ（２^{b}）上の誤り生成回路におけるｉバイトにおける誤訂正判定回路の一例の回路構成図である。

第２１図は、第２０図に示すｉバイトにおける誤訂正判定回路におけるビットごと排他的論理和回路の構成図である。

第２２図は、第２０図に示すｉバイトにおける誤訂正判定回路におけるビットごと選択回路の構成図である。

第２３図は、第２０図に示すｉバイトにおける誤訂正判定回路における１６入力ソート回路の構成図である。

第２４図は、第３図に示す（９０、６４）$[4_{1/8}EC]_{2}$符号に対する反転回路の一例であり、１バイトに対する反転回路の構成図である

。

（符号の説明）

1　シンドローム生成回路

2　符号化回路

3　回路（メモリ等の通信路）

4　復号回路

5　誤り訂正回路

6　H’復号回路

7　H”乗算回路

8　GF（2^r）上の並列復号回路

9　GF（2^b）上の誤り生成回路

10　反転回路

20　多入力パリティチェック回路

21　多入力パリティチェック回路

22　2入力排他的論理和回路

23　3入力パリティチェック回路

30　入力情報データ

31　送信語

32　受信語

33　シンドローム（S）

34　出力情報データ

35　訂正不可能誤り検出信号（UCE）

36　シンドローム（S）33の上位qビット（S_I）

37　シンドローム（S）33の下位3rビット（$S_{II}S_{III}S_{IV}$）

38　バイトごとの誤りの和（e_A）

３９　シンドローム（Ｓ）３３の上位ｑビットS_I３６とＨ”の転置（Ｈ”^{T}）の積（Ｓ’$_I$）

４０　ＧＦ（２^r）上のビット誤りポインタ

４１　訂正不可能誤り検出信号（ＤＳ（０））

４２　ＧＦ（２^b）上のビット誤りポインタ

４３　訂正不可能誤り検出信号（ＤＳ（１））

４４　２入力ＡＮＤゲート回路

５０　（ｉ、ｊ）に対する誤り生成回路

５１　ＧＦ（２^r）上の誤り算出回路

５２　（ｉ、ｊ）に対する誤り

５３　（ｉ、ｊ）に対する訂正可能信号

５４　２４入力ＯＲゲート回路

５５　６６入力ＮＯＲゲート回路

５６　２入力ＡＮＤゲート回路

６０　Ｓ’・$\tilde{H}_i^{\dagger T}$計算回路

６１　Ｓ’・$\tilde{H}_j^{\dagger T}$計算回路

６２　Ｓ’・$B_{(i,j)}^{\dagger\ T}$計算回路

６３　多入力パリティチェック回路

６４　多入力パリティチェック回路

６５　多入力パリティチェック回路

６６　１２入力ＮＯＲゲート回路

６７　２入力ＡＮＤゲート回路

６８　ビットごとＯＲ演算回路

６９　１１入力ＯＲゲート回路

７０　ｔビット誤り訂正復号回路

７１　復号結果ビット誤りポインタ（$\hat{e}$）

７２　ｉバイトにおける検出信号

７３　１２入力ソート回路

７４　ｉバイトにおける誤訂正判定回路

７５　２入力ＯＲゲート回路

７６　２入力ＡＮＤゲート回路

８０　ＮＯＴゲート回路

８１　３入力ＡＮＤゲート回路

８２　４入力ＡＮＤゲート回路

８３　５入力ＡＮＤゲート回路

８４　５入力ＯＲゲート回路

８５　６入力ＯＲゲート回路

８６　１５入力ＯＲゲート回路

９０　ビットごと排他的論理和回路

９１　１６入力ソート回路

９２　４入力ＯＲゲート回路

９３　１２入力ＮＯＲゲート回路

９４　ＮＯＴゲート回路

９５　３入力ＡＮＤゲート回路

９６　ビットごと選択回路

９７　多入力パリティチェック回路

９８　２入力マルチプレクサ回路

９９　２入力排他的論理和回路

１００　誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置

発明を実施するための最良の形態

以下、本発明を実施するための最良の形態について、図面を参照しながら詳細に説明する。

本発明を説明するための用語を次のように定義する。まず、本発明の実施形態において、対象とするディジタルデータとは、０と１との組合せの信号（２進化符号）であり、ビット誤りの発生とは、符号語中の任意のビットが、０→１または１→０になることを意味することとする。また、スポッティバイト誤りの発生とは、ｂビットからなるバイト中のｔビット（ｔ≦ｂ）までが０→１または１→０になることを意味することとする。

なお、本実施の形態で、「誤り」とある場合に、特に明記していない限り、前記「ビット誤り」、「スポッティバイト誤り」のすべてを包含していることとする。また、１バイト内に複数のスポッティバイト誤り（つまり、２以上のスポッティバイト誤り）が生じる誤りを『バイト内複数スポッティバイト誤り』と称する。更に、バイト内複数スポッティバイト誤りが限られた数のバイトに生じている場合、その誤りを『誤りバイト数を制限したバイト内複数スポッティバイト誤り』と称する。

ここで、まず、本発明に係る誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置１００の全体構成について説明する。第１図は、本発明のバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置１００（以下、スポッティバイト誤り訂正・検出装置１００と略す）の概略構成を示すブロック図である。第１図に示すように、スポッティバイト誤り訂正・検出装置１００は、符号化回路２、回路３、復号回路４等を備えた構成になっている。

符号化回路２は、対象のディジタルデータ（以下、入力情報データ３０と称する）に対して、誤りの訂正・検出を行う検査情報を生成する回路である。なお、検査情報は、任意個の検査ビットより構成されている。

回路３は、メモリ等の通信路に相当し、本実施形態では、回路３を経たデータには誤りが含まれる可能性が存在する構成としている。つまり、符号化回路２より出力されて回路３に入力されるデータには誤りが生じていないのに対して、回路３から出力されるデータには、誤りが発生している場合を含んでいる。

復号回路４は、シンドローム生成回路１と誤り訂正回路５とを備えた構成になっている。復号回路４は、受信語３２に誤りが含まれているか否かを検出し、その誤り箇所を特定して訂正を行うための回路である。復号回路４の詳細な構成については、後述する復号処理で説明する。

なお、後述する符号化処理及び復号処理では、ガロア体理論に基づいた行列演算を行うことから、符号化回路２、復号回路４、または通信路中における各データを、英字で付した行列、またはベクトルで表現する。具体的には、例えば、入力情報データＤ、符号（パリティ検査行列）Ｈ、検査情報Ｃ、送信語Ｖ、受信語Ｖ’、シンドロームＳと表している。

次に、本発明のスポッティバイト誤り訂正・検出装置１００の全体動作について説明する。第２図は、第１図に示された本発明のスポッティバイト誤り訂正・検出装置１００の全体動作の処理手順を示すフローチャートである。

第２図に示されるように、まず、符号化回路２では、入力情報データ（Ｄ）３０が入力されると、詳細については後述するパリティ検査行列

（Ｈ）を用いて検査情報（Ｃ）を生成する（ステップＳ２００）。次に、符号化回路２では、この検査情報（Ｃ）を入力情報データ（Ｄ）３０に付加して、送信語（Ｖ）３１としてメモリ等の通信路である回路３に送出する（ステップＳ２０１）。そして、回路３を経て誤りを含む可能性のある受信語（Ｖ’）３２は、復号回路４に入力される（ステップＳ２０２）。

次に、復号回路４では、まず、符号化回路２で用いたパリティ検査行列（Ｈ）を利用して、入力された受信語（Ｖ’）３２に誤りが発生しているか否かを調べる。具体的には、復号回路４のシンドローム生成回路１が、符号を表現しているパリティ検査行列（Ｈ）を転置させた転置行列（H^T）を、受信語（Ｖ’）３２に対して乗算することにより、シンドローム（Ｓ）を生成する（ステップＳ２０３）。シンドローム（Ｓ）３３の値は、受信語（Ｖ’）３２に応じて変化するため、受信語（Ｖ’）３２に誤りが含まれているかを判断することが可能となる。

そこで、復号回路４の誤り訂正回路５は、前記シンドローム（Ｓ）３３の値に基づいて、まず誤りの有無を検出し（ステップＳ２０４）、誤りが検出された場合に、その誤りが訂正可能か否かを判断し（ステップＳ２０５）、前記判断の結果、その誤りが訂正可能であれば、誤りの訂正を行う（ステップＳ２０６）。そして、誤り訂正回路５は、受信語Ｖ’に対してスポッティバイト誤りの訂正処理を施した場合に、訂正後のＶ’を受信語出力情報データ（Ｖ*）３４として出力する（ステップＳ２０７）。これに対して、ステップＳ２０５の判断の結果、訂正不可能なスポッティバイト誤り等が検出された場合には、誤り訂正回路５は、訂正不可能な誤り検出信号としてのＵＣＥ（Uncorrectable Error）信号３５を出力する（ステップＳ２０８）。

次に、前述した符号化回路２及び復号回路４で用いられるパリティ検査行列（Ｈ）の構成について、詳細に説明する。本発明のパリティ検査行列Ｈ（以下、Ｈマトリクス、符号マトリクス、または単に検査行列とも称する）は、制限された数のバイトに生じる複数のスポッティバイト誤りを訂正・検出する機能を有する符号である。

本実施の形態では、任意のp_1、p_2（$p_1 \leqq p_2$）、μ_1、μ_2（$\mu_1 \leqq \mu_2$）、ｔ（ｔは１以上ｂ以下の整数）、及びｂ（２以上の整数）に対して構成できる一般的な符号構成法を提示するとともに、その符号により実際に誤りの訂正・検出ができる方法を示し、またそのための符号化回路および復号回路を提示し、前記符号化回路及び復号回路によって誤りが具体的に訂正・検出できることを示す。

ここで、まず、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正し、かつp_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出することを具体的に示す。

以下では、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正し、かつp_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出する機能を有する、『誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号』を$[\mu_1 t/b\mathrm{EC}]_{p_1} - [\mu_2 t/b\mathrm{ED}]_{p_2}$符号と表す。例えば、２個のバイトに生じる３個のスポッティバイト誤り訂正符号の場合に、$[\mathrm{T}_{t/b}\mathrm{EC}]_2$符号と表し、$\mu_1 = \mu_2 = 3$、$p_1 = p_2 = 2$となる。

また、本発明の符号構成法において、ｔ＝ｂのとき、正確には後述するように、$t \geqq b / \lfloor (\mu_1 + \mu_2)/2 \rfloor$のとき、従来のバイト誤り制御符号であるリード・ソロモン符号（ＲＳ符号）に一致する。

いま、Ｋビットの入力情報データ（Ｄ）３０に対してＲビットの検査

情報（Ｃ）を付加し、ある機能を満足するＮ＝Ｋ＋Ｒビットの長さを有する送信語Ｖ（Ｎビットの行ベクトル）を構成することができたときに、この送信語（Ｖ）と、誤り制御符号を規定するＲ行Ｎ列の２元パリティ検査行列（Ｈ）との間には、$V \cdot H^T = 0$の関係が成立する。ここで、H^Tはパリティ検査行列（Ｈ）の行と列を入れ替えて生成した転置行列であり、また、右辺の０はＲビットの零行ベクトルを示している。

本実施形態では、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正し、かつp_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出する機能を有する、誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号（$[\mu_1 t/b\mathrm{EC}]_{p_1} - [\mu_2 t/b\mathrm{ED}]_{p_2}$符号）に用いるＲ行Ｎ列を有するパリティ検査行列（Ｈ）を、下記数１に示すような構成としている。

【数１】

$$H = \begin{bmatrix} H' & H' & \cdots & H' & \cdots & H' \\ \gamma^0 H'' & \gamma^1 H'' & \cdots & \gamma^i H'' & \cdots & \gamma^{n-1} H'' \\ \gamma^0 H'' & \gamma^2 H'' & \cdots & \gamma^{2i} H'' & \cdots & \gamma^{2(n-1)} H'' \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma^0 H'' & \gamma^{p_1+p_2-1} H'' & \cdots & \gamma^{(p_1+p_2-1)i} H'' & \cdots & \gamma^{(p_1+p_2-1)(n-1)} H'' \end{bmatrix}$$

ただし、Ｒ＝ｑ＋（$p_1 + p_2 - 1$）ｒ、Ｎ＝ｂｎ、０≦ｉ≦ｎ－１、$n = 2^r - 1$である。

上記数１を構成する行列Ｈ’は、下記数２に示すｑ次の列ベクトルh_j'（０≦ｊ≦ｂ－１）から構成されるｑ行ｂ列（ｑ≦ｂ）の２元行列であり、（$\mu_1 + \mu_2$）ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立である。

【数２】

$H' = [h_0', h_1', \cdots, h_{b-1}']$

ここで、$(\mu_1+\mu_2)t \geqq b$のとき、前記行列H'は単位行列を含む正則行列であり、$(\mu_1+\mu_2)t < b$のとき、前記行列H'は最小ハミング距離$(\mu_1+\mu_2)t+1$を有する符号の検査行列、すなわち、$(\mu_1+\mu_2)t$ビット誤り検出機能を有する$(b, b-q)$符号のパリティ検査行列に等しい。

また、上記数1を構成する行列H"は、下記数3に示すr次の列ベクトル$h_j"$（$0 \leqq j \leqq b-1$）から構成されるr行b列（$r \leqq b$）の2元行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor t$またはbのうちいずれか小さい値に等しいか大きい列が線形独立である。ここで、$\lfloor x\rfloor$はxを超えない最大の整数を表す。

【数3】

$H" = [h_0", h_1", \cdots, h_{b-1}"]$

ここで、$\lfloor(\mu_1+\mu_2)/2\rfloor t \geqq b$のとき、前記行列H"は単位行列を含む正則行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor t < b$のとき、前記行列H"は最小ハミング距離$\lfloor(\mu_1+\mu_2)/2\rfloor t+1$を有する符号の検査行列、すなわち、$\lfloor(\mu_1+\mu_2)/2\rfloor$ tビット誤り検出機能を有する（b、b－r）符号のパリティ検査行列に等しい。

次に、γをGF（2）のr次の拡大体GF（2^r）の原始元とするときに、$\gamma^i H"$は下記数4で定義することができる。

【数4】

$\gamma^i H" = [\gamma^i h_0", \gamma^i h_1", \cdots, \gamma^i h_{b-1}"]$

このとき、最大符号長Nは、$N = b(2^r-1)$である。

上記数1に示したパリティ検査行列（H）は、パラメータ$t \geqq b / \lfloor(\mu_1+\mu_2)/2\rfloor$とした場合に、H'及びH"は任意のb列が線形独立な行列

となるため、従来のｐ１バイト誤りを訂正するとともにｐ２バイト誤りを検出する機能を有する、ｐ１訂正ｐ２検出ＲＳ符号のパリティ検査行列と一致する。

また、本実施形態では、$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}$符号の伸長符号を定義することができる。ｐ１個のバイトに生じるμ１個のスポッティバイト誤りを訂正し、かつｐ２個のバイトに生じるμ２個のスポッティバイト誤りを検出する機能を有する、誤りバイト数を制限した伸長バイト内複数スポッティバイト誤り制御符号（伸長$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}$符号）に用いるＲ行Ｎ列を有するパリティ検査行列（Ｈ）は、下記数５に示すような構成とすることができる。

【数５】

$$H=\begin{bmatrix} H' & H' & \cdots & H' & \cdots & H' & H' & O_{q\times r'} \\ \gamma^0 H'' & \gamma^1 H'' & \cdots & \gamma^i H'' & \cdots & \gamma^{n-1} H'' & O_{r'\times b} & O_{r'\times r'} \\ \gamma^0 H'' & \gamma^2 H'' & \cdots & \gamma^{2i} H'' & \cdots & \gamma^{2(n-1)} H'' & O_{r'\times b} & O_{r'\times r'} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \gamma^0 H'' & \gamma^{p_1+p_2-1} H'' & \cdots & \gamma^{(p_1+p_2-1)i} H'' & \cdots & \gamma^{(p_1+p_2-1)(n-1)} H'' & O_{r'\times b} & I_{r'} \end{bmatrix}$$

ただし、ｒ’≧ｒであり、Ｒ＝ｑ＋（ｐ１＋ｐ２－１）ｒ’、Ｎ＝ｂ（ｎ＋１）＋ｒ’、０≦ｉ≦ｎ－１、ｎ＝２ｒ’－１であり、$I_{r'}$はｒ’次の単位行列で、Oは零行列である。

上記数５を構成する行列Ｈ’は、下記数６に示すｑ次の列ベクトルｈｊ’（０≦ｊ≦ｂ－１）から構成されるｑ行ｂ列（ｑ≦ｂ）の２元行列であり、（μ１＋μ２）ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立である。

【数６】

Ｈ’＝［ｈ０’、ｈ１’、・・・、ｈｂ－１’］

ここで、（$\mu_1+\mu_2$）t≧bのとき、前記行列H'は単位行列を含む正則行列であり、（$\mu_1+\mu_2$）t＜bのとき、前記行列H'は最小ハミング距離（$\mu_1+\mu_2$）t＋1を有する符号の検査行列、すなわち、（$\mu_1+\mu_2$）tビット誤り検出機能を有する（b、b－q）符号のパリティ検査行列に等しくなる。

また、上記数5を構成する行列H"は、下記数7に示すr次の列ベクトルh_j"（0≦j≦b－1）から構成されるr行b列（r≦b）の2元行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$tまたはbのうちいずれか小さい値に等しいか大きい列が線形独立である。ここで、$\lfloor x\rfloor$ははxを超えない最大の整数を表す。

【数7】

$$H" = [h_0", h_1", \cdots, h_{b-1}"]$$

ここで、$\lfloor(\mu_1+\mu_2)/2\rfloor$t≧bのとき、前記行列H"は単位行列を含む正則行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$t＜bのとき、前記行列H"は最小ハミング距離$\lfloor(\mu_1+\mu_2)/2\rfloor$t＋1を有する符号の検査行列、すなわち、$\lfloor(\mu_1+\mu_2)/2\rfloor$tビット誤り検出機能を有する（b、b－r）符号のパリティ検査行列に等しくなる。

次に、γをGF(2)のr'次の拡大体GF($2^{r'}$)の原始元とするときに、$\gamma^i H"$は下記数8で定義することができる。

【数8】

$$\gamma^i H" = [\gamma^i \Phi(h_0"), \gamma^i \Phi(h_1"), \cdots, \gamma^i \Phi(h_{b-1}")]$$

ここで、Φは加法のもとでGF(2^r)からGF($2^{r'}$)への単射準同型写像（injective homomorphism）であり、すなわち、Φ：GF(2^r)→GF($2^{r'}$)であることによって効果的に達成される。このとき

、最大符号長Ｎは、Ｎ＝ｂ・２$^{r'}$＋ｒ’である。数５のＨの２行目から下の行に用いるＨ”の行数ｒ’は、ｒより大きい値をとることができ、数５で表される符号は、（$\mu_1+\mu_2-1$）ビットずつ検査ビット数を増大させることにより、任意の符号長に対して構成できる符号であることに注意する必要がある。

上記数５に示したパリティ検査行列（Ｈ）は、パラメータｔ≧ｂ／$\lfloor(\mu_1+\mu_2)/2\rfloor$とした場合に、Ｈ’及びＨ”は任意のｂ列が線形独立な行列となるため、従来のp_1バイト誤りを訂正するとともにp_2バイト誤りを検出する機能を有する、p_1訂正p_2検出２次伸長ＲＳ符号のパリティ検査行列と一致する。

次に、上記数１に示したパリティ検査行列（Ｈ）で表される符号が、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正し、かつp_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出することを具体的に示す。一般に、パリティ検査行列Ｈ＝［H_0、H_1、・・・、H_{n-1}］が、以下に示す必要十分条件を満たしていることを示せばよい。ただし、H_iはＲ×ｂの小行列である。

【数９】

$$e_{i_1}\cdot H_{i_1}^T+\cdots+e_{i_{v_1}}\cdot H_{i_{v_1}}^T+e_{i_{v_1+1}}\cdot H_{i_{v_1+1}}^T+\cdots+e_{i_{v_1+v_2}}\cdot H_{i_{v_1+v_2}}^T$$
$$+\cdots+e_{i_{v_1+v_2+\cdots+v_{\lambda-1}+1}}\cdot H_{i_{v_1+v_2+\cdots+v_{\lambda-1}+1}}^T+\cdots+e_{i_{v_1+v_2+\cdots+v_\lambda}}\cdot H_{i_{v_1+v_2+\cdots+v_\lambda}}^T\neq 0$$

$$0<\sum_{x=1}^{v_1+\cdots+v_\lambda}w_M\left(e_{i_x}\right)\le\mu_1+\mu_2,\ 0<\sum_{x=1}^{\lambda}v_x\le p_1+p_2$$

$$0\le v_1\le p_1+p_2,\ 0\le v_2\le\min\left(\lfloor(\mu_1+\mu_2)/2\rfloor,p_1+p_2\right),$$
$$\cdots,\ 0\le v_\lambda\le\min\left(\lfloor(\mu_1+\mu_2)/\lambda\rfloor,p_1+p_2\right)$$

ここで、e_{i_x}は誤りベクトルｅにおけるｉ$_x$番目のバイトであり、$\lambda=\lceil b/t\rceil$、$\lfloor y\rfloor$はｙ以下である最大の整数、$\lceil z\rceil$はｚ以上である最小の整数をそれぞれ表す。

また、i_1、i_2、…、$i_{v_1+v_2+\cdots+v_\lambda}$（$0\le i_1$、$i_2$、…、$i_{v_1+v_2+\cdots+v_\lambda}\le n-$

１）は、すべて異なるものとし、$w_M(e)$は下記数１０で定義され、スポッティバイト誤りの個数を表している。ただし、$w_H(e_i)$はベクトルｅ$_i$におけるガロア体ＧＦ（２）上のハミング重みである。

【数１０】

$$w_M(e)=\sum_{i=0}^{n-1}\left\lceil \frac{w_H(e_i)}{t} \right\rceil$$

スポッティバイト誤り訂正・検出は、前述したシンドローム生成回路１が生成するシンドローム（Ｓ）３３を用いて実行する。ここで、シンドローム（Ｓ）３３と前記パリティ検査行列（Ｈ）との関係は、Ｓ＝Ｖ’・Ｈ^{T}で表される。いま、受信語（Ｖ’）３２が元々の送信語（Ｖ）３１に対して誤り（Ｅ）を含み、Ｖ’＝Ｖ＋Ｅで表されるとき、シンドローム（Ｓ）３３は、Ｓ＝Ｖ’・Ｈ^T＝（Ｖ＋Ｅ）・Ｈ^T＝Ｖ・Ｈ^T＋Ｅ・Ｈ^{T}となる。

前述したように、Ｖ・Ｈ^T＝０の関係が成立していることから、結局、シンドロームＳ＝Ｅ・Ｈ^{T}が成り立つ。つまり、シンドローム（Ｓ）３３は、送信語（Ｖ）３１に影響されずに誤り（Ｅ）のみで決まることから、シンドローム（Ｓ）３３に基づいてスポッティバイト誤り訂正を行うことが可能か否かは、パリティ検査行列（Ｈ）と誤り（Ｅ）とから計算されるＥ・Ｈ^{T}の値の結果によって決定できるものである。

上記数９に示した必要十分条件は、スポッティバイト誤りの訂正数μ_1及び検出数μ_2に対し、訂正・検出可能なすべてのバイト内複数スポッティバイト誤りのシンドロームと訂正可能な他のすべてのバイト内複数スポッティバイト誤りのシンドロームが異なることを示している。

すなわち、ｐ$_2$個以下のバイトに生じるμ_2個以下のバイト内複数ス

ポッティバイト誤りのシンドロームと他のｐ１個以下のバイトに生じるμ１個以下のバイト内複数スポッティバイト誤りのシンドロームが異なればよい。

ここで、ρ１（ρ１≦ｐ１）バイトに生じるℓ１（ℓ１≦μ１）個のスポッティバイト誤りの集合を$\Omega_i=\{e_{i_1},e_{i_2},\cdots,e_{i_{\rho 1}}\}$とする。ただし、$w_M(e_{i_x})\leqq\lambda$、$\sum_{x=1}^{\rho 1}w_M(e_{i_x})=\ell_1$である。

また、ρ２（ρ２≦ｐ２）バイトに生じるℓ２（ℓ２≦μ２）個のスポッティバイト誤りの集合を$\Omega_j=\{e_{j_1},e_{j_2},\cdots,e_{j_{\rho 2}}\}$とする。ただし、$w_M(e_{j_x})\leqq\lambda$、$\sum_{x=1}^{\rho 2}w_M(e_{j_x})=\ell_2$である。

このとき、$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}$符号は、Ω ｉ及びΩ ｊにおける誤りパターンにおいて、下記数１１の関係が成り立つ。

【数１１】

$$e_{i_1}\cdot H_{i_1}^T+e_{i_2}\cdot H_{i_2}^T+\cdots+e_{i_{\rho 1}}\cdot H_{i_{\rho 1}}^T\neq e_{j_1}\cdot H_{j_1}^T+e_{j_2}\cdot H_{j_2}^T+\cdots+e_{j_{\rho 2}}\cdot H_{j_{\rho 2}}^T$$

Ω ｉ及びΩ ｊにおけるスポッティバイト誤りは、同じバイトに誤りが生じているものがある。Ｖ（０≦Ｖ≦p_1）バイトにおいて、同じバイトに誤りが生じているとすると、下記数１２が得られる。

【数１２】

$$\begin{aligned}&e_{i_1}\cdot H_{i_1}^T+e_{i_2}\cdot H_{i_2}^T+\cdots+e_{i_V}\cdot H_{i_V}^T+e_{i_{V+1}}\cdot H_{i_{V+1}}^T+\cdots+e_{i_{\rho 1}}\cdot H_{i_{\rho 1}}^T\\&\qquad\neq e_{j_1}\cdot H_{j_1}^T+e_{j_2}\cdot H_{j_2}^T+\cdots++e_{j_V}\cdot H_{j_V}^T+e_{j_{V+1}}\cdot H_{j_{V+1}}^T+\cdots+e_{j_{\rho 2}}\cdot H_{j_{\rho 2}}^T\end{aligned}$$

よって、上記数１２より下記数１３が得られる。

【数１３】

$$\begin{aligned}&(e_{i_1}+e_{j_1})\cdot H_{i_1}^T+(e_{i_2}+e_{j_2})\cdot H_{i_2}^T+\cdots+(e_{i_V}+e_{j_V})\cdot H_{i_V}^T\\&\quad+e_{i_{V+1}}\cdot H_{i_{V+1}}^T+\cdots+e_{i_{\rho 1}}\cdot H_{i_{\rho 1}}^T+e_{j_V}\cdot H_{j_V}^T+e_{j_{V+1}}\cdot H_{j_{V+1}}^T+\cdots+e_{j_{\rho 2}}\cdot H_{j_{\rho 2}}^T\neq 0\end{aligned}$$

ここで、ｘ＝１、２、・・・、Ｖに対し、$w_M(e_{i_x}+e_{j_x})\leqq\lambda$である。上記数１３において、$(e_{i_x}+e_{j_x})$を$e_{i_x}$と置き換え、ρ１＋ρ２－Ｖ＝ｖ１＋ｖ１＋・・

・＋ｖ$_{\lambda}$とすると、上記数９に示した必要十分条件が成立する。

よって、数１に示すパリティ検査行列で表現される符号が、上記数９に示した必要十分条件を満たすことを示すことにより、ｐ$_{1}$個のバイトに生じるμ$_{1}$個のスポッティバイト誤りを訂正し、かつｐ$_{2}$個のバイトに生じるμ$_{2}$個のスポッティバイト誤りを検出することを具体的に示す。

シンドロームＳとして、下記数１４の関係が成立していると仮定する。

【数１４】

$$e_{i_1}\cdot\begin{bmatrix} H' \\ \gamma^{i_1}H'' \\ \gamma^{2i_1}H'' \\ \vdots \\ \gamma^{(p_1+p_2-1)i_1}H'' \end{bmatrix}^T+\cdots+e_{i_{v_1+1}}\cdot\begin{bmatrix} H' \\ \gamma^{i_{v_1+1}}H'' \\ \gamma^{2i_{v_1+1}}H'' \\ \vdots \\ \gamma^{(p_1+p_2-1)i_{v_1+1}}H'' \end{bmatrix}^T+\cdots$$

$$+\ e_{i_{v_1+v_2+\cdots+v_\lambda}}\cdot\begin{bmatrix} H' \\ \gamma^{i_{v_1+v_2+\cdots+v_\lambda}}H'' \\ \gamma^{2i_{v_1+v_2+\cdots+v_\lambda}}H'' \\ \vdots \\ \gamma^{(p_1+p_2-1)i_{v_1+v_2+\cdots+v_\lambda}}H'' \end{bmatrix}^T=\begin{bmatrix}0\\0\\0\\0\\0\end{bmatrix}^T$$

上記数１４の $\left(\sum_{y=1}^{v_1+v_2+\cdots+v_\lambda}e_{i_y}\right)\cdot H'^T=0$ において、Ｈ’は任意のＭｉｎ（（μ$_{1}$＋μ$_{2}$）ｔ、ｂ）列が線形独立な行列であることにより、下記数１５が得られる。

【数１５】

$$\sum_{y=1}^{v_1+v_2+\cdots+v_\lambda} e_{i_y} = 0$$

上記数１５において、右からＨ”Ｔをかけると、下記数１６が得られる。

【数１６】

$$\left(\sum_{y=1}^{v_1+v_2+\cdots+v_\lambda} e_{i_y}\right)\cdot H''^{T} = 0$$

上記数１４、数１６において、$e_{i_1}\cdot H''^{T}$、$e_{i_2}\cdot H''^{T}$、…、$e_{i_{v_1+v_2+\cdots+v_\lambda}}\cdot H''^{T}$をそれぞれｘ１、ｘ２、…、$x_{v_1+v_2+\cdots+v_\lambda}$とおくと、下記数１７が得られる。ここで、Ｈ”は任意のＭｉｎ（$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ、ｂ）列が線形独立な行列であることにより、ｘ１≠０、ｘ２≠０、…、$x_{v_1+v_2+\cdots+v_{\lfloor(\mu_1+\mu_2)/2\rfloor}}$≠０となる。

【数１７】

$$\begin{cases} x_1+x_2+\cdots+x_{v_1+v_2+\cdots+v_\lambda}=0 \\ \gamma^{i_1}x_1+\gamma^{i_2}x_2+\cdots+\gamma^{i_{v_1+v_2+\cdots+v_\lambda}}x_{v_1+v_2+\cdots+v_\lambda}=0 \\ \gamma^{2i_1}x_1+\gamma^{2i_2}x_2+\cdots+\gamma^{2i_{v_1+v_2+\cdots+v_\lambda}}x_{v_1+v_2+\cdots+v_\lambda}=0 \\ \cdots \\ \gamma^{(p_1+p_2-1)i_1}x_1+\gamma^{(p_1+p_2-1)i_2}x_2+\cdots+\gamma^{(p_1+p_2-1)i_{v_1+v_2+\cdots+v_\lambda}}x_{v_1+v_2+\cdots+v_\lambda}=0 \end{cases}$$

上記数１７において、上からｖ１＋ｖ２＋･･･＋ｖλ（≦ｐ１＋ｐ２）個の式を行列の形で表すと、下記数１８が得られる。

【数１８】

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \gamma^{i_1} & \gamma^{i_2} & \cdots & \gamma^{i_{v_1+v_2+\cdots+v_\lambda}} \\ \gamma^{2i_1} & \gamma^{2i_2} & \cdots & \gamma^{2i_{v_1+v_2+\cdots+v_\lambda}} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma^{(v_1+v_2+\cdots+v_\lambda-1)i_1} & \gamma^{(v_1+v_2+\cdots+v_\lambda-1)i_2} & \cdots & \gamma^{(v_1+v_2+\cdots+v_\lambda-1)i_{v_1+v_2+\cdots+v_\lambda}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{v_1+v_2+\cdots+v_\lambda} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

上記数１８中の$v_1+v_2+\cdots+v_\lambda$行$v_1+v_2+\cdots+v_\lambda$列の行列は、ヴァンデルモンドの行列の形をしているため、正則行列である。よって、上記数１８の両辺に左からこの行列の逆行列を乗算すると、下記数１９が得られる。

【数１９】

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{v_1+v_2+\cdots+v_\lambda} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

上記数１９は、$x_1=0$、$x_2=0$、…、$x_{v_1+v_2+\cdots+v_\lambda}=0$となり、前提である$x_1\neq 0$、$x_2\neq 0$、…、$x_{v_1+v_2+\cdots+v_{\lfloor(\mu_1+\mu_2)/2\rfloor}}\neq 0$に矛盾する。よって、数１で表される符号は、上記数９で示される必要十分条件を満たしている。

したがって、前述した証明結果より、上記数１に示すパリティ検査行列（Ｈ）は、受信語（Ｖ’）３２に含まれるp_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正し、かつp_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出する機能を有していることを証明することができた。

また、上記数５に示したパリティ検査行列（Ｈ）で表される伸長符号

も、同様に上記数９に示した必要十分条件を満たすことを示すことにより、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正し、かつp_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出することを示すことができる。

ここでは、具体的な符号パラメータを与えて、上記数１に示すパリティ検査行列（Ｈ）を構成できることを示す。なお、このパリティ検査行列（Ｈ）は、あくまで一例であって、本発明がこれに限定されないことは言うまでもない。

いま、読み書きデータ幅として、８ビットを有するメモリ素子を使用したメモリシステムを対象に、ｂ＝８、ｔ＝１、$\mu_1=\mu_2=4$、$p_1=p_2=2$のパラメータを与え、バイト内１ビットの誤りをスポッティバイト誤りとして、２バイトに生じる４個のバイト内複数スポッティバイト誤り（本例の場合、ビット誤り）を訂正する機能を有する$[4_{1/8}EC]_2$符号のパリティ検査行列（Ｈ）を構成することとする。

このとき、Ｍｉｎ（（$\mu_1+\mu_2$）ｔ、ｂ）＝Ｍｉｎ（８、８）＝８より、行列Ｈ’として、任意の８列が線形独立な行列を構成すればよい。この一例として、下記数２０に示す８×８（８行８列を意味する）単位行列を構成すればよい。

【数２０】

$$H' = I_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

また、Ｍｉｎ（$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ、ｂ）＝Ｍｉｎ（４、８）＝４より、行列Ｈ”として、任意の４列が線形独立な４ビット誤り検出符号の検査行列を構成すればよい。この一例として、下記数２１に示す６×８行列を示す。

【数２１】

$$H'' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

次に、γをＧＦ（２）上の６次の原始多項式ｇ（ｘ）＝ｘ^{6}＋ｘ＋１の根とすると、γは、６次の列ベクトルγ＝（０１０００００）^{T}と表現することができる。

つまり、γのべき表示γ^{i}とベクトル表示との対応を示すと、次のようになる。

γ^{0}＝（１０００００）^{T}、　γ^{1}＝（０１０００００）^{T}、　γ^{2}＝（００１０００）^{T}、

γ^{3}＝（０００１００）^{T}、　γ^{4}＝（０００００１０）^{T}、　γ^{5}＝（００

0 0 0 1) ^{T}、

γ^{6} = (1 1 0 0 0 0) ^{T}、 γ^{7} = (0 1 1 0 0 0) ^{T}、 γ^{8} = (0 0 1 1 0 0) ^{T}、

γ^{9} = (0 0 0 1 1 0) ^{T}、 γ^{10} = (0 0 0 0 1 1) ^{T}、 γ^{11} = (1 1 0 0 0 1) ^{T}、

γ^{12} = (1 0 1 0 0 0) ^{T}、 γ^{13} = (0 1 0 1 0 0) ^{T}、 γ^{14} = (0 0 1 0 1 0) ^{T}、

γ^{15} = (0 0 0 1 0 1) ^{T}、 γ^{16} = (1 1 0 0 1 0) ^{T}、 γ^{17} = (0 1 1 0 0 1) ^{T}、

γ^{18} = (1 1 1 1 0 0) ^{T}、 γ^{19} = (0 1 1 1 1 0) ^{T}、 γ^{20} = (0 0 1 1 1 1) ^{T}、

γ^{21} = (1 1 0 1 1 1) ^{T}、 γ^{22} = (1 0 1 0 1 1) ^{T}、 γ^{23} = (1 0 0 1 0 1) ^{T}、

γ^{24} = (1 0 0 0 1 0) ^{T}、 γ^{25} = (0 1 0 0 0 1) ^{T}、 γ^{26} = (1 1 1 0 0 0) ^{T}、

γ^{27} = (0 1 1 1 0 0) ^{T}、 γ^{28} = (0 0 1 1 1 0) ^{T}、 γ^{29} = (0 0 0 1 1 1) ^{T}、

γ^{30} = (1 1 0 0 1 1) ^{T}、 γ^{31} = (1 0 1 0 0 1) ^{T}、 γ^{32} = (1 0 0 1 0 0) ^{T}、

γ^{33} = (0 1 0 0 1 0) ^{T}、 γ^{34} = (0 0 1 0 0 1) ^{T}、 γ^{35} = (1 1 0 1 0 0) ^{T}、

γ^{36} = (0 1 1 0 1 0) ^{T}、 γ^{37} = (0 0 1 1 0 1) ^{T}、 γ^{38} = (1 1 0 1 1 0) ^{T}、

γ^{39} = (0 1 1 0 1 1) ^{T}、 γ^{40} = (1 1 1 1 0 1) ^{T}、 γ^{41} = (1 0 1 1 1 0) ^{T}、

γ^{42}＝（０１０１１１）^{T}、γ^{43}＝（１１１０１１）^{T}、γ^{44}＝（１０１１０１）^{T}、

γ^{45}＝（１００１１０）^{T}、γ^{46}＝（０１００１１）^{T}、γ^{47}＝（１１１００１）^{T}、

γ^{48}＝（１０１１００）^{T}、γ^{49}＝（０１０１１０）^{T}、γ^{50}＝（００１０１１）^{T}、

γ^{51}＝（１１０１０１）^{T}、γ^{52}＝（１０１０１０）^{T}、γ^{53}＝（０１０１０１）^{T}、

γ^{54}＝（１１１０１０）^{T}、γ^{55}＝（０１１１０１）^{T}、γ^{56}＝（１１１１１０）^{T}、

γ^{57}＝（０１１１１１）^{T}、γ^{58}＝（１１１１１１）^{T}、γ^{59}＝（１０１１１１）^{T}、

γ^{60}＝（１００１１１）^{T}、γ^{61}＝（１０００１１）^{T}、γ^{62}＝（１０００００１）^{T}

これより、上記数２１に示す行列Ｈ”は、Ｈ”＝［γ^{0} γ^{1} γ^{2} γ^{3} γ^{4} γ^{5} γ^{18} γ^{20}］と表すことができる。よって、上記数１に示したパリティ検査行列（Ｈ）は、Ｈ’及びＨ”を用いて、具体的な（５０４、４７８）［$4_{1/8}$ＥＣ］$_{2}$符号を構成することができる。この符号の最後から４１４列削除することによって得られた（９０、６４）［$4_{1/8}$ＥＣ］$_{2}$符号の検査行列を第３図に示す。

なお、第３図に示すように、パリティ検査行列（Ｈ）のＨ’及びＨ”に相当する行単位の箇所を明確化するため、その境界に実線を引いて示している。

いま、パリティ検査行列（Ｈ）の構成要素であるγ^{3}Ｈ”が、第３図に示す実際のビット値と対応しているかを具体的に確認する。前述した

ように、H”＝[$\gamma^0\gamma^1\gamma^2\gamma^3\gamma^4\gamma^5\gamma^{18}\gamma^{20}$]より、$\gamma^3$H”＝$\gamma^3$[$\gamma^0\gamma^1\gamma^2\gamma^3\gamma^4\gamma^5\gamma^{18}\gamma^{20}$]＝[$\gamma^3\gamma^4\gamma^5\gamma^6\gamma^7\gamma^8\gamma^{21}\gamma^{23}$]となる。このべき表示をベクトル表示（例えば、$\gamma^0$＝（１０００００）^{T}等）に変換すれば、第３図に示すａ部分に相当し、このａ部分は明らかにパリティ検査行列（Ｈ）のγ^3H”部分である。同様にして、パリティ検査行列（Ｈ）における他の構成要素についても、ベクトル表示に変換すれば、全体として第３図に示すビット表現の２６行９０列の構成となる。

ここで、$\mu_1=\mu_2=3$に対しては、Ｍｉｎ（（$\mu_1+\mu_2$）ｔ、ｂ）＝Ｍｉｎ（６、８）＝６より、行列Ｈ’として、任意の６列が線形独立な６ビット誤り検出符号の検査行列を構成すればよい。この一例として、下記数２２に示す７×８行列を示す。

【数２２】

$$H'=\begin{bmatrix}1&0&0&0&0&0&0&1\\0&1&0&0&0&0&0&1\\0&0&1&0&0&0&0&1\\0&0&0&1&0&0&0&1\\0&0&0&0&1&0&0&1\\0&0&0&0&0&1&0&1\\0&0&0&0&0&0&1&1\end{bmatrix}$$

また、Ｍｉｎ（$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ、ｂ）＝Ｍｉｎ（３、８）＝３より、行列Ｈ”として、任意の３列が線形独立な３ビット誤り検出符号の検査行列を構成すればよい。この一例として、下記数２３に示す４×８行列を示す。

【数２３】

$$H''=\begin{bmatrix}1&1&1&1&1&1&1&1\\0&0&0&0&1&1&1&1\\0&0&1&1&0&0&1&1\\0&1&0&1&0&1&0&1\end{bmatrix}$$

また、ｔ＝２に対しては、例えば、$\mu_1=\mu_2=3$、$p_1=p_2=2$のパラメータを有する［$T_{2/8}EC$］$_2$符号では、Ｍｉｎ（（$\mu_1+\mu_2$）ｔ、ｂ）＝Ｍｉｎ（１２、８）＝８より、行列Ｈ’として、任意の８列が線形独立な行列を構成すればよい。この一例として、上記数２０に示す８×８（８行８列を意味する）単位行列を構成すればよい。

また、Ｍｉｎ（$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ、ｂ）＝Ｍｉｎ（６、８）＝６より、行列Ｈ”として、任意の６列が線形独立な６ビット誤り検出符号の検査行列を構成すればよい。この一例として、下記数２４に示す７×８行列を示す。

【数２４】

$$H''=\begin{bmatrix}1&0&0&0&0&0&0&1\\0&1&0&0&0&0&0&1\\0&0&1&0&0&0&0&1\\0&0&0&1&0&0&0&1\\0&0&0&0&1&0&0&1\\0&0&0&0&0&1&0&1\\0&0&0&0&0&0&1&1\end{bmatrix}$$

なお、以上は最近最も使用されている、バイト幅ｂ＝８ビット入出力を有する半導体メモリ素子を対象として、パリティ検査行列Ｈ（符号）の一例を構成して説明したが、他にも多く使用されている多ビット入出力素子、例えば、ｂ＝１６ビット入出力を有する素子等に対しても、本発明の誤りバイト数を制限したバイト内複数スポッティバイト誤り制御

符号のパリティ検査行列Hを構成可能であることは明らかである。また、1以上の任意の整数値を有するtに対しても、そして、任意のp_1、p_2、μ_1、μ_2に対しても、本発明の誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号のパリティ検査行列Hを構成可能であることは明らかである。

このように、上記では、誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号のパリティ検査行列（H）を示したが、H'の構成をb次の単位行列にすることにより、バイト内複数スポッティバイト誤りの訂正能力を超えた1バイト誤りを検出する機能を有する、誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号を構成することができる。

すなわち、バイト誤り検出機能を有する誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号とは、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正するとともに、p_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出し、かつ1バイトに生じるバイト誤りを検出する機能を備える符号である。以下では、この符号を$[\mu_1 t/b\mathrm{EC}]_{p_1}-[\mu_2 t/b\mathrm{ED}]_{p_2}-\mathrm{S}_b\mathrm{ED}$符号と表す。

本実施形態では、$[\mu_1 t/b\mathrm{EC}]_{p_1}-[\mu_2 t/b\mathrm{ED}]_{p_2}-\mathrm{S}_b\mathrm{ED}$符号に用いるR行N列を有するパリティ検査行列（H）を、下記数25に示すような構成としている。

【数25】

$$H=\begin{bmatrix} I_b & I_b & \cdots & I_b & \cdots & I_b \\ \gamma^0 H'' & \gamma^1 H'' & \cdots & \gamma^i H'' & \cdots & \gamma^{n-1} H'' \\ \gamma^0 H'' & \gamma^2 H'' & \cdots & \gamma^{2i} H'' & \cdots & \gamma^{2(n-1)} H'' \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma^0 H'' & \gamma^{p_1+p_2-1} H'' & \cdots & \gamma^{(p_1+p_2-1)i} H'' & \cdots & \gamma^{(p_1+p_2-1)(n-1)} H'' \end{bmatrix}$$

ただし、Ｒ＝ｂ＋（ｐ$_1$＋ｐ$_2$－１）ｒ、Ｎ＝ｂｎ、０≦ｉ≦ｎ－１、ｎ＝２^r－１、Ｉ$_b$はｂ次の単位行列である。

上記数２５を構成する行列Ｈ”は、下記数２６に示すｒ次の列ベクトルｈ$_j$”（０≦ｊ≦ｂ－１）から構成されるｒ行ｂ列（ｒ≦ｂ）の２元行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立である。ここで、$\lfloor x\rfloor$はｘを超えない最大の整数を表す。

【数２６】

Ｈ”＝［ｈ$_0$”、ｈ$_1$”、・・・、ｈ$_{b-1}$”］

ここで、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ≧ｂのとき、前記行列Ｈ”は単位行列を含む正則行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＜ｂのとき、前記行列Ｈ”は最小ハミング距離$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＋１を有する符号の検査行列、すなわち、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔビット誤り検出機能を有する（ｂ、ｂ－ｒ）符号のパリティ検査行列に等しい。

次に、γをＧＦ（２）のｒ次の拡大体ＧＦ（２^r）の原始元とするときに、γ^{i}Ｈ”は下記数２７で定義することができる。

【数２７】

γ^{i}Ｈ”＝［γ^{i}ｈ$_0$”、γ^{i}ｈ$_1$”、…、γ^{i}ｈ$_{b-1}$”］

このとき、最大符号長Ｎは、Ｎ＝ｂ（２^r－１）である。

上記数２５に示したパリティ検査行列（Ｈ）は、パラメータｔ≧ｂ／

$\lfloor(\mu_1+\mu_2)/2\rfloor$とした場合に、H”は任意のｂ列が線形独立な行列となるため、従来のｐ$_1$バイト誤りを訂正するとともにｐ$_2$バイト誤りを検出する機能を有する、ｐ$_1$訂正ｐ$_2$検出ＲＳ符号のパリティ検査行列と一致する。

また、本実施形態では、$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}-S_bED$符号の伸長符号を定義することができる。伸長$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}-S_bED$符号に用いるＲ行Ｎ列を有するパリティ検査行列（Ｈ）は、下記数２８に示すような構成とすることができる。

【数２８】

$$H=\begin{bmatrix} I_b & I_b & \cdots & I_b & \cdots & I_b & I_b & O_{b\times r'} \\ \gamma^0 H'' & \gamma^1 H'' & \cdots & \gamma^i H'' & \cdots & \gamma^{n-1} H'' & O_{r'\times b} & O_{r'\times r'} \\ \gamma^0 H'' & \gamma^2 H'' & \cdots & \gamma^{2i} H'' & \cdots & \gamma^{2(n-1)} H'' & O_{r'\times b} & O_{r'\times r'} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \gamma^0 H'' & \gamma^{p_1+p_2-1} H'' & \cdots & \gamma^{(p_1+p_2-1)i} H'' & \cdots & \gamma^{(p_1+p_2-1)(n-1)} H'' & O_{r'\times b} & I_{r'} \end{bmatrix}$$

ただし、ｒ’≧ｒであり、Ｒ＝ｂ＋（ｐ$_1$＋ｐ$_2$－１）ｒ’、Ｎ＝ｂ（ｎ＋１）＋ｒ’、０≦ｉ≦ｎ－１、ｎ＝２$^{r'}$－１、Ｉ$_b$はｂ次の単位行列であり、$I_{r'}$はｒ’次の単位行列で、Oは零行列である。

上記数２８を構成する行列Ｈ”は、下記数２９に示すｒ次の列ベクトルｈ$_j$”（０≦ｊ≦ｂ－１）から構成されるｒ行ｂ列（ｒ≦ｂ）の２元行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立である。ここで、$\lfloor x\rfloor$はｘを超えない最大の整数を表す。

【数２９】

Ｈ”＝［ｈ$_0$”、ｈ$_1$”、・・・、ｈ$_{b-1}$”］

ここで、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ≧ｂのとき、前記行列Ｈ”は単位行列を含む正

則行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＜ｂのとき、前記行列Ｈ”は最小ハミング距離$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＋１を有する符号の検査行列、すなわち、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔビット誤り検出機能を有する（ｂ、ｂ－ｒ）符号のパリティ検査行列に等しくなる。

次に、γをＧＦ（２）のｒ’次の拡大体ＧＦ（$2^{r'}$）の原始元とするときに、γ^{i}Ｈ”は下記数３０で定義することができる。

【数３０】

$\gamma^{i}H'' = [\gamma^{i}\Phi(h_0''), \gamma^{i}\Phi(h_1''), \cdots, \gamma^{i}\Phi(h_{b-1}'')]$

ここで、Φは加法のもとでＧＦ（2^{r}）からＧＦ（$2^{r'}$）への単射準同型写像（injective homomorphism）であり、すなわち、Φ：ＧＦ（2^{r}）→ＧＦ（$2^{r'}$）であることによって効果的に達成される。このとき、最大符号長Ｎは、Ｎ＝ｂ・$2^{r'}$＋ｒ’である。数２８のＨの２行目から下の行に用いるＨ”の行数ｒ’は、ｒより大きい値をとることができ、数２８で表される符号は、（$\mu_1+\mu_2-1$）ビットずつ検査ビット数を増大させることにより、任意の符号長に対して構成できる符号であることに注意する必要がある。

上記数２８に示したパリティ検査行列（Ｈ）は、パラメータｔ≧ｂ／$\lfloor(\mu_1+\mu_2)/2\rfloor$とした場合に、Ｈ”は任意のｂ列が線形独立な行列となるため、従来のp_1バイト誤りを訂正するとともにp_2バイト誤りを検出する機能を有する、p_1訂正p_2検出２次伸長ＲＳ符号のパリティ検査行列と一致する。

次に、上記数２５に示したパリティ検査行列（Ｈ）で表される符号が、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正するとともに、p_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出

し、かつ１バイトに生じるバイト誤りを検出する機能を有していることを具体的に示す。パリティ検査行列Ｈ＝［Ｈ$_0$、Ｈ$_1$、・・・、Ｈ$_{n-1}$］が上記数９及び下記数３１に示す必要十分条件を満たしていることを示せばよい。ただし、Ｈ$_i$はＲ×ｂの小行列である。

【数３１】

$$e_{i_b}\cdot H_{i_b}^T+e_{i_1}\cdot H_{i_1}^T+\cdots+e_{i_{\nu_1}}\cdot H_{i_{\nu_1}}^T+e_{i_{\nu_1+1}}\cdot H_{i_{\nu_1+1}}^T+\cdots+e_{i_{\nu_1+\nu_2}}\cdot H_{i_{\nu_1+\nu_2}}^T$$
$$+\cdots+e_{i_{\nu_1+\nu_2+\cdots+\nu_{\lambda-1}+1}}\cdot H_{i_{\nu_1+\nu_2+\cdots+\nu_{\lambda-1}+1}}^T+\cdots+e_{i_{\nu_1+\nu_2+\cdots+\nu_\lambda}}\cdot H_{i_{\nu_1+\nu_2+\cdots+\nu_\lambda}}^T\neq 0$$

$$0<\sum_{x=1}^{\nu_1+\cdots+\nu_\lambda} w_M\left(e_{i_x}\right)\le\mu_1,0<\sum_{x=1}^{\lambda}\nu_x\le p_1$$

$$0\le\nu_1\le p_1,0\le\nu_2\le\min\left(\lfloor(\mu_1+\mu_2)/2\rfloor,p_1\right),\cdots,0\le\nu_\lambda\le\min\left(\lfloor(\mu_1+\mu_2)/\lambda\rfloor,p_1\right)$$

ここで、e_{i_b}は誤りベクトルｅにおけるｉ$_b$番目のバイトに生じるバイト誤りで、e_{i_x}は誤りベクトルｅにおけるｉ$_x$番目のバイトで、$\lambda=\lceil b/t\rceil$、$\lfloor y\rfloor$はｙ以下である最大の整数、$\lceil z\rceil$はｚ以上である最小の整数をそれぞれ表す。

また、ｉ$_b$、ｉ$_1$、ｉ$_2$、…、$i_{\nu_1+\nu_2+\cdots+\nu_\lambda}$（０≦ｉ$_b$、ｉ$_1$、ｉ$_2$、・・・、$i_{\nu_1+\nu_2+\cdots+\nu_\lambda}$≦ｎ－１）は、すべて異なるものとし、$w_M(e)$は上記数１０で定義され、スポッティバイト誤りの個数を表している。

上記数９に示した必要十分条件と同様に、上記数９及び上記数３１に示した条件を満たしていることを示すことができる。

また、上記数２８に示したパリティ検査行列（Ｈ）で表される伸長符号も、同様に上記数９及び上記数３１に示した必要十分条件を満たしていることを示すことができる。

このように、上記では、誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号のパリティ検査行列（Ｈ）、及びバイト誤り検出機能を有する誤りバイト数を制限したバイト内複数スポッティバイト

誤り制御符号のパリティ検査行列（Ｈ）の構成法の幾つかの例を示したが、次に、本発明のパリティ検査行列（Ｈ）を用いて行う符号化処理と復号処理の具体的な方法とそれぞれの回路構成について、上記数１に示したパリティ検査行列（Ｈ）で表される、誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号（$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}$符号）を例に取って説明する。

＜Ａ＞符号化の方法及びこれを実現する回路の構成について

まず、第１図に示す符号化回路２が行う符号化処理の具体的な方法について説明する。

符号化回路２は、入力情報データＤ（行ベクトル）３０と、第３図に示すパリティ検査行列（Ｈ）を基本行変形して得られるパリティ検査行列（Ｈ）の情報部$Ｈ_{R}$を用いて、検査情報Ｃ（行ベクトル）を生成する。この検査情報（Ｃ）の生成は、下記数３２により求めることができる。

【数３２】

$Ｃ＝Ｄ・Ｈ_{R}{}^{T}$

具体的には、符号化回路２は、入力情報データ（Ｄ）３０の情報ビット長６４ビット及び検査ビット長２６ビットを基に、パリティ検査行列Ｈである（５０４、４７８）$[４_{1/8}ＥＣ]_{2}$符号を構成した後に、後半４１４ビットに相当する列ベクトル４１４列を削除することにより、符号ビット長９０ビット、情報長６４ビットを有する第３図に示す（９０、６４）$[４_{1/8}ＥＣ]_{2}$符号を構成することができる。この符号のパリティ検査行列Ｈを基本行変形すると、組織符号としてのパリティ検査行列Ｈを構成することができる。第４図は、（９０、６４）$[４_{1/8}$

ＥＣ］$_2$組織符号のパリティ検査行列を示している。

第４図に示す（９０、６４）［$4_{1/8}$ＥＣ］$_2$組織符号のパリティ検査行列において、d_0～d_{63}が情報ビットに相当し、c_0～c_{25}が検査ビットに相当する。ここで、第４図に示す（９０、６４）［$4_{1/8}$ＥＣ］$_2$組織符号のパリティ検査行列において、情報ビットに相当するd_0～d_{63}の列ベクトルを集めた行列が、パリティ検査行列の情報部H_Rとなる。

ここで、符号化では第４図に示すパリティ検査行列を使用し、復号では第３図に示すパリティ検査行列を使用する。基本行変形して得られた行列は、もとの行列と等価であり、第４図に示すパリティ検査行列で表される符号は、第３図に示すパリティ検査行列で表される符号と、その性質は全く変化しない。これより、符号化においては、第４図に示すパリティ検査行列を使用することとする。

次に、符号化回路２は、上記数３２を基に、２６ビットの検査情報（Ｃ）を生成する。上記数３２に示すＣ＝Ｄ・H_R^Tから明らかなように、例えば、検査情報（Ｃ）における検査ビットc_0を生成するには、第４図に示す符号化用（９０、６４）［$4_{1/8}$ＥＣ］$_2$組織符号の行列の１行目において、c_0に対応するビットを除いて“１”が存在する箇所に対応するビットのＧＦ（２）上の和（すなわち、２を法とする和：ｍｏｄ２）をとる演算を行えばよい。

例えば、検査ビットc_0を求める場合に、c_0に対応するビットを除いて“１”が存在しているのは、d_1、d_4、d_5、d_6、d_8、d_{11}、d_{12}、d_{20}、d_{21}、d_{24}、d_{26}、d_{31}、d_{33}、d_{37}、d_{38}、d_{39}、d_{40}、d_{41}、d_{42}、d_{43}、d_{44}、d_{47}、d_{48}、d_{52}、d_{54}、d_{55}、d_{56}、d_{57}、d_{58}であり、これら２９個のビットについて、

２を法とする和（ｍｏｄ２）をとればよい。

これを実現するには、第５図に示すように、前記２９ビットを多入力パリティチェック回路２０（０）を構成すればよく、この多入力パリティチェック回路２０（０）の出力が検査ビットc_0となる。他の検査ビットc_1～c_{25}についても、検査ビットc_0と同様にして生成することができる。２６ビットの検査ビットc_0～c_{25}を並列に生成する符号化回路２（多入力パリティチェック回路２０（０）～２０（２５）からなる）の上位１４ビットの構成を第５図に、下位１２ビットの構成を第６図にそれぞれ示す。

次に、符号化回路２によって生成した検査情報（Ｃ）を、入力情報データ（Ｄ）３０に付加した符号語を下記数３３に示す。これが回路３に送られる送信語（Ｖ）（９０ビットのベクトル）となる。

【数３３】

$$V=[d_0、d_1、\cdots、d_{57}、d_{58}、c_0、d_{59}、c_1、\cdots、c_3、d_{60}、c_4、d_{61}、c_5、\cdots、c_9、d_{62}、c_{10}、d_{63}、c_{11}、\cdots、c_{25}]$$

＜Ｂ＞復号の方法及びこれを実現する回路の構成について

次に、送信語（Ｖ）３１に誤りが生じたとき（すなわち、受信語３２に誤りが含まれているとき）に、その誤りに対する訂正・検出処理が、第３図に示す（９０、６４）$[4_{1/8}EC]_2$符号を用いて生成されるシンドローム（Ｓ）３３の値に基づいて可能なことを示す。

第３図に示す（９０、６４）$[4_{1/8}EC]_2$符号の検査行列は、$p_1=p_2=2$を有することから、上記数１に示すように、パリティ検査行列Ｈ’、Ｈ”による４段構成を有する。これより、１段目より得られ

るｑ＝８ビットのシンドロームをS_I、２段目より得られるｒ＝６ビットのシンドロームをS_{II}、３段目より得られるｒ＝６ビットのシンドロームをS_{III}、４段目より得られるｒ＝６ビットのシンドロームをS_{IV}とする。

これらの情報を用いて、次のように誤りの訂正・検出を行う。受信語（Ｖ’）３２とパリティ検査行列（Ｈ）とから、シンドローム（Ｓ）３３は、下記数３４により求めることができる。

【数３４】

$$V' \cdot H^T = S = [S_I S_{II} S_{III} S_{IV}]$$

ここで、$S_I \in GF(2^q)$は、ｑ次の行ベクトルである。また、S_{II}、S_{III}、$S_{IV} \in GF(2^r)$は、それぞれｒ次の行ベクトルである。

上記数５に示す伸長符号の場合に、$S_I \in GF(2^q)$は、ｑ次の行ベクトルである。また、S_{II}、S_{III}、$S_{IV} \in GF(2^{r'})$は、それぞれｒ’次の行ベクトルである。

なお、シンドローム（Ｓ）３３は、例えば、i_I番目のバイトにe_Iの誤り、i_{II}番目のバイトにe_{II}の誤り（$w_M(e_I) + w_M(e_{II}) \leqq \mu_1 + \mu_2$）が生じているときに、下記数３５となる。

【数３５】

$$S = [e_I \cdot H'^T + e_{II} \cdot H'^T, \gamma^i e_I \cdot H''^T + \gamma^j e_{II} \cdot H''^T, \gamma^{2i} e_I \cdot H''^T + \gamma^{2j} e_{II} \cdot H''^T, \gamma^{3i} e_I \cdot H''^T + \gamma^{3j} e_{II} \cdot H''^T]$$

S_Iに注目すると、下記数３６が成り立つ。数３６において、Ｈ’は任意のＭｉｎ（（$\mu_1 + \mu_2$）ｔ、ｂ）列が線形独立な行列であるため、$e_I + e_{II}$（＝e_Aとする）を求めることができる。第３図に示す（９０、６４）$[4_{1/8}EC]_2$符号の場合に、Ｈ’は単位行列であるため

、$e_A = e_I + e_{II} = S_I$となる。

【数３６】

$(e_I + e_{II}) \cdot H'^T = S_I$

次に、$e_I + e_{II}$にH”^{T}を右から乗算し、乗算した結果をS’$_I \in GF(2^r)$とする。つまり、下記数３７の関係が成立する。

【数３７】

$S'_I = (e_I + e_{II}) \cdot H''^T$

ここで、$e_I \cdot H''^T$、$e_{II} \cdot H''^T$（$\in GF(2^r)$）をGF（2^r）上の誤りe’$_I$、e’$_{II}$とし、S’＝［S’$_I S_{II} S_{III} S_{IV}$］とすると、下記数３８の関係が成り立つ。

【数３８】

$S' = [S'_I S_{II} S_{III} S_{IV}] = [e'_I + e'_{II}, \gamma^i e'_I + \gamma^j e'_{II}, \gamma^{2i} e'_I + \gamma^{2j} e'_{II}, \gamma^{3i} e'_I + \gamma^{3j} e'_{II}]$

数３８に示すS’は、２バイト誤り訂正ＲＳ符号（距離ｄ＝５）において誤りe’$_I$、e’$_{II}$におけるシンドロームの式と一致する。よって、距離ｄ＝５を有するGF（2^r）上のＲＳ符号の復号法を用いることで、誤り位置i$_I$、i$_{II}$及びGF（2^r）上の誤りe’$_I$、e’$_{II}$を求めることができる。また、誤り位置が求まらない場合に、誤りの訂正が不可能として誤りを検出する。

次に、GF（2^r）上の誤りe’$_I$、e’$_{II}$から、GF（2^b）上の誤りe$_I$、e$_{II}$を求める。つまり、$e_I \cdot H''^T = e'_I$、$e_{II} \cdot H''^T = e'_{II}$の関係からe$_I$及びe$_{II}$を求める。求めた結果を$\hat{e}_I$、$\hat{e}_{II}$とする。

H”は任意のＭｉｎ（２ｔ、ｂ）列が線形独立な行列であるため、１バイトにｔビット以上の誤りが生じた場合に、その誤りは、誤訂正される（すなわち、$e_x \neq \hat{e}_x$となる）か、または制御能力外誤りとして検出

される。しかし、ｔビット以上の誤りが生じるバイトは、高々１つであるため、$e_x \neq \hat{e}_x$となるバイトは高々１つである。

そして、下記数３９により、誤訂正されているかどうかを判定することができる。つまり、下記数３９により、$e_x = \hat{e}_x$が成立するかどうかを判定することができる。すなわち、下記数３９が成立している場合に、$e_x = \hat{e}_x$となり誤訂正はされていないと判定され、下記数３９が成立していない場合に、$e_x \neq \hat{e}_x$となり誤訂正はされていると判定される。

【数３９】

$$w_M(\hat{e}_x + e_A) \leq \mu_1 - w_M(\hat{e}_x)$$

なお、数３９に示す関係は、$w_M(e)$が三角不等式を満足していることから、示すことができる。また、ＧＦ（２）上のハミング距離に対する誤訂正の判定は、非特許文献１１に詳しく述べられている。

最後に、誤訂正されている（すなわち、$e_x \neq \hat{e}_x$となっている）バイト、または制御能力外誤りとして検出されたバイトにおいて、上記において正しく訂正された誤りパターンおよびe_Aの排他的論理和をとることにより、誤訂正されているバイトにおける誤りパターンを復元することができる。

なお、任意のp_1、p_2、μ_1、μ_2に対し、つまり、前記で示した例と同様に、上記数３９により、誤訂正されているかどうかを判定し、高々１つである誤りバイトの誤りパターンを、正しく訂正された誤りパターンとe_Aとに基づいて、復元可能であることは明らかである。

ＲＳ符号に対する復号は、スポッティバイト誤りを訂正するバイト数p_1が小さい符号に対しては、並列復号法を用いて実現することができる。この並列復号法の一般的構成は、『バースト誤りパターン生成方法及びバーストおよびバイト誤り検出・訂正装置』（特許文献５参照）に

開示されている。

しかしながら、特許文献５に開示された並列復号法は、高速な復号が可能であるが、符号のスポッティバイト誤りを訂正するバイト数p_1が大きくなると、回路量が大きくなるという問題が生じる。スポッティバイト誤りを訂正するバイト数p_1が大きい符号に対しては、例えば、バーレカンプ・マッシィ法などの逐次復号法を用いることができるが、逐次復号法による復号は、逐次（シリアル）に行われるため、復号が遅いといった問題が生じる。なお、バーレカンプ・マッシィ法を用いたＲＳ符号の復号回路の構成は、非特許文献１２に開示されている。

特許文献５に開示された並列復号法によれば、例えば、$p_1 = p_2 = 2$の符号に対しては、次のような手順により復号する。

まず、４ｒ×ｎｒを有するＧＦ（2^r）上のＲＳ符号のパリティ検査行列$\tilde{H} = [\tilde{H}_0 \quad \tilde{H}_1 \quad \cdots \quad \tilde{H}_{n-1}]$に対し、４ｒ×２ｒの行列$\tilde{H}_{(i,j)} = [\tilde{H}_i \quad \tilde{H}_j]$を構成する。ただし、$\tilde{H}_i$は４ｒ×ｒの小行列である。

次に、この行列$\tilde{H}_{(i,j)}$に４ｒ×２ｒの行列Ｂ$_{(i,j)}$を付加し、４ｒ×４ｒの正則行列$A_{(i,j)} = [\tilde{H}_{(i,j)} \quad B_{(i,j)}]$を構成する。

そして、Ａ$_{(i,j)}$の逆行列$A^{-1}_{(i,j)}$を求めると、下記数４０の関係が成り立つ。

【数４０】

$$A^{-1}_{(i,j)} = \begin{bmatrix} \tilde{H}^{\dagger}_{(i,j)} \\ B^{\dagger}_{(i,j)} \end{bmatrix} = \begin{bmatrix} \tilde{H}^{\dagger}_{i} \\ \tilde{H}^{\dagger}_{j} \\ B^{\dagger}_{(i,j)} \end{bmatrix}$$

ここで、$\tilde{H}^{\dagger}_i$はｒ×４ｒの行列であり、$\tilde{H}^{\dagger}_i \cdot \tilde{H}_i = I_r$となる。ただし、Ｉ$_r$はｒ×ｒの単位行列である。また、$\tilde{H}^{\dagger}_j$はｒ×４ｒの行列であり、$\tilde{H}^{\dagger}_j \cdot \tilde{H}_j = I_r$

となる。また、$B^{\dagger}_{(i,j)}$は２ｒ×４ｒの行列であり、$B^{\dagger}_{(i,j)} \cdot B_{(i,j)} = I_{2r}$となる。ただし、$I_{2r}$は２ｒ×２ｒの単位行列である。また、$\tilde{H}^{\dagger}_{(i,j)}$は２ｒ×４ｒの行列であり、$\tilde{H}^{\dagger}_{(i,j)} \cdot \tilde{H}_{(i,j)} = I_{2r}$となる。

前記$\tilde{H}^{\dagger}_{i}$、$\tilde{H}^{\dagger}_{j}$、$B^{\dagger}_{(i,j)}$に対し、S’・$B^{\dagger T}_{(i,j)}$＝０となるバイト位置ｉ，ｊが誤りの位置i_{I}、i_{II}となり、S’・$\tilde{H}^{\dagger T}_{i}$、S’・$\tilde{H}^{\dagger T}_{j}$が誤りの大きさ$e'_{I}$、$e'_{II}$となる。これから、２バイトの誤りを訂正することができる。また、S’・$B^{\dagger T}_{(i,j)}$＝０となるバイト位置ｉ，ｊが存在しない場合に、誤りの訂正が不可能として誤りを検出する。

次に、本発明において、前述した$p_1 = p_2 = 2$の符号に対する並列復号方法を実現する復号回路４の概略構成を第７図に示す。

第１図に示したように、復号回路４は、シンドローム生成回路１と誤り訂正回路５とから構成されている。

また、第７図に示すように、誤り訂正回路５は、H’復号回路６と、H”乗算回路７と、ＧＦ（2^{r}）上の並列復号回路８と、ＧＦ（2^{b}）上の誤り生成回路９と、反転回路１０とから構成されている。

ここで、H’復号回路６は、シンドローム（Ｓ）３３の上位ｑビットS_{I}３６を基にして、バイトごとの誤りの和（e_{A}）３８を生成する。

また、H”乗算回路７は、バイトごとの誤りの和（e_{A}）３８を基にして、シンドローム（Ｓ）３３の上位ｑビットS_{I}３６とH”の転置（H”^{T}）の積（S'_{I}）３９を生成する。

次に、ＧＦ（2^{r}）上の並列復号回路８は、シンドローム（Ｓ）３３の上位ｑビットS_{I}３６とH”の転置（H”^{T}）の積（S'_{I}）３９、及びシンドローム（Ｓ）３３の下位３ｒビット$S_{II}S_{III}S_{IV}$３７を基にして、ＧＦ（2^{r}）上のビット誤りポインタ（ｅ’）４０、及びＧＦ（2^{r}）上の並列復号において誤り訂正が不可能とみなされたときに出力さ

れるＤＳ（０）（Detection Signal）４１を生成する。

そして、ＧＦ（2^b）上の誤り生成回路９は、バイトごとの誤りの和（e_A）３８、及びＧＦ（2^r）上のビット誤りポインタ（ｅ’）４０を基にして、ＧＦ（2^r）上の誤りをＧＦ（2^b）上の誤りに変換して、ＧＦ（2^b）上のビット誤りポインタ（ｅ）４２、及び誤り訂正が不可能とみなされたときに出力されるＤＳ（１）（Detection Signal）４３を出力する。

最後に、反転回路１０は、ＧＦ（2^b）上のビット誤りポインタ（ｅ）４２を基にして、誤りビットに該当する受信語（Ｖ’）３２のビット値を反転させることにより、受信語（Ｖ’）３２の誤りを訂正する。

次に、第７図に示される復号回路４で行われる動作手順の概略を説明する。

まず、シンドローム生成回路１では、Ｎビットからなる受信語（Ｖ’）３２が入力されると、Ｒビットのシンドローム（Ｓ）３３を生成する。

次に、生成されたＲビットのシンドローム（Ｓ）３３は、上位ｑビットS_I３６と、下位３ｒビット$S_{II}S_{III}S_{IV}$３７とに分けられる。シンドローム（Ｓ）３３の上位ｑビットS_I３６は、Ｈ’復号回路６に入力される。また、シンドローム（Ｓ）３３の下位３ｒビット$S_{II}S_{III}S_{IV}$３７は、ＧＦ（2^r）上の並列復号回路８に入力される。

そして、Ｈ’復号回路６では、詳細は後述するが、シンドローム生成回路１で生成されたシンドローム（Ｓ）３３の上位ｑビットS_I３６を基にして、ｂビットのバイトごとの誤りの和（e_A）３８を出力する。

そして、Ｈ”乗算回路７では、詳細は後述するが、Ｈ’復号回路６で生成されたバイトごとの誤りの和（e_A）３８を基にして、シンドロー

ム（Ｓ）３３の上位ｑビットS_I３６とＨ”の転置（Ｈ”^{T}）の積（Ｓ’$_I$）３９を出力する。

また、ＧＦ（２^r）上の並列復号回路８では、詳細は後述するが、Ｈ”乗算回路７で生成されたシンドローム（Ｓ）３３の上位ｑビットS_I３６とＨ”の転置（Ｈ”^{T}）の積（Ｓ’$_I$）３９、及びシンドローム生成回路１で生成されたシンドローム（Ｓ）３３の下位３ｒビット$S_{II}S_{III}S_{IV}$３７を基にして、ＧＦ（２^r）上の誤りを指摘する$r\lceil N/b\rceil$ビットのＧＦ（２^r）上のビット誤りポインタ$e'=\left(e'_0 \quad e'_1 \quad \cdots \quad e'_{r\lceil N/b\rceil-1}\right)$４０を生成する。ここで、$\lceil y\rceil$はｙを超える最小の整数を示す。また、ＧＦ（２^r）上の並列復号回路８では、ＧＦ（２^r）上の並列復号において、誤り訂正が不可能とみなされたときに、訂正不可能誤り検出信号（ＤＳ（０））４１を出力する。

そして、ＧＦ（２^b）上の誤り生成回路９では、詳細は後述するが、Ｈ’復号回路６で生成されたバイトごとの誤りの和（e_A）３８、及びＧＦ（２^r）上の並列復号回路８で生成されたＧＦ（２^r）上のビット誤りポインタ（ｅ’）４０を基にして、ＧＦ（２^b）上の誤りを指摘するＮビットのＧＦ（２^b）上のビット誤りポインタｅ＝（e_0、e_1、・・・、e_{N-1}）４２を生成する。また、ＧＦ（２^b）上の誤り生成回路９では、誤り訂正が不可能とみなされたときに、訂正不可能誤り検出信号（ＤＳ（１））４３を出力する。

次に、反転回路１０では、ＧＦ（２^b）上の誤り生成回路９で生成されたＮビットのＧＦ（２^b）上のビット誤りポインタ（ｅ）４２、及び受信語（Ｖ’）３２を基にして、ＧＦ（２^b）上のビット誤りポインタ（ｅ）４２に対応した受信語（Ｖ’）３２のビットを反転する。

最後に、反転回路１０では、Ｎビットの誤り訂正された受信語Ｖ*から、Ｋビットの情報語Ｄ*を取り出し、出力情報データ３４として出力する。

また、ＧＦ（２^{r}）上の並列復号回路８で生成された訂正不可能誤り検出信号（ＤＳ（０））４１と、ＧＦ（２^{b}）上の誤り生成回路９で生成された訂正不可能誤り検出信号（ＤＳ（１））４３の論理和をとることにより、訂正不可能誤り検出信号ＵＣＥ（Uncorrectable Error）３５の値を出力する。例えば、この訂正不可能誤り検出信号ＵＣＥ３５の値が“１”であれば、訂正能力を超えた誤りを検出したことを示している。

なお、上記数５に示すパリティ検査行列（Ｈ）で表される伸長符号（伸長$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}$符号）、上記数２５に示すパリティ検査行列（Ｈ）で表されるバイト誤り検出機能を有する誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号、及び上記数２８に示すパリティ検査行列（Ｈ）で表される伸長符号（伸長$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}-S_b ED$符号）についても、それらの復号方法及び回路構成は、上述した復号方法及び回路構成と基本的に同一である。

次に、シンドローム生成回路１、Ｈ’復号回路６、Ｈ”乗算回路７、ＧＦ（２^{r}）上の並列復号回路８、ＧＦ（２^{b}）上の誤り生成回路９、及び反転回路１０のそれぞれの具体的な構成例を、第３図に示す（９０、６４）［$4_{1/8}$ＥＣ］$_2$符号を用いて説明する。

＜ａ＞シンドローム生成回路１の構成

ここでは、シンドローム生成回路１の構成について詳細に説明する。

まず、シンドローム生成回路では、Ｎ＝９０（＝６４＋２６）ビットからなる受信語（Ｖ’）３２を、下記数４１に示すように表現したとき、この受信語（Ｖ’）３２を入力して、下記数４２に示す$Ｓ_{0}$、$Ｓ_{1}$、・・・、$Ｓ_{25}$からなるＲ＝２６ビットのシンドローム（Ｓ）３３を生成する。

前述したように、第３図に示す［$４_{1/8}ＥＣ］_{2}$符号のパリティ検査行列が４段構成を有することから、これに対応してシンドローム（Ｓ）３３も４個の部分に区分けし、$Ｓ_{I}$＝（$Ｓ_{0}Ｓ_{1}Ｓ_{2}Ｓ_{3}Ｓ_{4}Ｓ_{5}Ｓ_{6}Ｓ_{7}$）、$Ｓ_{II}$＝（$Ｓ_{8}Ｓ_{9}Ｓ_{10}Ｓ_{11}Ｓ_{12}Ｓ_{13}$）、$Ｓ_{III}$＝（$Ｓ_{14}Ｓ_{15}Ｓ_{16}Ｓ_{17}Ｓ_{18}Ｓ_{19}$）、$Ｓ_{IV}$＝（$Ｓ_{20}Ｓ_{21}Ｓ_{22}Ｓ_{23}Ｓ_{24}Ｓ_{25}$）とする。

【数４１】

$Ｖ’＝（ｖ_{0}’、ｖ_{1}’、\cdots、ｖ_{89}’）$

【数４２】

$Ｓ＝（Ｓ_{I}｜Ｓ_{II}｜Ｓ_{III}｜Ｓ_{IV}）$

$＝（Ｓ_{0}Ｓ_{1}Ｓ_{2}Ｓ_{3}Ｓ_{4}Ｓ_{5}Ｓ_{6}Ｓ_{7}｜Ｓ_{8}Ｓ_{9}Ｓ_{10}Ｓ_{11}Ｓ_{12}Ｓ_{13}｜Ｓ_{14}Ｓ_{15}Ｓ_{16}Ｓ_{17}Ｓ_{18}Ｓ_{19}｜Ｓ_{20}Ｓ_{21}Ｓ_{22}Ｓ_{23}Ｓ_{24}Ｓ_{25}）$

上記数３４に示したように、シンドロームＳ＝Ｖ’・$Ｈ^{T}$より、転置したパリティ検査行列（$Ｈ^{T}$）の各行は、入力する情報に対応し、例えば、行列（$Ｈ^{T}$）の第０行は、入力情報の$ｖ_{0}$’に対応している。

そこで、シンドローム（Ｓ）３３の各ビットを生成するには、符号であるパリティ検査行列（$Ｈ^{T}$）の各列方向に、“１”を有するビットに対応する受信語（Ｖ’）３２の受信情報をＧＦ（２）上で加算（ｍｏｄ２の計算）すればよい。

第８図及び第９図は、第３図に示した（９０、６４）［$４_{1/8}ＥＣ］_{2}$符号のパリティ検査行列を基に構成したシンドローム生成回路１を示

す図である。第８図では、S_0、S_1、・・・、S_{13}といったシンドローム（Ｓ）３３の上位１４ビット、また、第９図では、S_{14}、S_{15}、・・・、S_{25}といったシンドローム（Ｓ）３３の下位１２ビットをそれぞれ生成している。

第８図及び第９図に示すように、本実施例では、シンドローム生成回路１において、まず、２６個の多入力パリティチェック回路２１（ｐ）、（ｐ＝０、１、・・・、２５）に、第３図に示した（９０、６４）［$4_{1/8}$ＥＣ］$_2$符号のパリティ検査行列の対応する各行のビット“１”を有する受信情報（v_q’）、（ｑ＝０、１、・・・、８９）が入力する。

次に、各多入力パリティチェック回路２１（ｐ）、（ｐ＝０、１、・・・、２５）において、前記入力された受信情報（v_q’）に対して、２を法とする和（ｍｏｄ２）の計算を行い、その結果がS_pである。

具体的に、例えば、シンドローム（Ｓ）３３のビットS_0を生成するには、９０ビットの受信語v_0’、v_1’、・・・、v_{89}’のうち、第３図に示すパリティ検査行列の第０行目において、“１”が存在する箇所に相当するv_0’、v_8’、v_{16}’、v_{24}’、v_{32}’、v_{40}’、v_{48}’、v_{56}’、v_{64}’、v_{72}’、v_{80}’、v_{88}’の１２ビットについて、２を法とする和（ｍｏｄ２）をとればよい。第８図の多入力パリティチェック回路２１（０）は、前記１２ビットの受信情報を入力として、値S_0を出力するようになっている。シンドローム（Ｓ）３３の他のビットS_p、（ｐ＝１、・・・、２５）についても同様である。

＜ｂ＞Ｈ’復号回路６の構成

次に、Ｈ’復号回路６について説明する。

Ｈ’復号回路６を第１０図に示す。数３６に示した（e_I＋e_{II}）・Ｈ

$'^{T}=S_{I}$において、H'は任意のＭｉｎ（（$\mu_{1}+\mu_{2}$）ｔ、ｂ）列が線形独立な行列であるため、$e_{I}+e_{II}$（$=e_{A}$とする）を求めることができる。よって、論理合成を行うことにより、組み合わせ回路で実現することができる。

第３図に示す（９０、６４）［$4_{1/8}$ＥＣ］$_{2}$符号の場合、H'は単位行列であるため、$e_{A}=e_{I}+e_{II}=S_{I}$となる。すなわち、$e_{A}=\left(e_{A_0}e_{A_1}e_{A_2}e_{A_3}e_{A_4}e_{A_5}e_{A_6}e_{A_7}\right)$、$S_{I}=(S_{0}S_{1}S_{2}S_{3}S_{4}S_{5}S_{6}S_{7})$に対し、第１０図に示すように、$e_{A_0}=S_{0}$、$e_{A_1}=S_{1}$、$e_{A_2}=S_{2}$、$e_{A_3}=S_{3}$、$e_{A_4}=S_{4}$、$e_{A_5}=S_{5}$、$e_{A_6}=S_{6}$、$e_{A_7}=S_{7}$となる。

一方、H'が単位行列でない場合に、＜ｅ＞において後述する方法を用いる。

＜ｃ＞H"乗算回路７の構成

次に、H"乗算回路７について説明する。

H"乗算回路７を第１１図に示す。数３７に示した$S'_{I}=(e_{I}+e_{II})\cdot H"^{T}=e_{A}\cdot H"^{T}$を計算する。下記数４３に示すH"の対応する各行のビット“１”を有するe_{A}の情報が、H"乗算回路７に入力される。そして、H"乗算回路７では、前記入力されたe_{A}の情報に対し、ｍｏｄ２の計算を行い、その結果をS'_{p}（ｐ＝０、１、・・・、５）として出力する。

【数４３】

$$H''=\begin{bmatrix}1&0&0&0&0&0&1&0\\0&1&0&0&0&0&1&0\\0&0&1&0&0&0&1&1\\0&0&0&1&0&0&1&1\\0&0&0&0&1&0&0&1\\0&0&0&0&0&1&0&1\end{bmatrix}$$

具体的に、例えば、シンドローム（Ｓ）３３の上位ｑビットS_I３６とＨ”の転置（Ｈ”^{T}）の積（Ｓ’$_I$）３９のビットＳ’$_0$を生成するには、８ビットの$e_A=(e_{A_0}e_{A_1}e_{A_2}e_{A_3}e_{A_4}e_{A_5}e_{A_6}e_{A_7})$３８のうち、数４３に示すパリティ検査行列Ｈ”の第０行目において、“１”が存在する箇所に相当する２ビットe_{A_0}、e_{A_6}の値に対して、２を法とする和（ｍｏｄ２）をとればよい。

第１１図において、排他的論理和回路２２では、前記２ビットのe_Aの情報（つまり、２ビットe_{A_0}、e_{A_6}の値）が入力されて、値Ｓ’$_0$を出力していることを示している。

シンドローム（Ｓ）３３の上位ｑビットS_I３６とＨ”の転置（Ｈ”^{T}）の積（Ｓ’$_I$）３９のほかのビットＳ’$_p$についても、同様に、排他的論理和回路２２及び３入力パリティチェック回路２３を用いて、構成することができる。

ここで、Ｓ’は数４４に示す構成とする。

【数４４】

$$S'=(S'_I\mid S_{II}\mid S_{III}\mid S_{IV})$$
$$=(S'_0S'_1S'_2S'_3S'_4S'_5\mid S_8S_9S_{10}S_{11}S_{12}S_{13}\mid S_{14}S_{15}S_{16}S_{17}S_{18}S_{19}\mid S_{20}S_{21}S_{22}S_{23}S_{24}S_{25})$$
$$=(S'_0S'_1S'_2S'_3S'_4S'_5\mid S'_6S'_7S'_8S'_9S'_{10}S'_{11}\mid S'_{12}S'_{13}S'_{14}S'_{15}S'_{16}S'_{17}\mid S'_{18}S'$$

$_{19}S'$ $_{20}S'$ $_{21}S'$ $_{22}S'$ $_{23}$)

＜ｄ＞ＧＦ（２^{6}）上の並列復号回路８の構成

次に、ＧＦ（２^{6}）上の並列復号回路８について説明する。

ＧＦ（２^{6}）上のビット誤りポインタ４０を求めるために、並列復号法を用いる。ＧＦ（２^{6}）上の並列復号回路８の全体構成を第１２図に示す。

第１２図に示されるように、ＧＦ（２^{6}）上の並列復号回路８は、１２バイト中任意の２バイトの組$\binom{12}{2}$＝６６個を選び、ｉ、ｊ番目の２バイト（以下、（ｉ、ｊ）と表現することとする）に関する各々の誤りを生成する回路である（ｉ、ｊ）に対する誤り生成回路５０（０）・・・５０（６５）、（以下、代表として５０（ｍ）、０≦ｍ≦６５、と称する）と、ＧＦ（２^{6}）上のビット誤りポインタ（ｅ’）４０を出力するＧＦ（２^{6}）上の誤り算出回路５１とから構成されている。

まず、（ｉ、ｊ）に対する誤り生成回路５０（ｍ）は、バイト位置ｉ、ｊに対して、上述したＳ’・$\tilde{H}_i^{\dagger T}$、Ｓ’・$\tilde{H}_j^{\dagger T}$、Ｓ’・$\tilde{B}_{(i,j)}^{\dagger\ T}$を計算する回路である。要するに、（ｉ、ｊ）に対する誤り生成回路５０（０）～５０（６５）は、符号語の１２バイト中のｉ、ｊ（０≦ｉ＜ｊ≦１１）番目のバイトに対する前述したＳ’・$\tilde{H}_i^{\dagger T}$、Ｓ’・$\tilde{H}_j^{\dagger T}$、Ｓ’・$\tilde{B}_{(i,j)}^{\dagger\ T}$を計算し、$E'_{m,i}$、$E'_{m,j}$になる誤り５２（ｍ）、０≦ｍ≦６５、０≦ｉ＜ｊ≦１１（それぞれｒビットのベクトル）、及び訂正可能信号５３（ｍ）、０≦ｍ≦６５、を出力する。

また、ＧＦ（２^{6}）上の誤り算出回路５１は、ＧＦ（２^{6}）上のビット誤りポインタ（ｅ’）４０を出力する回路である。要するに、ＧＦ（

2^6）上の誤り算出回路５１は、誤り生成回路５０（ｍ）の出力である誤り$E'_{m,i}$、$E'_{m,j}$を入力とし、ＧＦ（2^6）上のビット誤りポインタ（ｅ’）４０を出力する。

ここで、ＧＦ（2^6）上の並列復号回路８の処理をより詳しく説明すると、第１２図に示すように、まず、２４ビットのＳ’（シンドローム（Ｓ）３３の上位ｑビットS_I３６とＨ”の転置（H''^T）の積（S'_I）３９、及びシンドローム（Ｓ）３３の下位３ｒビット$S_{II}S_{III}S_{IV}$３７）を（ｉ、ｊ）に対する誤り生成回路５０（ｍ）に入力する。

次に、（ｉ、ｊ）に対する誤り生成回路５０（ｍ）は、誤り$E'_{m,i}$、$E'_{m,j}$を生成し、そして、生成した誤り$E'_{m,i}$、$E'_{m,j}$をＧＦ（2^6）上の誤り算出回路５１に入力する。次に、ＧＦ（2^6）上の誤り算出回路５１は、ＧＦ（2^6）上のビット誤りポインタ（ｅ’）４０を出力する。

また、２４ビットのＳ’（シンドローム（Ｓ）３３の上位ｑビットS_I３６とＨ”の転置（H''^T）の積（S'_I）３９、及びシシンドローム（Ｓ）３３の下位３ｒビット$S_{II}S_{III}S_{IV}$３７）を２４ビット入力ＯＲゲート回路５４に入力する。

一方、（ｉ、ｊ）に対する誤り生成回路５０（ｍ）によって生成される訂正可能信号５３（ｍ）を、６６入力ＮＯＲゲート回路５５に入力する。

２４ビット入力ＯＲゲート回路５４及び６６入力ＮＯＲゲート回路５５からそれぞれ出力された２本の信号を２入力ＡＮＤゲート回路５６に入力する。２入力ＡＮＤゲート回路５６は、訂正不可能誤り検出信号（ＤＳ（０））４１を出力する。これらの回路により、シンドロームが非零且つ訂正可能信号がすべて０の場合に、誤りを検出する。

次に、（ｉ、ｊ）に対する誤り生成回路５０（ｍ）の構成例を具体的に示す。例として、（０、１）に対する誤り生成回路５０（０）を第１３図に示す。第１３図に示されるように、（ｉ、ｊ）に対する誤り生成回路５０（ｍ）は、Ｓ’・$\tilde{H}_i^{\dagger T}$計算回路６０と、Ｓ’・$\tilde{H}_j^{\dagger T}$計算回路６１と、Ｓ’・$\tilde{B}_{(i,j)}^{\dagger\ T}$計算回路６２とを備える。

ここで、$\tilde{H}$は下記数４５に示すＧＦ（２^6）上のＲＳ符号の検査行列であり、符号長７２ビットと検査ビット長２４ビットとを有する。ただし、γはＧＦ（２^6）の元であり、下記数４６に示すように、生成多項式ｇ（ｘ）＝ｘ^6＋ｘ＋１で定義される２元の６×６随伴行列（companion matrix）で表現される。

【数４５】

$$\tilde{H}=\begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\ 1 & \gamma & \gamma^2 & \gamma^3 & \gamma^4 & \gamma^5 & \gamma^6 & \gamma^7 & \gamma^8 & \gamma^9 & \gamma^{10} & \gamma^{11}\\ 1 & \gamma^2 & \gamma^4 & \gamma^6 & \gamma^8 & \gamma^{10} & \gamma^{12} & \gamma^{14} & \gamma^{16} & \gamma^{18} & \gamma^{20} & \gamma^{22}\\ 1 & \gamma^3 & \gamma^6 & \gamma^9 & \gamma^{12} & \gamma^{15} & \gamma^{18} & \gamma^{21} & \gamma^{24} & \gamma^{27} & \gamma^{30} & \gamma^{33}\end{bmatrix}$$

【数４６】

$$\gamma=\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1\\ 1 & 0 & 0 & 0 & 0 & 1\\ 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}$$

例えば、（０、１）に対する誤り生成回路５０（０）では、$\tilde{H}_0$、$\tilde{H}_1$は、数４５の第０列、第１列に相当し、それぞれ下記数４７に示す２４×６行列となる。

【数４７】

$$\tilde{H}_0=\begin{bmatrix}1\\1\\1\\1\end{bmatrix}=\left[\begin{array}{cccccc}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\\hline 1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\\hline 1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\\hline 1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\end{array}\right]$$

$$\tilde{H}_1=\begin{bmatrix}1\\\gamma\\\gamma^2\\\gamma^3\end{bmatrix}=\left[\begin{array}{cccccc}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\\hline 0&0&0&0&0&1\\1&0&0&0&0&1\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\\hline 0&0&0&0&1&0\\0&0&0&0&1&1\\1&0&0&0&0&1\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\\hline 0&0&0&1&0&0\\0&0&0&1&1&0\\0&0&0&0&1&1\\1&0&0&0&0&1\\0&1&0&0&0&0\\0&0&1&0&0&0\end{array}\right]$$

次に、$\tilde{H}_0$、$\tilde{H}_1$から２４×１２の行列$\tilde{H}_{(0,1)}=[\tilde{H}_0\quad\tilde{H}_1]$を構成し、$\tilde{H}_{(0,1)}$に２４×１２の行列Ｂ$_{(0,1)}$を付加することにより、下記数４８に示す２４×２４の正則行列$A_{(0,1)}=[\tilde{H}_{(0,1)}\quad B_{(0,1)}]$を構成する。ここで、Ｂ$_{(0,1)}$はＡ$_{(0,1)}$が正則行列となるように求めた２４×１２の行列であり、一例である。

【数４８】

$$A_{(0,1)}=\left[\tilde{H}_0\quad\tilde{H}_1\quad B_{(0,1)}\right]=$$

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

次に、A$_{(0,1)}$の逆行列$A_{(0,1)}^{-1}$を求めると、下記数４９の関係が成り立つ。$\tilde{H}_0^{\dagger}$、$\tilde{H}_1^{\dagger}$、$B_{(0,1)}^{\dagger}$は、それぞれ下記数５０、数５１、数５２に示す行列となる。

【数４９】

$$A_{(0,1)}^{-1}=\begin{bmatrix}\tilde{H}_{(0,1)}^{\dagger}\\ B_{(0,1)}^{\dagger}\end{bmatrix}=\begin{bmatrix}\tilde{H}_0^{\dagger}\\ \tilde{H}_1^{\dagger}\\ B_{(0,1)}^{\dagger}\end{bmatrix}$$

【数５０】

$\tilde{H}_0^{\dagger} =$

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

【数５１】

$\tilde{H}_1^{\dagger} =$

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

【数５２】

$\tilde{B}_{(0,1)}^{\dagger} =$

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

S’・$\tilde{H}_0^{\dagger T}$計算回路６０では、$\tilde{H}_0^{\dagger T}$の各列が入力する情報に対応し、

例えば、$\tilde{H}_0^{\dagger T}$の第０行が入力情報のＳ’$_0$に対応している。そこで、Ｓ’・$\tilde{H}_0^{\dagger T}$の各ビットを生成するには、$\tilde{H}_0^{\dagger}$の各行方向に、“１”を有するビットに対応するシンドロームＳ’の情報をＧＦ（２）上で加算（ｍｏｄ２の計算）すればよい。

要するに、Ｓ’・$\tilde{H}_0^{\dagger T}$計算回路６０は、まず、前記$\tilde{H}_0^{\dagger T}$の対応する各行のビット“１”を有するシンドローム情報（Ｓ’$_p$）、（ｐ＝０、１、･･･、２３）をそれぞれ６個の多入力パリティチェック回路６３（ｕ）、（ｕ＝０、１、･･･、５）に入力する。次に、これらのシンドローム情報についてｍｏｄ２の計算を行い、その結果としてＥ’$_{0,0}$、Ｅ’$_{0,1}$を出力する。

具体的に、例えば、Ｅ’$_{0,0}$の０ビット目を生成するには、２４ビットのシンドロームＳ’＝Ｓ’$_0$、Ｓ’$_1$、･･･、Ｓ’$_{23}$のうち、上記数５０に示す$\tilde{H}_0^{\dagger}$の第０行目において“１”が存在する箇所に相当する５ビットｓ’$_0$、ｓ’$_5$、ｓ’$_9$、ｓ’$_{21}$、ｓ’$_{23}$の値について、２を法とする和（ｍｏｄ２）をとればよい。第１３図の多入力パリティチェック回路６３（０）は、この５ビットｓ’$_0$、ｓ’$_5$、ｓ’$_9$、ｓ’$_{21}$、ｓ’$_{23}$のシンドロームが入力される。また、多入力パリティチェック回路６３（０）は、５２（０）であるＥ’$_{0,0}$の０ビット目を出力する。なお、Ｅ’$_{0,0}$の他のビットについても同様である。

また、第１３図に示されるように、Ｓ’・$\tilde{H}_1^{\dagger T}$計算回路６１は、６個の多入力パリティチェック回路６４（ｗ）、（ｗ＝０、１、･･･、５）で構成され、そして、Ｓ’・$\tilde{B}_{(0,1)}^{\dagger\ T}$計算回路６２は、１２個の多入力パリティチェック回路６５（ｚ）、（ｚ＝０、１、･･･、１１）で構成される。これらの計算回路は、Ｓ’・$\tilde{H}_0^{\dagger T}$計算回路６０と同様の方法で構成することができる。

ここで、（０、１）に対する誤り生成回路５０（０）の処理をより詳しく説明すると、第１３図に示すように、２４ビットのシンドロームＳ’を多入力パリティチェック回路６３（ｕ）、（ｕ＝０、１、・・・、５）に入力することにより、$E'_{0,0}$を生成する。また、２４ビットのシンドロームＳ’を多入力パリティチェック回路６４（ｗ）、（ｗ＝０、１、・・・、５）に入力することにより、$E'_{0,1}$を生成する。

また、２４ビットのシンドロームＳ’を多入力パリティチェック回路６５（ｚ）、（ｚ＝０、１、・・・、１１）に入力する。多入力パリティチェック回路６５（ｚ）、（ｚ＝０、１、・・・、１１）から出力される１２ビットの信号を１２入力ＮＯＲゲート回路６５に入力し、（０、１）に対する訂正可能信号５３（０）を出力する。これらの回路は、$S' \cdot \widetilde{B}^{\dagger}_{(0,1)}{}^{T}=0$となるバイトの組（ｉ、ｊ）が誤りのバイトの組であるため、$S' \cdot \widetilde{B}^{\dagger}_{(0,1)}{}^{T}=0$であることを検出し、バイトの組（ｉ、ｊ）で訂正可能であることを意味する。

また、（０、１）に対する訂正可能信号５３（０）と多入力パリティチェック回路６３、多入力パリティチェック回路６４からの出力信号を２入力ＡＮＤゲート回路６７に入力し、$E'_{0,0}$及び$E'_{0,1}$を出力する。これらの回路は、誤りが生じているバイトの組である場合に、誤りを出力し、そうでない場合に、０のパターンを出力するようにしている。

以上より、（０、１）に対する誤り生成回路５０（０）は、誤り$E'_{0,0}$、$E'_{0,1}$５２（０）、及び（０、１）に対する訂正可能信号５３（０）を最終的に出力する。ここでは、（０、１）に対する誤り生成回路５０（０）を例にして説明したが、他のバイト（ｉ、ｊ）に対しても同様である。

次に、ＧＦ（２^{6}）上の誤り算出回路５１の構成例を具体的に示す。ＧＦ（２^{6}）上の誤り算出回路５１を第１４図に示す。

第１４図に示されるように、ＧＦ（２^{6}）上の誤り算出回路５１は、ｉ番目のバイトに対応するすべての誤りＥ’$_{m、i}$をビットごとＯＲ演算回路６８に入力し、ＧＦ（２^{6}）上のビット誤りポインタ（ｅ’）４０を最終的に出力する。これらの回路は、ｉ番目に対応するバイトの組（ｉ、ｊ）からの誤りＥ’$_{m、i}$を統合し、ｉ番目の誤りｅ$_{i}$’を出力するようにしている。

具体的に、例えば、ＧＦ（２^{6}）上のビット誤りポインタ（ｅ’）４０のｅ’$_{6}$を出力するには、（０、１）に対する誤り生成回路５０（０）、（１、２）に対する誤り生成回路５０（１１）、・・・、及び（１、１１）に対する誤り生成回路５０（２０）から出力されるＥ’$_{0、1}$、Ｅ’$_{11、1}$、・・・、Ｅ’$_{20、1}$の０ビット目をビットごとのＯＲ演算回路６８に入力することにより得られる。ビットごとのＯＲ演算回路６８では、６ビットを有するＥ’$_{0、1}$、Ｅ’$_{11、1}$、・・・、Ｅ’$_{20、1}$の０ビット目をそれぞれ１１入力ＯＲ回路６９に入力することにより、Ｅ’$_{0、1}$、Ｅ’$_{11、1}$、・・・、Ｅ’$_{19、1}$の０ビット目を統合し、ｅ$_{6}$’を出力している。

以上では、ＧＦ（２^{6}）上のビット誤りポインタ（ｅ’）４０のｅ’$_{6}$を例にして説明したが、他のビット値ｅ’$_{0}$、・・・、ｅ’$_{65}$についても、まったく同様である。

＜ｅ＞ＧＦ（２^{8}）上の誤り生成回路９の構成

次に、ＧＦ（２^{8}）上の誤り生成回路９について説明する。ＧＦ（２^{8}）上の誤り生成回路９の全体構成を第１５図に示す。

第１５図に示すように、ＧＦ（２^{8}）上の誤り生成回路９は、ｔビット誤り訂正復号回路７０と、１２入力ソート回路７３と、ｉバイトにおける誤訂正判定回路７４とを備えており、ＧＦ（２^{6}）上のビット誤りポインタ（ｅ’）４０と、バイトごとの誤りの和（e_A）３８とを入力とし、ＧＦ（２^{8}）上のビット誤りポインタ（ｅ）４２と、訂正不可能誤り検出信号（ＤＳ（１））４３とを出力する回路である。

ここで、ｔビット誤り訂正復号回路（本例では、２ビット誤り訂正復号回路）７０では、ＧＦ（２^{6}）上のビット誤りポインタ（ｅ’）４０の１バイトを入力とし、復号結果ビット誤りポインタ（$\hat{e}$）７１の１バイトと、訂正能力をこえた誤りを検出する信号であるｉバイトにおける検出信号７２とを出力する。

そして、１２入力ソート回路７３は、ｔビット誤り訂正復号回路７０から出力されるｉバイトにおける検出信号７２を入力とし、入力される１２ビットの情報中の重み（１の数）を数え、その重みの数だけ出力の上位ビットから１を連続して表記する回路である。例えば、１２ビットの入力情報中に１が６個存在する場合、この回路の出力１２ビットのうち上位６ビットはすべて１を、残りの６ビットはすべて０を出力する回路である。

最後に、ｉバイトにおける誤訂正判定回路７４では、すべてのバイトのｔビット誤り訂正復号回路７０から出力される復号結果ビット誤りポインタ（$\hat{e}$）７１と、ｉバイトにおける検出信号７２と、バイトごとの誤りの和（e_A）３８とを入力とし、ＧＦ（２^{8}）上のビット誤りポインタ（ｅ）４２の１バイトを出力する。

ここで、ＧＦ（２^{8}）上の誤り生成回路９の処理をより詳しく説明すると、第１５図に示すように、ＧＦ（２^{6}）上のビット誤りポインタ（

e'）４０を６ビットずつに分け、２ビット誤り訂正復号回路７０にそれぞれ入力する。

そして、２ビット誤り訂正復号回路７０から出力されたｉバイトにおける検出信号７２は、１２入力ソート回路７３に入力される。１２入力ソート回路７３から出力された信号において、２バイト以上の２ビット誤り訂正復号回路７０において誤りを検出した場合に、制御能力外の誤りが生じたとして誤りを検出する（一方、１バイトの２ビット誤り訂正復号回路７０において誤りを検出した場合に、詳しくは後述するように、ｉバイトにおける誤訂正判定回路７３において誤りの訂正を行う）。よって、１２入力ソート回路７３から出力された信号において、その上位２ビット目の信号を訂正不可能誤り検出信号（ＤＳ（１））４３として出力する。

また、すべてのバイトの２ビット誤り訂正復号回路７０から出力された復号結果ビット誤りポインタ（$\hat{e}$）７１と、ｉバイトにおける検出信号７２と、バイトごとの誤りの和（e_A）３８とを、それぞれｉバイトにおける誤訂正判定回路７４に入力する。

そして、すべてのｉバイトにおける誤訂正判定回路７４では、ｉバイトにおける検出信号７２から出力された信号を集め、９０ビットのＧＦ（2^b）上のビット誤りポインタ４２を出力する。

次に、２ビット誤り訂正復号回路７０の構成例を具体的に示す。

例として、１バイト目に対する２ビット誤り訂正復号回路７０において、１バイト目に対する復号結果ビット誤りポインタ（$\hat{e}$）７１の上位４ビットを出力する回路を第１６図に、１バイト目に対する復号結果ビット誤りポインタ（$\hat{e}$）７１の下位４ビットを出力する回路を第１７図に、そして、訂正能力をこえた誤りが検出されたときに“１”を出力す

る１バイトにおける検出信号７２を出力する回路（つまり、後述する検出信号出力部）を第１８図にそれぞれ示す。

１バイト目に対する２ビット誤り訂正復号回路７０は、ＧＦ（2^6）上のビット誤りポインタ（ｅ’）４０の１バイト目を入力とし、復号結果ビット誤りポインタ（$\hat{e}$）７１の１バイト目及び１バイトにおける検出信号７２を出力する回路である。

上述したような２ビット誤り訂正復号回路７０では、Ｈ”により復号した結果である誤り$\hat{e}_I$及びＧＦ（2^6）上の誤りe_I’に対し、$\hat{e}_I \rightarrow e_I$’は単射の関係となっている。よって、論理合成を行うことにより、組み合わせ回路で実現することができる。

ただし、Ｈ”は任意の４列が線形独立な行列であるため、すなわち、２ビット誤り訂正符号の検査行列であるため、$[4_{1/8}EC]_2$符号において、１バイトに３ビットや４ビットの誤りが生じた場合、Ｈ”で復号した結果が誤訂正されるか、または制御能力外誤りとして検出される。１バイトに３ビットや４ビットの誤りが誤訂正された場合に、詳しくは後述するように、誤訂正判定を行うことにより、誤訂正されているかどうかを判別することができる。

具体的に、例えば、第１６図において、情報１バイト目では、$\hat{e}_I \cdot H''^T = e_I'$であり、Ｈ”は、上記数２４に示す任意の４列が線形独立な行列であるため、すなわち、２ビット誤り訂正符号の検査行列であるため、重みが２以下のＨ”により復号した結果である誤り$\hat{e}_I$、及びＧＦ（2^6）上の誤りe_I’は一意に定まる。

よって、重みが２以下のＨ”により復号した結果である誤り$\hat{e}_I = (\hat{e}_8\hat{e}_9\hat{e}_{10}\hat{e}_{11}\hat{e}_{12}\hat{e}_{13}\hat{e}_{14}\hat{e}_{15})$、及びＧＦ（$2^6$）上の誤り$e_I' = (e_6' \; e_7' \; e_8' \; e_9' \; e_{10}' \; e_{11}')$は、下記表１に示す関係を有する。

【表１】

$\hat{e}_8\hat{e}_9\hat{e}_{10}\hat{e}_{11}\hat{e}_{12}\hat{e}_{13}\hat{e}_{14}\hat{e}_{15}$	$e_6', e_7', e_8', e_9', e_{10}', e_{11}'$	$\hat{e}_8\hat{e}_9\hat{e}_{10}\hat{e}_{11}\hat{e}_{12}\hat{e}_{13}\hat{e}_{14}\hat{e}_{15}$	$e_6', e_7', e_8', e_9', e_{10}', e_{11}'$
0 0 0 0 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 1 0 0 0 0 0 0	0 1 0 0 0 0	0 0 1 0 0 0 0 0	0 0 1 0 0 0
0 0 0 1 0 0 0 0	0 0 0 1 0 0	0 0 0 0 1 0 0 0	0 0 0 0 1 0
0 0 0 0 0 1 0 0	0 0 0 0 0 1	0 0 0 0 0 0 1 0	1 1 1 1 0 0
0 0 0 0 0 0 0 1	0 0 1 1 1 1	1 1 0 0 0 0 0 0	1 1 0 0 0 0
1 0 1 0 0 0 0 0	1 0 1 0 0 0	1 0 0 1 0 0 0 0	1 0 0 1 0 0
1 0 0 0 1 0 0 0	1 0 0 0 1 0	1 0 0 0 0 1 0 0	1 0 0 0 0 1
1 0 0 0 0 0 1 0	0 1 1 1 0 0	1 0 0 0 0 0 0 1	1 0 1 1 1 1
0 1 1 0 0 0 0 0	0 1 1 0 0 0	0 1 0 1 0 0 0 0	0 1 0 1 0 0
0 1 0 0 1 0 0 0	0 1 0 0 1 0	0 1 0 0 0 1 0 0	0 1 0 0 0 1
0 1 0 0 0 0 1 0	1 0 1 1 0 0	0 1 0 0 0 0 0 1	0 1 1 1 1 1
0 0 1 1 0 0 0 0	0 0 1 1 0 0	0 0 1 0 1 0 0 0	0 0 1 0 1 0
0 0 1 0 0 1 0 0	0 0 1 0 0 1	0 0 1 0 0 0 1 0	1 1 0 1 0 0
0 0 1 0 0 0 0 1	0 0 0 1 1 1	0 0 0 1 1 0 0 0	0 0 0 1 1 0
0 0 0 1 0 1 0 0	0 0 0 1 0 1	0 0 0 1 0 0 1 0	1 1 1 0 0 0

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

ここで、ＧＦ（２^{6}）上の誤りｅ$_{I}$’は３７パターンあるが、残りの２^{6}－３７＝２７パターンに対する$\hat{e}_{I}$は、訂正能力外誤り（制御能力外誤り）として検出し、$\hat{e}_{I}$の出力はドントケア＊とする。

例えば、上記表１に示されたＨ”により復号した結果である誤り$\hat{e}_{I}$とＧＦ（２^{6}）上の誤りｅ$_{I}$’の対応関係から、情報１バイト目の１ビット目の値$\hat{e}_{8}$は、ｅ$_{I}$’＝（ｅ$_{6}$’ ｅ$_{7}$’ ｅ$_{8}$’ ｅ$_{9}$’ ｅ$_{10}$’ ｅ$_{11}$’）＝（１００００００）、（１１０００００）、（１０１０００）、（１００１００）、（０１１１００）、（１０００１０）、（１０１１１１）、（１００００１）のときに、１となる。

また、上記表１に示されたＨ”により復号した結果である誤り$\hat{e}_{I}$とＧＦ（２^{6}）上の誤りｅ$_{I}$’の対応関係において、残った２７パターンのＧＦ（２^{6}）上の誤りｅ$_{I}$’＝（ｅ$_{6}$’ ｅ$_{7}$’ ｅ$_{8}$’ ｅ$_{9}$’ ｅ$_{10}$’ ｅ$_{11}$’）＝（１００１１０）、（１１０１１０）、（０１０１１０）、（０１１１１０）、（１０１１１０）、（１１００１０）、（０１１０１０）、（１１１０１０）、（１０１０１０）、（１０００１１）、（０１００１１）、（０１１０１１）、（１１１０１１）、（１０１０１１）、（１００１１１）、（１１０１１１）、（０１０１１１）、（１１１１１１）、（１００１０１）、（１１０１０１）、（０１０１０１）、（０

１１１０１）、（１０１１０１）、（１１０００１）、（０１１００１）、（１１１００１）、（１０１００１）に対する情報１バイト目の１ビット目の値$\hat{e}_8$は、ドントケア＊とする。

これらの論理関数を簡単化すると、情報１バイト目の１ビット目の値$\hat{e}_8$は、下記数５３に示す論理式で表される。

【数５３】

$$\hat{e}_8 = e'_6\bar{e}'_7\bar{e}'_8 \vee e'_6\bar{e}'_8\bar{e}'_9\bar{e}'_{10} \vee \bar{e}'_6e'_7e'_8\bar{e}'_9\bar{e}'_{11} \vee e'_6\bar{e}'_7e'_8\bar{e}'_9 \vee e'_6\bar{e}'_7e'_8e'_{11}$$

ただし、$\bar{e}'_x$はe'_xの否定で、$e'_6\bar{e}'_7\bar{e}'_8$は$e'_6 \wedge \bar{e}'_7 \wedge \bar{e}'_8$であり、また、∨は論理和を、∧は論理積を表す。

よって、第１６図に示されるように、まず、ＧＦ（2^6）上のビット誤りポインタ４０e’$_6$、e’$_7$、e’$_8$、e’$_9$、e’$_{10}$、e’$_{11}$をＮＯＴゲート回路８０に入力し、$\bar{e}'_x$を生成する。そして、$e'_6,\bar{e}'_7,\bar{e}'_8$を３入力ＡＮＤゲート回路８１に、$e'_6,\bar{e}'_8,\bar{e}'_9,\bar{e}'_{10}$を４入力ＡＮＤゲート回路８２に、$\bar{e}'_6,e'_7,e'_8,\bar{e}'_9,\bar{e}'_{11}$を５入力ＡＮＤゲート回路８３に、$e'_6,\bar{e}'_7,e'_8,\bar{e}'_9$を４入力ＡＮＤゲート回路８２に、$e'_6,\bar{e}'_7,e'_8,e'_{11}$を４入力ＡＮＤゲート回路８２にそれぞれ入力する。それぞれのＡＮＤゲート回路から出力された信号を５入力ＯＲゲート回路８４に入力することにより、情報１バイト目の１ビット目の値$\hat{e}_8$が得られる。

第１６図及び第１７図に示されるように、他のビットの値$\hat{e}_9$、・・・、$\hat{e}_{15}$も、ＮＯＴゲート回路８０、３入力ＡＮＤゲート回路８１、４入力ＡＮＤゲート回路８２、５入力ＡＮＤゲート回路８３、５入力ＯＲゲート回路８４、６入力ＯＲゲート回路８５により、同様にして得られる。

また、ｔビット誤り訂正復号回路（本例では、２ビット誤り訂正復号回路）７０の検出信号出力部も同様にして、構成することができる。検

出信号出力部は、ＧＦ（２^{6}）上の誤りe_I'から写像されることのないH"により復号した結果である誤り$\hat{e}_I$を訂正不可能な誤りとして検出する。例えば、１バイト目に対する２ビット誤り訂正復号回路７０の検出信号出力部を第１８図に示す。

具体的に、例えば、情報１バイト目において、上記表１に示されたH"により復号した結果である誤り$\hat{e}_I$とＧＦ（２^{6}）上の誤りe_I'の対応関係において、残った２７パターンのＧＦ（２^{6}）上の誤りe_I'＝（e_6' e_7' e_8' e_9' e_{10}' e_{11}'）＝（１００１１０）、（１１０１１０）、（０１０１１０）、（０１１１１０）、（１０１１１０）、（１１００１０）、（０１１０１０）、（１１１０１０）、（１０１０１０）、（１０００１１）、（０１００１１）、（０１１０１１）、（１１１０１１）、（１０１０１１）、（１００１１１）、（１１０１１１）、（０１０１１１）、（１１１１１１）、（１００１０１）、（１１０１０１）、（０１０１０１）、（０１１１０１）、（１０１１０１）、（１１０００１）、（０１１００１）、（１１１００１）、（１０１００１）に対する検出信号の値を１にし、誤りを検出する。

よって、これらの論理関数を簡単化すると、検出信号の値は、下記数５４に示す論理式で表される。

【数５４】

$$e'_6\bar{e}'_8e'_9e'_{11} \vee e'_7\bar{e}'_8e'_9e'_{10} \vee e'_7\bar{e}'_8e'_9e'_{11} \vee e'_7e'_8\bar{e}'_9e'_{10} \vee e'_6e'_8\bar{e}'_9e'_{10} \vee e'_6\bar{e}'_7e'_9e'_{10}\bar{e}'_{11}$$
$$\vee e'_6\bar{e}'_7\bar{e}'_8e'_{10}e'_{11} \vee e'_6e'_7\bar{e}'_8e'_{10}\bar{e}'_{11} \vee e'_6e'_7\bar{e}'_9\bar{e}'_{10}e'_{11} \vee \bar{e}'_6e'_7\bar{e}'_8e'_{10}e'_{11} \vee \bar{e}'_6e'_7e'_8e'_{10}\bar{e}'_{11}$$
$$\vee \bar{e}'_6e'_7e'_8\bar{e}'_{10}e'_{11} \vee e'_6e'_7e'_8e'_{10}e'_{11} \vee e'_6\bar{e}'_7e'_8\bar{e}'_{10}e'_{11}$$

よって、第１８図に示されるように、まず、ＧＦ（２^{6}）上のビット誤りポインタ４０e'$_6$、e'$_7$、e'$_8$、e'$_9$、e'$_{10}$、e'$_{11}$をＮＯＴゲート回路８０に入力し、$\bar{e}'_x$を生成する。

そして、$e'_6\bar{e}'_8e'_9e'_{11}, e'_7\bar{e}'_8e'_9e'_{10}, e'_7\bar{e}'_8e'_9e'_{11}, e'_7e'_8\bar{e}'_9e'_{10}, e'_6e'_8\bar{e}'_9e'_{10}, e'_6\bar{e}'_7e'_9e'_{10}\bar{e}'_{11}$, $e'_6\bar{e}'_7\bar{e}'_8e'_{10}e'_{11}, e'_6e'_7\bar{e}'_8e'_{10}\bar{e}'_{11}, e'_6e'_7\bar{e}'_9\bar{e}'_{10}e'_{11}, \bar{e}'_6e'_7\bar{e}'_8e'_{10}e'_{11}, \bar{e}'_6e'_7e'_8e'_{10}\bar{e}'_{11}$ $, \bar{e}'_6e'_7e'_8\bar{e}'_{10}e'_{11}, e'_6e'_7e'_8e'_{10}e'_{11}, e'_6\bar{e}'_7e'_8\bar{e}'_{10}e'_{11}$を合計１４個の４入力ＡＮＤゲート８２及び５入力ＡＮＤゲート８３にそれぞれ入力する。

最後に、それぞれのＡＮＤゲート回路から出力された信号を、１４入力ＯＲ回路８６に入力することにより、１バイトにおける検出信号７２を出力する。

なお、以上では、１バイト目におけるｔビット誤り訂正復号回路７０を例として詳細に説明したが、他のバイトについてもまったく同様である。

次に、１２入力ソート回路７３の構成例を第１９図に示す。

第１９図に示すように、１２ビットの入力情報がソートされ、２入力ＯＲゲート回路７５、及び２入力ＡＮＤゲート回路７６が対になって構成された２入力２出力回路をセルとして、１２入力に対してこのセルを多段に接続して構成している。このソート回路は、一般に、ｎビット入力中に１の数がｘ個存在すれば、ｎビットの上位ｘ個に１を出力し、残りの出力はすべて０を出力する回路である。なお、この回路の一般的な構成は、非特許文献１３に述べられている。

第１９図に示すように、１２入力１２出力の回路である１２入力ソート回路７３では、このセルを３９個（すなわち、７８ゲート）使用し、最大９段のセル（９ゲート段）で実現することが可能である。例えば、１２ビットの入力情報として（００１０１１００１０１１）のような重み６の情報のとき、１２入力ソート回路７３の出力は、（１１１１１１０００００００）となる。

次に、ｉバイトにおける誤訂正判定回路７４の構成例を第２０図に示

す。

例えば、１バイト目に対して、具体的に構成した１バイトにおける誤訂正判定回路７４の構成例を第２０図に示す。第２０図に示すように、１バイトにおける誤訂正判定回路７４は、ビットごと排他的論理和回路９０と、１６入力ソート回路９１と、ビットごと選択回路９６とを備えている。

ここで、１バイトにおける誤訂正判定回路７４の処理をより詳しく説明すると、第２０図に示すように、１バイト目以外の復号結果ビット誤りポインタ（$\hat{e}$）７１ $\hat{e}_0 \cdots \hat{e}_7 \hat{e}_{16} \cdots \hat{e}_{95}$ と、バイトごとの誤りの和（e_A）３８とをビットごと排他的論理和回路９０に入力する。また、１バイト目の復号結果ビット誤りポインタ（$\hat{e}$）７１ $\hat{e}_8 \cdots \hat{e}_{15}$ と、バイトごとの誤りの和（e_A）３８とをビットごと排他的論理和回路９０に入力する。

次に、バイトごとの誤りの和（e_A）３８と、ビットごと排他的論理和回路９０の出力とを１６入力ソート回路９１に入力する。上記数３９に示した関係より、下記数５５に示す関係が成り立つ。

【数５５】

$$w_H\left(\hat{e}_x + e_A\right) + w_H\left(\hat{e}_x\right) \leq \mu_1 \cdot t$$

よって、１６入力ソート回路９１の出力において、１の数が４個以下である場合、１を出力するようにする。すなわち、１６入力ソート回路９１の出力において、上位４ビットを４入力ＯＲゲート回路９２に、下位１２ビットを１２入力ＮＯＲゲート回路９３にそれぞれ入力する。また、１バイトにおける検出信号をＮＯＴゲート回路９４に入力する。そして、４入力ＯＲゲート回路９２からの出力と、１２入力ＮＯＲゲート回路９３からの出力と、ＮＯＴゲート回路９４からの出力とを３入力Ａ

ＮＤゲート回路９５に入力する。

最後に、３入力ＡＮＤゲート回路９５の出力を制御信号として、ビットごと排他的論理和回路９０の出力と、１バイト目の復号結果ビット誤りポインタ（$\hat{e}$）７１ $\hat{e}_8 \cdots \hat{e}_{15}$ とをビットごと選択回路９６に入力し、ＧＦ（2^8）上のビット誤りポインタ（ｅ）４２の１バイト目（$e_8 \cdots e_{15}$）を出力する。その結果、正しく訂正されている場合に、$e_x = \hat{e}_x$ を出力し、誤訂正されている、または制御能力外誤りとして検出された場合に、その他のバイトにおいて正しく訂正された誤りパターンと e_A の排他的論理和をとることにより、誤訂正されているバイトにおける誤りパターンを復元することができる。

次に、ビットごと排他的論理和回路９０の構成例を第２１図に示す。第２１図に示すように、それぞれ８ビットを有する２つの信号の各ビットどうしの排他的論理和をとる回路である。

ここで、ビットごと排他的論理和回路９０の処理をより詳しく説明すると、第２１図に示すように、８ビットの信号の中で０ビット目どうしを多入力排他的論理和回路９７に入力する。多入力パリティチェック回路９７から出力された各信号をまとめて、最終的に８ビットの信号として出力する。

次に、１６入力ソート回路９１の構成例を第２３図に示す。第２３図に示されるように、１６入力ソート回路９１では、１６ビットの入力情報がソートされ、２入力ＯＲゲート回路７５、及び２入力ＡＮＤゲート回路７６が対になって構成された２入力２出力回路をセルとして、１６入力に対してこのセルを多段に接続して構成している。このソート回路は、一般に、ｎビット入力中に１の数がｘ個存在すれば、ｎビットの上位ｘ個に１を出力し、残りの出力はすべて０を出力する回路である。

また、第２３図に示すように、１６入力１６出力の回路である１６入力ソート回路９１では、このセルを６３個（すなわち、１２６ゲート）使用し、最大１０段のセル（１０ゲート段）で実現することが可能である。

最後に、ビットごと選択回路９６の構成例を具体的に示す。

第２２図に示すように、ビットごと選択回路９６は、それぞれ８ビットを有する２つの信号の各ビットどうしを制御信号（つまり、３入力ＡＮＤゲート回路９５の出力である）に従って選択する回路である。

ここで、ビットごと選択回路９６の処理をより詳しく説明すると、第２２図に示すように、８ビットの信号の中で０ビット目どうしを２入力マルチプレクサ９８に入力する。２入力マルチプレクサ９８では、制御信号が０の場合に、上の信号を出力し、制御信号が１の場合に、下の信号を出力する。０ビット目以外の信号に関しても同様に、制御信号により選択を行う。２入力マルチプレクサ９８から出力された各信号をまとめて、最終的に８ビットの信号として出力する。

以上では、ｂ＝８、ｔ＝１、$\mu_1=\mu_2=4$、$p_1=p_2=2$といった符号パラメータを有する［$4_{1/8}$ＥＣ］$_2$符号を例として説明したが、前記以外の符号パラメータを有する符号についても、同様に上述した回路を構成することができることは明らかである。

（ｆ）反転回路１０の構成

最後に、反転回路１０の構成について説明する。

例えば、１バイト目に対して、具体的に構成した反転回路１０を第２４図に示す。第２４図に示す反転回路１０は、誤りの箇所を具体的に指摘されたビット位置に対して、受信したビット値の反転を行って受信語

（Ｖ’）３２を訂正する回路である。第２４図において、v_8’～v_{15}’は、受信語（Ｖ’）３２の１バイト目に相当する８ビットであり、v_8^*～v_{15}^*は訂正後の出力ビットである。

反転回路１０は、複数の排他的論理和回路を有している。第２４図に示された１バイト目における反転回路１０において、合計８個の排他的論理和回路９９は、８入力ビットv_8’～v_{15}’に対して、対応するビット誤りポインタe_i（ｉ＝８、９、・・・、１５）との排他的論理和をとり、例えば、ビット誤りポインタが“１”のとき、入力ビット値が反転されて訂正されることになる。

なお、反転回路１０は、各バイトにそれぞれ対応するようにするため、Ｎ＝９０ビットの受信語（Ｖ’）に対しては、９０個の排他的論理和回路が必要となる。反転回路１０は、最後に、Ｎ＝９０ビットの受信語（Ｖ’）から、下記数５６に示す６４ビットの訂正語Ｄ*を得る。

【数５６】

$$D^* = (v_0^* v_1^* \cdots v_{57}^* v_{58}^* v_{60}^* v_{64}^* v_{66}^* v_{72}^* v_{74}^*)$$

以上では、第３図に示すｂ＝８、ｔ＝１、$\mu_1=\mu_2=4$、$p_1=p_2=2$といった符号パラメータを有する$[4_{1/8}EC]_2$符号を例にとり、誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号における符号化回路及び復号回路の具体的な構成を詳細に説明した。ここで、ＧＦ（2^r）上の並列復号回路８において、例えば、前記バーレカンプ・マッシィ法を用いた逐次復号を行う場合に、誤り位置多項式および誤り評価多項式から、誤り位置および誤りを求める既存の手法を用いればよい。この手法の実現手段は、ハードウェアでもソフトウェアでもよい。

また、前記以外の符号パラメータを有する、誤りバイト数を制限した

バイト内複数スポッティバイト誤り制御符号に対しては、次のような考えで構成することができる。

まず、符号化回路は、２元で表現したパリティ検査行列により、Ｒビットの検査情報を生成すればよく、全く同様に構成することができる。

また、復号は、前述したように、p_1訂正p_2検出ＲＳ符号の復号により得られたＧＦ（2^r）上の誤りｅ’において、各バイトに対しＨ”の復号、及び誤訂正判定を行うことにより、ＧＦ（2^b）上の誤りを求めることができ、同様に復号回路を構成することができる。

さらに、上記数５に示されたパリティ検査行列で表現される、誤りバイト数を制限した伸長バイト内複数スポッティバイト誤り制御符号に対しては、次のような考えで構成することができる。

まず、符号化回路は、２元で表現したパリティ検査行列により、Ｒビットの検査情報を生成すればよく、全く同様に構成することができる。

また、復号は、前記並列復号法やバーレカンプ・マッシィ法を用いた逐次復号法により、ＧＦ（$2^{r'}$）上のp_1訂正p_2検出ＲＳ符号の復号をすることにより、また、ＧＦ（$2^{r'}$）上の誤りをＧＦ（2^b）上の誤りに前記の例と同様に変換することにより、同様に復号することができる。

よって、復号回路も前記の例における前記ＧＦ（2^r）上の並列復号回路８をＧＦ（$2^{r'}$）上で並列復号を行う回路に変更することにより、また、前記の例における前記ＧＦ（2^b）上の誤り生成回路９において、ｒビットをｒ’ビットに変更することにより、復号回路を同様に構成できることは明らかである。

また、上記数２５及び上記数２８に示されたパリティ検査行列で表現される、バイト誤り検出機能を有する誤りバイト数を制限したバイト内

複数スポッティバイト誤り制御符号に対しては、次のような考えで構成することができる。

まず、符号化回路は、２元で表現したパリティ検査行列により、Ｒビットの検査情報を生成すればよく、全く同様に構成することができる。

また、復号は、前述したように、シンドロームを計算した後に、Ｈ”が$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立であるため、$S_I \neq 0$かつ$S_{II}=S_{III}=S_{IV}=0$の場合、バイト誤りが生じているとして検出を行う。また、前述したp_1訂正p_2検出ＲＳ符号の復号により得られたＧＦ（2^r）上の誤りｅ’において、誤りバイト数が１かつS_Iのハミング重みがμ_1ｔより大きい場合、バイト誤りが生じているとして検出を行うことにより、同様に復号することができる。

よって、復号回路も前記の例におけるＨ”乗算回路にS_I以外のシンドロームを新たに入力し、$S_I \neq 0$かつS_I以外のシンドロームが０の場合、検出信号を出力する。また、ＧＦ（2^b）上の誤り生成回路にS_Iを新たに入力し、ＧＦ（2^b）上の誤り生成回路において、ＧＦ（2^r）上の誤りｅ’において誤りバイト数が１かどうかを調べ、前記ソート回路を用いてS_Iのハミング重みがμ_1ｔより大きいかどうかを調べることにより、復号回路を同様に構成できることは明らかである。

前述したように、本発明では、バイト中にｔビットまでの誤りを有するスポッティバイト誤りに対し、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正し、かつp_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出する能力を有する誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号の一般性のある符号構成を与えるとともに、誤りの訂正・検出が実現できる回路である符号化回路及び

復号回路の具体的な構成を示して、検査情報が具体的に生成できること、及び誤りが具体的に訂正・検出できることを示した。

また、本発明はｔ＝１としたとき、限られた数のバイトに生じたランダムビット誤りを制御することができる機能を備えた誤りバイト数を制限したランダムビット誤り訂正・検出符号となることを示した。

また、本発明では、訂正能力を超えた１バイト誤りを検出する機能を有する誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号の一般性のある符号構成を与えるとともに、誤りの訂正・検出が実現できる回路である符号化回路及び復号回路の具体的な構成を示して、検査情報が具体的に生成できること、及び誤りが具体的に訂正・検出できることを示した。

以上の説明では、主としてメモリ装置に注目して本発明の適用を述べたが、本発明は、それに限定されるものではなく、例えば、このようなバイト単位で物理的に独立した回路と、モジュールと、装置とから構成され、この中で限定した数の誤りが生じる可能性の高い一般の情報システム等に適用可能である。また、光通信回路やバス線回路等の通信または伝送のための回路、装置またはシステムにも適用可能である。

また、本発明の目的は、本実施の形態の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置の機能を実現するソフトウェアのプログラムコードを記述した記憶媒体を、システム或いは装置に供給し、そのシステム或いは装置のコンピュータ（又はＣＰＵやＭＰＵ）が記憶媒体に格納されたプログラムコードを読み出して実行することによっても、達成されることは言うまでもない。

この場合、記憶媒体から読み出されたプログラムコード自体が本実施の形態の機能を実現することとなり、そのプログラムコードを記憶した

記憶媒体及び当該プログラムコードは本発明を構成することとなる。プログラムコードを供給するための記憶媒体としては、ＲＯＭ、フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、ＣＤ－ＲＯＭ、ＣＤ－Ｒ、磁気テープ、不揮発性のメモリカード等を用いることができる。

また、コンピュータが読み出したプログラムコードを実行することにより、前記本実施の形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼動しているＯＳ等が実際の処理の一部又は全部を行い、その処理によって本実施の形態の機能が実現される場合も含まれることは言うまでもない。

産業上の利用可能性

以上説明したように、本発明に係る、誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法及び装置によれば、符号を表現するパリティ検査行列と入力情報データとを基に生成した検査情報を前記入力情報データに付加した送信語が、情報伝送路中で発生した誤りを有する可能性のある受信語に対し、前記パリティ検査行列を基にシンドロームを生成するとともに、このシンドロームを基に前記受信語の誤りを訂正または検出するようにしているので、１バイト中の１ビット、２ビット、３ビット等のｔビットまでの誤りであるスポッティバイト誤りに対し、任意に制限された数のバイトに生じた、任意の数のバイト内複数スポッティバイト誤り（誤りの傾向を調査して、設計者がバイトの数及びスポッティバイト誤りの個数を任意に決めることができる）を訂正・検出する機能を有する、誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号を実現することができた。

これにより、誤りを検出または訂正する符号であるパリティ検査行列の構成や復号の処理手順を、誤りの発生状況ごとに対応させる必要がなく統一的に扱うことが可能となり、本発明の符号機能におけるｔ値、ｂ値、p_1値、p_2値、μ_1値、μ_2値を任意に設定するだけで、パリティ検査行列Ｈや復号手順を統一的に扱うことが可能となり、誤りの発生状態に柔軟に対応できる符号機能を提供することができる。

さらに、ｂビットであるバイト長よりも小さな整数ビット長ｔを選ぶことにより、従来のバイト誤り制御符号（ｔ＝ｂに相当）と比較して、検査ビット数を大幅に減少することが可能となる。これにより、符号長に対する検査ビット数の占める割合を小さくさせて、符号化率を格段に向上することが可能となり、高効率且つ高信頼性のあるデータ転送を実現することができる。

要するに、従来の符号技術では存在しない、『制限された数のバイトに生じた、多重のバイト内複数スポッティバイト誤りを制御できる』といった優れた機能を備える本発明の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法及び装置によれば、従来のリードソロモン符号（Reed-Solomon 符号）による多重バイト誤り制御符号と比較して、検査ビット数を少なくして符号化率を向上させることができ、また、特に、ｔ＝１とした場合、従来のＢＣＨ符号による多重ビット誤り制御符号と比較して、検査ビット数を少なくして符号化率を向上させることができ、さらに、ビット誤りの発生状態に柔軟に対応できる符号機能を提供することができるといった顕著な効果を奏する。

＜参考文献一覧＞

特許文献１：

特開２００４－７２１７号公報

特許文献２：

特願２００３－４１６９７８号明細書

特許文献３：

特願２００４－２３９３９２号明細書

特許文献４：

特願２００４－２８７８１０号明細書

特許文献５：

特開２００２－３７４１７５号公報

非特許文献１：

エス.ジェイ.ホン（S.J.Hong）・エイ.エム.パテル（A.M.Patel）共著,『ア ゼネラル クラス オフ マックシマル コードズ フォー コンピュータ アプリケーションズ “A general Class of Maximal Codes for Computer Applications”』,ＩＥＥＥ トランスアクションズ オン コンピュータズ（IEEE Transactions on Computers）,第 C-21 巻,第 12 号, p.1322-1331,1972 年

非特許文献２：

イー.フジワラ（E.Fujiwara）,『オッドーウエイトーコラム ｂーアジェイスント エラー コレクティング コードズ “Odd-Weight-Column

b-Adjacent Error Correcting Codes”』，トランスアクションズ オブ ザ ＩＥＣＥ ジャパン（Transactions of the IECE Japan），第 E61 巻，第 10 号，p.781-787，1978 年

非特許文献３：

エス.カネダ（S.Kaneda）・イー.フジワラ（E.Fujiwara）共著，『シングル バイト エラー コレクティングーダブル バイト エラー ディテクティング コードズ フォー メモリ システムズ “Single Byte Error Correcting-Double Byte Error Detecting Codes for Memory Systems”』，ＩＥＥＥ トランスアクションズ オン コンピュータズ（IEEE Transactions on Computers），第 C-31 巻，第 7 号，p.596-602，1982 年 7 月

非特許文献４：

シー.エル.チェン（C.L.Chen）・エル.イー.グロバシュ（L.E.Grosbach）共著，『フォールトートレラント メモリ デザイン イン ザ ＩＢＭ アプリケーション システム／４００™ “Fault-Tolerant Memory Design in the IBM Application System/400™”』，プロシーディングズ オフ アニュアル インターナショナル シンポジウム オン フォールトートレラント コンピューティング（Proceedings of Annual International Symposium on Fault-Tolerant Computing），p.393-400，1991 年 6 月

非特許文献５：

エス.カネダ（S.Kaneda），『ア クラス オフ オッドーウエイトーコ

ラム ＳＥＣ－ＤＥＤ－ＳｂＥＤ コードズ フォー メモリ システム アプリケーションズ “A class of Odd-Weight-Column SEC-DED-SbED Codes for Memory System Applications”』，ＩＥＥＥ トランスアクションズ オン コンピュータズ（IEEE Transactions on Computers），第 C-33 巻，第 8 号，p. 737-739，1984 年 8 月

非特許文献６：

ティ．アール．エヌ．ラオ（T.R.N.Rao）・エイジ フジワラ（Eiji Fujiwara）共著，『エラー コントロール コーディング フォー コンピュータ システムズ “Error Control Coding for Computer Systems”』，プレンティスーホール（Prentice-Hall），1989 年

非特許文献７：

イー．フジワラ（E.Fujiwara）・エム．ハマダ（M.Hamada）共著，『シングル ｂ－ビット バイト エラー コレクティング アンド ダブル ビット エラー ディテクティング コードズ フォー メモリ システムズ “Single b-Bit Byte Error Correcting and Double Bit Error Detecting Codes for Memory Systems”』，ＩＥＩＣＥ トランスアクションズ オン ファンダメンタルズ（IEICE Transactions on Fundamentals），第 E76-A 巻，第 9 号，p. 1442-1448，1993 年 9 月

非特許文献８：

エム．ハマダ（M.Hamada）・イー．フジワラ（E.Fujiwara）共著，『アクラス オフ エラー コントロール コードズ フォー バイト オーガナイゼド メモリ システムズ－ＳｂＥＣ－（Ｓｂ＋Ｓ）ＥＤ コー

ドズー “A Class of Error Control Codes for Byte Organized Memory Systems – SbEC-(Sb+S)ED Codes –”』，ＩＥＥＥ トランスアクションズ オン コンピュータズ（IEEE Transactions on Computers），第46巻，第1号，p.105-109，1997年1月

非特許文献９：

ジー.ウマネサン（G.Umanesan）・イー.フジワラ（E.Fujiwara）共著，『ランダム ダブル ビット エラー コレクティングーシングル バイト エラー コレクティング（ＤＥＣ－ＳｂＥＣ） コードズ フォー メモリ システムズ “Random Double Bit Error Correcting – Single Byte Error Correcting (DEC-SbEC) Codes for Memory Systems”』，ＩＥＩＣＥ トランスアクションズ オン ファンダメンタルズ（IEICE Transactions on Fundamentals），第E85-A巻，第1号，p.273-276，2002年1月

非特許文献１０：

ジー.ウマネサン（G.Umanesan）・イー.フジワラ（E.Fujiwara）共著，『ア クラス オフ コードズ フォー コレクティング シングル スポッティ バイト エラーズ “A Class of Codes for Correcting Single Spotty Byte Errors”』，ＩＥＩＣＥ トランスアクションズ オン ファンダメンタルズ（IEICE Transactions on Fundamentals），第E86-A巻，第3号，p.704-714，2003年3月

非特許文献１１：

今井秀樹・藤谷宏共著，『復号の簡単な誤り訂正符号の一構成法』，電子

通信学会論文誌,1979年

非特許文献１２：

エス.ビィー.ウィッカー（S.B.Wicker）・ヴィ.ケイ.バールガバ（V.K.Bhargava）共著,『リードソロモン コードズ アンド ゼア アプリケーションズ “Reed-Solomon Codes and Their Applications”』,ＩＥＥＥ プレス（IEEE Press）,1994年

非特許文献１３：

エス.ジェイ.ピーストラク(S.J.Piestrak),『ザ ミニマル テストセット フォー マルチアウトプット スレッショウルド サーキットズ インプレメンテド アズ ソーティング ネットワークス “The minimal Test Set for Multioutput Threshold Circuits Implemented as Sorting Networks”』,ＩＥＥＥ トランスアクションズ オン コンピュータズ(IEEE Transactions on Computers),第42巻,第6号,p.700-712,1993年6月

請求の範囲

1．入力情報データを基に送信語を生成する符号化手段と、情報伝送路中で誤りが発生した前記送信語を受信語として入力して前記誤りを訂正または検出する復号手段とを備える誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置であって、

前記符号化手段は、スポッティバイト誤り制御符号を表現するパリティ検査行列と、前記入力情報データとを基に生成した検査情報を前記入力情報データに付加することにより、前記送信語を生成し、

前記復号手段は、前記パリティ検査行列を基に前記受信語のシンドロームを生成するシンドローム生成手段と、前記シンドローム生成手段により生成されたシンドロームを基に前記受信語の誤りを訂正または検出する誤り訂正手段とを備えることを特徴とする誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

2．前記入力情報データが、b（bは2以上の整数）ビットを1バイトとし、複数のバイトから構成される場合に、ここで、バイト内のtビット（$1 \leqq t \leqq b$）までの誤りをスポッティバイト誤りと称し、1バイト内に複数の前記スポッティバイト誤りが生じる誤りをバイト内複数スポッティバイト誤りと称することを前提とし、

前記パリティ検査行列で表現される前記スポッティバイト誤り制御符号は、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正し、かつp_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出する請求の範囲第1項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

3．前記ｔ値、ｂ値、p_1値、p_2値、μ_1値、μ_2値を任意に設定し、ただし、$p_1 \leqq p_2$及び$\mu_1 \leqq \mu_2$が成立する請求の範囲第２項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

4．前記情報伝送路は、情報通信システムである請求の範囲第１項乃至請求の範囲第３項のいずれかに記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

5．前記情報伝送路は、メモリシステムである請求の範囲第１項乃至請求の範囲第３項のいずれかに記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

6．前記情報伝送路は、バス線回路である請求の範囲第１項乃至請求の範囲第３項のいずれかに記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

7．前記パリティ検査行列は、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正し、かつp_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出する機能を有する、誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号（$[\mu_1 t/b\mathrm{EC}]_{p_1} - [\mu_2 t/b\mathrm{ED}]_{p_2}$符号）に用いる、Ｒ行Ｎ列を有する次の行列Ｈであり、

$$H=\begin{bmatrix} H' & H' & \cdots & H' & \cdots & H' \\ \gamma^0 H'' & \gamma^1 H'' & \cdots & \gamma^i H'' & \cdots & \gamma^{n-1} H'' \\ \gamma^0 H'' & \gamma^2 H'' & \cdots & \gamma^{2i} H'' & \cdots & \gamma^{2(n-1)} H'' \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma^0 H'' & \gamma^{p_1+p_2-1} H'' & \cdots & \gamma^{(p_1+p_2-1)i} H'' & \cdots & \gamma^{(p_1+p_2-1)(n-1)} H'' \end{bmatrix}$$

ただし、Ｒ＝ｑ＋（ｐ$_1$＋ｐ$_2$－１）ｒ、Ｎ＝ｂｎ、０≦ｉ≦ｎ－１、ｎ＝２^r－１であり、行列Ｈ’は、Ｈ’＝［ｈ$_0$’、ｈ$_1$’、・・・、ｈ$_{b-1}$’］に示すｑ次の列ベクトルｈ$_j$’（０≦ｊ≦ｂ－１）から構成されるｑ行ｂ列（ｑ≦ｂ）の２元行列であり、（$\mu_1+\mu_2$）ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立であり、（$\mu_1+\mu_2$）ｔ≧ｂのとき、前記行列Ｈ’は単位行列を含む正則行列であり、（$\mu_1+\mu_2$）ｔ＜ｂのとき、前記行列Ｈ’は最小ハミング距離（$\mu_1+\mu_2$）ｔ＋１を有する符号の検査行列、すなわち、（$\mu_1+\mu_2$）ｔビット誤り検出機能を有する（ｂ、ｂ－ｑ）符号のパリティ検査行列に等しくなり、また、行列Ｈ”は、Ｈ”＝［ｈ$_0$”、ｈ$_1$”、・・・、ｈ$_{b-1}$”］に示すｒ次の列ベクトルｈ$_j$”（０≦ｊ≦ｂ－１）から構成されるｒ行ｂ列（ｒ≦ｂ）の２元行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立であり、ここで、$\lfloor x\rfloor$はｘを超えない最大の整数を表し、また、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ≧ｂのとき、前記行列Ｈ”は単位行列を含む正則行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＜ｂのとき、前記行列Ｈ”は最小ハミング距離$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＋１を有する符号の検査行列、すなわち、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔビット誤り検出機能を有する（ｂ、ｂ－ｒ）符号のパリティ検査行列に等しくなり、γはＧＦ（２）のｒ次の拡大体ＧＦ（２^r）の原始元であって、γ^{i}Ｈ”＝［γ^{i}ｈ$_0$”、γ^{i}ｈ$_1$”、…、γ^{i}ｈ$_{b-1}$”］が成立するように構成される請求の範囲第２項乃至請

求の範囲第６項のいずれかに記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

８．前記パリティ検査行列は、ｐ$_1$個のバイトに生じるμ_1個のスポッティバイト誤りを訂正し、かつｐ$_2$個のバイトに生じるμ_2個のスポッティバイト誤りを検出する機能を有する、誤りバイト数を制限した伸長バイト内複数スポッティバイト誤り制御符号（伸長$[\mu_1 t/b\mathrm{EC}]_{p_1}-[\mu_2 t/b\mathrm{ED}]_{p_2}$符号）に用いる、Ｒ行Ｎ列を有する次の行列Ｈであり、

$$H=\begin{bmatrix} H' & H' & \cdots & H' & \cdots & H' & H' & O_{q\times r'} \\ \gamma^0 H'' & \gamma^1 H'' & \cdots & \gamma^i H'' & \cdots & \gamma^{n-1} H'' & O_{r'\times b} & O_{r'\times r'} \\ \gamma^0 H'' & \gamma^2 H'' & \cdots & \gamma^{2i} H'' & \cdots & \gamma^{2(n-1)} H'' & O_{r'\times b} & O_{r'\times r'} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \gamma^0 H'' & \gamma^{p_1+p_2-1} H'' & \cdots & \gamma^{(p_1+p_2-1)i} H'' & \cdots & \gamma^{(p_1+p_2-1)(n-1)} H'' & O_{r'\times b} & I_{r'} \end{bmatrix}$$

ただし、ｒ’≧ｒであり、Ｒ＝ｑ＋（ｐ$_1$＋ｐ$_2$－１）ｒ’、Ｎ＝ｂ（ｎ＋１）＋ｒ’、０≦ｉ≦ｎ－１、ｎ＝２$^{r'}$－１であり、$I_{r'}$はｒ’次の単位行列で、Oは零行列であり、行列Ｈ’は、Ｈ’＝［ｈ$_0$’、ｈ$_1$’、・・・、ｈ$_{b-1}$’］に示すｑ次の列ベクトルｈ$_j$’（０≦ｊ≦ｂ－１）から構成されるｑ行ｂ列（ｑ≦ｂ）の２元行列であり、（$\mu_1+\mu_2$）ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立であり、（$\mu_1+\mu_2$）ｔ≧ｂのとき、前記行列Ｈ’は単位行列を含む正則行列であり、（$\mu_1+\mu_2$）ｔ＜ｂのとき、前記行列Ｈ’は最小ハミング距離（$\mu_1+\mu_2$）ｔ＋１を有する符号の検査行列、すなわち、（$\mu_1+\mu_2$）ｔビット誤り検出機能を有する（ｂ、ｂ－ｑ）符号のパリティ検査行列に等しくなり、また、行列Ｈ”は、Ｈ”＝［ｈ$_0$”、ｈ$_1$

”、・・・、h_{b-1}”］に示すｒ次の列ベクトルh_j”（$0 \leqq j \leqq b-1$）から構成されるｒ行ｂ列（$r \leqq b$）の２元行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor t$またはｂのうちいずれか小さい値に等しいか大きい列が線形独立であり、ここで、$\lfloor x\rfloor$はｘを超えない最大の整数を表し、また、$\lfloor(\mu_1+\mu_2)/2\rfloor t \geqq b$のとき、前記行列Ｈ”は単位行列を含む正則行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor t < b$のとき、前記行列Ｈ”は最小ハミング距離$\lfloor(\mu_1+\mu_2)/2\rfloor t+1$を有する符号の検査行列、すなわち、$\lfloor(\mu_1+\mu_2)/2\rfloor t$ビット誤り検出機能を有する（ｂ、ｂ－ｒ）符号のパリティ検査行列に等しくなり、γはＧＦ（２）のｒ’次の拡大体ＧＦ（$2^{r'}$）の原始元であって、$\gamma^i H$”＝［$\gamma^i \Phi$（h_0”）、$\gamma^i \Phi$（h_1”）、・・・、$\gamma^i \Phi$（h_{b-1}”）］が成立するように構成され、ここで、Φは加法のもとでＧＦ（2^r）からＧＦ（$2^{r'}$）への単射準同型写像（injective homomorphism）であり、すなわち、Φ：ＧＦ（2^r）→ＧＦ（$2^{r'}$）である請求の範囲第２項乃至請求の範囲第６項のいずれかに記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

９．前記パリティ検査行列は、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正するとともに、p_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出し、かつ訂正能力を超えた１バイト誤りを検出する機能を有する、誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号（$[\mu_1 t/b\mathrm{EC}]_{p_1}-[\mu_2 t/b\mathrm{ED}]_{p_2}-\mathrm{S}_b\mathrm{ED}$符号）に用いる、Ｒ行Ｎ列を有する次の行列Ｈであり、

$$H=\begin{bmatrix} I_b & I_b & \cdots & I_b & \cdots & I_b \\ \gamma^0 H'' & \gamma^1 H'' & \cdots & \gamma^i H'' & \cdots & \gamma^{n-1} H'' \\ \gamma^0 H'' & \gamma^2 H'' & \cdots & \gamma^{2i} H'' & \cdots & \gamma^{2(n-1)} H'' \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma^0 H'' & \gamma^{p_1+p_2-1} H'' & \cdots & \gamma^{(p_1+p_2-1)i} H'' & \cdots & \gamma^{(p_1+p_2-1)(n-1)} H'' \end{bmatrix}$$

ただし、Ｒ＝ｂ＋（ｐ$_1$＋ｐ$_2$－１）ｒ、Ｎ＝ｂｎ、０≦ｉ≦ｎ－１、ｎ＝２^r－１、Ｉ$_b$はｂ次の単位行列であり、行列Ｈ”は、Ｈ”＝［ｈ$_0$”、ｈ$_1$”、・・・、ｈ$_{b-1}$”］に示すｒ次の列ベクトルｈ$_j$”（０≦ｊ≦ｂ－１）から構成されるｒ行ｂ列（ｒ≦ｂ）の２元行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立であり、ここで、$\lfloor x\rfloor$はｘを超えない最大の整数を表し、ここで、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ≧ｂのとき、前記行列Ｈ”は単位行列を含む正則行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＜ｂのとき、前記行列Ｈ”は最小ハミング距離$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＋１を有する符号の検査行列、すなわち、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔビット誤り検出機能を有する（ｂ、ｂ－ｒ）符号のパリティ検査行列に等しくなり、γはＧＦ（２）のｒ次の拡大体ＧＦ（２^r）の原始元であって、γ^{i}Ｈ”＝［γ^{i}ｈ$_0$”、γ^{i}ｈ$_1$”、…、γ^{i}ｈ$_{b-1}$”］が成立するように構成される請求の範囲第２項乃至請求の範囲第６項のいずれかに記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

１０．前記パリティ検査行列は、ｐ$_1$個のバイトに生じるμ_1個のスポッティバイト誤りを訂正するとともに、ｐ$_2$個のバイトに生じるμ_2個のスポッティバイト誤りを検出し、かつ訂正能力を超えた１バイト誤りを検出する機能を有する、誤りバイト数を制限した伸長バイト内複数ス

ポッティバイト誤り制御符号（伸長$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}-S_bED$符号）に用いる、Ｒ行Ｎ列を有する次の行列Ｈであり、

$$H=\begin{bmatrix} I_b & I_b & \cdots & I_b & \cdots & I_b & I_b & O_{b\times r'} \\ \gamma^0 H'' & \gamma^1 H'' & \cdots & \gamma^i H'' & \cdots & \gamma^{n-1} H'' & O_{r'\times b} & O_{r'\times r'} \\ \gamma^0 H'' & \gamma^2 H'' & \cdots & \gamma^{2i} H'' & \cdots & \gamma^{2(n-1)} H'' & O_{r'\times b} & O_{r'\times r'} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \gamma^0 H'' & \gamma^{p_1+p_2-1} H'' & \cdots & \gamma^{(p_1+p_2-1)i} H'' & \cdots & \gamma^{(p_1+p_2-1)(n-1)} H'' & O_{r'\times b} & I_{r'} \end{bmatrix}$$

ただし、ｒ’≧ｒであり、Ｒ＝ｂ＋（ｐ$_1$＋ｐ$_2$－１）ｒ’、Ｎ＝ｂ（ｎ＋１）＋ｒ’、０≦ｉ≦ｎ－１、ｎ＝２$^{r'}$－１、Ｉ$_b$はｂ次の単位行列であり、$I_{r'}$はｒ’次の単位行列で、Oは零行列であり、行列Ｈ”は、Ｈ”＝［ｈ$_0$”、ｈ$_1$”、・・・、ｈ$_{b-1}$”］に示すｒ次の列ベクトルｈ$_j$”（０≦ｊ≦ｂ－１）から構成されるｒ行ｂ列（ｒ≦ｂ）の２元行列であり、

ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立であり、ここで、$\lfloor x \rfloor$はｘを超えない最大の整数を表し、ここで、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ≧ｂのとき、前記行列Ｈ”は単位行列を含む正則行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＜ｂのとき、前記行列Ｈ”は最小ハミング距離$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＋１を有する符号の検査行列、すなわち、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔビット誤り検出機能を有する（ｂ、ｂ－ｒ）符号のパリティ検査行列に等しくなり、γはＧＦ（２）のｒ’次の拡大体ＧＦ（２$^{r'}$）の原始元であって、γ^{i}Ｈ”＝［γ^{i}Φ（ｈ$_0$”）、γ^{i}Φ（ｈ$_1$”）、・・・、γ^{i}Φ（ｈ$_{b-1}$”）］が成立するように構成され、ここで、Φは加法のもとでＧＦ（２^r）からＧＦ（２$^{r'}$）への単射準同型写像（injective homomorphism）であり、すなわち、Φ：ＧＦ（２^r）→ＧＦ（２$^{r'}$）である請求の範囲第２項乃至請求の範囲第６項のいずれかに記載の誤りバ

イト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

１１．前記誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号（$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}$符号）は、前記ｔ値を１に設定した場合（ｔ＝１）に、ｐ$_1$個のバイトに生じるμ_1個のランダムビット誤りを訂正し、かつｐ$_2$個のバイトに生じるμ_2個のランダムビット誤りを検出する機能を有する、誤りバイト数を制限したランダムビット誤り制御符号となる請求の範囲第７項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

１２．前記誤りバイト数を制限した伸長バイト内複数スポッティバイト誤り制御符号（伸長$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}$符号）は、前記ｔ値を１に設定した場合（ｔ＝１）に、ｐ$_1$個のバイトに生じるμ_1個のランダムビット誤りを訂正し、かつｐ$_2$個のバイトに生じるμ_2個のランダムビット誤りを検出する機能を有する、誤りバイト数を制限したランダムビット誤り制御符号となる請求の範囲第８項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

１３．前記誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号（$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}-S_b ED$符号）は、前記ｔ値を１に設定した場合（ｔ＝１）に、ｐ$_1$個のバイトに生じるμ_1個のランダムビット誤りを訂正するとともに、ｐ$_2$個のバイトに生じるμ_2個のランダムビット誤りを検出し、かつ訂正能力を超えた１バイト誤りを検出する機能を有する、誤りバイト数を制限したランダムビット誤り制御符号となる請求の範囲第９項に記載の誤りバイト数を制限したバイト内複数

スポッティバイト誤り訂正・検出装置。

１４．前記誤りバイト数を制限した伸長バイト内複数スポッティバイト誤り制御符号（伸長$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}-S_bED$符号）は、前記ｔ値を１に設定した場合（ｔ＝１）に、p_1個のバイトに生じるμ_1個のランダムビット誤りを訂正するとともに、p_2個のバイトに生じるμ_2個のランダムビット誤りを検出し、かつ訂正能力を超えた１バイト誤りを検出する機能を有する、誤りバイト数を制限したランダムビット誤り制御符号となる請求の範囲第１０項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

１５．前記パリティ検査行列（Ｈ）は、$t \geqq b/\lfloor(\mu_1+\mu_2)/2\rfloor$とした場合に、前記行列Ｈ’及び前記行列Ｈ”は任意のｂ列が線形独立な行列となり、p_1バイト誤りを訂正するとともにp_2バイト誤りを検出する機能を有するＲＳ符号のパリティ検査行列と一致する請求の範囲第７項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

１６．前記パリティ検査行列（Ｈ）は、$t \geqq b/\lfloor(\mu_1+\mu_2)/2\rfloor$とした場合に、前記行列Ｈ’及び前記行列Ｈ”は任意のｂ列が線形独立な行列となり、p_1バイト誤りを訂正するとともにp_2バイト誤りを検出する機能を有する２次伸長ＲＳ符号のパリティ検査行列と一致する請求の範囲第８項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

１７．前記パリティ検査行列（Ｈ）は、ｔ≧ｂ／$\lfloor(\mu_1+\mu_2)/2\rfloor$とした場合に、前記行列Ｈ”は任意のｂ列が線形独立な行列となり、p_1バイト誤りを訂正するとともにp_2バイト誤りを検出する機能を有するＲＳ符号のパリティ検査行列と一致する請求の範囲第９項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

１８．前記パリティ検査行列（Ｈ）は、ｔ≧ｂ／$\lfloor(\mu_1+\mu_2)/2\rfloor$とした場合に、前記行列Ｈ”は任意のｂ列が線形独立な行列となり、p_1バイト誤りを訂正するとともにp_2バイト誤りを検出する機能を有する２次伸長ＲＳ符号のパリティ検査行列と一致する請求の範囲第１０項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

１９．前記誤り訂正手段は、前記シンドロームを基に前記受信語の何れのビットが誤っているかを検出するビット誤りポインタを生成し、生成された前記ビット誤りポインタを基に、誤りビットに該当する前記受信語のビット値を反転させることにより、前記受信語の誤りを訂正し、また、前記受信語のビット誤りを訂正できないことを検出した場合に、訂正不可能誤り検出信号を出力する請求の範囲第１項乃至請求の範囲第１８項のいずれかに記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

２０．前記誤り訂正手段は、Ｈ’復号手段と、Ｈ”乗算手段と、ＧＦ（2^r）上の並列復号手段と、ＧＦ（2^b）上の誤り生成手段と、反転手段とから構成される請求の範囲第７項乃至請求の範囲第１８項のいずれ

かに記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

２１．前記Ｈ’復号手段は、前記シンドローム生成手段で生成された前記シンドロームの上位ｑビットを基にして、バイトごとの誤りの和（e_A）を生成し、

前記Ｈ”乗算手段は、前記Ｈ’復号手段で生成された前記バイトごとの誤りの和（e_A）を基にして、前記バイトごとの誤りの和（e_A）とＨ”の転置行列の積を生成し、

前記ＧＦ（2^r）上の並列復号手段は、前記Ｈ”乗算手段で生成された前記バイトごとの誤りの和（e_A）とＨ”の転置行列の積、及び前記シンドローム生成手段で生成された前記シンドロームの上位ｑビットを除く残りの下位ビットを基にして、ＧＦ（2^r）上のビット誤りポインタ、及びＧＦ（2^r）上の並列復号において誤り訂正が不可能とみなされたときに出力される第１の訂正不可能誤り検出信号を生成し、

前記ＧＦ（2^b）上の誤り生成手段は、前記Ｈ’復号手段で生成された前記バイトごとの誤りの和（e_A）、及び、前記ＧＦ（2^r）上の並列復号手段で生成された前記ＧＦ（2^r）上のビット誤りポインタを基にして、ＧＦ（2^b）上のビット誤りポインタ、及び誤り訂正が不可能とみなされたときに出力される第２の訂正不可能誤り検出信号を生成し、

前記反転手段は、前記ＧＦ（2^b）上の誤り生成手段で生成された前記ＧＦ（2^b）上のビット誤りポインタを基にして、前記ＧＦ（2^b）上のビット誤りポインタに対応した前記受信語のビット値を反転させることにより、前記受信語の誤りを訂正する請求の範囲第２０項に記載の

誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

２２．前記誤り訂正手段では、前記ＧＦ（２^{r}）上の並列復号手段で生成された前記第１の訂正不可能誤り検出信号と、前記ＧＦ（２^{b}）上の誤り生成手段で生成された前記第２の訂正不可能誤り検出信号の論理和をとることにより、訂正能力を超えた誤りを検出したことを示す第３の訂正不可能誤り検出信号を出力する請求の範囲第２１項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

２３．前記ＧＦ（２^{b}）上の誤り生成手段は、

前記ＧＦ（２^{r}）上のビット誤りポインタ（ｅ’）の１バイトから、復号結果ビット誤りポインタ（$\hat{e}$）の１バイト及びｉバイトにおける検出信号を出力するｔビット誤り訂正復号手段と、

入力された複数の前記ｉバイトにおける検出信号の重みの数だけを出力の上位ビットから１を連続して出力する多入力ソート手段と、

誤訂正判定を行い、前記ＧＦ（２^{b}）上のビット誤りポインタ（ｅ）の１バイトを出力するｉバイトにおける誤訂正判定手段と、

を備えた請求の範囲第２１項又は請求の範囲第２２項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

２４．前記ＧＦ（２^{b}）上の誤り生成手段は、ｔビット誤り訂正復号手段と、多入力ソート手段と、ｉバイトにおける誤訂正判定手段とを備えており、

前記ｔビット誤り訂正復号手段は、前記ＧＦ（２^{r}）上のビット誤り

ポインタ（e'）の所定の1バイトを入力とし、復号結果ビット誤りポインタ（$\hat{e}$）の前記所定の1バイトに対応する1バイトを前記iバイトにおける誤訂正判定手段に出力するとともに、訂正能力をこえた誤りを検出する信号であるiバイトにおける検出信号を前記多入力ソート手段と前記iバイトにおける誤訂正判定手段に出力する手段であり、

前記多入力ソート手段は、入力される複数の前記iバイトにおける検出信号の重み（1の数）を数え、その重みの数だけを出力の上位ビットから1を連続して出力する手段であり、

前記iバイトにおける誤訂正判定手段は、すべてのバイトの前記tビット誤り訂正復号手段から出力される前記復号結果ビット誤りポインタ（$\hat{e}$）、前記iバイトにおける検出信号、及び前記バイトごとの誤りの和（e_A）を入力とし、誤訂正判定を行い、前記GF（2^b）上のビット誤りポインタ（e）の1バイトを出力する手段である請求の範囲第21項又は請求の範囲第22項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

25．前記GF（2^b）上の誤り生成手段では、前記多入力ソート手段から出力された信号において、その上位2ビット目の信号を前記第2の訂正不可能誤り検出信号として出力し、また、前記iバイトにおける誤訂正判定手段から出力された信号を集め、符号ビット長のビット数を有する前記GF（2^b）上のビット誤りポインタを出力する請求の範囲第23項又は請求の範囲第24項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

26．前記iバイトにおける誤訂正判定手段では、前記誤訂正判定が次

の誤訂正判定数式によって行われ、

$$w_{\mathrm{M}}(\hat{e}_{\mathrm{x}}+e_{\mathrm{A}})\leq \mu_1 - w_{\mathrm{M}}(\hat{e}_{\mathrm{x}})$$

ただし、$w_{\mathrm{M}}(e)$は$w_{\mathrm{M}}(e)=\sum_{i=0}^{n-1}\left\lceil \frac{w_{\mathrm{H}}(e_i)}{t} \right\rceil$で定義され、スポッティバイト誤りの個数を表し、$w_{\mathrm{H}}(e_i)$はベクトル$e_i$におけるガロア体GF（2）上のハミング重みであり、

前記誤訂正数式が成立している場合に、$e_x=\hat{e}_x$となり誤訂正はされていないと判定され、前記誤訂正数式が成立していない場合に、$e_x\neq\hat{e}_x$となり誤訂正はされていると判定されるようにする請求の範囲第25項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

27．前記$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}-S_bED$符号、

及び前記伸長$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}-S_bED$符号に対しては、

前記H”乗算手段にS_I以外のシンドロームを新たに入力し、$S_I\neq 0$かつS_I以外のシンドロームが0の場合に、検出信号を出力し、

また、前記GF（2^b）上の誤り生成手段にS_Iを新たに入力し、前記GF（2^b）上の誤り生成手段において、GF（2^r）上の誤りe’において誤りバイト数が1かどうかを調べ、前記多入力ソート手段を用いてS_Iのハミング重みが$\mu_1 t$より大きいかどうかを調べることにより、前記復号手段を構成する請求の範囲第26項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出装置。

28．入力情報データを基に送信語を生成する符号化処理ステップと、情報伝送路中で誤りが発生した前記送信語を受信語として入力して前記

誤りを訂正または検出する復号処理ステップとを有する誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法であって、

前記符号化処理ステップは、スポッティバイト誤り制御符号を表現するパリティ検査行列と、前記入力情報データとを基に生成した検査情報を前記入力情報データに付加することにより、前記送信語を生成し、

前記復号処理ステップは、前記パリティ検査行列を基に前記受信語のシンドロームを生成するシンドローム生成処理ステップと、前記シンドローム生成処理ステップにより生成されたシンドロームを基に前記受信語の誤りを訂正または検出する誤り訂正処理ステップとを有し、

前記入力情報データが、b（bは2以上の整数）ビットを1バイトとし、複数のバイトから構成される場合に、ここで、バイト内のtビット（$1 \leqq t \leqq b$）までの誤りをスポッティバイト誤りと称し、1バイト内に複数の前記スポッティバイト誤りが生じる誤りをバイト内複数スポッティバイト誤りと称することを前提とし、

前記パリティ検査行列で表現される前記スポッティバイト誤り制御符号は、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正し、かつp_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出し、

前記t値、b値、p_1値、p_2値、μ_1値、μ_2値を任意に設定し、ただし、$p_1 \leqq p_2$及び$\mu_1 \leqq \mu_2$が成立し、

前記情報伝送路は、情報通信システム、メモリシステム、又はバス線回路の何れかであることを特徴とする誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法。

29．前記パリティ検査行列は、p_1個のバイトに生じるμ_1個のスポ

ッティバイト誤りを訂正し、かつｐ２個のバイトに生じるμ２個のスポッティバイト誤りを検出する機能を有する、誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号（$[\mu_1 t/b\mathrm{EC}]_{p_1}-[\mu_2 t/b\mathrm{ED}]_{p_2}$符号）に用いる、Ｒ行Ｎ列を有する次の行列Ｈであり、

$$H=\begin{bmatrix} H' & H' & \cdots & H' & \cdots & H' \\ \gamma^0 H'' & \gamma^1 H'' & \cdots & \gamma^i H'' & \cdots & \gamma^{n-1} H'' \\ \gamma^0 H'' & \gamma^2 H'' & \cdots & \gamma^{2i} H'' & \cdots & \gamma^{2(n-1)} H'' \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma^0 H'' & \gamma^{p_1+p_2-1} H'' & \cdots & \gamma^{(p_1+p_2-1)i} H'' & \cdots & \gamma^{(p_1+p_2-1)(n-1)} H'' \end{bmatrix}$$

ただし、Ｒ＝ｑ＋（ｐ１＋ｐ２－１）ｒ、Ｎ＝ｂｎ、０≦ｉ≦ｎ－１、ｎ＝２ｒ－１であり、行列Ｈ’は、Ｈ’＝［ｈ０’、ｈ１’、・・・、ｈｂ－１’］に示すｑ次の列ベクトルｈｊ’（０≦ｊ≦ｂ－１）から構成されるｑ行ｂ列（ｑ≦ｂ）の２元行列であり、（μ１＋μ２）ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立であり、（μ１＋μ２）ｔ≧ｂのとき、前記行列Ｈ’は単位行列を含む正則行列であり、（μ１＋μ２）ｔ＜ｂのとき、前記行列Ｈ’は最小ハミング距離（μ１＋μ２）ｔ＋１を有する符号の検査行列、すなわち、（μ１＋μ２）ｔビット誤り検出機能を有する（ｂ、ｂ－ｑ）符号のパリティ検査行列に等しくなり、また、行列Ｈ”は、Ｈ”＝［ｈ０”、ｈ１”、・・・、ｈｂ－１”］に示すｒ次の列ベクトルｈｊ”（０≦ｊ≦ｂ－１）から構成されるｒ行ｂ列（ｒ≦ｂ）の２元行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立であり、ここで、$\lfloor x\rfloor$はｘを超えない最大の整数を表し、また、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ≧ｂのとき、前記行列Ｈ”は単位行列を含む正則行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＜ｂのとき、前記行列Ｈ”は最小ハミング距離$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＋１を有する符号の検査行

列、すなわち、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔビット誤り検出機能を有する（ｂ、ｂ－ｒ）符号のパリティ検査行列に等しくなり、γはＧＦ（２）のｒ次の拡大体ＧＦ（２^{r}）の原始元であって、γ^{i}Ｈ”＝［γ^{i}ｈ$_{0}$”、γ^{i}ｈ$_{1}$”、…、γ^{i}ｈ$_{b-1}$”］が成立するように構成される請求の範囲第２８項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法。

３０．前記パリティ検査行列は、ｐ$_{1}$個のバイトに生じるμ_1個のスポッティバイト誤りを訂正し、かつｐ$_{2}$個のバイトに生じるμ_2個のスポッティバイト誤りを検出する機能を有する、誤りバイト数を制限した伸長バイト内複数スポッティバイト誤り制御符号（伸長$[\mu_1 t/b\mathrm{EC}]_{p_1}-[\mu_2 t/b\mathrm{ED}]_{p_2}$符号）に用いる、Ｒ行Ｎ列を有する次の行列Ｈであり、

$$H=\begin{bmatrix} H' & H' & \cdots & H' & \cdots & H' & H' & O_{q\times r'} \\ \gamma^0 H'' & \gamma^1 H'' & \cdots & \gamma^i H'' & \cdots & \gamma^{n-1} H'' & O_{r'\times b} & O_{r'\times r'} \\ \gamma^0 H'' & \gamma^2 H'' & \cdots & \gamma^{2i} H'' & \cdots & \gamma^{2(n-1)} H'' & O_{r'\times b} & O_{r'\times r'} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \gamma^0 H'' & \gamma^{p_1+p_2-1} H'' & \cdots & \gamma^{(p_1+p_2-1)i} H'' & \cdots & \gamma^{(p_1+p_2-1)(n-1)} H'' & O_{r'\times b} & I_{r'} \end{bmatrix}$$

ただし、ｒ’≧ｒであり、Ｒ＝ｑ＋（ｐ$_{1}$＋ｐ$_{2}$－１）ｒ’、Ｎ＝ｂ（ｎ＋１）＋ｒ’、０≦ｉ≦ｎ－１、ｎ＝２$^{r'}$－１であり、$I_{r'}$はｒ’次の単位行列で、Oは零行列であり、行列Ｈ’は、Ｈ’＝［ｈ$_{0}$’、ｈ$_{1}$’、・・・、ｈ$_{b-1}$’］に示すｑ次の列ベクトルｈ$_{j}$’（０≦ｊ≦ｂ－１）から構成されるｑ行ｂ列（ｑ≦ｂ）の２元行列であり、（$\mu_1+\mu_2$）ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立であり、（$\mu_1+\mu_2$）ｔ≧ｂのとき、前記行列Ｈ’は単位行列を含む正

則行列であり、（$\mu_1+\mu_2$）t＜bのとき、前記行列H'は最小ハミング距離（$\mu_1+\mu_2$）t＋1を有する符号の検査行列、すなわち、（$\mu_1+\mu_2$）tビット誤り検出機能を有する（b、b－q）符号のパリティ検査行列に等しくなり、また、行列H"は、H"＝[h_0"、h_1"、・・・、h_{b-1}"]に示すr次の列ベクトルh_j"（0≦j≦b－1）から構成されるr行b列（r≦b）の2元行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$tまたはbのうちいずれか小さい値に等しいか大きい列が線形独立であり、ここで、$\lfloor x\rfloor$はxを超えない最大の整数を表し、また、$\lfloor(\mu_1+\mu_2)/2\rfloor$t≧bのとき、前記行列H"は単位行列を含む正則行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$t＜bのとき、前記行列H"は最小ハミング距離$\lfloor(\mu_1+\mu_2)/2\rfloor$t＋1を有する符号の検査行列、すなわち、$\lfloor(\mu_1+\mu_2)/2\rfloor$tビット誤り検出機能を有する（b、b－r）符号のパリティ検査行列に等しくなり、γはGF(2)のr'次の拡大体GF($2^{r'}$)の原始元であって、γ^iH"＝[$\gamma^i\Phi(h_0")$、$\gamma^i\Phi(h_1")$、・・・、$\gamma^i\Phi(h_{b-1}")$]が成立するように構成され、ここで、Φは加法のもとでGF(2^r)からGF($2^{r'}$)への単射準同型写像（injective homomorphism）であり、すなわち、Φ：GF(2^r)→GF($2^{r'}$)である請求の範囲第28項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法。

31．前記パリティ検査行列は、p_1個のバイトに生じるμ_1個のスポッティバイト誤りを訂正するとともに、p_2個のバイトに生じるμ_2個のスポッティバイト誤りを検出し、かつ訂正能力を超えた1バイト誤りを検出する機能を有する、誤りバイト数を制限したバイト内複数スポッティバイト誤り制御符号（$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}-S_bED$符号）に用

いる、Ｒ行Ｎ列を有する次の行列Ｈであり、

$$H=\begin{bmatrix} I_b & I_b & \cdots & I_b & \cdots & I_b \\ \gamma^0 H'' & \gamma^1 H'' & \cdots & \gamma^i H'' & \cdots & \gamma^{n-1} H'' \\ \gamma^0 H'' & \gamma^2 H'' & \cdots & \gamma^{2i} H'' & \cdots & \gamma^{2(n-1)} H'' \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma^0 H'' & \gamma^{p_1+p_2-1} H'' & \cdots & \gamma^{(p_1+p_2-1)i} H'' & \cdots & \gamma^{(p_1+p_2-1)(n-1)} H'' \end{bmatrix}$$

ただし、Ｒ＝ｂ＋（ｐ$_1$＋ｐ$_2$－１）ｒ、Ｎ＝ｂｎ、０≦ｉ≦ｎ－１、ｎ＝２^r－１、Ｉ$_b$はｂ次の単位行列であり、行列Ｈ”は、Ｈ”＝［ｈ$_0$”、ｈ$_1$”、・・・、ｈ$_{b-1}$”］に示すｒ次の列ベクトルｈ$_j$”（０≦ｊ≦ｂ－１）から構成されるｒ行ｂ列（ｒ≦ｂ）の２元行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立であり、ここで、$\lfloor x\rfloor$はｘを超えない最大の整数を表し、ここで、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ≧ｂのとき、前記行列Ｈ”は単位行列を含む正則行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＜ｂのとき、前記行列Ｈ”は最小ハミング距離$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＋１を有する符号の検査行列、すなわち、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔビット誤り検出機能を有する（ｂ、ｂ－ｒ）符号のパリティ検査行列に等しくなり、γはＧＦ（２）のｒ次の拡大体ＧＦ（２^r）の原始元であって、γ^{i}Ｈ”＝［γ^{i}ｈ$_0$”、γ^{i}ｈ$_1$”、…、γ^{i}ｈ$_{b-1}$”］が成立するように構成される請求の範囲第２８項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法。

３２．前記パリティ検査行列は、ｐ$_1$個のバイトに生じるμ_1個のスポッティバイト誤りを訂正するとともに、ｐ$_2$個のバイトに生じるμ_2個のスポッティバイト誤りを検出し、かつ訂正能力を超えた１バイト誤りを検出する機能を有する、誤りバイト数を制限した伸長バイト内複数ス

ポッティバイト誤り制御符号（伸長$[\mu_1 t/bEC]_{p_1}-[\mu_2 t/bED]_{p_2}-S_bED$符号）に用いる、Ｒ行Ｎ列を有する次の行列Ｈであり、

$$H=\begin{bmatrix} I_b & I_b & \cdots & I_b & \cdots & I_b & I_b & O_{b\times r'} \\ \gamma^0 H'' & \gamma^1 H'' & \cdots & \gamma^i H'' & \cdots & \gamma^{n-1} H'' & O_{r'\times b} & O_{r'\times r'} \\ \gamma^0 H'' & \gamma^2 H'' & \cdots & \gamma^{2i} H'' & \cdots & \gamma^{2(n-1)} H'' & O_{r'\times b} & O_{r'\times r'} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \gamma^0 H'' & \gamma^{p_1+p_2-1} H'' & \cdots & \gamma^{(p_1+p_2-1)i} H'' & \cdots & \gamma^{(p_1+p_2-1)(n-1)} H'' & O_{r'\times b} & I_{r'} \end{bmatrix}$$

ただし、ｒ’≧ｒであり、Ｒ＝ｂ＋（ｐ$_1$＋ｐ$_2$－１）ｒ’、Ｎ＝ｂ（ｎ＋１）＋ｒ’、０≦ｉ≦ｎ－１、ｎ＝２$^{r'}$－１、Ｉ$_b$はｂ次の単位行列であり、$I_{r'}$はｒ’次の単位行列で、Oは零行列であり、行列Ｈ”は、Ｈ”＝［ｈ$_0$”、ｈ$_1$”、・・・、ｈ$_{b-1}$”］に示すｒ次の列ベクトルｈ$_j$”（０≦ｊ≦ｂ－１）から構成されるｒ行ｂ列（ｒ≦ｂ）の２元行列であり、

ｔまたはｂのうちいずれか小さい値に等しいか大きい列が線形独立であり、ここで、$\lfloor x \rfloor$はｘを超えない最大の整数を表し、ここで、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ≧ｂのとき、前記行列Ｈ”は単位行列を含む正則行列であり、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＜ｂのとき、前記行列Ｈ”は最小ハミング距離$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔ＋１を有する符号の検査行列、すなわち、$\lfloor(\mu_1+\mu_2)/2\rfloor$ｔビット誤り検出機能を有する（ｂ、ｂ－ｒ）符号のパリティ検査行列に等しくなり、γはＧＦ（２）のｒ’次の拡大体ＧＦ（２$^{r'}$）の原始元であって、γ^{i}Ｈ”＝［γ^{i}Φ（ｈ$_0$”）、γ^{i}Φ（ｈ$_1$”）、・・・、γ^{i}Φ（ｈ$_{b-1}$”）］が成立するように構成され、ここで、Φは加法のもとでＧＦ（２^r）からＧＦ（２$^{r'}$）への単射準同型写像（injective homomorphism）であり、すなわち、Φ：ＧＦ（２^r）→ＧＦ（２$^{r'}$）である請求の範囲第２８項に記載の誤りバイト数を制限したバイト内複数

スポッティバイト誤り訂正・検出方法。

３３．前記誤り訂正処理ステップは、前記シンドロームを基に前記受信語の何れのビットが誤っているかを検出するビット誤りポインタを生成し、生成された前記ビット誤りポインタを基に、誤りビットに該当する前記受信語のビット値を反転させることにより、前記受信語の誤りを訂正し、また、前記受信語のビット誤りを訂正できないことを検出した場合に、訂正不可能誤り検出信号を出力する請求の範囲第２８項乃至請求の範囲第３２項のいずれかに記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法。

３４．前記誤り訂正処理ステップは、H’復号処理ステップと、H”乗算処理ステップと、ＧＦ（2^r）上の並列復号処理ステップと、ＧＦ（2^b）上の誤り生成処理ステップと、反転処理ステップとを有する請求の範囲第２９項乃至請求の範囲第３２項のいずれかに記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法。

３５．前記H’復号処理ステップは、前記シンドローム生成処理ステップで生成された前記シンドロームの上位ｑビットを基にして、バイトごとの誤りの和（e_A）を生成し、

前記H”乗算処理ステップは、前記H’復号処理ステップで生成された前記バイトごとの誤りの和（e_A）を基にして、前記バイトごとの誤りの和（e_A）とH”の転置行列の積を生成し、

前記ＧＦ（2^r）上の並列復号処理ステップは、前記H”乗算処理ステップで生成された前記バイトごとの誤りの和（e_A）とH”の転置行

列の積、及び前記シンドローム生成処理ステップで生成された前記シンドロームの上位ｑビットを除く残りの下位ビットを基にして、ＧＦ（２^{r}）上のビット誤りポインタ、及びＧＦ（２^{r}）上の並列復号において誤り訂正が不可能とみなされたときに出力される第１の訂正不可能誤り検出信号を生成し、

前記ＧＦ（２^{b}）上の誤り生成処理ステップは、前記Ｈ’復号処理ステップで生成された前記バイトごとの誤りの和（ｅ$_{A}$）、及び、前記ＧＦ（２^{r}）上の並列復号処理ステップで生成された前記ＧＦ（２^{r}）上のビット誤りポインタを基にして、ＧＦ（２^{b}）上のビット誤りポインタ、及び誤り訂正が不可能とみなされたときに出力される第２の訂正不可能誤り検出信号を生成し、

前記反転処理ステップは、前記ＧＦ（２^{b}）上の誤り生成処理ステップで生成された前記ＧＦ（２^{b}）上のビット誤りポインタを基にして、前記ＧＦ（２^{b}）上のビット誤りポインタに対応した前記受信語のビット値を反転させることにより、前記受信語の誤りを訂正する請求の範囲第３４項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法。

３６．前記誤り訂正処理ステップでは、前記ＧＦ（２^{r}）上の並列復号処理ステップで生成された前記第１の訂正不可能誤り検出信号と、前記ＧＦ（２^{b}）上の誤り生成処理ステップで生成された前記第２の訂正不可能誤り検出信号の論理和をとることにより、訂正能力を超えた誤りを検出したことを示す第３の訂正不可能誤り検出信号を出力する請求の範囲第３５項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法。

３７．前記ＧＦ（２^{b}）上の誤り生成処理ステップは、

前記ＧＦ（２^{r}）上のビット誤りポインタ（ｅ’）の１バイトから、復号結果ビット誤りポインタ（$\hat{e}$）の１バイト及びｉバイトにおける検出信号を出力するｔビット誤り訂正復号処理ステップと、

入力された複数の前記ｉバイトにおける検出信号の重みの数だけを出力の上位ビットから１を連続して出力する多入力ソート処理ステップと、

誤訂正判定を行い、前記ＧＦ（２^{b}）上のビット誤りポインタ（ｅ）の１バイトを出力するｉバイトにおける誤訂正判定処理ステップと、

を有する請求の範囲第３５項又は請求の範囲第３６項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法。

３８．前記ＧＦ（２^{b}）上の誤り生成処理ステップは、ｔビット誤り訂正復号処理ステップと、多入力ソート処理ステップと、ｉバイトにおける誤訂正判定処理ステップとを有し、

前記ｔビット誤り訂正復号処理ステップは、前記ＧＦ（２^{r}）上のビット誤りポインタ（ｅ’）の所定の１バイトを入力とし、復号結果ビット誤りポインタ（$\hat{e}$）の前記所定の１バイトに対応する１バイトを前記ｉバイトにおける誤訂正判定処理ステップに出力するとともに、訂正能力をこえた誤りを

検出する信号であるｉバイトにおける検出信号を前記多入力ソート処理ステップと前記ｉバイトにおける誤訂正判定処理ステップに出力し、

前記多入力ソート処理ステップは、入力される複数の前記ｉバイトにおける検出信号の重み（１の数）を数え、その重みの数だけを出力の上

位ビットから１を連続して出力し、

前記ｉバイトにおける誤訂正判定処理ステップは、すべてのバイトの前記ｔビット誤り訂正復号処理ステップから出力される前記復号結果ビット誤りポインタ（$\hat{e}$）、前記ｉバイトにおける検出信号、及び前記バイトごとの誤りの和（e_A）を入力とし、誤訂正判定を行い、前記ＧＦ（2^b）上のビット誤りポインタ（ｅ）の１バイトを出力する請求の範囲第３５項又は請求の範囲第３６項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法。

３９．前記ＧＦ（2^b）上の誤り生成処理ステップでは、前記多入力ソート処理ステップから出力された信号において、その上位２ビット目の信号を前記第２の訂正不可能誤り検出信号として出力し、また、前記ｉバイトにおける誤訂正判定処理ステップから出力された信号を集め、符号ビット長のビット数を有する前記ＧＦ（2^b）上のビット誤りポインタを出力する請求の範囲第３７項又は請求の範囲第３８項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法。

４０．前記ｉバイトにおける誤訂正判定処理ステップでは、前記誤訂正判定が次の誤訂正判定数式によって行われ、

$$w_M(\hat{e}_x + e_A) \le \mu_1 - w_M(\hat{e}_x)$$

ただし、$w_M(e)$は$w_M(e) = \sum_{i=0}^{n-1} \left\lceil \frac{w_H(e_i)}{t} \right\rceil$で定義され、スポッティバイト誤りの個数を表し、$w_H(e_i)$はベクトルｅ$_i$におけるガロア体ＧＦ（２）上のハミング重みであり、

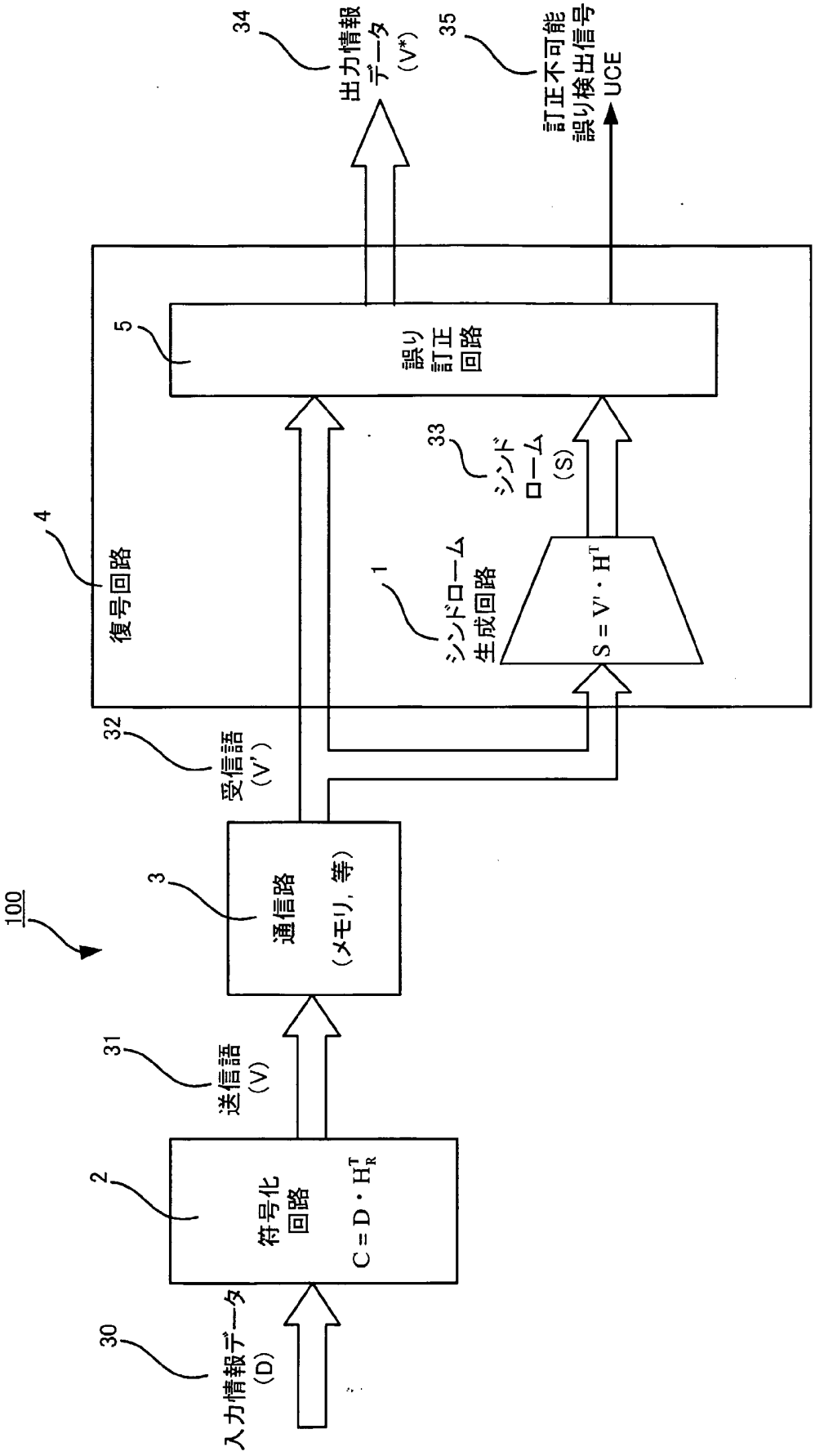
前記誤訂正数式が成立している場合に、$e_x = \hat{e}_x$となり誤訂正はされていないと判定され、前記誤訂正数式が成立していない場合に、$e_x \neq \hat{e}_x$となり誤訂正はされていると判定されるようにする請求の範囲第39項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法。

41．前記$[\mu_1 t/bEC]_{p_1} - [\mu_2 t/bED]_{p_2} - S_b ED$符号、
及び前記伸長$[\mu_1 t/bEC]_{p_1} - [\mu_2 t/bED]_{p_2} - S_b ED$符号に対しては、

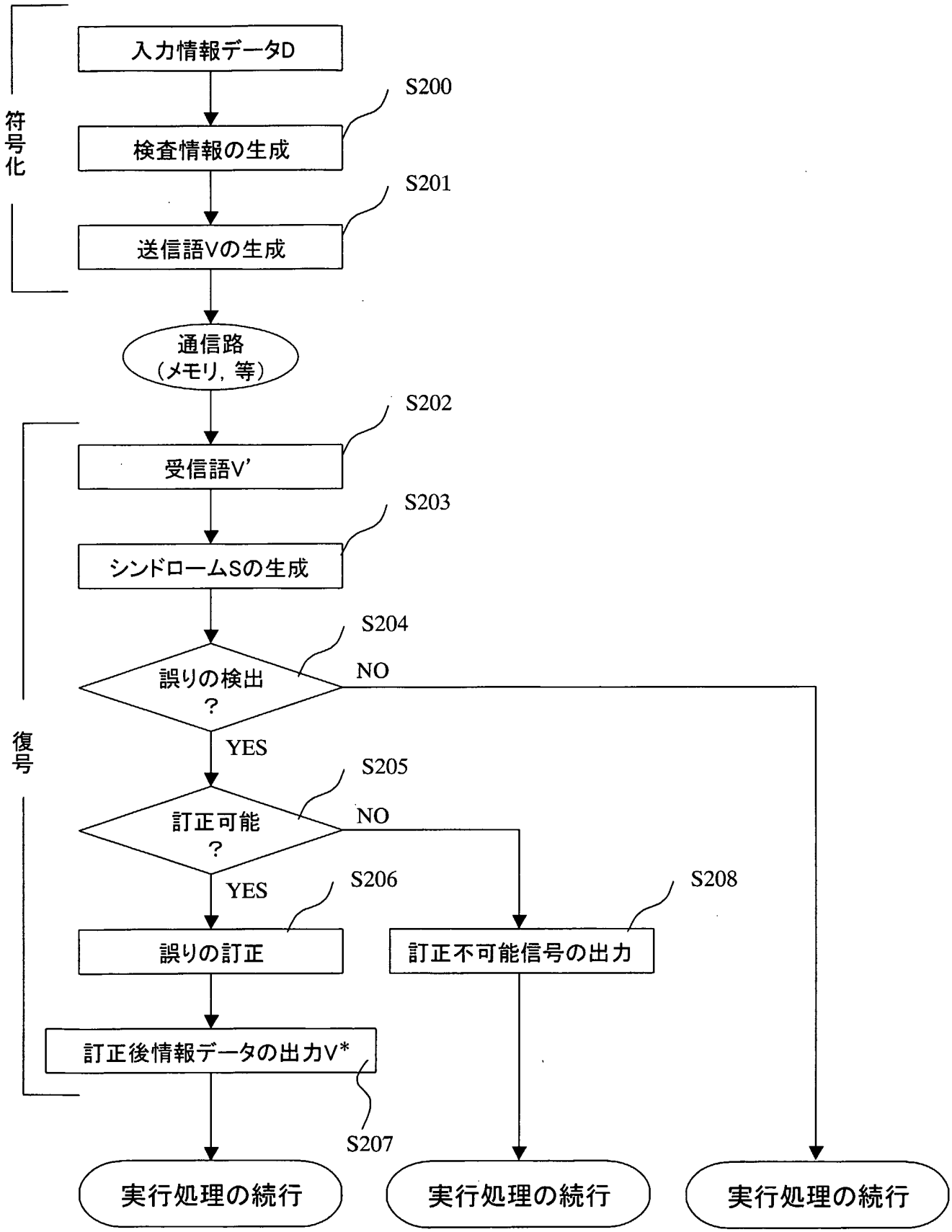
前記H”乗算処理ステップにS_I以外のシンドロームを新たに入力し、$S_I \neq 0$かつS_I以外のシンドロームが0の場合に、検出信号を出力し、

また、前記GF（2^b）上の誤り生成処理ステップにS_Iを新たに入力し、前記GF（2^b）上の誤り生成処理ステップにおいて、GF（2^r）上の誤りe’において誤りバイト数が1かどうかを調べ、前記多入力ソート処理ステップでS_Iのハミング重みが$\mu_1 t$より大きいかどうかを調べることにより、復号処理を行う請求の範囲第40項に記載の誤りバイト数を制限したバイト内複数スポッティバイト誤り訂正・検出方法。

第1図


100
30
入力情報データ
(D)
2
符号化
回路
C = D・H_R^T
31
送信語
(V)
3
通信路
(メモリ, 等)
32
受信語
(V')
4
復号回路
1
シンドローム
生成回路
S = V'・H^T
33
シンド
ローム
(S)
5
誤り
訂正
回路
34
出力情報
データ
(V*)
35
訂正不可能
誤り検出信号
UCE

第2図

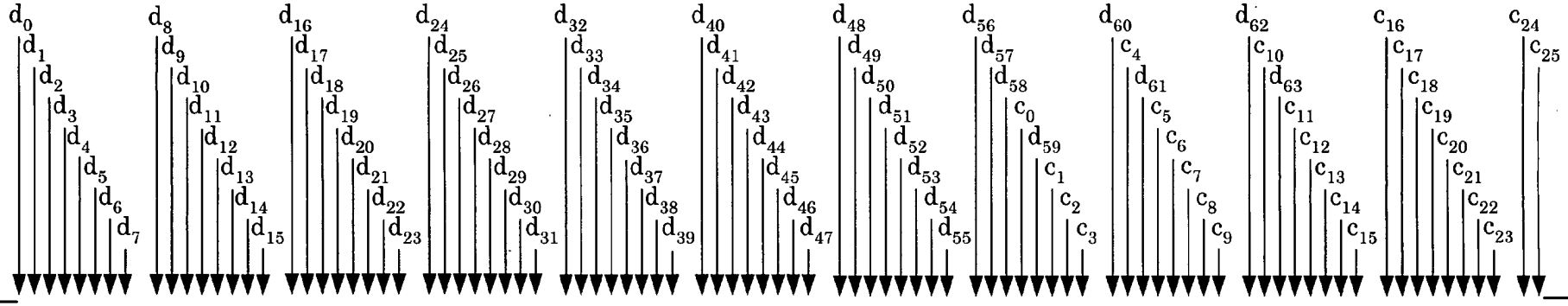

入力情報データD
S200
検査情報の生成
S201
送信語Vの生成
符号化
通信路
（メモリ，等）
S202
受信語V'
S203
シンドロームSの生成
S204
誤りの検出
？
NO
YES
S205
訂正可能
？
NO
YES
S206
S208
誤りの訂正
訂正不可能信号の出力
訂正後情報データの出力V*
S207
復号
実行処理の続行
実行処理の続行
実行処理の続行

第3図

10000000	10000000	10000000	10000000	10000000	10000000	10000000	10000000	10000000	10000000	10000000	10
01000000	01000000	01000000	01000000	01000000	01000000	01000000	01000000	01000000	01000000	01000000	01
00100000	00100000	00100000	00100000	00100000	00100000	00100000	00100000	00100000	00100000	00100000	00
00010000	00010000	00010000	00010000	00010000	00010000	00010000	00010000	00010000	00010000	00010000	00
00001000	00001000	00001000	00001000	00001000	00001000	00001000	00001000	00001000	00001000	00001000	00
00000100	00000100	00000100	00000100	00000100	00000100	00000100	00000100	00000100	00000100	00000100	00
00000010	00000010	00000010	00000010	00000010	00000010	00000010	00000010	00000010	00000010	00000010	00
00000001	00000001	00000001	00000001	00000001	00000001	00000001	00000001	00000001	00000001	00000001	00
10000010	00000101	00001001	00010011	00100011	01000010	10000111	00001100	00011010	00110000	01100001	11
01000010	10000111	00001100	00011010	00110000	01100001	11000101	10001011	00010110	00101010	01010001	10
00100011	01000010	10000111	00001100	00011010	00110000	01100001	11000101	10001011	00010110	00101010	01
00010011	00100011	01000010	10000111	00001100	00011010	00110000	01100001	11000101	10001011	00010110	00
00001001	00010011	00100011	01000010	10000111	00001100	00011010	00110000	01100001	11000101	10001011	00
00000101	00001001	00010011	00100011	01000010	10000111	00001100	00011010	00110000	01100001	11000101	10
10000010	00001001	00100011	10000111	00011010	01100001	10001011	00101010	10100100	10011101	01111011	11
01000010	00001100	00110000	11000101	00010110	01010001	01001110	00111100	11110101	11010011	01000111	00
00100011	10000111	00011010	01100001	10001011	00101010	10100100	10011101	01111011	11101010	10100001	10
00010011	01000010	00001100	00110000	11000101	00010110	01010001	01001110	00111100	11110101	11010011	01
00001001	00100011	10000111	00011010	01100001	10001011	00101010	10100100	10011101	01111011	11101010	10
00000101	00010011	01000010	00001100	00110000	11000101	00010110	01010001	01001110	00111100	11110101	11
10000010	00010011	10000111	00110000	10001011	01010001	10011101	11110101	10100001	00011111	11100110	00
01000010	00011010	11000101	00101010	01001110	01111011	11010011	10001110	01110010	10010001	10010100	10
00100011	00001100	01100001	00010110	10100100	00111100	11101010	01000111	00111001	11001001	01001011	01
00010011	10000111	00110000	10001011	01010001	10011101	11110101	10100001	00011111	11100110	00100110	00
00001001	01000010	00011010	11000101	00101010	01001110	01111011	11010011	10001110	01110010	10010001	10
00000101	00100011	00001100	01100001	00010110	10100100	00111100	11101010	01000111	00111001	11001001	01

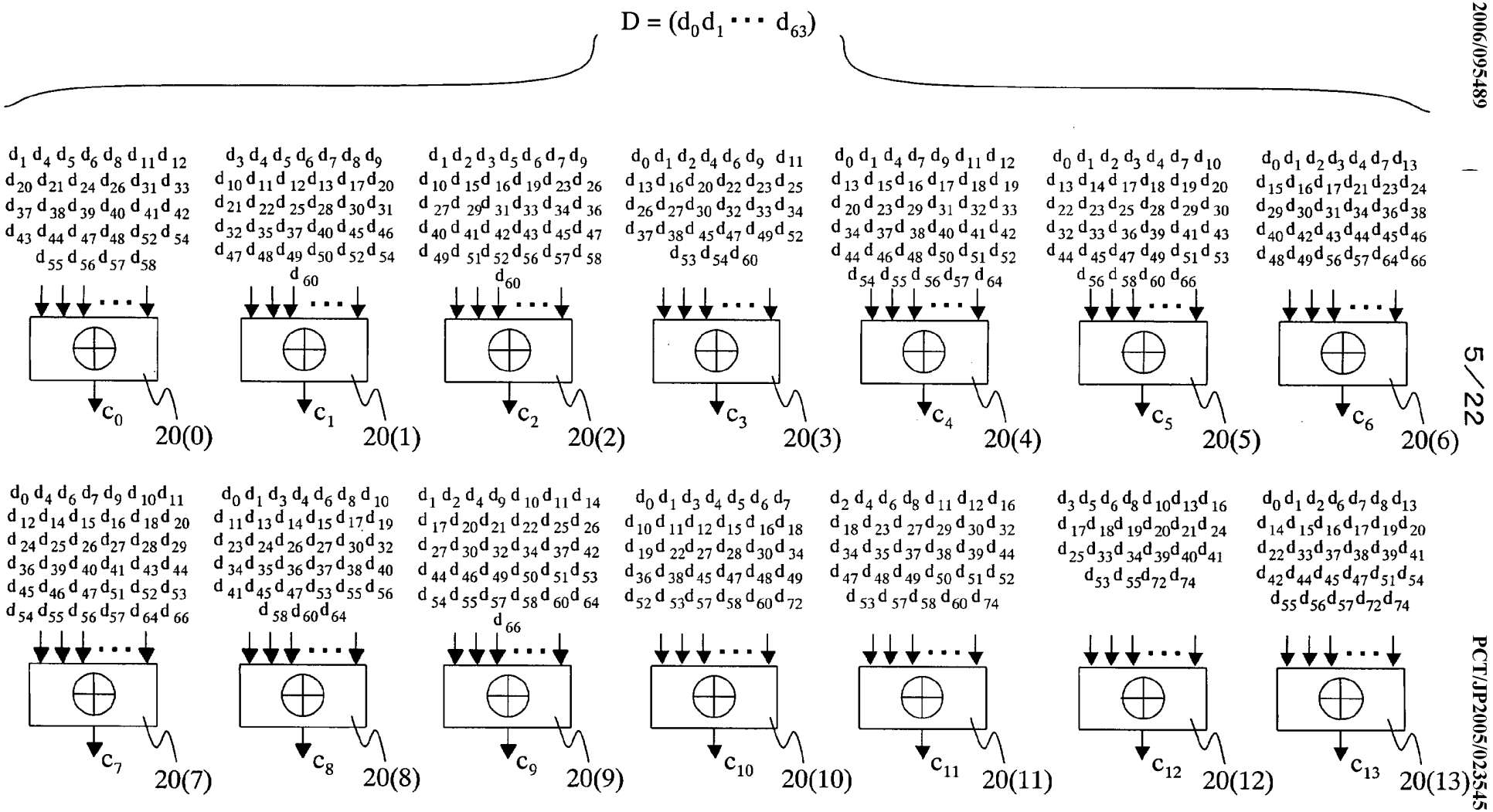
a部分

第4図



01001110	10011000	00001100	10100001	01000111	11111001	10001011	11110000	00000000	00000000	00000000	00
00011111	11111100	01001110	01001011	10010100	10000111	11101010	00001100	00000000	00000000	00000000	00
01110111	01100001	10010001	00110101	01101000	11110101	01011000	11101010	00000000	00000000	00000000	00
11101010	01010100	10001011	01110010	11100110	00000101	01001110	00001001	00000000	00000000	00000000	00
11001001	01011101	11111001	00000101	11100110	11101010	10111011	11000000	11000000	00000000	00000000	00
11111001	00100110	01111011	01001110	11001001	01011101	01010100	10101000	00110000	00000000	00000000	00
11111001	00000101	11000101	10000111	00101010	10111110	11000000	11000000	10101000	00000000	00000000	00
10001011	01111011	10101000	11111100	00001001	11011111	00011111	11000000	10100100	00000000	00000000	00
11011010	10110111	01010001	10110010	10111110	11000101	00000101	10101000	10000010	00000000	00000000	00
01101000	01110010	01001110	01110010	10100100	00101010	01110111	01101000	10100001	00000000	00000000	00
11011111	00111001	10110010	00011010	00101010	00000101	11001100	01101000	00000000	11000000	00000000	00
00101010	10011000	10100001	00010110	10110111	00001001	11111100	01101000	00000000	00110000	00000000	00
00010110	10100100	11111100	11000000	01100001	11000000	00000101	00000000	00000000	10101000	00000000	00
11100011	10000111	11011010	00000000	01000111	01101101	00010011	11000000	00000000	10100100	00000000	00
10011101	01110111	10110111	11101111	11101010	10000010	10100001	11000000	00000000	10000010	00000000	00
10010100	01100001	00010110	01100001	11100011	01111110	00100110	01101000	00000000	10100001	00000000	00
10001100	00110111	01110000	10101010	11010001	10110101	10001001	11101000	10000000	10000000	10000000	00
01101010	01001100	00111011	11010001	10010011	01001100	00011101	00101000	10000000	10000000	01000000	00
00100000	00100000	00100000	00100000	00100000	00100000	00100000	00100000	00100000	00100000	00100000	00
10001101	00110110	11000110	11101001	00101001	10111101	00110011	00100000	00100000	00100000	00010000	00
11100111	10101001	00110001	01001111	01000011	01110110	11001101	11001000	10100000	10100000	00001000	00
01110011	00000100	00111000	10110011	11011110	00110001	11100010	00001000	10100000	10100000	00000100	00
00110010	10100011	01110101	01101010	00111110	10110000	11111110	10000000	10000000	10000000	00000010	00
00010111	01000110	11010010	01100000	10100000	01010000	00011110	00001000	10100000	10100000	00000001	00
00001100	10110111	11110000	00101010	01010001	00110101	00001001	01101000	00000000	00000000	00000000	10
00111100	01101000	00110000	10001110	00011111	11100011	00101010	11000000	00000000	00000000	00000000	01

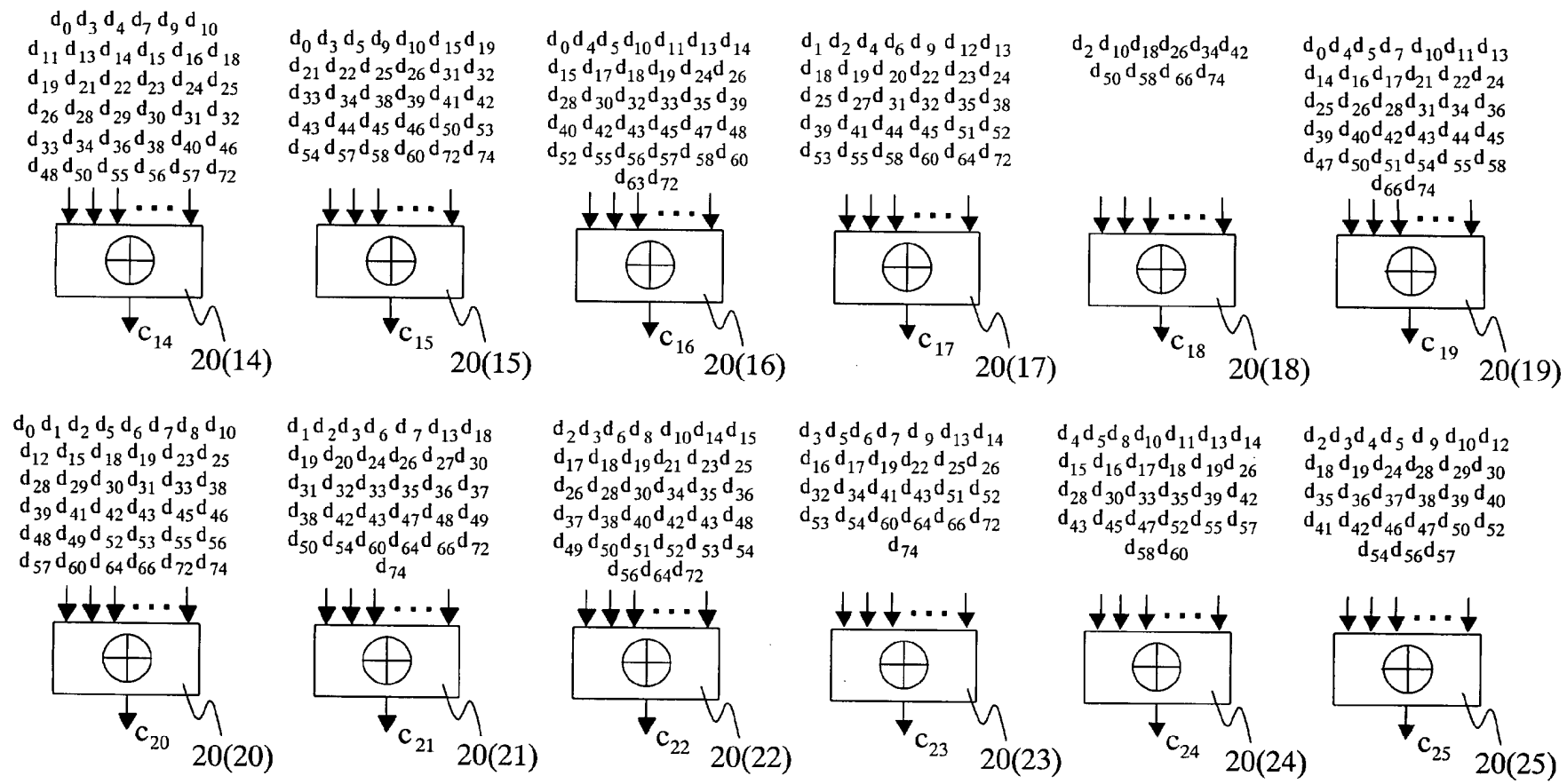
第5図



$$V = (d_0 d_1 \cdots d_{57} d_{58} c_0 d_{59} c_1 \cdots c_3 d_{60} c_4 d_{61} c_5 \cdots c_9 d_{62} c_{10} d_{63} c_{11} \cdots c_{25})$$

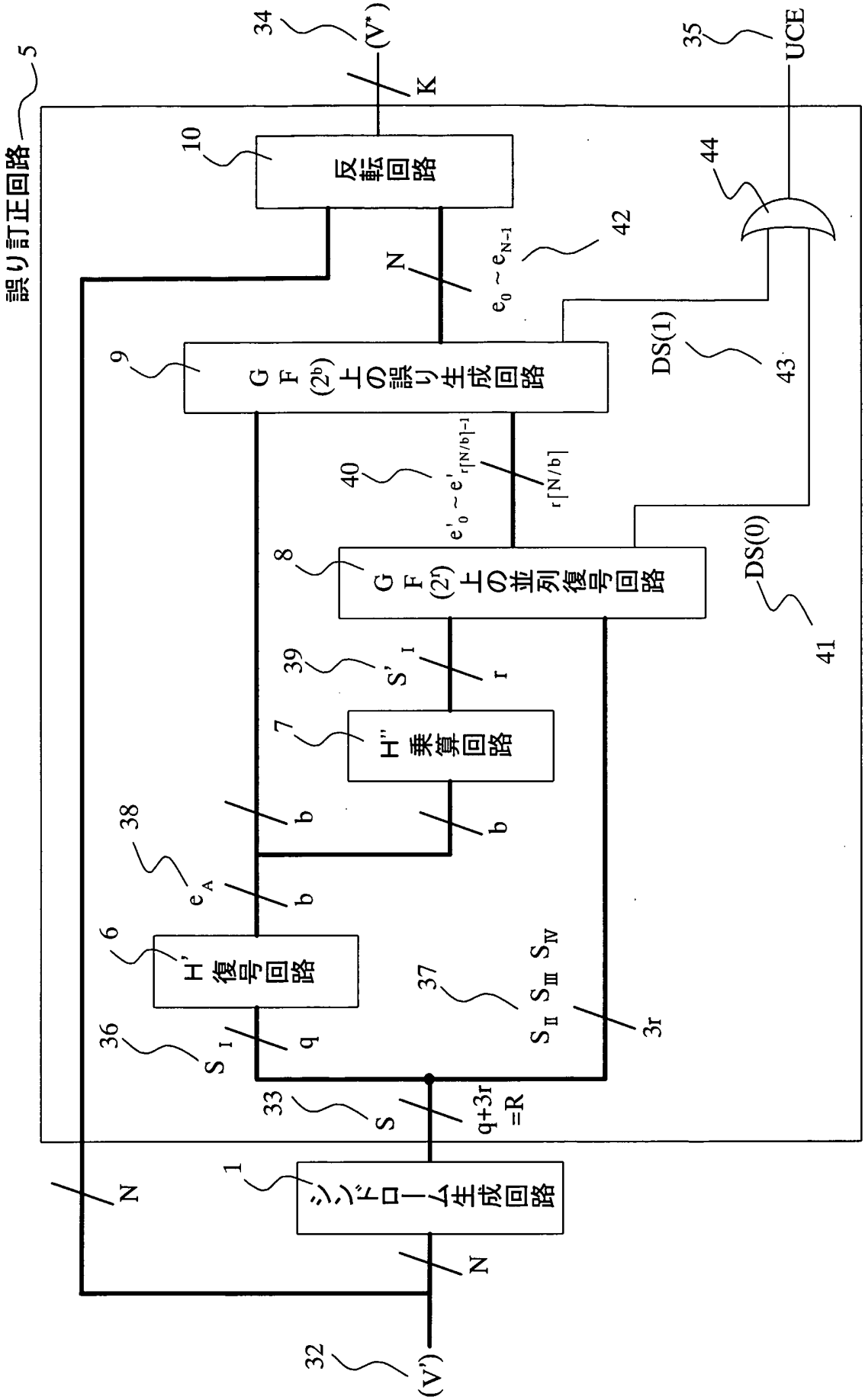
第6図

$D = (d_0 d_1 \cdots d_{63})$

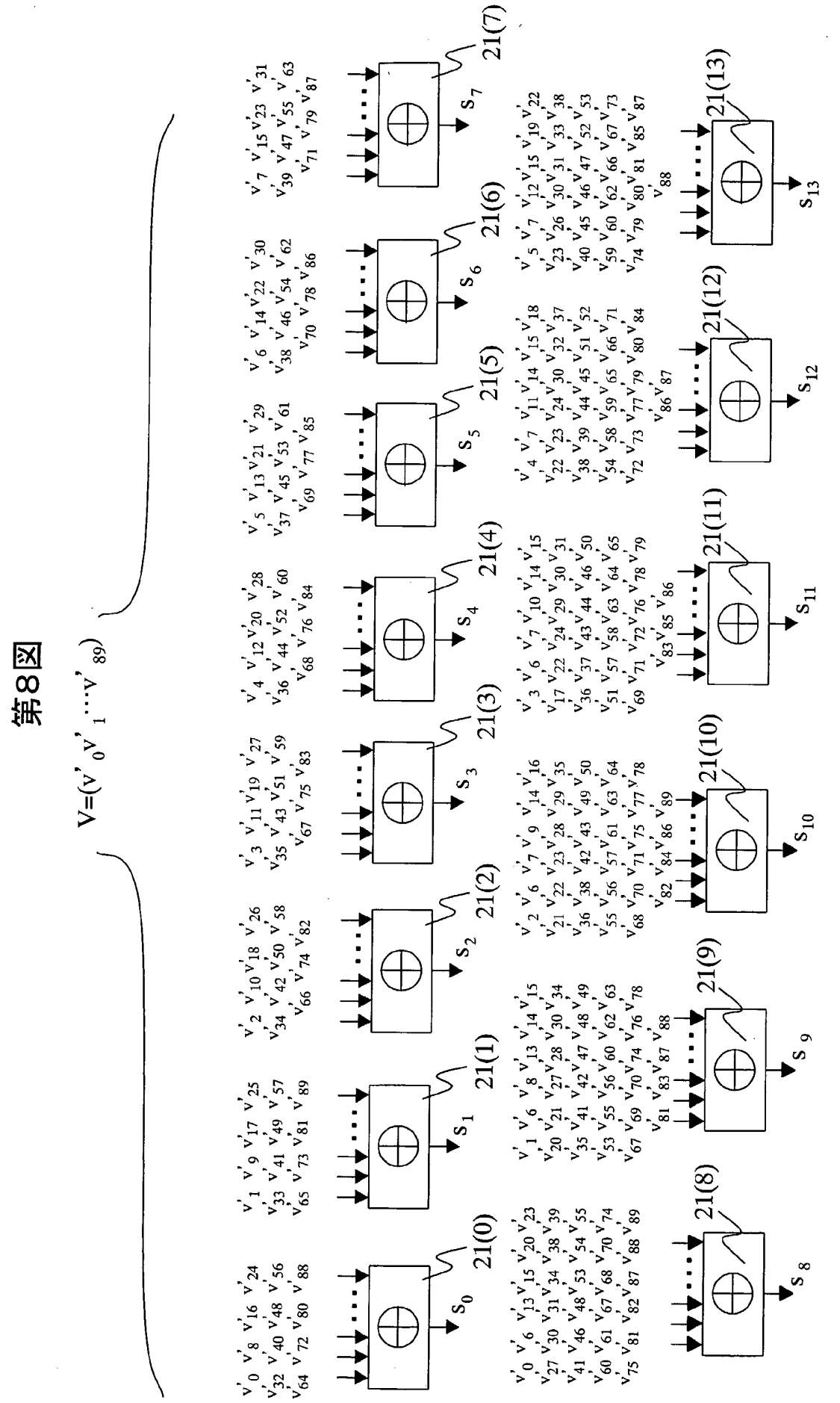


$V = (d_0 d_1 \cdots d_{57} d_{58} c_0 d_{59} c_1 \cdots c_3 d_{60} c_4 d_{61} c_5 \cdots c_9 d_{62} c_{10} d_{63} c_{11} \cdots c_{25})$

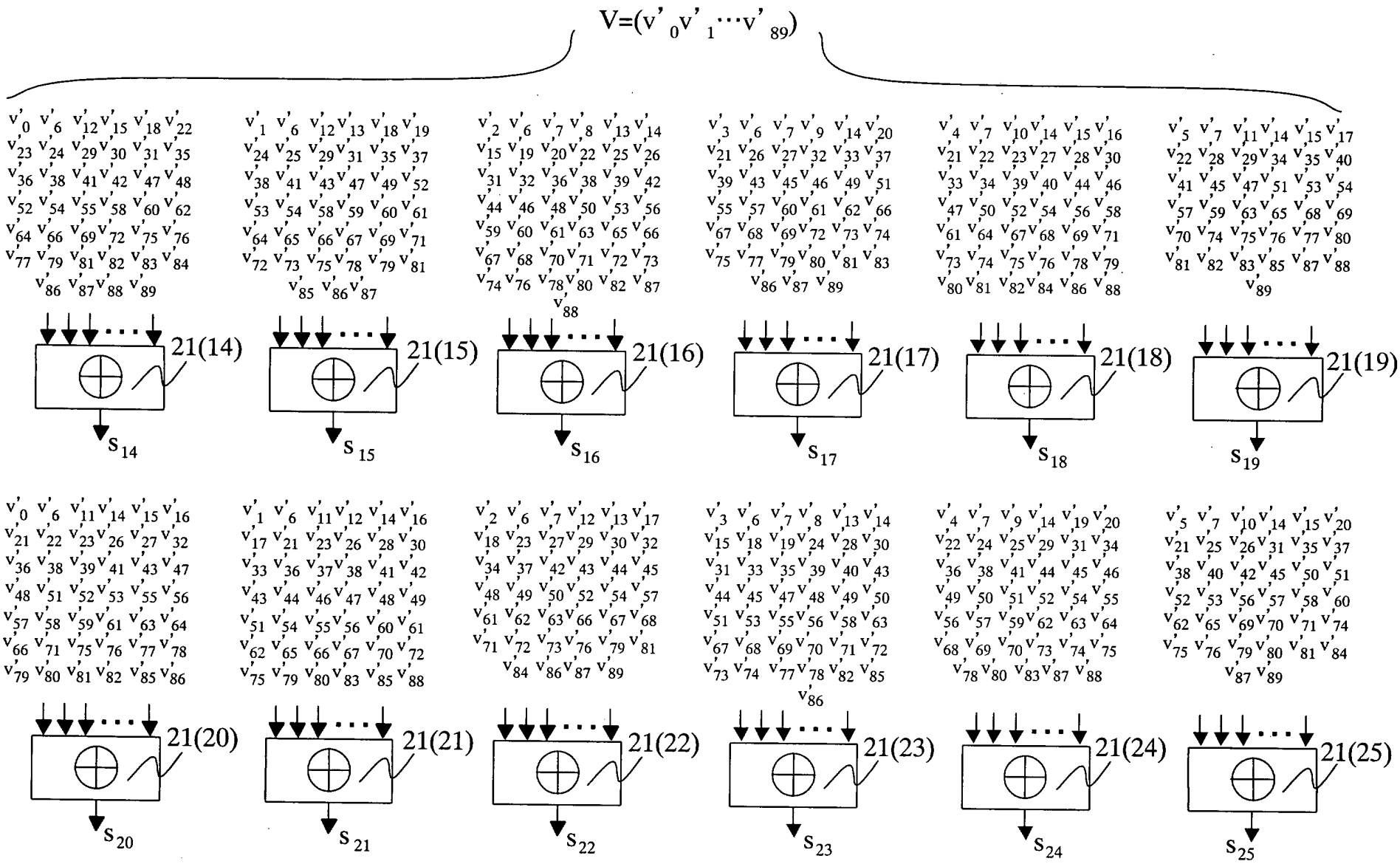
第7図


誤り訂正回路
5
34
(V*)
K
35
UCE
10
反転回路
44
N
42
$e_0 \sim e_{N-1}$
DS(1)
43
9
GF(2^b)上の誤り生成回路
40
$e'_0 \sim e'_{r[N/b]-1}$
$r_{[N/b]}$
8
GF(2^r)上の並列復号回路
DS(0)
41
39
S'_I
r
7
H'' 乗算回路
b
38
e_A
b
6
H' 復号回路
37
S_{II} S_{III} S_{IV}
3r
36
S_I
q
33
S
q+3r
=R
1
シンドローム生成回路
N
N
32
(V')

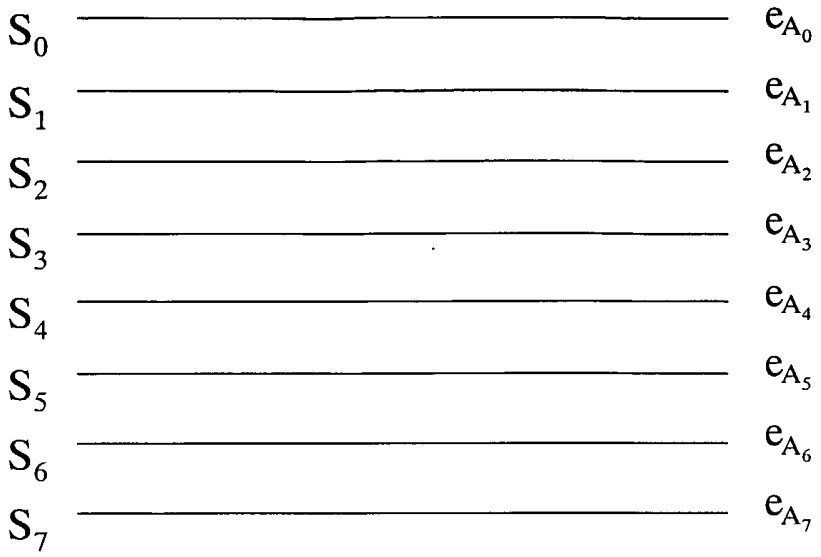
第8図


V=(v'0 v'1 ···v'89)
21(0)
21(1)
21(2)
21(3)
21(4)
21(5)
21(6)
21(7)
21(8)
21(9)
21(10)
21(11)
21(12)
21(13)
S0
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13

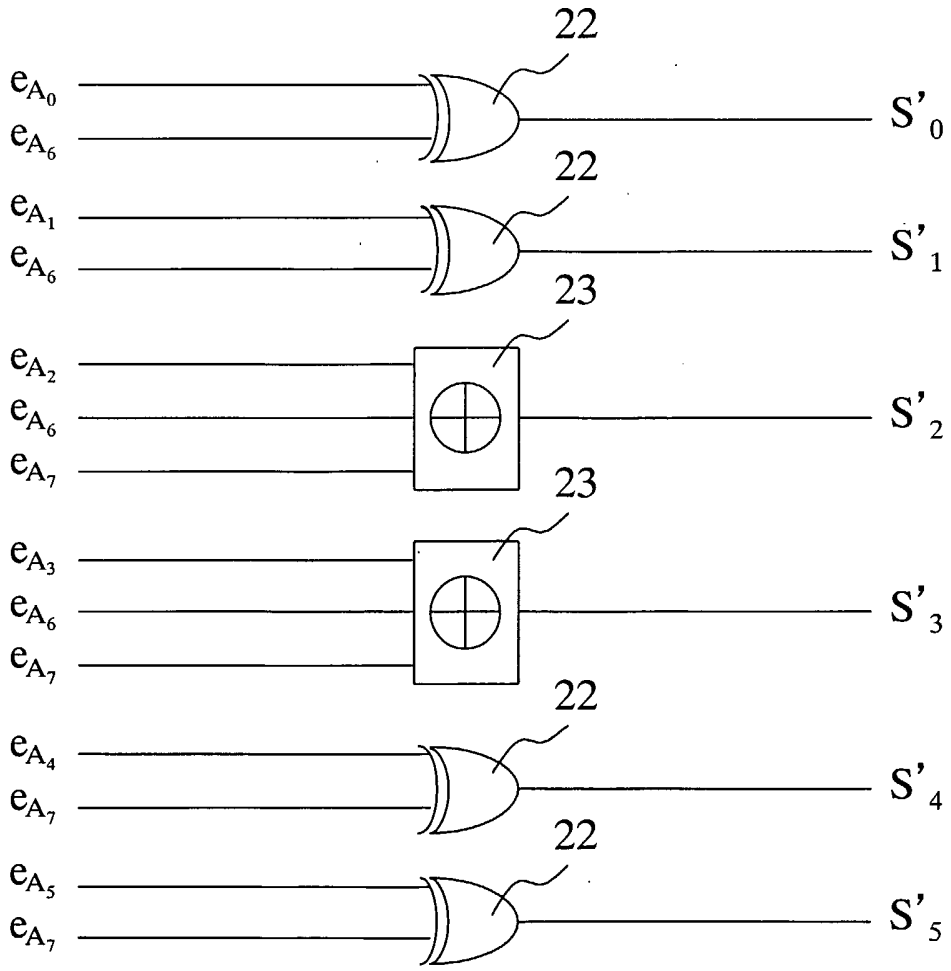
第9図


V=(v'0v'1…v'89)
v'0 v'6 v'12 v'15 v'18 v'22
v'23 v'24 v'29 v'30 v'31 v'35
v'36 v'38 v'41 v'42 v'47 v'48
v'52 v'54 v'55 v'58 v'60 v'62
v'64 v'66 v'69 v'72 v'75 v'76
v'77 v'79 v'81 v'82 v'83 v'84
v'86 v'87 v'88 v'89
v'1 v'6 v'12 v'13 v'18 v'19
v'24 v'25 v'29 v'31 v'35 v'37
v'38 v'41 v'43 v'47 v'49 v'52
v'53 v'54 v'58 v'59 v'60 v'61
v'64 v'65 v'66 v'67 v'69 v'71
v'72 v'73 v'75 v'78 v'79 v'81
v'85 v'86 v'87
v'2 v'6 v'7 v'8 v'13 v'14
v'15 v'19 v'20 v'22 v'25 v'26
v'31 v'32 v'36 v'38 v'39 v'42
v'44 v'46 v'48 v'50 v'53 v'56
v'59 v'60 v'61 v'63 v'65 v'66
v'67 v'68 v'70 v'71 v'72 v'73
v'74 v'76 v'78 v'80 v'82 v'87
v'88
v'3 v'6 v'7 v'9 v'14 v'20
v'21 v'26 v'27 v'32 v'33 v'37
v'39 v'43 v'45 v'46 v'49 v'51
v'55 v'57 v'60 v'61 v'62 v'66
v'67 v'68 v'69 v'72 v'73 v'74
v'75 v'77 v'79 v'80, v'81 v'83
v'86 v'87 v'89
v'4 v'7 v'10 v'14 v'15 v'16
v'21 v'22 v'23 v'27 v'28 v'30
v'33 v'34 v'39 v'40 v'44 v'46
v'47 v'50 v'52 v'54 v'56 v'58
v'61 v'64 v'67 v'68 v'69 v'71
v'73 v'74 v'75 v'76 v'78 v'79
v'80 v'81 v'82 v'84 v'86 v'88
v'5 v'7 v'11 v'14 v'15 v'17
v'22 v'28 v'29 v'34 v'35 v'40
v'41 v'45 v'47 v'51 v'53 v'54
v'57 v'59 v'63 v'65 v'68 v'69
v'70 v'74 v'75 v'76 v'77 v'80
v'81 v'82 v'83 v'85 v'87 v'88
v'89
21(14)
21(15)
21(16)
21(17)
21(18)
21(19)
s14
s15
s16
s17
s18
s19
v'0 v'6 v'11 v'14 v'15 v'16
v'21 v'22 v'23 v'26 v'27 v'32
v'36 v'38 v'39 v'41 v'43 v'47
v'48 v'51 v'52 v'53 v'55 v'56
v'57 v'58 v'59 v'61 v'63 v'64
v'66 v'71 v'75 v'76 v'77 v'78
v'79 v'80 v'81 v'82 v'85 v'86
v'1 v'6 v'11 v'12 v'14 v'16
v'17 v'21 v'23 v'26 v'28 v'30
v'33 v'36 v'37 v'38 v'41 v'42
v'43 v'44 v'46 v'47 v'48 v'49
v'51 v'54 v'55 v'56 v'60 v'61
v'62 v'65 v'66 v'67 v'70 v'72
v'75 v'79 v'80 v'83 v'85 v'88
v'2 v'6 v'7 v'12 v'13 v'17
v'18 v'23 v'27 v'29 v'30 v'32
v'34 v'37 v'42 v'43 v'44 v'45
v'48 v'49 v'50 v'52 v'54 v'57
v'61 v'62 v'63 v'66 v'67 v'68
v'71 v'72 v'73 v'76 v'79 v'81
v'84 v'86 v'87 v'89
v'3 v'6 v'7 v'8 v'13 v'14
v'15 v'18 v'19 v'24 v'28 v'30
v'31 v'33 v'35 v'39 v'40 v'43
v'44 v'45 v'47 v'48 v'49 v'50
v'51 v'53 v'55 v'56 v'58 v'63
v'67 v'68 v'69 v'70 v'71 v'72
v'73 v'74 v'77 v'78 v'82 v'85
v'86
v'4 v'7 v'9 v'14 v'19 v'20
v'22 v'24 v'25 v'29 v'31 v'34
v'36 v'38 v'41 v'44 v'45 v'46
v'49 v'50 v'51 v'52 v'54 v'55
v'56 v'57 v'59 v'62 v'63 v'64
v'68 v'69 v'70 v'73 v'74 v'75
v'78 v'80 v'83 v'87 v'88
v'5 v'7 v'10 v'14 v'15 v'20
v'21 v'25 v'26 v'31 v'35 v'37
v'38 v'40 v'42 v'45 v'50 v'51
v'52 v'53 v'56 v'57 v'58 v'60
v'62 v'65 v'69 v'70 v'71 v'74
v'75 v'76 v'79 v'80 v'81 v'84
v'87 v'89
21(20)
21(21)
21(22)
21(23)
21(24)
21(25)
s20
s21
s22
s23
s24
s25

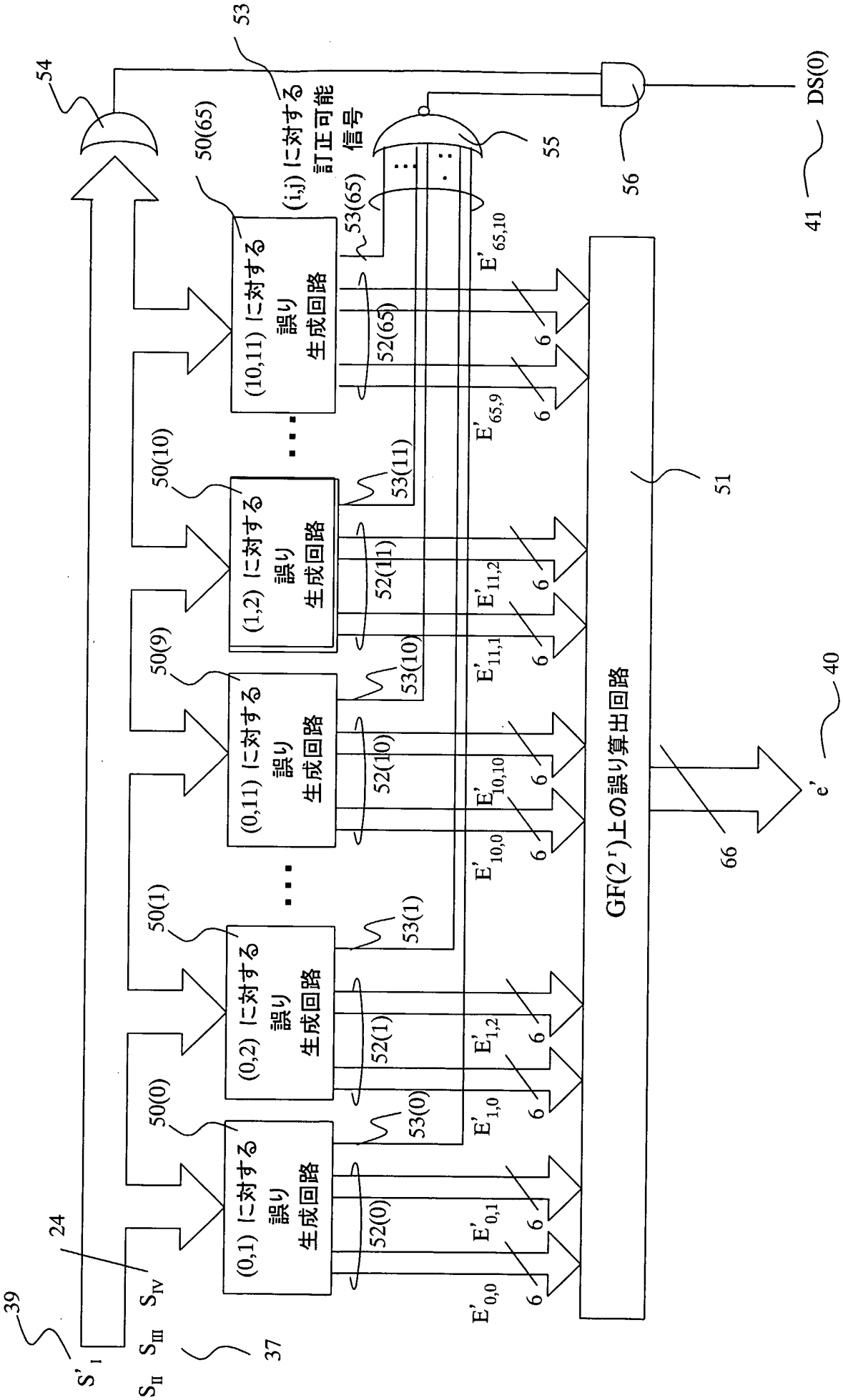
第10図


S0
S1
S2
S3
S4
S5
S6
S7
eA0
eA1
eA2
eA3
eA4
eA5
eA6
eA7

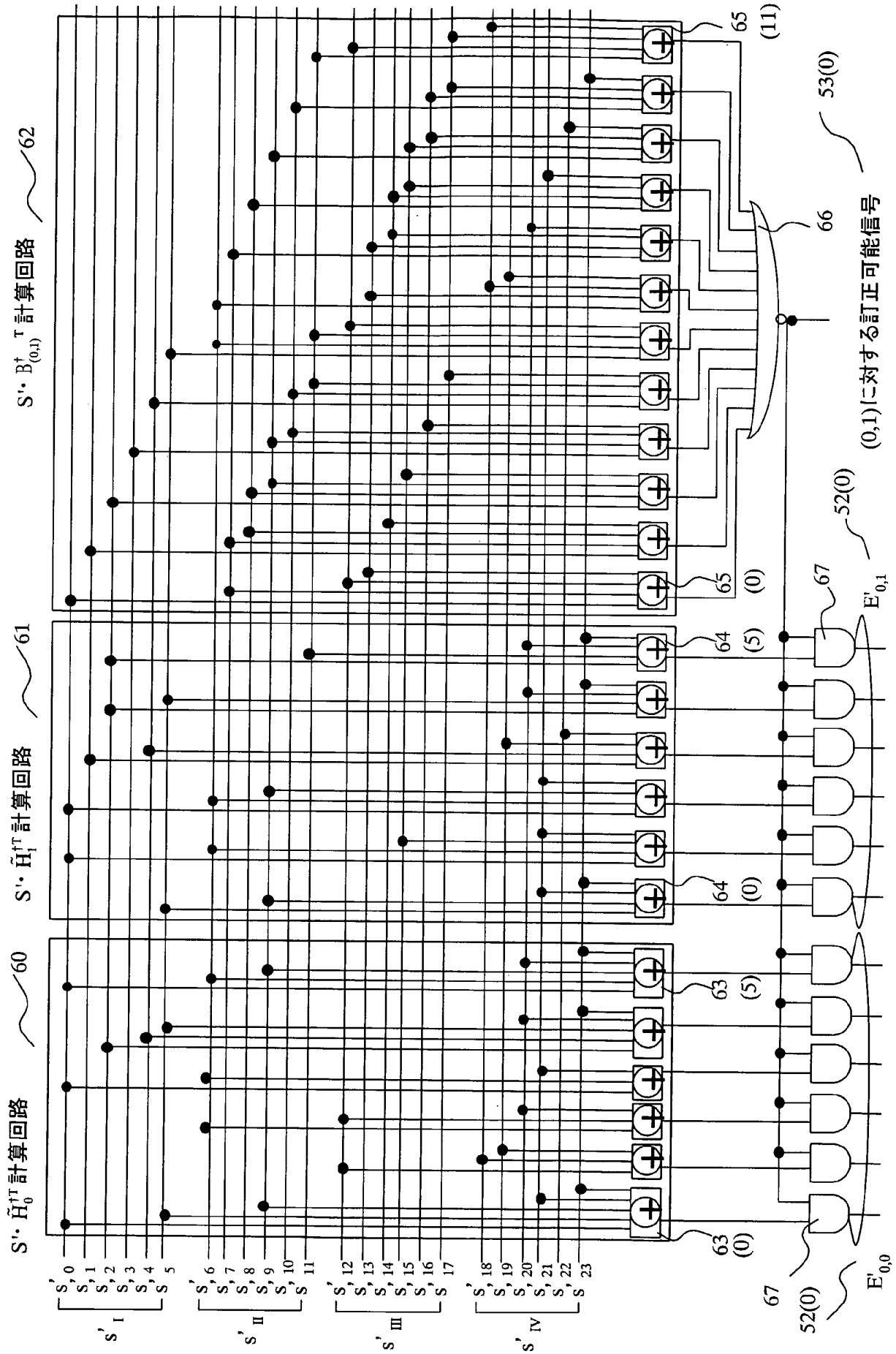
第11図


22
23
eA0
eA6
eA1
eA2
eA7
eA3
eA4
eA5
S'0
S'1
S'2
S'3
S'4
S'5

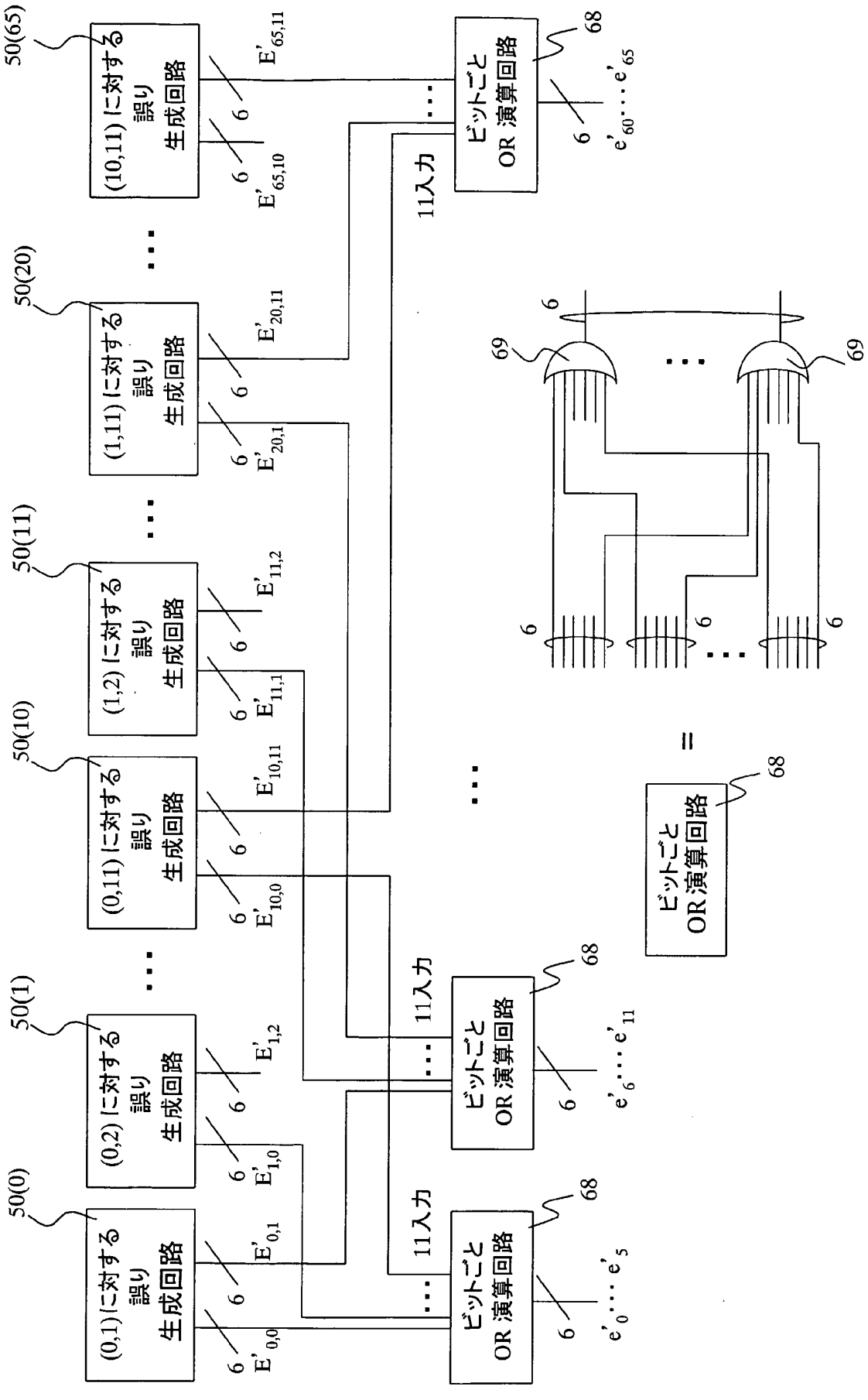
第12図


53
54
(i,j) に対する
訂正可能
信号
50(65)
(10,11) に対する
誤り
生成回路
53(65)
52(65)
E'65,10
E'65,9
50(10)
(1,2) に対する
誤り
生成回路
53(11)
52(11)
E'11,2
E'11,1
50(9)
(0,11) に対する
誤り
生成回路
53(10)
52(10)
E'10,10
E'10,0
50(1)
(0,2) に対する
誤り
生成回路
53(1)
52(1)
E'1,2
E'1,0
50(0)
(0,1) に対する
誤り
生成回路
53(0)
52(0)
E'0,1
E'0,0
55
56
DS(0)
41
51
GF(2r)上の誤り算出回路
40
e'
66
6
24
39
S'I
SII
SIII
SIV
37

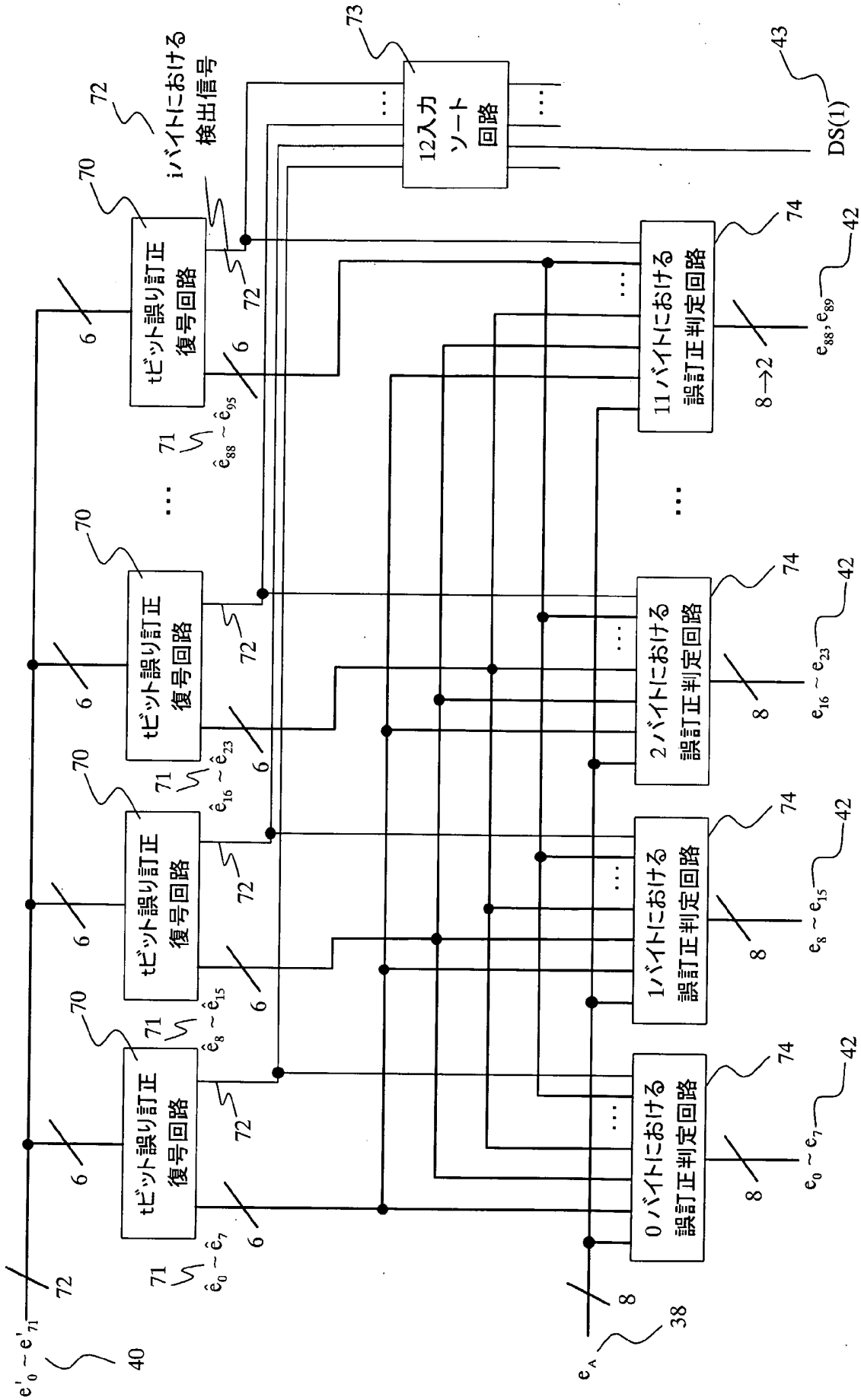
第13図


S'・H̃0^tT計算回路
60
S'・H̃1^tT計算回路
61
S'・B(0,1)^t T計算回路
62
S'0 S'1 S'2 S'3 S'4 S'5
S'6 S'7 S'8 S'9 S'10 S'11
S'12 S'13 S'14 S'15 S'16 S'17
S'18 S'19 S'20 S'21 S'22 S'23
S'I
S'II
S'III
S'IV
63 (0)
63 (5)
64 (0)
64 (5)
65 (0)
65 (11)
66
67
52(0)
53(0)
E'0,0
E'0,1
(0,1)に対する訂正可能信号

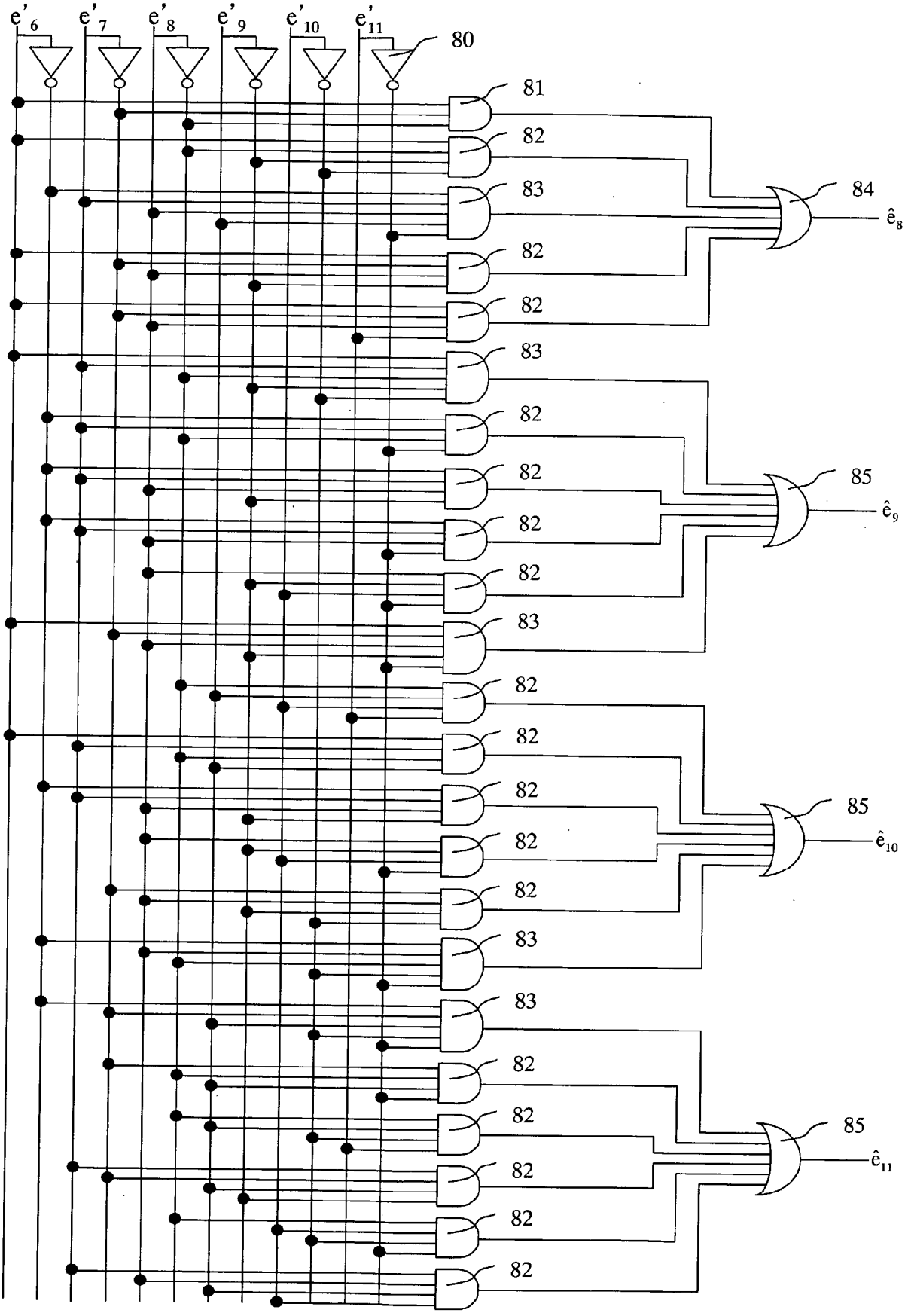
第14図


50(0)
(0,1)に対する誤り生成回路
50(1)
(0,2)に対する誤り生成回路
50(10)
(0,11)に対する誤り生成回路
50(11)
(1,2)に対する誤り生成回路
50(20)
(1,11)に対する誤り生成回路
50(65)
(10,11)に対する誤り生成回路
11入力
ビットごとOR演算回路
68
69
6
=

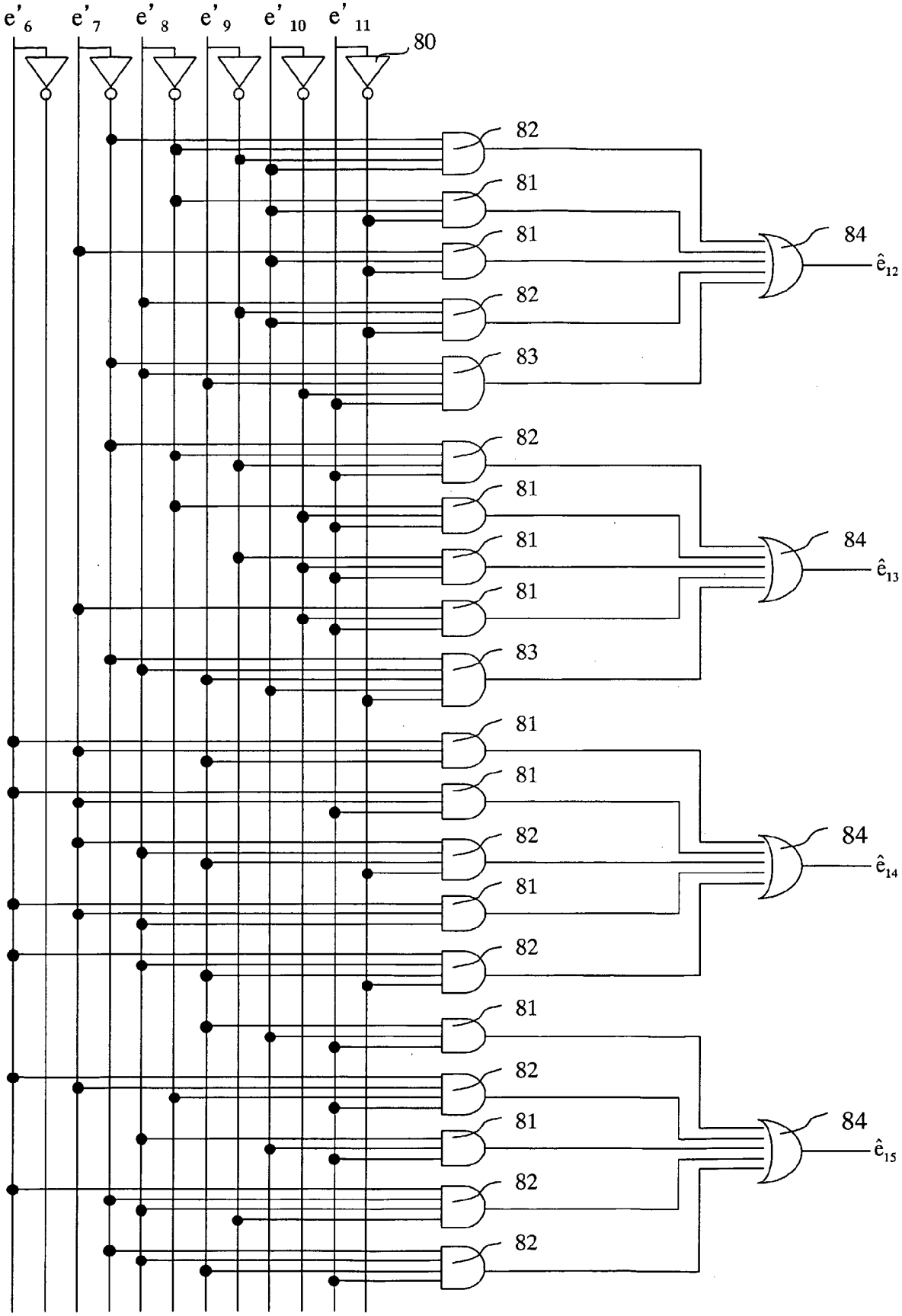
第15図


tビット誤り訂正復号回路
iバイトにおける検出信号
12入力ソート回路
DS(1)
11バイトにおける誤訂正判定回路
2バイトにおける誤訂正判定回路
1バイトにおける誤訂正判定回路
0バイトにおける誤訂正判定回路
8→2

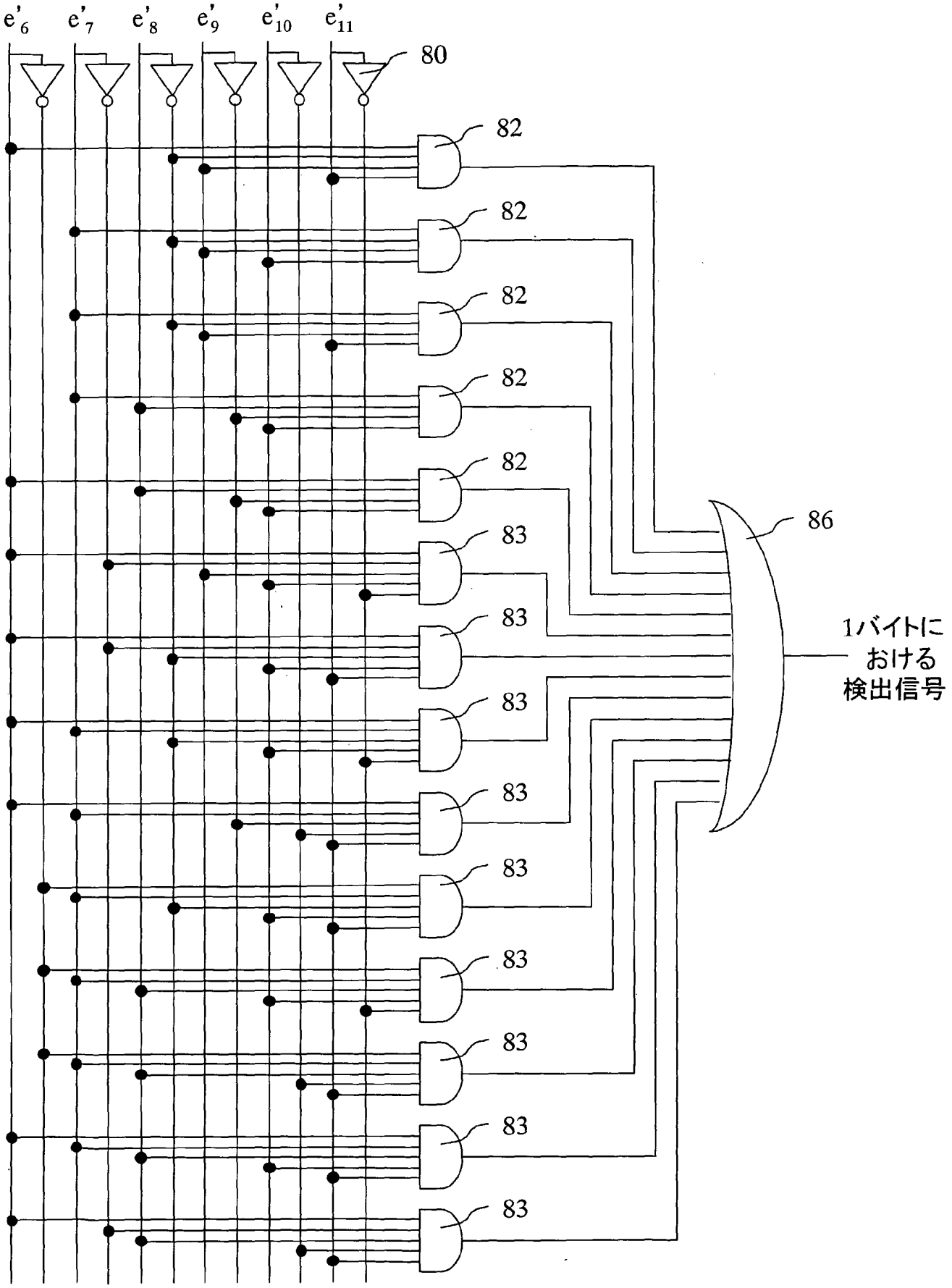
第16図



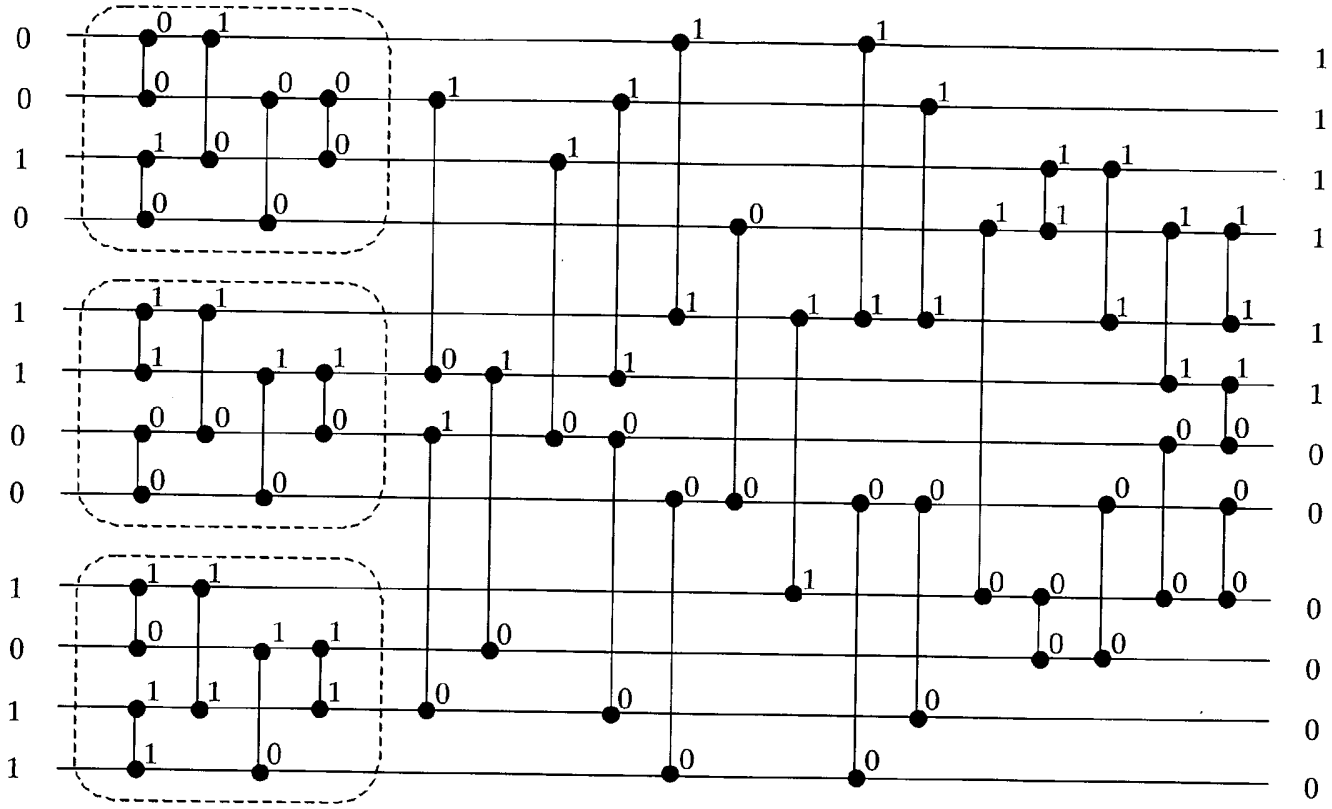
第17図

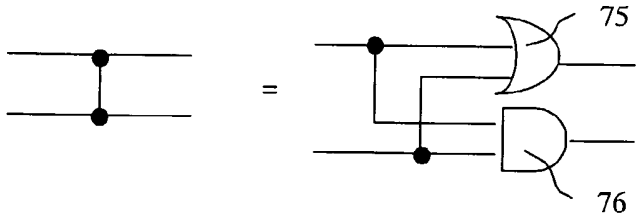

e'6
e'7
e'8
e'9
e'10
e'11
80
82
81
81
82
83
84
ê12
82
81
81
81
83
84
ê13
81
81
82
81
82
84
ê14
81
82
81
82
82
84
ê15

第18図

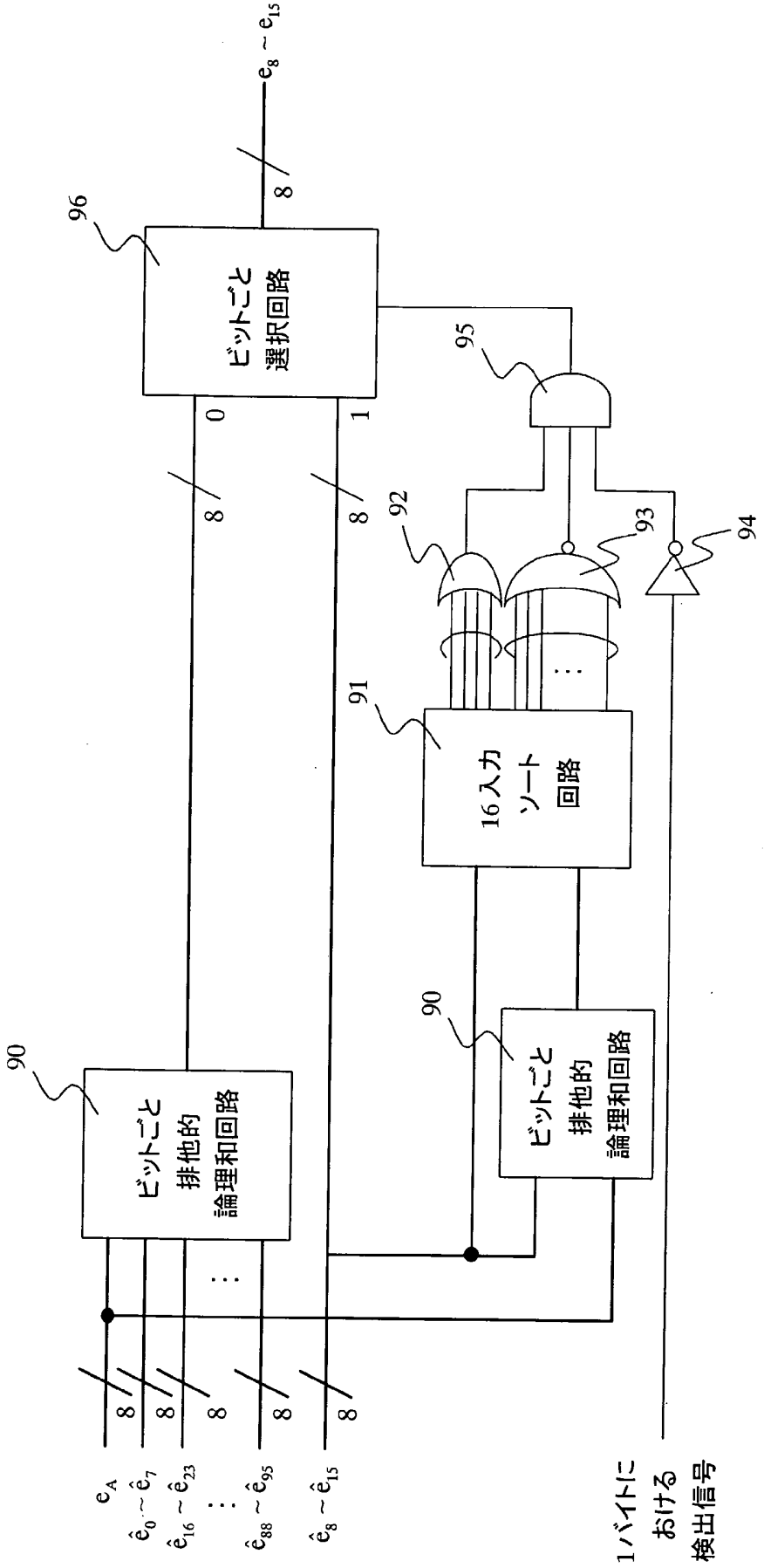

e'6
e'7
e'8
e'9
e'10
e'11
80
82
82
82
82
82
83
83
83
83
83
83
83
83
86
1バイトにおける検出信号

第19図

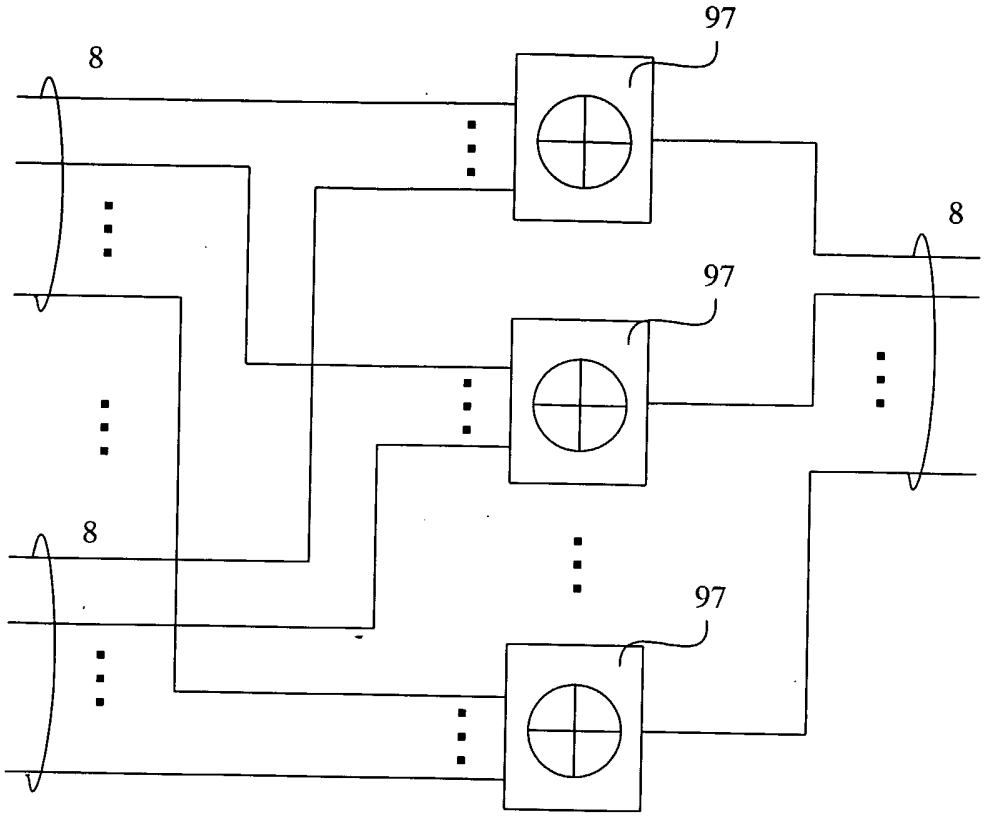




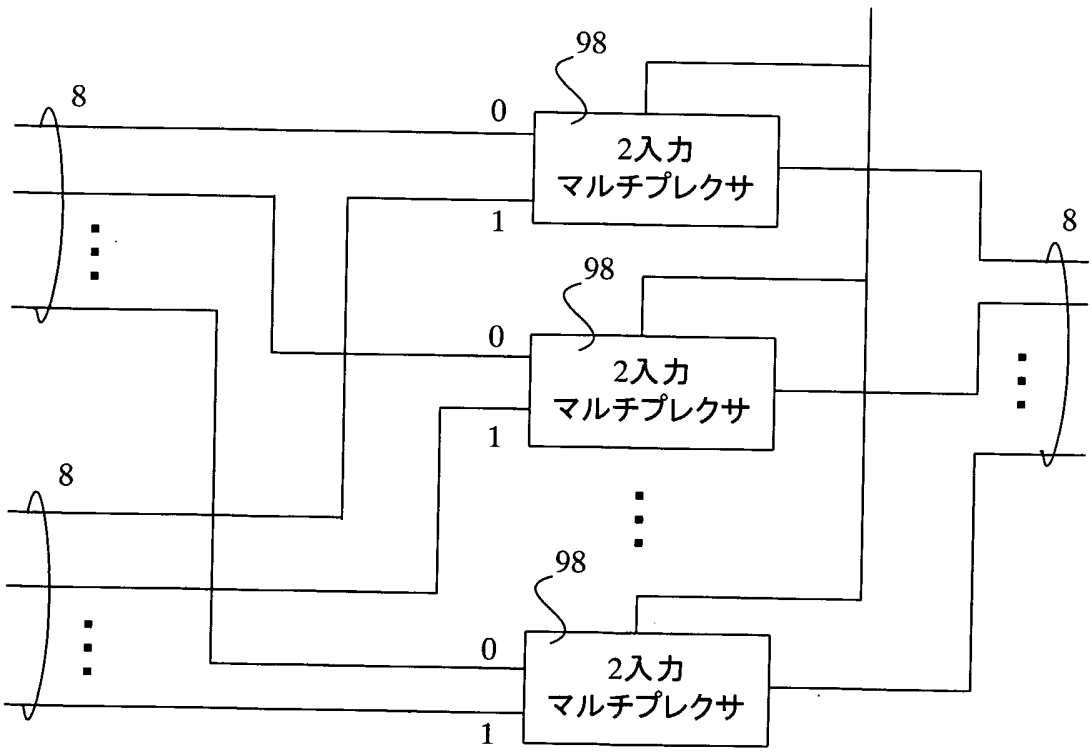
第20図


e_A
$\hat{e}_0 \sim \hat{e}_7$
$\hat{e}_{16} \sim \hat{e}_{23}$
$\hat{e}_{88} \sim \hat{e}_{95}$
$\hat{e}_8 \sim \hat{e}_{15}$
8
90
ビットごと
排他的
論理和回路
91
16入力
ソート
回路
92
93
94
95
96
ビットごと
選択回路
0
1
$e_8 \sim e_{15}$
1バイトに
おける
検出信号

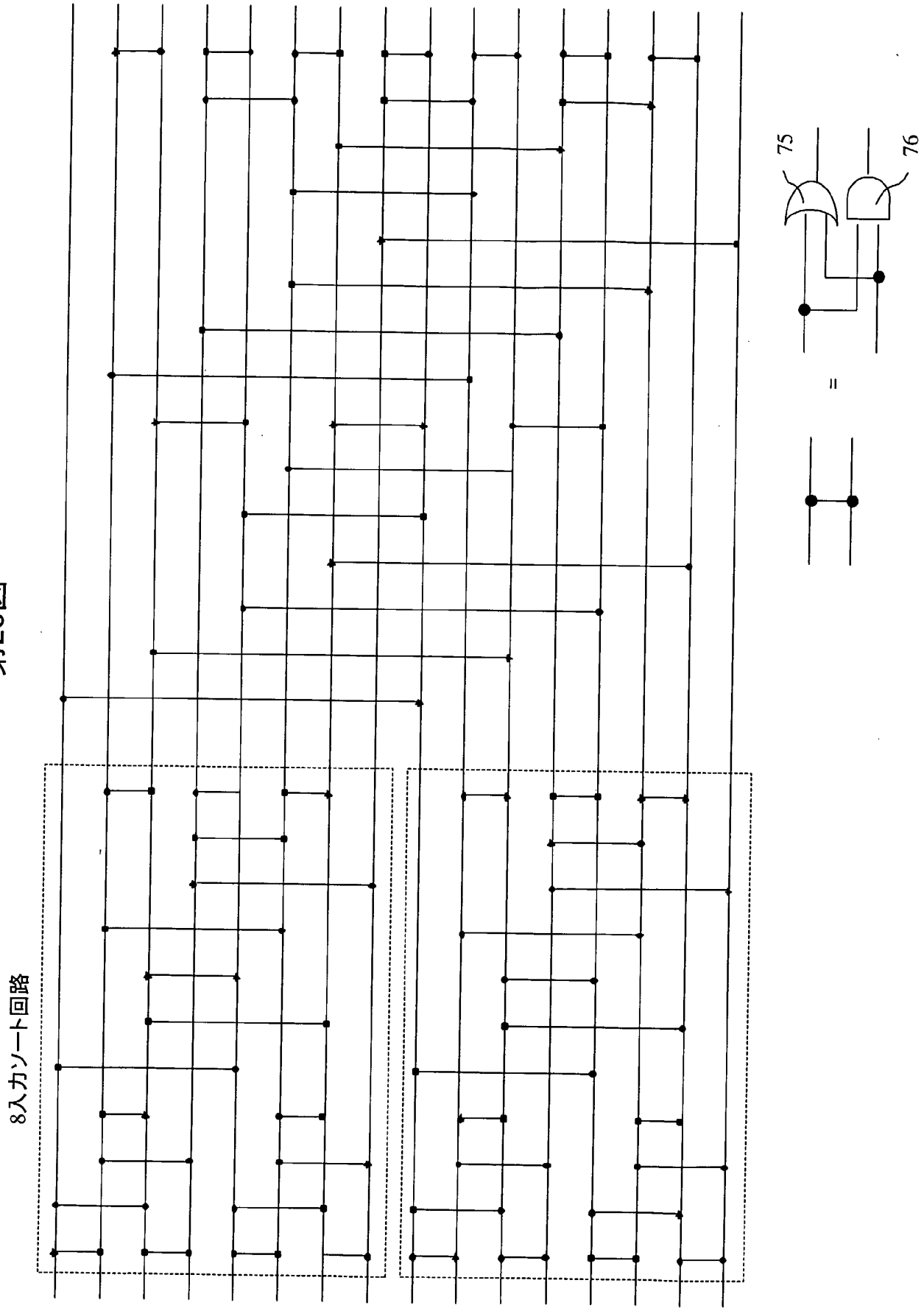
第21図


97
8
8
97
8
97

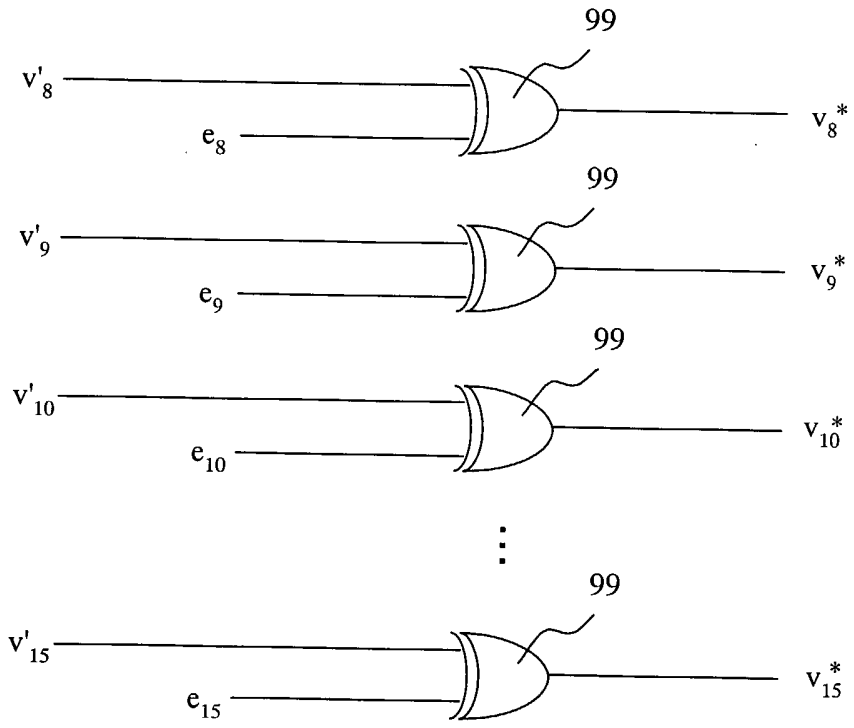
第22図


98
8
0
2入力
マルチプレクサ
1
8
98
0
2入力
マルチプレクサ
1
8
98
0
2入力
マルチプレクサ
1

第23図



第24図



INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2005/023545

A. CLASSIFICATION OF SUBJECT MATTER
G06F12/16(2006.01), ***G06F11/10***(2006.01), ***H03M13/15***(2006.01)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F12/16(2006.01), ***G06F11/10***(2006.01), ***H03M13/15***(2006.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1922-1996 Jitsuyo Shinan Toroku Koho 1996-2006
Kokai Jitsuyo Shinan Koho 1971-2006 Toroku Jitsuyo Shinan Koho 1994-2006

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2004-007217 A (Zaidan Hojin Rikogaku Shinkokai), 08 January, 2004 (08.01.04), Full text; all drawings (Family: none)	1-41
A	Toshihiko KASHIYAMA, Kazuyoshi SUZUKI, Eiji FUJIWARA, "Spotty Byte Ayamari Seigyo Fugo", FIT2004 Dai 3 Kai Forum on Information Technology (LC-006), 2004.09, pages 43 to 45	1-41
A	Ganesan Umanesan & Eiji FUJIWARA, A Class of Random Multiple Bits in a Byte Error Correcting (St/bEC) Codes for Semiconductor Memory Systems, Proceedings of the 2002 Pacific Rim International Symposium on Dependable Computing (PRDC'02), 2002.12, pages 247 to 254	1-41

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
15 March, 2006 (15.03.06)	28 March, 2006 (28.03.06)
Name and mailing address of the ISA/ Japanese Patent Office Facsimile No.	Authorized officer Telephone No.

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2005/023545

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,A	JP 2005-176244 A (Tokyo Institute of Technology), 30 June, 2005 (30.06.05), Full text; all drawings (Family: none)	1-41
T	Kazuyoshi SUZUKI, Mamoru SHIMIZU, Toshihiko KASHIYAMA, and Eiji FUJIWARA, A Class of Error Control Codes for M-Spotty Byte Errors Occurred in a Limited Number of Bytes, Proceedings. International Symposium on Information Theory 2005(ISIT 2005), 04 September, 2005 (04.09.05), pages 247 to 254	1-41

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２００５／０２３５４５

Ａ．　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int.Cl.　*G06F12/16(2006.01), G06F11/10(2006.01), H03M13/15(2006.01)*

Ｂ．　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Int.Cl.　*G06F12/16(2006.01), G06F11/10(2006.01), H03M13/15(2006.01)*

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２００６年
日本国実用新案登録公報	１９９６－２００６年
日本国登録実用新案公報	１９９４－２００６年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

Ｃ．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
A	JP 2004-007217 A (財団法人理工学振興会) 2004.01.08, 全文, 全図 (ファミリーなし)	1-41
A	樫山俊彦, 鈴木一克, 藤原英二, スポッティバイト誤り制御符号, FIT2004 第３回情報科学技術フォーラム (LC-006), 2004.09, pp. 43-45	1-41

☑ Ｃ欄の続きにも文献が列挙されている。　　☐ パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー

「Ａ」特に関連のある文献ではなく、一般的技術水準を示すもの

「Ｅ」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「Ｌ」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）

「Ｏ」口頭による開示、使用、展示等に言及する文献

「Ｐ」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献

「Ｔ」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「Ｘ」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Ｙ」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「＆」同一パテントファミリー文献

国際調査を完了した日	国際調査報告の発送日
１５．０３．２００６	２８．０３．２００６

国際調査機関の名称及びあて先
日本国特許庁（ＩＳＡ／ＪＰ）
郵便番号１００－８９１５
東京都千代田区霞が関三丁目４番３号

特許庁審査官（権限のある職員）　５Ｋ　８６２５
藤井　浩
電話番号　０３－３５８１－１１０１　内線　３５５６

Ｃ（続き）．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
A	Ganesan Umanesan & Eiji Fujiwara, A Class of Random Multiple Bits in a Byte Error Correcting (St/bEC) Codes for Semiconductor Memory Systems, Proceedings of the 2002 Pacific Rim International Symposium on Dependable Computing (PRDC'02), 2002.12, pp.247-254	1-41
P,A	JP 2005-176244 A (国立大学法人東京工業大学) 2005.06.30, 全文,全図　(ファミリーなし)	1-41
T	Kazuyoshi Suzuki, Mamoru Shimizu, Toshihiko Kashiyama, and Eiji Fujiwara, A Class of Error Control Codes for M-Spotty Byte Errors Occurred in a Limited Number of Bytes, Proceedings. International Symposium on Information Theory 2005(ISIT 2005), 2005.09.04, pp.247-254	1-41