

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4190658号
(P4190658)

(45) 発行日 平成20年12月3日(2008.12.3)

(24) 登録日 平成20年9月26日(2008.9.26)

(51) Int.Cl.		F I
G06F 3/06	(2006.01)	G06F 3/06 540
G06F 11/10	(2006.01)	G06F 11/10 330Q
G11B 20/18	(2006.01)	G11B 20/18 542Z
H03M 13/00	(2006.01)	G11B 20/18 570Z
		H03M 13/00

請求項の数 14 (全 20 頁)

(21) 出願番号 特願平11-135424
 (22) 出願日 平成11年5月17日(1999.5.17)
 (65) 公開番号 特開2000-47833(P2000-47833A)
 (43) 公開日 平成12年2月18日(2000.2.18)
 審査請求日 平成18年3月31日(2006.3.31)
 (31) 優先権主張番号 09/079757
 (32) 優先日 平成10年5月15日(1998.5.15)
 (33) 優先権主張国 米国(US)

(73) 特許権者 591179352
 クォンタム・コーポレーション
 QUANTUM CORPORATION
 アメリカ合衆国、95110 カリフォル
 ニア州、サン・ノゼ、テクノロジー・ドラ
 イブ、1650、スイート・800
 (74) 代理人 100064746
 弁理士 深見 久郎
 (74) 代理人 100085132
 弁理士 森田 俊雄
 (74) 代理人 100091409
 弁理士 伊藤 英彦
 (74) 代理人 100096781
 弁理士 堀井 豊

最終頁に続く

(54) 【発明の名称】 符号化のためのシステムおよび復号器

(57) 【特許請求の範囲】

【請求項1】

k のデータディスクからの m ビットデータシンボルを符号化して r の冗長ディスクのための m ビット冗長シンボルを生成することを、k のデータディスク各々からの対応するシンボルを符号化することにより行なう、符号化のためのシステムであって、前記システムは、

(A) k のデータディスクの対応するセクタからデータを受取るための手段と、

(B) (m + 1) ビット冗長シンボルを、以下の式すなわち

【数1】

$$c_y = \alpha^{y*(k-1)} i_{k-1} + \alpha^{y*(k-2)} i_{k-2} + \dots + \alpha^{y*(1)} i_1 + i_0$$

として生成するための符号化手段とを含み、式中 y = 0, 1, ... r - 1 であり、i_j は j 番目のデータディスクからのデータであり、 α は m + 1 を素数とした GF(2^m) の要素の (m + 1) ビット表現であり、前記手段は各 c_y について、

(i) 乗算を y ビット循環シフトとして行なうためのシフトレジスタと、

(ii) シフトレジスタの内容に次のデータディスクからの対応するデータを加算するための加算器とを備え、前記システムはさらに、

(C) (m + 1) ビット冗長シンボルを m ビット冗長シンボルに変換するための変換手

段と、

(D) r の冗長ディスクの対応するセクタに m ビット冗長シンボルをストアするための手段とを含む、符号化のためのシステム。

【請求項 2】

符号化手段は m ビットセクタを符号化する、請求項 1 に記載のシステム。

【請求項 3】

シフトレジスタは、1つのシフトで y ビット循環シフトの結果を生成するフィードバック経路を含む $(m+1)$ ビットシフトレジスタである、請求項 1 に記載のシステム。

【請求項 4】

$(k+r)$ トラックテープの k のトラックからの m ビットデータシンボルを符号化して r の冗長トラックのための m ビット冗長シンボルを生成することを、 k のデータトラック各々からの対応するシンボルを符号化することにより行なう、符号化のためのシステムであって、

(A) k のデータトラックの対応するブロックからデータを受取るための手段と、

(B) $(m+1)$ ビット冗長シンボルを、以下の式すなわち

【数 2】

$$c_y = \alpha^{y*(k-1)} i_{k-1} + \alpha^{y*(k-2)} i_{k-2} + \dots + \alpha^{y*(1)} i_1 + i_0$$

10

20

として生成するための符号化手段とを含み、式中 $y = 0, 1, \dots, r-1$ であり、 i_j は j 番目のデータトラックからのデータであり、 α は $m+1$ を素数とした $GF(2^m)$ の要素の $(m+1)$ ビット表現であり、前記手段は各 c_r について、

(i) 乗算を y ビット循環シフトとして行なうためのシフトレジスタと、

(ii) シフトレジスタの内容に次のデータトラックからの対応するデータを加算するための加算器とを備え、前記システムはさらに、

(C) $(m+1)$ ビット冗長シンボルを m ビット冗長シンボルに変換するための変換手段と、

(D) m ビット冗長シンボルを r の冗長トラックの対応するブロックにストアするための手段とを含む、符号化のためのシステム。

30

【請求項 5】

符号化手段は m ビットブロックを符号化する、請求項 4 に記載のシステム。

【請求項 6】

シフトレジスタは、1つのシフトにおいて y ビット循環シフトの結果を生成するフィードバック経路を含む $(m+1)$ ビットシフトレジスタである、請求項 4 に記載のシステム。

【請求項 7】

r の冗長ディスクの対応するセクタにストアされた冗長シンボルを用いて k のデータディスクの対応するセクタにおける誤りを訂正するための復号器であって、

(A) k のデータディスクからデータを受取るための手段と、

(B) $(m+1)$ ビット冗長シンボルを、以下の式すなわち

【数 3】

$$c_y = \alpha^{y*(k-1)} i_{k-1} + \alpha^{y*(k-2)} i_{k-2} + \dots + \alpha^{y*(1)} i_1 + i_0$$

40

として生成するための復号化手段とを含み、式中 $y = 0, 1, \dots, r-1$ であり、 i_j は j 番目のデータディスクからの誤りのないデータであり、または j 番目のディスクのデータに誤りがあればすべて 0 であり、 α は $m+1$ を素数とした $GF(2^m)$ の要素の $(m+1)$ ビット表現であり、前記手段は各 c_r について、

(i) 乗算を y ビット循環シフトとして行なうためのシフトレジスタと、

50

(ii) シフトレジスタの内容に次のデータディスクからのデータを加算するための加算器とを備え、前記復号器はさらに、

(C) z を誤りのあるデータを含むディスクの数として、シンドローム $S_j, S_{j+1}, \dots, S_{j+z-1}$ を生成するためのシンドローム発生器を含み、前記シンドローム発生器は、
【数 4】

$$S_j = (\alpha^j)^{e_0} * i'_{e_0} + (\alpha^j)^{e_1} * i'_{e_1} + \dots + (\alpha^j)^{e_z} * i'_{e_z}$$

を生成し、式中 $j = 0, 1, \dots, r - 1$ であり、 i'_{e_z} は z 番目の誤りのあるディスクと関連する誤りパターンであり、 S_j は j 番目の冗長ディスクと関連するシンドロームであり、前記復号器はさらに、

(D) シンドロームと関連する誤り値を決定するための手段を含み、前記手段は、重み 1 の $(m + 1)$ ビットシンボル $(i^j)^p$ の乗法逆元を $(i^j)^{-p}$ として決定するための手段を含み、前記復号器はさらに、

(E) $(m + 1)$ ビット誤り値を m ビット誤り値に変換するための変換手段と、

(F) m ビット誤り値を用いて誤りのある m ビットシンボルを訂正するための手段とを含む、復号器。

【請求項 8】

誤り値を決定するための手段はさらに、 $A = a^h + b^v$ 、重み 2 の $(m + 1)$ ビットシンボルの乗法逆元を、 $A^{-1} = (a^{m+1-h}) * b^{-v-h}$ として決定するための手段を含み、式中 a は $a^{m-1} + a^{m-3} + \dots + a^3 + a^1$ である、請求項 7 に記載の復号器。

【請求項 9】

誤り値を決定するための手段はまた、 $A = a^a + b^b + c^c$ 、重み 3 の $(m + 1)$ ビットシンボルの乗法逆元を $(a^c * D)^{b-c}$ として決定するための手段を含み、式中 D は $a^a + b^b + c^c + 1$ の逆数であり、 $a^a = (a - c) / (b - c)$ である、請求項 8 に記載の復号器。

【請求項 10】

誤り値を決定するための手段はさらに、 $\dots 1 \dots 0 0 1 1$ の形式の重み 3 の $(m + 1)$ ビットシンボルの乗法逆元の $(m - 1)$ 要素テーブルを含み、前記手段はテーブルを投入してシンボル D を決定する、請求項 9 に記載の復号器。

【請求項 11】

r の冗長トラックの対応するブロックにストアされた冗長シンボルを用いて複数トラックテープシステムの k のデータトラックの対応するブロックにおける誤りを訂正するための復号器であって、

(A) k のデータトラックからデータを受取るための手段と、

(B) $(m + 1)$ ビット冗長シンボルを、以下の式すなわち

【数 5】

$$c_y = \alpha^{y*(k-1)} i_{k-1} + \alpha^{y*(k-2)} i_{k-2} + \dots + \alpha^{y*(1)} i_1 + i_0$$

として生成するための復号化手段とを含み、式中、 $y = 0, 1, \dots, r - 1$ であり、 i_j は j 番目のデータトラックからの誤りのないデータであり、または j 番目のトラックのデータに誤りがあればすべて 0 であり、 α は $m + 1$ を素数とする $GF(2^m)$ の要素の $(m + 1)$ ビット表現であり、前記手段は各 c_y について、

(i) 乗算を r ビット循環シフトとして行なうためのシフトレジスタと、

(ii) シフトレジスタの内容に次のデータトラックからのデータを加算するための加算器とを備え、前記復号器はさらに、

(C) シンドローム $S_j, S_{j+1}, \dots, S_{j+z-1}$ を生成するためのシンドローム発生器を含み、式中 z は誤りのあるデータを含むトラックの数であり、シンドローム発生器は、

10

20

30

40

50

【数 6】

$$S_j = (\alpha^j)^{e_0} * i'_{e_0} + (\alpha^j)^{e_1} * i'_{e_1} + \dots + (\alpha^j)^{e_z} * i'_{e_z}$$

を生成し、式中 $j = 0, 1, \dots, r - 1$ であり、 i'_{e_z} は z 番目の誤りのあるトラックに関連する誤りパターンであり、 S_j は j 番目の冗長トラックに関連するシンドロームであり、前記復号器はさらに、

(D) シンドロームに関連する誤り値を決定するための手段を含み、前記手段は重み 1 の $(m + 1)$ ビットシンボル $(i'_j)^p$ の乗法逆元を $(i'_j)^{m+1-jp}$ として決定するための手段を含み、前記復号器はさらに、

10

(E) $(m + 1)$ ビットエラー値を m ビットエラー値に変換するための変換手段と、
(F) m ビット誤り値を用いて誤りのある m ビットシンボルを訂正するための手段とを含む、復号器。

【請求項 1 2】

誤り値を決定するための手段はさらに、 $A = h + v$ 、重み 2 の $(m + 1)$ ビットシンボルの乗法逆元を、 $A^{-1} = (h^{-1} + v^{-1}) * v^{-h}$ として決定するための手段を含み、式中 h は $m-1 + m-3 + \dots + 3 + 1$ である、請求項 1 1 に記載の復号器。

【請求項 1 3】

誤り値を決定するための手段はまた、 $A = a + b + c$ 、重み 3 の $(m + 1)$ ビットシンボルの乗法逆元を $(a + c) * D^{b-c}$ として決定するための手段を含み、式中 D は $a^{-1} + c^{-1} + 1$ の逆数であり、 $a^{-1} = (a - c) / (b - c)$ である、請求項 1 2 に記載の復号器。

20

【請求項 1 4】

誤り値を決定するための手段はさらに、 $\dots 1 \dots 0 0 1 1$ の形式の重み 3 の $(m + 1)$ ビットシンボルの乗法逆元の $(m - 1)$ 要素テーブルを含み、前記手段はテーブルを投入してシンボル D を決定する、請求項 9 に記載の復号器。

【発明の詳細な説明】

【0001】

【発明の背景】

複数の独立したディスクを備え、冗長データをストアするためにこれらのディスクのうち 2 つ以上を使用するシステムは一般に、RAID - 6 システムと呼ばれており、RAID は、「廉価または独立ディスクの冗長アレイ (redundant array of inexpensive or independent disks)」の頭字語である。「 k 」のデータディスクおよび「 r 」の冗長ディスクを備えるシステムにおいては、 r の冗長ディスクに記録されたデータは、 k のデータディスクに記録されたデータを操作することにより生成される。典型的には、 r の冗長ディスクのデータは、 k のデータディスクのデータの誤り訂正符号化の結果である。

30

【0002】

k のデータディスクのデータはデータ符号語の形式を取り、符号語各々がデータシンボルおよび関連する誤り訂正符号 (ECC) シンボルを含む。ECC シンボルは、ECC に従い従来の態様でデータシンボルを符号化することによって生成される。 r の冗長ディスクのセクタは、冗長符号語を含む。この符号語におけるシンボルは、 k のデータディスク各々からの対応するシンボルを符号化することによって生成される。この符号化は、 k の対応するシンボルの「カラム」の符号化と考えることができる。本明細書において混乱を避けるために、 k のシンボルのカラムを符号化するのに使用される ECC は、「カラム ECC」と呼ぶことにする。典型的には、カラム ECC は、 k のデータディスクに対するデータ符号語を生成するために使用される ECC と同じではない。

40

【0003】

r の冗長ディスクにストアされたシンボルは、含まれる ECC シンボルを用いて訂正できないデータ符号語における誤りを訂正するために使用される。システムは、誤りを訂正するために、カラム ECC に従い、誤りのあるデータシンボルを含むカラムを復号化する。

50

【 0 0 0 4 】

システムは、冗長ディスクのためのデータを生成するために、 k の対応するセクタにストアされたデータを m ビットシンボルに分割する。このシステムは次に、カラムECCに従い k のセクタ各々からの対応する m ビットシンボルを符号化して、 r の対応する m ビット冗長シンボル生成する。これらの r のシンボルはそれぞれ、 r の冗長ディスク各々の対応するセクタにおいて記録される。

【 0 0 0 5 】

システム的设计者は、カラムの符号化および復号化に必要な時間を最短にするために、比較的大きなシンボルサイズ、すなわち「 m 」についての値を選択しようとするものである。こうすると、1セクタ当りに必要なカラム符号化/復号化動作数が減少する。符号化および復号化動作には、複数のガロア体乗算が必要である。従来のシステムでは、 m のサイズが増大するほど、ガロア体乗算の複雑度も大幅に増大する。さらに、各復号化動作は、ガロア体 $GF(2^m)$ の要素の乗法逆元の決定を含む。 m が大きい場合、乗法逆元を発見する演算には時間がかかり、かつ/または、大きな、 $2^m - 1$ 要素の、ルックアップテーブルを必要とするかもしれない。したがって、 m の値は、システムの複雑さと符号化/復号化に要する全時間との歩み寄りにおいて選択される。

【 0 0 0 6 】

複数トラックテープシステムのトラックを、RAID-6システムにおいて複数のディスクを使用するのと同じやり方で使用してもよい。したがって、データ符号語は k のトラックにストアされ、関連の冗長シンボルは残りの r のトラックにストアされる。 $k + r$ のディスクに対して使用されるのと同じ符号化/復号化技術を、 $k + r$ のトラックに対して用いる。このようにして、カラム符号化に際し m ビットシンボルのサイズを選択する上で、同じ歩み寄りを行なう。

【 0 0 0 7 】

【 発明の概要 】

本発明は、 m について選択された値の組のうち1つを使用し、ガロア体乗算を循環シフトおよび排他的OR演算の組合せとして行なう、RAID-6または複数トラックテープシステムのための符号化/復号化システムである。このシステムはまた、以下でより詳細に述べるように、ある乗法逆演算を、本質的には体の種々の要素を2の選択された累乗に累乗することによって決定する。このシステムを用いると、 m の値を、システムが実行する乗算または逆演算の複雑性を増すことなく、任意に大きなものとして選択することができる。

【 0 0 0 8 】

このシステムは、 $(m + 1)$ ビットシンボルを用い、乗算および逆演算を行なう。 $m + 1$ は素数である。 m の値は、体 $GF(2^m)$ を以下の既約多項式によって生じさせることができるように選択される。

【 0 0 0 9 】

$$g(x) = x^m + x^{m-1} + \dots + x^2 + x + 1$$

$GF(2^m)$ の2つの要素 $a(x)$ および $b(x)$ の乗算を、循環シフトされた要素 $b(x)$ の $(m + 1)$ ビット版の排他的ORを取ることによって行なうことができる。さらに、以下で述べるように、重み1、2または3の $GF(2^m)$ の要素 $c(x)$ の乗法逆元を、 $(m + 1)$ ビット表現を用いて、本質的には体の特定の要素を2の関連の累乗に累乗することによって容易に決定することができる。このように、既知の先行技術のシステムよりも、復号化動作の複雑性は低く、かつ/または要する時間も短い。

【 0 0 1 0 】

このシステムを用いると、 m の値を、たとえばディスクのセクタまたはテープのブロックの大きさになるように選択できる。このようにして、カラム誤り訂正符号化/復号化を、1セクタまたはブロック当たり1回行なうことができる。

【 0 0 1 1 】

本発明についての以下の説明では、添付の図面を参照する。

【 0 0 1 2 】

【実施例の詳細な説明】

本明細書では、 k のデータディスクおよび r の冗長ディスクを備えた複数ディスクシステムと関連付けて、本発明について説明を行なう。本発明を、 k のデータトラックおよび r の冗長トラックを備える複数トラックテープシステムについて使用することもできる。

【 0 0 1 3 】

従来の RAID - 6 システムでは、 k のセクタの各々における i 番目の m ビットシンボル、すなわち i 番目のカラムと関連する r の冗長シンボルは、典型的には以下のようにして生成される。

【 0 0 1 4 】

【数 7】

$$C_{0i} = i_0 + i_1 + i_2 + \dots + i_{k-1}$$

$$C_{1i} = (\alpha^{1*0}) * i_0 + (\alpha^{1*1}) * i_1 + (\alpha^{1*2}) * i_2 + \dots + (\alpha^{1*(k-1)}) * i_{k-1}$$

$$C_{2i} = (\alpha^{2*0}) * i_0 + (\alpha^{2*1}) * i_1 + (\alpha^{2*2}) * i_2 + \dots + (\alpha^{2*(k-1)}) * i_{k-1}$$

...

$$C_{(r-1)i} = (\alpha^{(r-1)*0}) * i_0 + (\alpha^{(r-1)*1}) * i_1 + (\alpha^{(r-1)*2}) * i_2 + \dots + (\alpha^{(r-1)*(k-1)}) * i_{k-1}$$

10

20

【 0 0 1 5 】

式中、 $C_{(r-1)i}$ は、 r 番目の冗長ディスクの i 番目のカラムに対する冗長シンボルであり、 i_j は、 j 番目のディスクの i 番目のカラムにストアされたシンボルであり、「+」および「*」は、ガロア体加算および乗算を表わし、「 α 」は $GF(2^m)$ の要素である。このように、各カラムに対し冗長シンボルを生成するために行なわなければならない、複数の、時間の掛かるガロア体乗算がある。

【 0 0 1 6 】

本発明のシステムは、選択された数の体 $GF(2^m)$ のうち 1 つを用い、ガロア体乗算を簡素化している。選択された m に対し、 $m+1$ を素数とすると、ガロア体 $GF(2^m)$ を、既約多項式、

$$g(x) = x^m + x^{m-1} + \dots + x^2 + x + 1$$

によって生じさせることができ、要素 2 は、体 $GF(m+1)$ の原始元である。このような体は、本明細書に引用により援用する、J. K. ウルフ (J.K. Wolf) による、Discrete Mathematics 106/107(1992) 497-502 頁において発表された、「 m のある値に対し $GF(2^m)$ において乗算するための効率的な回路 (Efficient Circuits For Multiplying In $GF(2^m)$ For Certain Values of m)」と題された文献において論じられている。以下で述べるように、このシステムはこういった体を利用し、 $GF(2^m)$ の原始元が既知であることを必要としない。このことは、より大きな体を用いるときには重要となる。なぜなら、原始元を決定することは困難であり得るからである。 $m < 32987$ の場合の m の値は、図 1 から 3 の表に列記されている。

【 0 0 1 7 】

同時係属中の、「 $GF(2^{w+i})$ のシンボルの $(w+i+1)$ ビット表現を用いた修正リードソロモン誤り訂正システム (MODIFIED REED-SOLOMON ERROR CORRECTION SYSTEM USING $(w+i+1)$ -bit REPRESENTATIONS OF SYMBOLS OF $GF(2^{w+i})$)」と題された米国特許出願第 08/786,894 号において述べられているように、 $GF(2^m)$ の要素は、補数である 2 つの $(m+1)$ ビットシンボル $a(x)$ および $b(x)$ によって表現することができる。すなわち、 $a(x) + b(x) = 0$ である。たとえば、 $GF(2^4)$ の要素を 2 つの

30

40

50

5 ビットシンボルのいずれかによって表現することができる。

【 0 0 1 8 】

【表 1】

要素	4 ビット表現	5 ビット表現
0	0000	00000 または 11111
β^0	0001	00001 または 11110
β^1	0011	00011 または 11100
β^2	0101	00101 または 11010
β^3	1111	01111 または 10000
...
β^{14}	1010	01010 または 10101

10

【 0 0 1 9 】

m ビットシンボルの (m + 1) ビット表現は、先行ゼロを追加するかまたは先行 1 を追加し、残りの m ビットを補足することによって生成される。明らかに、一方の (m + 1) ビット表現は、他方の (m + 1) ビット表現よりも重み、すなわち 1 の数が少ない。

20

【 0 0 2 0 】

2 つの (m + 1) ビットシンボルの乗算は、一方のシンボルの循環シフトされたコピーと、他方のシンボルの係数またはビットとの排他的 OR を取ることによって行なうことができる。例として GF (2⁴) の要素を用いると、2 つの要素 a (x) および b (x) の乗算すなわち a (x) * b (x) は、a (x) = a₄ a₃ a₂ a₁ a₀、b (x) = b₄ b₃ b₂ b₁ b₀ の場合以下のようにして行なわれる。

【 0 0 2 1 】

【数 8】

$$\begin{aligned}
 & b_0 * (a_4 a_3 a_2 a_1 a_0) \\
 & + b_1 * (a_3 a_2 a_1 a_0 a_4) \\
 & + b_2 * (a_2 a_1 a_0 a_4 a_3) \\
 & + b_3 * (a_1 a_0 a_4 a_3 a_2) \\
 & + b_4 * (a_0 a_4 a_3 a_2 a_1)
 \end{aligned}$$

30

【 0 0 2 2 】

b (x) の係数がゼロならば、この係数と a (x) を循環シフトしたものの積はゼロであり、したがって計算の必要がない。このようにして、乗算を、b (x) についての 2 つの (m + 1) ビット表現の重みの低い方を選択することによって簡単にすることができる。実際に、この乗算を、2 つのシンボルの重みの低い方として b (x) を選択することによってさらに簡単にすることができる。例として、b (x) = 0 0 1 0 1 の場合、乗算は、a (x) * b (x) = (a₄ a₃ a₂ a₁ a₀) + (a₂ a₁ a₀ a₄ a₃) となる。最も簡単な乗算は、1 という 1 つの係数と m のゼロの値の係数とを有する (m + 1) ビットシンボル b (x) を含む。この積はしたがって、a (x) の (m + 1) ビット表現を循環シフトさせたものとなる。

40

【 0 0 2 3 】

50

この乗算は、 m のサイズが増大しても複雑にならない。したがって、 m を少なくともたとえばセクタのビット数の大きさとして選択することができる。このことは、1セクタにつき1つのカラム符号化動作を行なうことを意味している。

【0024】

(1) 符号化

上記のように、従来のRAID-6システムでは、 k のセクタの各々における i 番目の m ビットシンボルと関連する、すなわち i 番目のカラムと関連する r の冗長シンボルは、典型的には以下のようにして生成される。式中、 $y = 0, 1, \dots, r - 1$ である。

【0025】

【数9】

$$c_{yi} = (\alpha^{y*0}) * i_0 + (\alpha^{y*1}) * i_1 + (\alpha^{y*2}) * i_2 + \dots + (\alpha^{y*(k-1)}) * i_{k-1}$$

10

【0026】

現システムは、好ましくは、少なくともディスクセクタにおけるビット数の大きさの m の値を用いる。したがって、1つの符号化動作により、冗長ディスクにおける対応するセクタについてセクタサイズのシンボルが生まれる。要素は、 $m - 1$ の先行ゼロを有する00000...10として選択される。冗長ディスクの数が $m + 1$ 以下であれば、冗長シンボルに対し等式の組において必要なの累乗は以下ようになる。乗算は循環シフトである。

20

【0027】

【数10】

$$\alpha^0 = 000 \dots 01$$

$$\alpha^1 = 000 \dots 10$$

$$\alpha^2 = 000 \dots 100$$

⋮

$$\alpha^m = 100 \dots 00$$

30

【0028】

次に図4を参照すると、 c_r についての等式における k 番目の積は、回路100_rにおいて、 k 番目のディスクの対応するセクタからのデータの $(m + 1)$ ビット表現を $(m + 1)$ ビットシフトレジスタ110_rに与え、シンボルを左に「 j 」ビットだけ循環シフトすることによって計算される。 $j = r - 1, r - 2, \dots, 1, 0$ である。この積を、次に、加算器120_rにおいて、 $(k - 1)$ 番目のディスクの対応するシンボルからのデータに加算する。再びシフトレジスタは j ビットだけシフトされ、 $(k - 2)$ 番目のディスクの対応するセクタからのデータがシフトレジスタの内容に加算され、以降同様の動作を行ない、 c_r の $(m + 1)$ ビット表現を生成する。

40

【0029】

$c_0 = i_0 + (\alpha^{0*1}) i_1 + (\alpha^{0*2}) i_2 + \dots + (\alpha^{0*(k-1)}) i_{k-1}$ を生成する回路100₀については、図5においてより詳細に示している。 k 番目のディスクのセクタからのデータの $(m + 1)$ ビット表現は、シフトレジスタ110₀に与えられ、シフトレジスタは $j = 0$ ビットシフトされる。 $(k - 1)$ 番目のディスクのセクタ情報の $(m + 1)$ ビット表現は、それぞれのビットを組合せる加算器サブ回路121_{0} - 121_mを含む加算器120_rにおいてシフトレジスタの内容に加算され、シフトレジスタはしたがって、 $(\alpha^{0*1}) * i_{k-1} + i_{k-2}$ を含む。シフトレジスタを再び $j = 0$ ビットシフトし、 $(k - 2)$ 番目}

50

のディスクの対応するセクタからのデータの $(m + 1)$ ビット表現を次に、シフトレジスタの内容に加算する。すなわちシフトレジスタは以下を含む。

【 0 0 3 0 】

【数 1 1】

$$\alpha^0((\alpha^{0*1})i_{k-1} + i_{k-2}) + i_{k-3}$$

または

$$(\alpha^{0*2})i_{k-1} + (\alpha^0)i_{k-2} + i_{k-3}$$

10

【 0 0 3 1 】

シフトレジスタは再び $j = 0$ ビットシフトされ、 $(k - 3)$ 番目のディスクの対応するセクタのデータの $(m + 1)$ ビット表現をシフトレジスタの内容に加算する。すなわちレジスタは以下を含む。

【 0 0 3 2 】

【数 1 2】

$$\alpha^0((\alpha^{0*2})i_{k-1} + (\alpha^{0*1})i_{k-2} + i_{k-3}) + i_{k-4}$$

または

$$(\alpha^{0*3})i_{k-1} + (\alpha^{0*2})i_{k-2} + (\alpha^0)i_{k-3} + i_{k-4}$$

20

【 0 0 3 3 】

同様の動作を、残りのデータディスクの対応するセクタからのデータがシフトレジスタに加算されるまで行なう。結果、レジスタは $(m + 1)$ ビット冗長シンボル c_0 を含む。

【 0 0 3 4 】

冗長シンボル c_0 の $(m + 1)$ ビット表現を次に、変換器 1 3 0 において m ビットシンボルに変換する。変換器 1 3 0 は、 $(m + 1)$ ビット表現の先頭ビットと残り m ビットの各々との排他的 OR を取る。先頭ビットがゼロであれば、残りの m ビットは変化なしであり、この演算では先頭ビットを無視した場合と同じ結果が生まれる。先頭ビットが 1 であれば、排他的 OR 演算は、 m ビットを補足し、先頭ビットは無視される。

30

【 0 0 3 5 】

セクタデータの $(m + 1)$ ビット表現は、先頭ゼロビットを追加したセクタの内容である。このようにして、セクタから読出した m ビットを直接、 m ビットを $(m + 1)$ ビット表現に変換するという別のステップを行わずに、シフトレジスタの適切なロケーションに与えることができる。

【 0 0 3 6 】

図 6 に示すように、 c_1 を生成するための回路 1 0 0₁ も同様にシフトレジスタ 1 1 0₁ および加算器 1 2 0₁ を含む。 k 番目のデータディスクのセクタからのデータをシフトレジスタに与え、レジスタは $j = 1$ ビットシフトされ、 $(k - 1)$ 番目のディスクの対応するセクタからのデータがシフトレジスタの内容に加算される。 $(k - 1)$ 番目のシンボルのビット 0 は k 番目のシンボルのビット m と組合せられ、 $(k - 1)$ 番目のシンボルのビット 1 は k 番目のシンボルのビット 0 と組合せられ、以降同様に行なって、

40

$$i_{k-1} + i_{k-2}$$

を生成する。次に、シフトレジスタ 1 1 0₁ の内容を $j = 1$ ビットシフトし、 $(k - 2)$ 番目のデータディスクの対応するセクタからのデータをシフトレジスタの内容に加算して以下を生成する。

【 0 0 3 7 】

【数 1 3】

50

$$\alpha^j(\alpha^j i_{k-1} + i_{k-2}) + i_{k-3}$$

または

$$(\alpha^{j*2})i_{k-1} + (\alpha^{j*1})i_{k-2} + i_{k-3}$$

【 0 0 3 8 】

以降同様に行なう。kのディスクすべてからのデータをシフトレジスタに加算した後、内容は変換器130に与えられる。変換器は(m+1)ビットシンボルをmビットシンボル c_1 に変換する。

10

【 0 0 3 9 】

j > 1に対するシフトレジスタ110_jは、1つのクロックサイクルにおいてjビットシフトを行なうように設定されたフィードバック経路を有してもよい。こうして、単一のシフト演算により適切なビット操作を行なってシフトレジスタ内容を i^j で乗算する。このようにして、冗長シンボル $c_0 \dots c_{r-1}$ を、同数のクロックサイクルで並行して生成することができる。

【 0 0 4 0 】

次に図7を参照して、 c_2 を生成するための回路は、1クロックサイクルで2ビット循環シフトの結果を生成するようにされたフィードバック経路を備えたシフトレジスタ110₂を含む。シフトレジスタ110₂は、ビットmをロケーション112_mからロケーション112₁に与え、ロケーション112_{m-1}のビットm-1をロケーション112₀に与え、残りのビットm-2、...、0それぞれをロケーション112_{m-2}、...、112₀からロケーション112_m、...、112₂に与える。(k-1)番目のディスクの対応するセクタからのビットを次にシフトレジスタの内容に加算し、(k-1)番目のセクタのビット0がk番目のセクタのビットm-1と組合され、(k-1)番目のセクタのビット1がk番目のセクタのビットmと組合され、(k-1)番目のセクタのビット2がk番目のセクタのビット0と組合され、以降同様に行なわれるようにする。したがって、シフトレジスタは

20

$$i^j i_{k-1} + i_{k-2}$$

を含み、j = 2である。次にシフトレジスタの内容を適切にフィードバックして内容を i^j で乗算する。これは、(m+1)ビット内容を左にjビットだけ循環シフトするのと同じである。(k-2)番目のデータディスクの対応するセクタからのデータもまたレジスタに加算し、すべてのデータディスクの対応するセクタからのデータがレジスタに加算されるまで同様に行なう。次にシフトレジスタの内容を、mビット冗長シンボル c_2 を生成する変換器130に与える。

30

【 0 0 4 1 】

好ましくは、 c_3, c_4, \dots, c_r に対する回路は、上記のように単一クロックサイクルで適切なjビットシフトを生じさせるフィードバック経路と同様に構成される。

【 0 0 4 2 】

(2) 復号化

上記のように、このシステムは、検出されたがデータ符号語におけるECCシンボルを用いて訂正することができない誤りを訂正する。したがって、誤りのロケーションはわかっている。以下で図8を参照してより詳細に述べるように、現システムを用いて1つまたは複数の誤りを訂正することには、一般的に、重み1、2または3の要素の逆数を発見することが含まれる。

40

【 0 0 4 3 】

重み1のGF(2^m)要素、すなわち1つの非ゼロ係数を有する要素は、1つの1およびmの0、または1つの0およびmの1を含む(m+1)ビット表現に相当する。i番目のビットで1つの1である係数を有する要素 $i = 000 \dots 01 \dots 00$ について、乗法逆元 i^{-1} は、 i^i で乗算されたときに $0 = 1$ を生成する要素である。

【 0 0 4 4 】

50

シンボル α^0 は $(m + 1)$ ビットシンボル α^{m+1} と同じである。 $(m + 1)$ ビット表現を用いると、 α^i の乗法逆元は、 α^i の i 番目の係数を $(m + 1)$ 番目の位置に循環シフトする要素である。したがって逆数は α^{m+1-i} である。 $m = 4$ の場合、シンボル $\alpha^2 = 00100$ は、 $\alpha^{5-2} = \alpha^3$ の乗法逆元を有する。したがって、 $\alpha^2 * \alpha^3 = \alpha^5 = \alpha^0$ である。3 ビットを左に循環シフトすることによって α^2 を α^3 で乗算することができる。

【0045】

重み 2 の要素すなわち 2 つの 1 の係数および $m - 1$ の 0 の係数を備える要素に対し、乗法逆元は、

$$0000\dots0011 = \alpha^1 + \alpha^0$$

の逆数から決定される。このシンボルの乗法逆元は、 $0101\dots1010 : 0000\dots$ 10

$$0011 * 0101\dots1010 = 0101\dots1010 + 1010\dots0100 = 1111\dots1110$$

または $0000\dots0001$ である。したがって逆数は、シンボル $\alpha^{m-1} + \alpha^{m-3} + \alpha^{m-5} + \dots + \alpha^3 + \alpha^1$ であり、このシンボルを α^{-1} と定めると、

$$(\alpha^1 + \alpha^0) * (\alpha^{-1} + \alpha^0) = \alpha^0$$

が得られる。

【0046】

和 $\alpha^1 + \alpha^0$ を 2 の累乗に累乗すると以下が生成される。

【0047】

【数14】

20

$$(\alpha^1 + \alpha^0)^{2^0} = \alpha^{2^0} + \alpha^0 = \alpha^1 + \alpha^0$$

$$(\alpha^1 + \alpha^0)^{2^1} = \alpha^{2^1} + \alpha^0 = \alpha^2 + \alpha^0$$

$$(\alpha^1 + \alpha^0)^{2^2} = \alpha^{2^2} + \alpha^0 = \alpha^4 + \alpha^0$$

...

$$(\alpha^1 + \alpha^0)^{2^m} = \alpha^{2^m} + \alpha^0 = \alpha^{2^m} + \alpha^0$$

30

【0048】

2 を $GF(m + 1)$ の原始元とすると、 $2^i \bmod (m + 1)$ は 1 と m との間の何らかの値となり得る。したがって、いかなる要素 $\alpha^t + \alpha^0$ も、 $(\alpha^1 + \alpha^0)^t$ として表わすことができる。この t は $0 \leq t < m$ の何らかの値について $2^t \equiv 1 \pmod{m+1}$ である。 $(\alpha^1 + \alpha^0)^t$ の逆数は以下ようになる。

【0049】

【数15】

$$[(\alpha^1 + \alpha^0)^t]^{-1} = [\alpha^1 + \alpha^0]^{t*(-1)} = \delta^t$$

40

または

$$(\alpha^{m-1})^t + \dots + (\alpha^3)^t + (\alpha^1)^t$$

【0050】

式中、指数はモジュロ $m + 1$ である。 $(m + 1)$ ビットシンボルを用いると、 α^i を 2 の累乗に累乗することは、 $(m + 1)$ ビットシンボルのビットの置換として行なわれる。 i 番目のビットの係数すなわち累乗 $t = 2^n$ に累乗した α^i の α^i は以下のとおりである。

【0051】

50

【数 1 6】

$$X^{(2^n * i) \bmod m+1}$$

【0 0 5 2】

何らかの重み 2 の要素 $A = \alpha^h + \alpha^v$ 、 $h < v$ の逆数を決定するために、このシステムは、 $\alpha^{h^*} = A$ の $\alpha^0 + \alpha^{v-h}$ を用いる。したがって A の逆数は以下ようになる。

【0 0 5 3】

【数 1 7】

$$\sigma^{-1} = [\alpha^0 + \alpha^{v-h}]^{-1}$$

または

$$\delta^{v-h} = \alpha^{1*(v-h)} + \alpha^{3*(v-h)} + \dots + \alpha^{(m-3)*(v-h)} + \alpha^{(m-1)*(v-h)}$$

$$A^{-1} = (\alpha^h * \sigma)^{-1} = (\alpha^h)^{-1} * \sigma^{-1}$$

【0 0 5 4】

要素 α^h は重み 1 を有するため以下ようになる。

【0 0 5 5】

【数 1 8】

$$A^{-1} = (\alpha^{m+1-h}) * \delta^{v-h}$$

【0 0 5 6】

量 α^{v-h} は置換として計算され、積は置換されたビットを $(m+1-h)$ ビット左に循環シフトさせたものである。

【0 0 5 7】

$m = 4$ 、 $A = 10100$ または $\alpha^2 + \alpha^4$ 、 $h = 2$ 、および $v = 4$ の場合の例は以下のようにになる。

【0 0 5 8】

【数 1 9】

$$\sigma = \alpha^0 + \alpha^{(4-2)} = \alpha^0 + \alpha^2$$

$$\sigma^{-1} = \delta^2$$

または

$$\alpha^{1*2} + \alpha^{3*2} = \alpha^2 + \alpha^{6(\bmod 5)} = \alpha^2 + \alpha^1$$

$$A^{-1} = (\alpha^{5-2}) * (\alpha^2 + \alpha^1) =$$

$$\alpha^3 * (\alpha^2 + \alpha^1) =$$

$$\alpha^0 + \alpha^4 = 10001$$

【0 0 5 9】

次に挙げるのはこれを検査するための乗法逆元である。

$$A * A^{-1} = 10100 * 10001 =$$

10

20

30

40

50

1 0 1 0 0 + 0 1 0 1 0 = 1 1 1 1 0 = 0 0 0 0 1

重み 3 の要素について、このシステムは 0 0 0 ... 0 0 0 1 1 1 , 0 0 0 ... 0 0 1 0 1 1 , 0 0 0 ... 0 1 0 0 1 1 , (以降同様) の形式の要素の逆数の (m - 1) 要素ルックアップテーブルを用いる。何らかの重み 3 の要素 A の逆数を発見するために ($A = \alpha^a + \alpha^b + \alpha^c$ かつ $a > b > c$)、このシステムは最初に A を、 α^{-c} すなわち $\alpha^{(m+1-c)}$ で乗算し、以下を形成する。

【 0 0 6 0 】

【数 2 0】

$$A * \alpha^{-c} = \alpha^{a'} + \alpha^{b'} + 1 = \gamma$$

$$a' = a - c$$

$$b' = b - c$$

10

【 0 0 6 1 】

これは以下と同じである。

【 0 0 6 2 】

【数 2 1】

$$\gamma = (\alpha^{a''} + \alpha^1 + 1)^{b'}$$

$$a'' * b' = a' \quad \text{または} \quad a'' = \frac{a'}{b'} \pmod{m+1}$$

20

【 0 0 6 3 】

次に、 $\alpha^{a''} + \alpha^1 + 1$ の逆数をルックアップテーブルから検索する。この逆数は要素 D であり、 $\alpha^{a''}$ の逆数は $D^{b'}$ である。A の逆数はしたがって以下のとおりとなる。

【 0 0 6 4 】

【数 2 2】

$$\alpha^c * D^{b'} =$$

$$\alpha^c * D^{b-c}$$

30

【 0 0 6 5 】

式中 D^{b-c} は置換として計算され、 α^c による乗算は循環シフトである。

(3) システム

次に図 8 を参照して、符号器 6 1 0 は、k のデータディスク 6 1 2 からの対応するセクタを符号化して、r の冗長ディスク 6 1 4 に記録するために r の冗長セクタを生成する。この符号器は図 4 - 7 を参照して先に述べた回路 1 0 0_r を用いて、冗長シンボル c_0, c_1, \dots, c_{r-1} を生成する。

40

【 0 0 6 6 】

復号器 6 2 0 および誤り訂正プロセッサ 6 3 0 は、k のデータディスクにストアされたデータにおける誤りを訂正する。この誤りは、データディスクにストアされた ECC シンボルを用い従来のやり方で検出される。

【 0 0 6 7 】

復号器 6 2 0 は、誤りのない m ビットセクタの (m + 1) ビット表現を用い、誤りのあるセクタについては m + 1 のゼロを用いてカラムを復号化する。したがって誤りのあるセクタはカラム復号化については読出されない。q の誤りのあるセクタがあるならば (q は 1

50

つ以上の冗長セクタを含み得る)、このシステムは q の1次方程式の組を解いて誤りのある情報を回復する。

【0068】

この復号器は回路100_rを用いて冗長シンボル C_r を以下のようにして c'_r として再生する。

【0069】

【数23】

$$c'_r = i'_0 + (\alpha^{r*1})i'_1 + \dots + \alpha^{r*k-1}i'_{k-1} \quad 10$$

【0070】

式中、何らかの $i = 0, 1, \dots, k - 1$ に対する i'_i は、 i 番目のデータディスクから読出した誤りのないデータであり、もしデータが誤りを含んでいればすべて0である。復号器は、シンドローム発生器621において従来の態様で、誤りのない冗長ディスクに対応するシンドローム S_j 、 $j = 0, 1 \dots r - 1$ を生成する。

【0071】

1つの誤りを訂正するためのシンドロームは、

$$S_j = (\alpha^j)^p * i'_p \quad 20$$

であり、 i'_p は、 p 番目のディスクと関連する誤りパターンである。誤りの値は、

$$i'_p = S_j * (\alpha^j)^{-1}$$

を解く、誤り値プロセッサ622によって回復される。 α^j の重みは1であるので、上記のように、 $(\alpha^j)^p$ の重みは1であり $(\alpha^j)^{-1}$ は (α^{m+1-jp}) である。プロセッサはこのようにして容易に、重み1のサブプロセッサ624を用いて乗法逆元を決定し、逆数を循環シフトとして S_j で乗算する。

【0072】

二重の誤りを訂正するためのシンドロームは以下のとおりである。

【0073】

【数24】

$$S_j = (\alpha^j)^p * i'_p + (\alpha^j)^q i'_q$$

$$S_{j+1} = (\alpha^{j+1})^p * i'_p + (\alpha^{j+1})^q i'_q \quad 30$$

【0074】

式中 i'_q は q 番目のディスクと関連する誤りパターンである。誤りの値はしたがって以下のようにして決定される。

【0075】

【数25】

40

$$i'_p = \frac{(S_j * (\alpha^{j+1})^q) + (S_{j+1} * (\alpha^j)^q)}{[(\alpha^j)^p * (\alpha^{j+1})^q] + [(\alpha^{j+1})^p * (\alpha^{j+1})^q]}$$

$$i'_q = \frac{(S_j * (\alpha^{j+1})^p) + (S_{j+1} * (\alpha^j)^q)}{[(\alpha^j)^p * (\alpha^{j+1})^q] + [(\alpha^{j+1})^p * (\alpha^{j+1})^q]}$$

10

【 0 0 7 6 】

分母は、 α^a の累乗の 2 つの積の和、すなわち $\alpha^a + \alpha^b$ である。この和の重みは、もし $a = b$ であれば 0 であり、もし $a \neq b$ であれば 2 である。重み 2 の分母については、 $\alpha^a + \alpha^b$ の逆数は上記のように、重み 2 のサブプロセッサ 6 2 5 を用いて容易に決定することができる。

【 0 0 7 7 】

システムは、3 つ以上の誤りを訂正するために、連続冗長ディスクに対応するシンドロームを用いる。式の組を解くために、システムは以下の行列式の逆数を決定する。

20

【 0 0 7 8 】

【 数 2 6 】

$$D = \begin{vmatrix} \alpha^{j * e_0} & \alpha^{j * e_1} & \dots & \alpha^{j * e_z} \\ \alpha^{(j+1) * e_0} & \alpha^{(j+1) * e_1} & \dots & \alpha^{(j+1) * e_z} \\ \vdots & \vdots & \dots & \vdots \\ \alpha^{(j+z) * e_0} & \alpha^{(j+z) * e_1} & \dots & \alpha^{(j+z) * e_z} \end{vmatrix}$$

30

または

$$D = J * \begin{vmatrix} 1 & 1 & \dots & 1 \\ \alpha^{e_0} & \alpha^{e_1} & \dots & \alpha^{e_z} \\ \vdots & \vdots & \dots & \vdots \\ \alpha^{z * e_0} & \alpha^{z * e_1} & \dots & \alpha^{z * e_z} \end{vmatrix} = JV$$

式中

$$J = [(\alpha^{j * e_0}) * (\alpha^{j * e_1}) * \dots * (\alpha^{j * e_z})]$$

40

であり、V は既知のヴァンデルモンドの行列式。

【 0 0 7 9 】

ヴァンデルモンドの行列式の値は、すべての e_0, a, e_{z-1} および e_1, b, e_z について以下の項の積に等しい。

【 0 0 8 0 】

【 数 2 7 】

$$(\alpha^{e_a} + \alpha^{e_b})$$

【 0 0 8 1 】

したがって、行列式 D の値は、因子の逆数の積である以下の逆数である。

【 0 0 8 2 】

【 数 2 8 】

$$J * (\alpha^{e_0} + \alpha^{e_1}) * (\alpha^{e_0} + \alpha^{e_2}) * \dots * (\alpha^{e_{z-1}} + \alpha^{e_z})$$

10

【 0 0 8 3 】

因子 J は、重み 1 のアルファの累乗の連続の積であり、J はしたがって、重み 1 のアルファの累乗である。J の逆数は、 α^{m+1-j} として容易に決定することができる。ヴァンデルモンドの行列式の因子は各々重み 2 であり、その逆数は上記のように重み 2 のサブプロセッサ 6 2 5 を用いて容易に決定される。

【 0 0 8 4 】

行列式 J V は、 $\text{Det}_j / J V$ によって決定される誤りの値を解くために用いられる分母である。この Det_j は、j 番目のカラムを適切なシンδροームで置き換えた行列式 D である。したがって Det_j は項

20

$$(\alpha^{a_j} + \alpha^{b_j}) * S_j$$

の和に等しく、各積は、 S_j を循環シフトさせたもの 2 つの和として計算される。

【 0 0 8 5 】

z の連続するシンδροームを生成するのに十分な数の連続した誤りのない冗長ディスクがなければ、分母における行列式はヴァンデルモンドの行列式ではない。生成される因子はしたがって、2 よりも大きな重みを有する。重み 3 の因子については、重み 3 のサブプロセッサ 6 2 6 が上記のように逆数を決定する。3 よりも大きな重みを有する項については、重み > 3 のサブプロセッサ 6 2 7 がユークリッドの互除法を用いて反復的に $a(x) * b(x) + q(x) * (x^m + x^{m+1} + \dots + 1 + x + 1)$ を解く。b(x) は逆数である。

30

【 0 0 8 6 】

代わりに、このシステムは、カラム ECC について非原始のリードソロモン符号を用いてもよく、カラム符号器および復号器はしたがって、本明細書に引用により援用する、Fettweis および Hassner による、「ハードウェアの複雑性が低い複合リードソロモン符号器およびシンδροーム発生器 (A Combined Reed-Solomon Encoder and Syndrome Generator With Small Hardware Complexity)」、(Proceeding of the 1992 IEEE International Symposium On Circuits and Systems, San Diego, California, May 10-13 1993, pp.1871-1874) において述べられているように、ハードウェアを共有することができる。たとえば、符号器および復号器は、乗算器、シフトレジスタおよび加算器を共有してもよい。非原始のリードソロモン符号を用いるならば、最大で全ディスクは m であり、すなわち $k + r = m$ である。

40

【 0 0 8 7 】

非原始のリードソロモン符号は、 $t + 1$ の連続する根を有する生成多項式

$$g(x) = (x + \alpha^0) (x + \alpha^1) \dots (x + \alpha^{t-1})$$

を有する。非原始のリードソロモン符号については、シンδροームは、単にデータディスクではなくディスクすべてからの検索されたシンボルに基づき決定される。したがってこのシンδροームは、以下のようなになる。

【 0 0 8 8 】

【 数 2 9 】

$$S_j = i'_0 + (\alpha^{j*1})i'_1 + \dots + (\alpha^{j*k+r-2})i'_{k+r-2} + (\alpha^{j*k+r-1})i'_{k+r-1}$$

$$j = 0, 1, \dots, t$$

【0089】

いかなる数の誤り z r についても、常に r の連続するシンδροームが利用できる。実際に、最初の z のシンδροームを使用することができる。誤りの値を解くために逆数が必要とされる行列式はしたがってヴァンデルモンドの行列式である。したがって上記のように、行列式の逆数は容易に、各々が重み 2 を有する、因子 $a + b$ の逆数をとるとともに乗算することによって発見される。誤りの値を決定するための計算は本質的には、いかなる数の誤りについても同じである。こうして、システムは容易に r の誤りまで、重みが 2 よりも大きな何らかの要素の逆数を発見せずとも、訂正することができる。

10

【0090】

上記のシステムは、 m ビットシンボルのカラムを $(m + 1)$ シンボルとして符号化し、 m はディスクセクタまたはテープブロックにおけるビット数と同じまたはこれよりも大きなものとして選択することができる。カラム符号化についての既知の先行技術のシステムでは、このような大きな m ビットシンボルの使用は、必要なガロア体乗算器、および $GF(2^m)$ における乗法逆元を発見するための回路が複雑なために本質的に禁じられている。このシステムはこうした演算を簡単にしており、したがって本質的に任意の大きな値の m を使用することができる。上記のように、 m の値は、必要に応じてより大きな m の値を含むように拡張できる、図 1 から 3 の表から選択される。

20

【0091】

上記の説明は、本発明の特定的な実施例に限定されている。しかしながら、循環シフト演算を行なうシフトレジスタの種々の構成の使用、ソフトウェア、ハードウェアまたはファームウェアにおいて種々の演算を行なうこと、または種々の単一目的のプロセッサを 1 つ以上の多重目的のプロセッサに組合せることなどの変形および修正を本発明に施し、その利点のいくつかがまたはすべてを得ることができることが明らかであろう。したがって、前掲の請求項の目的は、本発明の精神および範囲内のこうしたすべての変形および修正を網羅することである。

30

【図面の簡単な説明】

【図 1】 m の選択された値の表の図である。

【図 2】 図 1 に続く表の図である。

【図 3】 図 2 に続く表の図である。

【図 4】 冗長シンボル c_r を生成するための回路の機能ブロック図である。

【図 5】 冗長シンボル c_0 を生成するための回路の機能ブロック図である。

【図 6】 冗長シンボル c_1 を生成するための回路の機能ブロック図である。

【図 7】 冗長シンボル c_2 を生成するための好ましい回路の機能ブロック図である。

【図 8】 図 4 から 7 の回路を組み込んだ符号化 / 復号化システムの機能ブロック図である。

40

【符号の説明】

6 1 0 符号器

6 2 0 復号器

6 3 0 誤り訂正プロセッサ

【図1】

既約多項式 $x^m + x^{m-1} + x^{m-2} + \dots + x^3 + x^2 + x + 1$ に $m=2$ の
 場合 (2^m) に対応する m の値

2	4	10	12	18	28	36	52	58	60
66	82	100	106	130	138	148	162	172	178
180	196	210	226	268	292	316	346	348	372
378	388	418	420	442	460	466	490	508	522
540	546	556	562	586	612	618	652	658	660
676	700	708	756	772	786	796	820	826	828
852	858	876	882	906	940	946	1018	1060	1090
1108	1116	1122	1170	1186	1212	1228	1236	1258	1276
1282	1290	1300	1306	1372	1380	1426	1450	1452	1482
1492	1498	1522	1530	1548	1570	1618	1620	1636	1666
1668	1692	1732	1740	1746	1786	1860	1866	1876	1900
1906	1930	1948	1972	1978	1986	1996	2026	2028	2052
2068	2082	2098	2130	2140	2212	2220	2236	2242	2266
2268	2292	2308	2332	2338	2356	2370	2388	2436	2458
2466	2476	2530	2538	2548	2556	2578	2620	2658	2676
2682	2692	2698	2706	2740	2788	2796	2802	2818	2836
2842	2850	2860	2908	2938	2956	2962	3010	3018	3036
3066	3082	3186	3202	3252	3298	3306	3322	3346	3370
3412	3460	3466	3468	3490	3498	3516	3532	3538	3546
3556	3570	3580	3612	3636	3642	3658	3676	3690	3700
3708	3732	3778	3796	3802	3850	3852	3876	3906	3916
3922	3930	3946	3988	4002	4012	4018	4020	4090	4092
4098	4132	4138	4156	4218	4228	4242	4252	4258	4260
4282	4348	4356	4362	4372	4396	4450	4482	4492	4506
4516	4546	4602	4620	4636	4690	4722	4786	4788	4812
4876	4932	4956	4972	4986	5002	5010	5050	5058	5076
5098	5106	5146	5170	5178	5188	5226	5260	5308	5332
5386	5442	5476	5482	5500	5506	5556	5562	5572	5650
5658	5682	5692	5700	5716	5740	5748	5778	5812	5826
5842	5850	5868	5922	5938	5986	6010	6028	6052	6066
6100	6130	6172	6196	6202	6210	6228	6268	6276	6298
6316	6322	6372	6378	6388	6396	6468	6490	6546	6618
6636	6652	6658	6690	6700	6708	6732	6762	6778	6780
6802	6826	6828	6868	6882	6898	6906	6916	6946	6948
6970	7012	7018	7026	7042	7068	7108	7186	7210	7218
7228	7236	7242	7252	7282	7306	7330	7348	7410	7450
7458	7476	7498	7506	7516	7522	7540	7546	7548	7572
7588	7602	7620	7642	7668	7690	7716	7756	7788	7828
7852	7876	7882	7900	7906	7932	7948	8052	8068	8092
8116	8122	8146	8170	8178	8218	8220	8236	8242	8268
8290	8292	8362	8386	8428	8442	8466	8538	8562	8572
8596	8626	8668	8676	8698	8730	8740	8746	8802	

(図2に続く)

【図2】

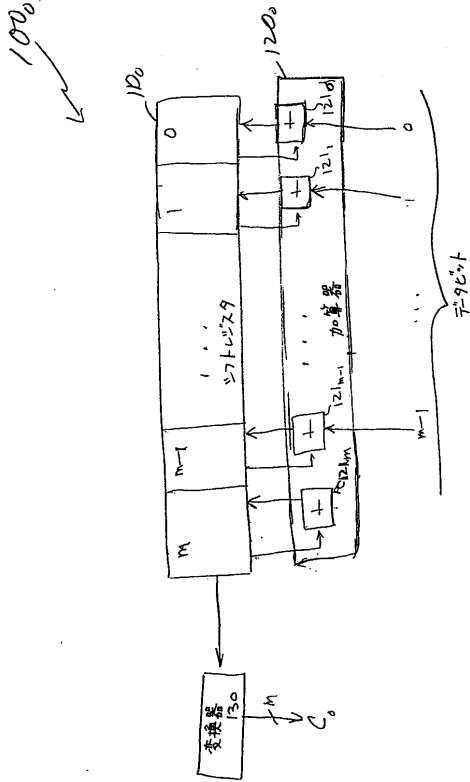
8818	8820	8836	8860	8866	8922	8932	8962	8970	9010
9028	9058	9172	9180	9202	9220	9226	9282	9292	9322
9340	9348	9370	9396	9418	9420	9436	9466	9490	9532
9538	9546	9586	9612	9618	9628	9642	9660	9676	9732
9748	9802	9850	9858	9882	9900	9906	9922	9940	9948
10036	10066	10068	10090	10092	10098	10132	10138	10140	10162
10180	10252	10258	10266	10300	10330	10356	10426	10458	10476
10498	10500	10588	10612	10666	10690	10708	10722	10732	10788
10836	10852	10858	10860	10866	10882	10890	10908	10948	10972
10978	10986	11002	11026	11068	11082	11092	11130	11170	11196
11212	11260	11316	11436	11442	11482	11548	11578	11586	11620
11676	11698	11716	11778	11788	11812	11820	11826	11866	11908
11932	11938	11980	11986	12010	12042	12106	12148	12156	12196
12202	12210	12226	12250	12252	12268	12276	12300	12322	12346
12372	12378	12412	12436	12490	12538	12546	12588	12610	12612
12618	12636	12652	12658	12738	12756	12762	12780	12820	12828
12898	12906	12916	12922	12940	12978	13036	13042	13108	13146
13162	13186	13228	13290	13330	13338	13396	13410	13450	13468
13476	13522	13612	13618	13626	13690	13708	13722	13756	13762
13828	13858	13876	13882	13900	13906	13930	13932	13996	14010
14050	14106	14172	14220	14242	14340	14386	14388	14410	14418
14460	14532	14548	14556	14620	14626	14652	14652	14668	14698
14716	14722	14740	14746	14770	14796	14812	14820	14826	14842
14850	14866	14868	14890	14922	14938	14946	14956	15012	15052
15060	15076	15082	15090	15100	15106	15130	15138	15148	15172
15186	15226	15258	15268	15298	15330	15348	15372	15412	15426
15442	15460	15580	15628	15660	15662	15682	16348	16348	16362
15772	15786	15796	15802	15858	15906	15922	15970	16066	16068
16138	16186	16188	16228	16252	16300	16332	16338	16348	16362
16380	16420	16426	16452	16476	16492	16546	16572	16602	16618
16650	16692	16746	16762	16786	16828	16842	16882	16900	16930
16962	16978	16980	17010	17020	17026	17052	17076	17092	17098
17122	17188	17202	17298	17316	17332	17386	17388	17442	17476
17482	17508	17572	17578	17596	17626	17668	17746	17748	17788
17826	17836	17850	17890	17908	17922	17938	17956	17980	17988
18012	18042	18058	18076	18148	18180	18228	18250	18268	18412
18426	18442	18460	18492	18586	18636	18700	18730	18748	18756
18772	18786	18858	18868	18898	18916	18946	18972	18978	19012
19036	19050	19068	19138	19140	19156	19162	19180	19218	19236
19258	19300	19332	19372	19378	19386	19402	19426	19468	19482
19500	19506	19540	19570	19596	19602	19660	19708	19716	19738
19762	19812	19852	19866	19890	19948	19962	19972	19978	19996
20028	20050	20106	20116	20122	20148	20172	20218	20260	20268
20322	20340	20356	20442	20476	20506	20508	20532	20548	20610
20626	20692	20716	20730	20748	20770	20772	20788	20898	20938
20962	20980	21010	21012	21018	21058	21066	21100	21106	21138

(図3に続く)

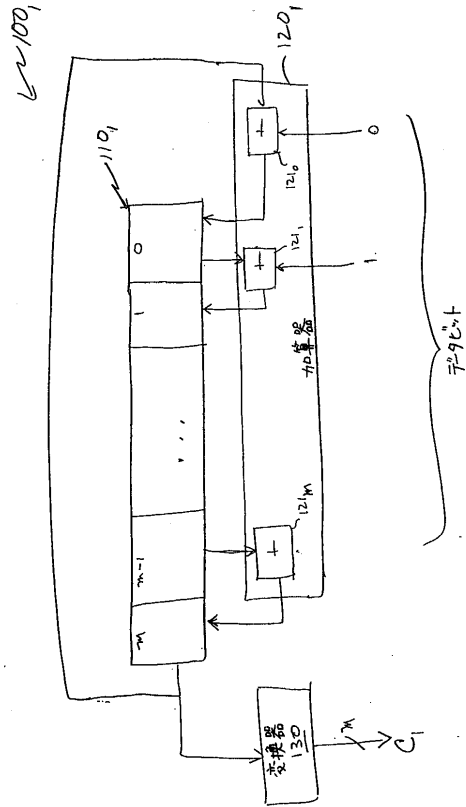
【図3】

21156	21162	21178	21186	21210	21220	21226	21268	21316	21340
21346	21378	21396	21418	21466	21490	21492	21522	21556	21562
21586	21588	21610	21612	21660	21682	21700	21738	21772	21802
21820	21858	21892	22002	22012	22036	22066	22090	22108	22122
22132	22156	22170	22188	22228	22258	22276	22282	22306	22348
22396	22468	22482	22500	22530	22548	22612	22618	22620	22636
22642	22668	22690	22708	22716	22738	22786	22852	22858	22860
22876	22900	22972	23002	23020	23026	23028	23052	23098	23116
23188	23196	23250	23332	23338	23356	23370	23458	23508	23530
23538	23548	23562	23602	23626	23628	23668	23740	23746	23788
23812	23818	23826	23868	23898	23908	23916	23956	24018	24028
24042	24076	24082	24106	24108	24178	24196	24202	24228	24316
24370	24378	24412	24418	24442	24498	24508	24532	24546	24658
24676	24682	24690	24708	24732	24748	24762	24780	24820	24850
24858	24906	24916	24922	24970	24978	24988	25012	25036	25146
25162	25188	25218	25228	25236	25242	25252	25306	25348	25356
25372	25452	25468	25522	25540	25578	25588	25602	25642	25666
25692	25716	25732	25746	25770	25866	25930	25932	25996	26002
26020	26052	26098	26106	26140	26170	26188	26236	26250	26260
26266	26308	26338	26356	26386	26458	26500	26538	26556	26572
26596	26626	26668	26682	26692	26698	26716	26722	26812	26820
26860	26946	26986	27010	27042	27058	27060	27066	27076	27090
27106	27178	27196	27252	27258	27298	27396	27436	27508	27538
27580	27610	27652	27700	27732	27772	27778	27802	27826	27850
27892	27900	27916	27940	27946	28018	28026	28050	28098	28108
28122	28162	28180	28210	28228	28276	28282	28306	28308	28348
28386	28410	28428	28476	28492	28498	28516	28540	28546	28548
28570	28572	28578	28596	28602	28618	28642	28660	28722	28770
28812	28836	28842	28858	28908	28932	28948	28978	29020	29026
29058	29076	29100	29122	29130	29146	29172	29220	29242	29250
29332	29338	29362	29386	29388	29410	29428	294		

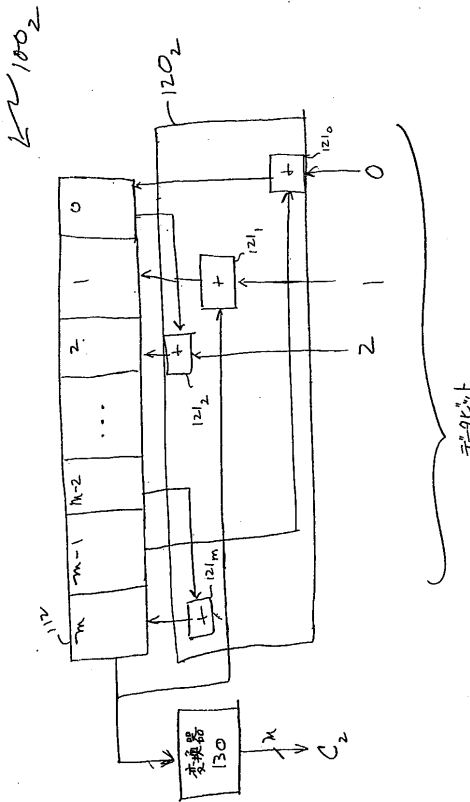
【図5】



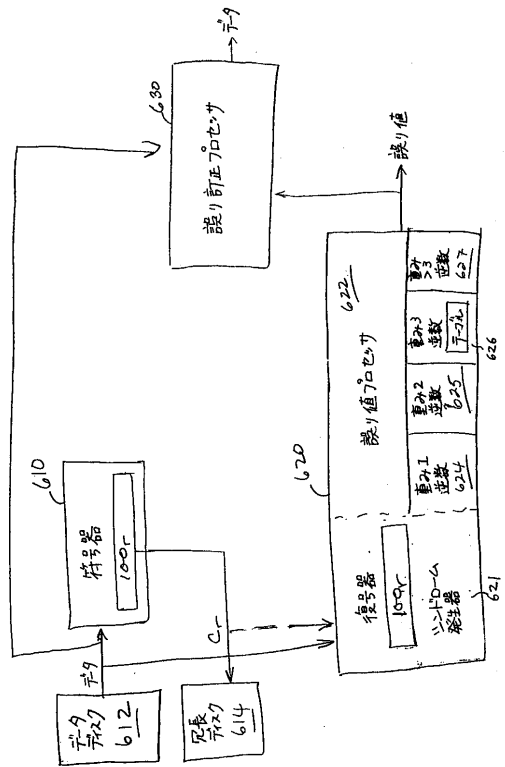
【図6】



【図7】



【図8】



フロントページの続き

(72)発明者 リィ・ジィ・ウエン

アメリカ合衆国、01545 マサチューセッツ州、シュルーズベリー、ブルックデール・サークル、95

審査官 矢頭 尚之

(56)参考文献 特開平10-111767(JP,A)

特開平9-251725(JP,A)

(58)調査した分野(Int.Cl., DB名)

H03M 13/00

G06F 3/06

G06F 11/10

G11B 20/18