

(19) 日本国特許庁(JP)

## (12) 公表特許公報(A)

(11) 特許出願公表番号

特表2005-516458

(P2005-516458A)

(43) 公表日 平成17年6月2日(2005.6.2)

(51) Int.CI.<sup>7</sup>

**H03M 13/15**  
**G06F 11/10**  
**H03M 13/29**  
**H04L 1/00**

F 1

H03M 13/15  
G06F 11/10 330P  
G06F 11/10 330Q  
HO3M 13/29  
HO4L 1/00

テーマコード(参考)

5B001  
5J065  
5K014

HO4L 1/00 B

審査請求 未請求 予備審査請求 未請求 (全 24 頁)

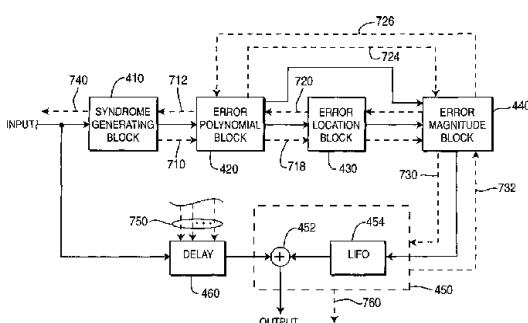
(21) 出願番号 特願2003-563105 (P2003-563105)  
(86) (22) 出願日 平成15年1月22日 (2003.1.22)  
(85) 翻訳文提出日 平成16年9月7日 (2004.9.7)  
(86) 國際出願番号 PCT/US2003/001913  
(87) 國際公開番号 WO2003/063363  
(87) 國際公開日 平成15年7月31日 (2003.7.31)  
(31) 優先権主張番号 10/055,114  
(32) 優先日 平成14年1月23日 (2002.1.23)  
(33) 優先権主張国 米国(US)

(71) 出願人 501263810  
トムソン ライセンシング ソシエテ ア  
ノニム  
Thomson Licensing S  
.A.  
フランス国、エフ-92100 プロー  
ニュ ビヤンクール、ケ アルフォンス  
ル ガロ、46番地  
(74) 代理人 100087321  
弁理士 渡辺 勝徳  
(74) 代理人 100115864  
弁理士 木越 力  
(72) 発明者 リトワイン、ジュニア、ルイス ロバート  
アメリカ合衆国 ニュージャージ州 プレ  
インズボロ ピーオー・ボックス 934  
最終頁に続く

(54) 【発明の名称】復号器内構成要素ブロックによるメッセージ処理

## (57) 【要約】

符号語からシンドローム生成器(410)でシンドロームが計算され、シンドロームに基づいて誤り多項式生成器(420)で誤り多項式が生成され、誤り多項式から誤り位置生成器(430)で誤り位置が決定され、誤り多項式から誤り度合生成器(440)で誤り度合が計算され、位置および誤り度合に応答して誤り訂正符号語生成器(450)により符号語が訂正される復号器および復号方法を説明する。1つまたは複数の構成要素(410、420、430、440、450)が不活動メッセージ(640、650)を生成して、次の符号語に対応するデータを処理する機能を知らせる復号器内ブロック・メッセージ処理方式を説明する。チェン・ブロック(930)を用いて、チェン/フォーニー・ブロック(940)により実施される誤り位置および度合の計算とは別に、指定された符号語に対応する誤りの数を決定する二重チェン・サーチ・ブロック(930、940)の実施形態を説明する。追加のガロア体加算器(1140)を使用して符号語と誤りベクトルの同期をとり、それにより、LIFOレジスタ(952)と共に実施される



**【特許請求の範囲】****【請求項 1】**

シンドローム生成ブロック(410)、誤り多項式ブロック(420)、誤り位置ブロック(430)および誤り度合ブロック(440)を含めて、少なくとも4つのブロックを有する巡回線形ブロック符号誤り訂正復号器において、符号語のデータの誤り訂正を実施する方法であって、

前記シンドローム生成ブロック(410)で、前記符号語からシンドロームを計算する動作と、

前記誤り多項式ブロック(420)で、前記シンドロームから誤り多項式を生成する動作と、  
10

前記誤り位置ブロック(430)で、前記誤り多項式から誤り位置を決定する動作と、

前記誤り度合ブロック(440)で、前記誤り多項式から誤り度合を計算する動作とを含み、

前記少なくとも4つのブロックの少なくとも1つのブロックが、前記少なくとも4つのブロックの他のブロックに非活動メッセージ(640、650)を送信するように適合されることを特徴とする、方法。

**【請求項 2】**

前記誤り位置および前記誤り度合に応答して、前記符号語を訂正する動作を更に含む、請求項1に記載の方法。

**【請求項 3】**

前記非活動メッセージが受信準備完了メッセージ(640)である、請求項1に記載の方法。  
20

**【請求項 4】**

前記非活動メッセージが送信準備完了メッセージ(650)である、請求項1に記載の方法。

**【請求項 5】**

前記誤り多項式ブロックがユークリッド・アルゴリズム・ブロックである、請求項1に記載の方法。

**【請求項 6】**

前記誤り多項式ブロックがバーレカンプ・マッシィ・アルゴリズム・ブロックである、  
30 請求項1に記載の方法。

**【請求項 7】**

前記誤り位置ブロックがチェン・サーチ・ブロックである、請求項1に記載の方法。

**【請求項 8】**

前記誤り度合ブロックがフォーニー・アルゴリズム・ブロックである、請求項1に記載の方法。

**【請求項 9】**

前記誤り位置ブロックおよび前記誤り度合ブロックに関連する前記動作が、チェン/フォーニー組合せブロック内で実施される、請求項1に記載の方法。

**【請求項 10】**

前記巡回線形ブロック符号誤り訂正復号器がリード・ソロモン復号器である、請求項1に記載の方法。  
40

**【請求項 11】**

符号語中のデータの誤り検出を実施する復号器であって、

前記符号語からシンドロームを計算する手段(410)と、

前記シンドロームから誤り多項式を生成する手段(420)と、

前記誤り多項式から誤り位置を決定する手段(430)と、

前記誤り多項式から誤り度合を計算する手段(440)とを備え、

前記シンドロームを計算する前記手段(410)、前記誤り多項式を生成する前記手段(420)、前記誤り位置を決定する前記手段(430)および前記誤り度合を計算する  
50

前記手段(440)を含む群から選択される少なくとも1つの構成要素内に、復号器内非活動メッセージ(640、650)を送信するように適合させることが組み込まれることを特徴とする、復号器。

【請求項12】

前記符号語、前記誤り位置および前記誤り度合を入力として受信することに応答して、前記符号語を訂正する手段(450)を更に備える、請求項11に記載の復号器。

【請求項13】

前記復号器がリード・ソロモン復号器である、請求項11に記載の復号器。

【請求項14】

前記復号器内非活動メッセージが受信準備完了メッセージ(640)である、請求項1 10  
1に記載の復号器。

【請求項15】

前記復号器内非活動メッセージが送信準備完了メッセージ(650)である、請求項1  
1に記載の復号器。

【請求項16】

誤り多項式を生成する前記手段(420)がユークリッド・アルゴリズム・ブロックである、請求項11に記載の復号器。

【請求項17】

誤り位置を生成する前記手段(430)がチェン・サーチ・ブロックである、請求項1  
1に記載の復号器。

【請求項18】

誤り度合を計算する前記手段(440)がフォーニー・アルゴリズム・ブロックである、  
請求項11に記載の復号器。

【請求項19】

符号語中のデータの誤り検出を実施する復号器であって、  
前記符号語からシンドロームを計算するシンドローム生成器(410)と、  
前記シンドロームから誤り多項式を生成する誤り多項式生成器(420)と、  
前記誤り多項式から誤り位置を決定する誤り位置生成器(430)と、  
前記誤り多項式から誤り度合を計算する誤り度合生成器(440)とを備え、  
前記シンドローム生成器(410)、前記誤り多項式生成器(420)、前記誤り位置  
生成器(430)および前記誤り度合生成器(440)を含む群から選択される少なくとも  
1つの構成要素内に、復号器内非活動メッセージ(640、650)を送信するように  
適合させることが組み込まれることを特徴とする、復号器。

【請求項20】

前記符号語、前記誤り位置および前記誤り度合を入力として受信することに応答して、  
前記符号語を訂正する誤り訂正符号語生成器(450)を更に備える、請求項19に記載  
の復号器。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ディジタル通信システムの誤り検出および訂正に関する。

【背景技術】

【0002】

本特許出願は、2002年1月23日に本願と同時出願の「DUAL CHIEN SEARCH BLOCKS IN AN ERROR-CORRECTING DECODER」という名称の米国特許出願第10/055,076号(整理番号PU020003)および2002年1月23日に本願と同時出願の「CHIEN SEARCH CELL FOR AN ERROR-CORRECTING DECODER」という名称の米国特許出願第10/055,470号(整理番号PU020001)に関連する。

【0003】

現在のデジタル通信システムの重要な機能は、誤り制御符号化処理 (error control coding) である。誤り制御符号化処理は、デジタル・システムにおける誤りの検出 (detecting) および訂正 (correcting) に関する技術を扱う通信の分野である。一般に、誤り検出 / 訂正方式は、デジタル・データの送信時またはデジタル・データの記憶により、データに誤りが生じないようにするか、或いは、データに誤りが生じた場合には、生じた誤りの訂正が行われるようにすることが望まれるときに必ず用いられる。データの誤り検出および / または訂正を行う機能は、データに冗長性を附加することにより実現される。送信または記憶するデータ中に冗長ビットを含めると、元の符号化していない信号またはフィールドよりも多くのビットからなる符号化された信号またはフィールドが得られる。

10

#### 【0004】

誤り検出 / 訂正に頻繁に用いられる 1 方式は、いわゆるリード・ソロモン (R S : Reed - Solomon) 符号の使用によるものである。リード・ソロモン符号は、非 2 値組織的線形ブロック符号である。非 2 値符号は、幾つかのビットからなるシンボルを扱う。リード・ソロモン符号などの非 2 値符号は、これらの符号による訂正がシンボル・レベルで行われるので、バースト誤りを訂正するのに適している。リード・ソロモン符号などの組織的符号は、不变形式のメッセージ・シンボルを含む符号語を生成する。符号器は、これらのメッセージ・シンボルに可逆的な数学関数を適用して、冗長またはパリティ・シンボルを生成する。次いで、このパリティ・シンボルをメッセージ・シンボルに付加することにより符号語を形成する。リード・ソロモン符号は、任意の有効符号語を巡回シフトしても別の有効符号語が生成されるので、巡回符号とみなされる。巡回符号は、それを実施する効率的で安価な復号技術が存在するので一般に普及している。最後に、リード・ソロモン符号は、任意の 2 つの有効符号語を追加すると、別の有効符号語が得られるので線形とみなされる。

20

#### 【0005】

典型的なリード・ソロモン復号器は、(i) シンドローム (syndrome) 生成ブロック、(ii) 誤り多項式ブロック、(iii) 誤り位置ブロック、(iv) 誤り度合ブロック、(v) 誤り訂正ブロック、および(vi) 遅延ブロックという主要な構成要素ブロックからなる。シンドローム生成ブロックは、符号語を受信し、この符号語からシンドロームを生成するのに用いる。このシンドロームを用いて、誤り多項式ブロックで誤り多項式を生成する。この誤り多項式を誤り位置ブロックおよび誤り度合ブロックに渡し、そこで符号語の誤りの位置および大きさがそれぞれ決まる。この誤りの位置および大きさから誤りベクトルを生成する。特定の符号語に対応する誤りベクトルを用いて、誤り訂正ブロックで、受信し遅延させたバージョンの符号語を訂正する。遅延ブロックは、誤りベクトルを生成する復号プロセスが実施される間、受信した符号語を記憶するメモリからなる。

30

#### 【0006】

従来技術の復号器は、これらのブロックを用いて「伝送パイプライン」を形成する。即ち、ある 1 つのブロックへの入力は前のブロックの出力のみによって決まり、1 つの構成要素ブロックから前のブロックまたは後続のブロックへのフィードバックはない。従って、一度に 1 つの符号語しか処理することができず、前の符号語の処理が完了するまで、処理すべき次の符号語には取りかからない。

40

#### 【発明の開示】

#### 【0007】

##### (発明の概要)

符号語中のデータの誤り検出を実施する方法および装置である本発明では、従来技術の効率の悪さが解決される。データの誤り検出を実施する復号器は、符号語からシンドロームを計算するシンドローム生成器、シンドロームから誤り多項式を生成する誤り多項式生成器、誤り多項式から誤り位置を決定する誤り位置生成器、および誤り多項式から誤り度合を計算する誤り度合生成器からなる。この復号器は、シンドローム生成器、誤り多項式

50

生成器、誤り位置生成器および誤り度合生成器を含む群から選択された少なくとも1つの構成要素から復号器内非活動メッセージを送信するように適合されることを特徴とする。

### 【0008】

添付の図面と併せて本発明の以下の説明を参照されたい。

### 【発明を実施するための最良の形態】

### 【0009】

図1を参照すると、本発明で実施する原理による誤り検出／訂正方式を組み込んだデジタル・データ伝送システム100を示すブロック図が示されている。一般に、誤り検出／訂正方式は、デジタル・データの送信時またはデジタル・データの記憶により、データに誤りが生じないようにするか、或いは、データに誤りが生じた場合には、生じた誤りの訂正が行われるようにすることが望まれるときに必ず用いられる。データの誤り検出および／または訂正を行う機能は、データに冗長性を附加することにより実現される。送信または記憶するデータ中に冗長ビットを含めると、元の符号化していない信号またはフィールドよりも多くのビットからなる符号化された信号またはフィールドが得られる。この追加のオーバーヘッドを許容する見返りとして、誤りを検出する機能、或いは、誤りを検出し訂正する機能が得られる。誤り制御符号化処理を用いて得られる性能の改善は、符号化利得について評価されることが多い。符号化処理を行わない通信システムで、30dBのSNR (Signal-to-Noise Ratio: 信号対雑音比) で所定のBER (Bit Error Rate: ビット誤り率) が実現されるとする。このシステムに3dBの符号化利得の誤り制御符号化方式を加える場合、この符号化処理を行うシステムは、27dBという更に低いSNRでこのBERを実現することができる。或いは、このシステムを30dBのSNRで稼働させる場合、符号化処理を行うシステムにより実現されるBERは、符号化処理を行わないシステムが33dBのSNRで実現するものと同じBERとなる。符号化利得の威力は、通信システムが、(i) 符号化処理を行わずに所望のBERを維持することが可能であったSNRよりも低いSNRでそのBERを維持できるか、或いは、(ii) 符号化処理を行わないシステムが所定のSNRで得ることができるはずのものよりも高いBERを実現できることである。

### 【0010】

例として、符号器110の機能は、データ源からデジタル・データを受信し、チャネルを介してデータを送信するか、または記憶媒体にデータを記憶するか(図には、全体としてチャネルまたは記憶装置115として示す)、或いは、他の方法でこのデータを操作または処理することである。データの処理、送信または記憶時に、データにノイズまたは誤り125が導入され、そのためしばしば、データが破損するか、或いは元のデジタル・データから変質した形式になる。復号器120は、デジタル・データの規定部分が破損したかどうかを検出し訂正するか、或いは、単に検出するように機能する。

### 【0011】

図2を参照すると、送信機／チャネル／受信機環境の状況で利用可能な様々な誤り検出／訂正プロセスを示す流れ図が示されている。このような状況で説明するが、当業者には、このような誤り検出／訂正プロセスが、ブロードキャスト送信、デジタル・データ記憶、または(データ・フィールド、パケット、ストリームなどの形態にあろうが)デジタル・データを処理または操作する任意の他のプロセスにも同様に適用されることが明らかであろう。網羅的または排他的ではなく単なる例として、(i)テープ、CD(コンパクト・ディスク)、DVD(デジタル多用途ディスク)、バーコードなどの様々な記憶装置(ただし、これらに限定されるものではない)、(ii)(携帯電話、双方向トランシーバ、マイクロ波リンクなどの)無線または携帯通信、(iii)、衛星通信、(iv)デジタル・ラジオ、DTV(デジタル・テレビジョン)、DVB(デジタル・ビデオ放送)など、(v)ケーブル、V.pcm、ADSL、xDSLなどに用いるモ뎀(ただし、これらに限定されるものではない)という技術／装置が、誤り検出／訂正方式を用いて、それらの性能、完全性および信頼性を改善させることができる。

### 【0012】

10

20

30

40

50

ステップ210によれば、最初にリンクを確立し、送信チャネル・パラメータを調停した後で、送信元は、送信に適当な形式でデジタル・データを処理する。ステップ215によれば、送信元は送信を行う前に、少なくとも部分的に送信デジタル・データの値に基づいて、ある程度のデータ冗長度が得られる誤り符号を生成する。ステップ220によれば、この生成された誤り符号を、添付、附加、多重化その他の方法でデジタル・データと共に含め、送信機から受信機に送信する。ステップ225によれば、このデジタル・データおよび誤り符号を受信機で受信する。ステップ230によれば、必要な場合には、受信機で初期信号処理を実施する。ステップ235によれば、受信機は、用いられている誤り制御符号化方式に従って誤り符号の冗長ビットにアクセスし、その中に含まれる情報を処理する。ステップ240によれば、受信したデジタル・データと処理された冗長ビットの照合が肯定的であった場合、このデータは破損していないとみなされる。ステップ245によれば、受信機において、このデジタル・データの更なる信号処理を（それがある場合には）継続して行う。

#### 【0013】

しかし、処理された冗長ビットが、受信したデジタル・データが破損していることを示す（少なくとも1つのビット誤りを含む）場合、ステップ250によれば、このデータ誤りの値を求めて、この誤りが、用いられている誤り制御方式で訂正可能かどうかを判定する。即ち、幾つかの誤り制御方式は、誤り検出しか行うことができず、これらの誤りを訂正することができる特質およびタイプの冗長データは含んでいない。他の誤り制御方式は、検出および訂正機能を共に有し得るにも関わらず、誤り検出機能しか利用しないことがある。これは、個々のデータ信号、メッセージまたはパケットの正確さがさほど重要ではなく、一貫し時宜を得たデータの配信が最も重要であるときにしばしば用いられる方式である。このような応用例は、会話、音声およびビデオ用途の同期ストリーム・データである。或いは、誤り訂正方式を用いて誤りを検出し訂正するときでも、検出された誤りの数または連續して生じた誤りが、誤り訂正機能を上回る場合（即ち、誤り符号ビットにより提供される冗長情報では不足の場合）、このデータは絶対に復元不可能である。ステップ255によれば、データが訂正可能な場合、受信機でデータ誤りを訂正し、更なる信号処理を（それがある場合には）継続する。しかし、データが訂正不可能な場合、ステップ260によれば、データの正確さが本質的なことであるかどうかを決める評価を実施する。純粋なデータ・システムではそうであるように、データの正確さが本質的なことである場合、単一のビット誤りが決定的なものである可能性が大きく、ステップ265によれば、送信元に再送要求を送り返す。しかし、（会話、音声またはビデオなどの）同期的な性質のデータ・メッセージの場合がそうであるように、訂正不可能なデータの正確さが本質的なことではない場合、ステップ270によれば、破損し訂正不可能なデータを単に破棄し、次の連続したデータ・メッセージを処理する。

#### 【0014】

図3を参照すると、様々な誤り符号の分類および方式の階層図が示されている。誤り符号310は、(i)自動再送要求(A R Q : A u t o m a t i c R e t r a n s m i s s i o n R e q u e s t )または検出符号315、および(ii)前方誤り訂正(F E C : F o r w a r d E r r o r C o r r e c t i o n )符号320という2つの基本的な分類に分けることができる。A R Qは、検出だけを行うタイプの符号であり、受信機により送信時の誤りを検出することができるが、誤りを訂正することはできない。受信機は、受信されたデータを調べ、誤りが検出された受信データを送信し直す要求を出さなければならない。このような再送は、貴重な帯域を横取りすることになるので、一般に、A R Q符号は「クリーン」な（誤りの可能性がより低い）送信媒体に用いられる。最も一般的な例の1つは、単純なパリティ・チェック325であり、R A M中のデータ誤りを検出するのにしばしば用いられる。別の例は、C R C（巡回冗長検査）330であり、例えば、イーサネット（登録商標）上の送信における誤りを検出するのに用いられる。誤りが検出された場合、そのメッセージが再送されることになる。イーサネット（登録商標）は主に電線上を送信されるので、誤りの確率は他の幾つかの媒体よりも少ない。C R Cおよび

A R Q は、誤り検出符号方式の单なる 2 つの例であり、当業者には他の誤り検出符号方式が知られている。単に誤りを検出するだけでそれらを訂正しない誤り符号は、加えられる冗長度が誤り訂正符号よりもかなり少ない。更に、誤り検出用の復号器は、誤り訂正用の復号器よりもはるかに簡単である。一般に、誤り検出符号方式を利用するシステムは、データの再送で生じるオーバーヘッドに関して帯域許容性が高い。即ち、データを再送しても、全体的なシステムのスループットに大きく影響しない。

#### 【 0 0 1 5 】

ノイズの多い媒体では、所定の送信中に誤りが生じる確率が高いので、自動再送要求 (A R Q ) 法を用いると常時データの再送が生じることになり、それにより、システムのスループットが許容できないレベルまで落ちる。このような例では、誤り訂正符号により、その名前が示すとおり、受信側で誤りを検出するだけでなく、誤りの訂正を行うこともできる。こうすると、データを再送する必要が減り、用いられる誤り訂正方法により訂正することができる数よりも誤りの数が多いときだけデータの再送が必要になる。誤り訂正是、受信機が送信元に再送を要求する機会がない一方向通信にも用いられる。このような一方向経路の例には、一部の衛星送信および磁気テープ記憶媒体が含まれる。

#### 【 0 0 1 6 】

誤り訂正符号は、2 つの主要な副分類に分けることができる。第 1 はブロック符号 3 3 5 であり、第 2 は畳込み符号 3 4 0 である。ブロック符号 3 3 5 は、定義された有限長のメッセージ・ブロックを扱う誤り訂正符号であり、しばしば用いられている。ブロック符号 3 3 5 は、この副分類では、ブロック単位のメッセージ・シンボルを処理し、次いで、ブロック単位の符号語シンボルを出力する符号器を利用してこの名前がついている。一般に、ブロック符号は、2 値符号 3 4 5 および非 2 値符号 3 5 5 の 2 つのタイプに分類することができる。2 値符号 3 4 5 の一例はハミング符号 3 5 0 であり、1 文字当たり 4 つの情報ビットおよび 3 つの検査ビットを有するという特徴がある。非 2 値符号の例は、リード・ソロモン符号 3 6 0 である。

#### 【 0 0 1 7 】

対照的に、畳込み符号 3 4 0 用の符号器は、連続したストリームのメッセージ・シンボルを扱い、同時に、符号化された連続出力ストリームを生成する。これらの符号は、その符号化プロセスを、メッセージ・シンボルと符号器のインパルス応答の畳込み処理とみなすことができることからこの名前がついている。畳込み符号 3 4 0 の例には、T C M (トレリス (T r e l l i s ) 符号化変調) 3 6 5 および 2 値畳込み符号化処理 3 7 0 の 2 つがある。

#### 【 0 0 1 8 】

図 4 は、R S (リード・ソロモン) 復号器の例を示すブロック図である。前に説明したように、リード・ソロモン符号は、非 2 値組織的線形ブロック符号である。非 2 値符号は、幾つかのビットからなるシンボルを扱う。非 2 値符号用の一般的なシンボル・サイズは、8 ビット即ち 1 バイトである。リード・ソロモン符号などの非 2 値符号は、これらの符号の訂正がシンボル・レベルで行われるので、バースト誤りを訂正するのに適している。復号プロセスでシンボルを扱うことにより、これらの符号は、1 個のビット誤りを伴うシンボルを訂正することができるのと同じように簡単に、8 個の誤りのバーストを伴うシンボルを訂正する。リード・ソロモン符号などの組織的符号は、不变形式のメッセージ・シンボルを含む符号語を生成する。符号器は、これらのメッセージ・シンボルに可逆的な数学関数を適用して、冗長またはパリティ・シンボルを生成する。次いで、このパリティ・シンボルをメッセージ・シンボルに付加することにより符号語を形成する。リード・ソロモン符号は、任意の有効符号語を巡回シフトしても別の有効符号語が生成されるので、巡回符号とみなされる。巡回符号は、それを実施する効率的で安価な復号技術が存在するので一般に普及している。最後に、リード・ソロモン符号は、任意の 2 つの有効符号語を追加すると、別の有効符号語が得られるので線形とみなされる。

#### 【 0 0 1 9 】

誤り制御符号の理論では、有限体またはガロア体として知られている数学的な構造体を

10

20

30

40

50

用いる。ガロア体は、有限個の元を含む 1 つの組である。この組に対する加算および乗算演算が定義される。これらの演算は、通常の算術から予想されるように動作する。例えば、加算の単位元はゼロであり、乗算の単位元は 1 である。リード・ソロモン符号は、この符号の巡回且つ線形な性質を司るガロア体の数学的処理を用いて実施され、 $q = p^m$  次のガロア体上で演算が行われる。ただし、 $p$  は正の素数であり、 $m$  は正の整数である。 $q$  次のガロア体は  $G F ( q )$  で表し、 $q$  個の別個の元を含んでいる。

#### 【 0 0 2 0 】

所定のリード・ソロモン符号は、それを  $( n , k )$  符号と称することにより示す。パラメータ  $n$  は、符号語中のシンボルの数で表した符号語長を示す。パラメータ  $k$  は、符号語中のメッセージ・シンボルの数を示す。従って、加えられるパリティ・シンボルの数は  $n - k$  個になる。この符号の誤り訂正機能は、 $t = ( n - k ) / 2$  である。この符号は、 $T$  個の誤りを検出し訂正する。ただし、 $0 \leq t \leq T$  である。この符号語は、メッセージ・シンボルに基づくものであり、リード・ソロモン符号器により生成される。リード・ソロモンは組織的符号 ( systematic code : システマティック・コード ) なので、 $n$  個のメッセージ・シンボルはそのまま送信され、これらのメッセージ・シンボルに  $n - k$  個のパリティ・シンボルが付加されて符号語が形成される。送信される符号語に冗長度を加えるこのパリティ・シンボルの値は、メッセージ・シンボルにより決まる。受信機の復号器はこの冗長度を利用して、誤りを検出し訂正する。

#### 【 0 0 2 1 】

受信機の復号器で、シンドローム生成ブロック 410 への入力として符号語を受信する。復号器により実施される最初のステップはシンドローム計算であり、( しばしばシンドローム生成モジュールまたは単にシンドローム生成器とも称する ) シンドローム生成ブロック 410 により実施される。このシンドロームは  $n - k$  個のシンボルからなり、これらの値は受信した符号語から計算する。このシンドロームは、誤りベクトルのみにより決まり、送信された符号語とは無関係である。即ち、各誤りベクトルは固有のシンドローム・ベクトルを有するが、受信された多くの異なる符号語は、それらの誤りパターンが同じ場合には同じシンドロームを有することになる。最初にシンドロームを計算する理由は、そうすると誤りベクトルの探索範囲が狭まるからである。まず、シンドロームが分かると、妥当な誤りベクトルの数が、全部で  $2^n$  個の可能な誤りベクトルから  $2^{n-k}$  個の誤りベクトルに減る。

#### 【 0 0 2 2 】

シンドローム生成ブロック 410 がシンドロームを計算する 1 つの方法は、ガロア体代数を用いて、受信した符号語を生成多項式で割ることによるものである。この除算の剩余をシンドローム多項式  $s ( x )$  と呼ぶ。実際のシンドローム・ベクトル  $S ( x )$  は、 $\sim^{n-k}$  で  $s ( x )$  の値を求ることにより計算する。ただし、この方法は、ハードウェアの観点から最も効率的でないことがあり、ハードウェア内で頻繁に用いられる代替方法では、直接  $\sim^{n-k}$  で受信済み符号語  $R ( x )$  の値を求める。シンドローム生成ブロック 410 は、 $\sim^{n-k}$  で受信した符号語  $R$  の値を求ることにより、即ち  $R ( \sim^{n-k} )$  によりシンドローム  $S$  を計算する。リード・ソロモン符号では、 $n - k = 2t$  であり、従って、計算しようとする  $2t$  個のシンドローム値  $[ S_1 \ S_2 \ S_3 \dots S_{(2t)} ]$  がある。一般に、これらの値は、第 1 シンドローム生成器が  $\sim^2$  で受信済み符号語の値を求めて  $S_1$  を形成し、次のシンドローム生成器が  $\sim^2$  で受信済み符号語の値を求めて  $S_2$  を形成し、以下同様に行って、並列に計算する。

#### 【 0 0 2 3 】

シンドローム生成ブロック 410 によりシンドロームが計算されると、この値は、誤り多項式ブロック 420 に渡される。そこで、このシンドロームを用いて誤り位置多項式を生成する。このプロセスでは、 $t$  個の未知数の連立方程式を解くことが必要である。これらの計算を行うのに、バーレカンプ・マッサイ ( Berlekamp-Massay ) アルゴリズムまたはユークリッド・アルゴリズムを含めて、幾つかの高速アルゴリズムが利用可能である。これらのアルゴリズムは、リード・ソロモン符号の特別な行列構造を利用

10

20

30

40

50

し、必要な計算の労力を大きく減らす。

【0024】

誤り多項式ブロック420は、誤り位置多項式を（それが決定された後で）誤り位置ブロック430および誤り度合ブロック440に渡す。誤り位置ブロック430は、この誤り位置多項式の根を見つけて誤り位置を決定する。一般に、これは、チェン・サーチ・アルゴリズムまたはチェン・セルを利用して行われる。誤り位置ブロック430により決定された誤り位置は、前に決定された誤り位置多項式と共に誤り度合ブロック440に渡される。誤り度合ブロック440は、 $t$ 個の未知数を有する連立方程式を解くことにより誤り度合を決定する。誤り度合ブロック440を実施する際に用いる高速で広く用いられているアルゴリズムは、フォーニー（Forney）アルゴリズムである。

10

【0025】

計算された誤り位置および誤り度合は、誤り訂正ブロック450に転送されて、符号語が破損している場合には、符号語が訂正されて復元される。誤り位置および誤り度合の組合せを、しばしば誤りベクトルと称する。この誤りベクトルは、符号語と同じサイズであり、誤りに対応する位置で非ゼロ値を含んでいる。他の全ての位置にはゼロが入る。誤り訂正ブロック450への別の入力は遅延ブロック460の出力である。遅延ブロック460は、受信した符号語を取り込み、この受信した同じ符号語を遅延させて出力する。一実施形態では、誤り訂正ブロック450は、（最後に入力されたものが最初に出力される）LIFOブロック454と共にガロア体加算器452を使用して実施する。受信した符号語中の誤りは、ガロア体加算器452を使用して、この受信した符号語に誤りベクトルを加えることにより訂正される。LIFOブロック454を使用するのは、誤りベクトルが受信した符号語とは逆の順序で生成され、従って、受信した符号語または誤りベクトルの何れかにLIFO演算を適用して、両方のベクトル中のバイトの順序を整合させなければならないからである。誤り訂正ブロック450の出力は、復号器による元の符号語の推定値である。

20

【0026】

図5は、RS（リード・ソロモン）復号器の実施形態の例を示すブロック図である。入力は受信した符号語であり、それをシンドローム生成ブロック510に転送する。シンドローム生成ブロック510によりシンドロームS（ $\times$ ）が計算されると、その値は、ユークリッド・アルゴリズム・ブロック520に渡される。ユークリッド・アルゴリズムによりシンドロームS（ $\times$ ）が処理され、それにより、誤り位置多項式（ $\times$ ）および誤り度合多項式（ $\times$ ）が生成される。即ち、ユークリッド・アルゴリズムにより、キー（Key）方程式と称する以下の方程式が解かれる。

30

【0027】

【数1】

$$\Lambda(x) [1 + S(x)] = \Omega(x) \bmod x^{2t+1} \quad \text{式1}$$

リード・ソロモン復号処理で用いるアルゴリズムは、2つの多項式の最大公約数（GCD）を求めるのにユークリッド・アルゴリズムに基づいている。ユークリッド・アルゴリズムは、周知の反復多項式除算アルゴリズムである。

40

【0028】

誤り位置多項式（ $\times$ ）が計算されると、この多項式の値を求めてその根を見つける必要がある。チェン・サーチ・アルゴリズムを用いて、これらの根を見つける。チェン・サーチは、全ての可能な入力値に対して多項式の値を求める総当たり式アルゴリズムであり、次いで、どの出力がゼロに等しいかを決める。位置*i*で誤りが生じる場合、次式がゼロに等しくなる。

【0029】

## 【数2】

$$\sum_{j=0}^t A_j \alpha^{-j} = 0, \text{ ただし, } i=0..(n-1) \quad \text{式2}$$

## 【0030】

チェン・サーチにより、全ての  $i$  および  $j$  の値について式2の値が求まり、この式がゼロに等しくなる回数が数えられる。ゼロの位置は誤りの位置であり、ゼロの数は誤りのあるシンボルの数である。

## 【0031】

ある実施形態の例では、( $t + 1$ )個のチェン・サーチ・ステージがあり、これらはハードウェアで実施される。これらのステージ(ステージは、乗算器、マルチプレクサおよびレジスタからなる)はそれぞれ、上記チェン・サーチ方程式中の  $j$  について異なる値を示す。このサーチは、 $n$ クロック・サイクルに亘って実行され(上記方程式中の異なる  $i$  の値が、各クロック・サイクルを表す)、加算器出力がゼロに等しいかどうか検査される。加算器出力がゼロに等しい場合、ゼロ検出ブロックが1を出力することになり、そうでない場合には、ゼロが出力されることになる。このように、チェン・サーチ・ブロックの出力は、「0」または「1」の何れかの値をとる  $n$  個のビット列になる。「1」はそれぞれ、誤りのあるシンボルの位置を表す。

## 【0032】

第1クロック・サイクルで、マルチプレクサが、誤り位置多項式の係数をレジスタに送る。残りの( $n - 1$ )個のクロック・サイクルに亘って、乗算器出力は、マルチプレクサを介してレジスタに送られることになる。これらの乗算器のベキ指数は負の値を有する。ただし、これらの値は、モジュロ演算子を用いてあらかじめ計算することができる。

$i$  のベキ指数は、 $(-i \bmod n) = (-i \bmod 255)$  に等しい。例えば、 $-1$  は  $254$  に等しく、 $-2$  は  $253$  に等しい。以下同様である。

## 【0033】

フォーニー・アルゴリズムを用いて、誤りの値  $Y_i$  を計算する。これらの値を計算するために、フォーニー・アルゴリズムでは、誤り位置多項式 ( $x$ ) および誤り度合多項式 ( $x$ ) を用いる。誤りの値を求める式は次のようにある。

## 【0034】

## 【数3】

$$Y_i = \frac{\Omega(x)}{\Lambda'(x)} \quad x = \alpha^i \quad (\text{ただし, } \alpha^i \text{ は } \Lambda(x) \text{ の根}) \quad \text{式3}$$

## 【0035】

形式的導関数 ( $x$ ) の計算は、実際には極めて簡単である。例えば、 $(x) = 4X^3 + 3X^2 + X + 2$  と仮定する。こうすると、 $(x)$  は次式のようになる。

## 【0036】

## 【数4】

$$\begin{aligned} \Lambda'(x) &= 3\alpha^4 X^2 + 2\alpha^3 X + \alpha \\ &= (\alpha^4 + \alpha^4 + \alpha^4) X^2 + (\alpha^3 + \alpha^3) X + \alpha \\ &= \alpha^4 X^2 + \alpha \end{aligned} \quad \text{式4}$$

$X$  の奇数の指数の係数をとり、それらを次に低い指数(偶数)の  $X$  に割り当てることに

10

20

30

40

50

より導関数を形成する。

【0037】

次いで、チェン・サーチに用いたのと同じタイプのハードウェアを用いて、  
 多項式と共に  $(\times)$  多項式の値を求める。 $(\times)$  の値を求めるには、 $\alpha_0$  の係数に  
 $\alpha_1$  の係数に  $\alpha^{-1}$  を掛けたものを加え、続けて、 $\alpha_2$  の係数に  $\alpha^{-2}$  を掛けたもの、以下  
 同様に、 $\alpha_t$  の係数に  $\alpha^{-t}$  を掛けたものまで加える。次いで、これらの乗算器の出力を  
 合計する。

【0038】

次いで、逆数の乗算器を用いて分子に分母を掛ける。この逆数の乗算器は、分母の逆数  
 を見つけ出すルックアップ・テーブルを含んでいる。例えば、分母が  $\alpha^3$  の場合、逆数は  
 $\alpha^{-3}$  になる。次いで、これを次式のように表すことができる。

【0039】

【数5】

$$\alpha^{-l} = \alpha^{(-l \bmod n)} = \alpha^{(-3 \bmod 255)} = \alpha^{252}. \quad \text{式5}$$

【0040】

チェン・サーチおよびフォーニー・アルゴリズムには共に同じタイプのハードウェアが必要なので、チェン／フォーニー・ブロック530として示すように、これら2つの機能を同じブロック内で組み合わせることができる。この実施形態では、チェン・サーチの出力のところで2つの加算器を用いる。第1加算器は偶数ステージの値を合計し、他方の加算器は奇数ステージの値を合計する。最終的なチェン・サーチ出力を形成するには、次いで、これら2つの加算器の出力を合計し、ゼロ検出ブロックが根の位置を検出する。フォーニー・アルゴリズムでは、奇数ステージ用加算器の出力も用いる。奇数ステージの合計は、フォーニー方程式の分母を表す。この合計値の逆数をとり、次いで、誤り度合多項式の値を求ることにより形成された分子の値を掛ける。この出力とゼロ検出出力のアンド(AND)演算を行う。というのは、これらの誤りの値は、実際の誤り位置でしか有効でないからである（そうでない場合には、誤りの値はゼロにしようとするである）。

【0041】

このように、チェン／フォーニー・ブロック530は、誤り位置多項式および誤り度合多項式を用いて誤りベクトルを生成し、それを誤り訂正ブロック540に転送する。この誤りベクトルは、符号語と同じサイズであり、誤りに対応する位置で非ゼロ値を含んでいる。他の全ての位置にはゼロが入る。誤り訂正ブロック540への別の入力は遅延ブロック550の出力である。遅延ブロック550は、受信した符号語を取り込み、この受信した同じ符号語を遅延させて出力する。この実施形態では、誤り訂正ブロック540は、LIFOブロック542と共にガロア体加算器544を使用して実施する。受信した符号語中の誤りは、ガロア体加算器544を使用して、この受信した符号語に誤りベクトルを加えることにより訂正される。LIFOブロック542を使用するのは、誤りベクトルが受信した符号語とは逆の順序で生成され、従って、受信した符号語または誤りベクトルの何れかにLIFO演算を適用して、両方のベクトルについてバイトの順序を同期させなければならないからである。誤り訂正ブロック540の出力は、復号器による元の符号語の推定値である。

【0042】

復号器内構成要素ブロック・メッセージ処理

前に説明したように、リード・ソロモン復号器は、少なくとも4つの主要な構成要素即ち機能ブロックを有する。それらは、シンドローム生成器、誤り多項式ブロック、誤り位置ブロックおよび誤り度合ブロックである。復号器が誤りを検出するのに加えてそれを訂正する場合、復号器は誤り訂正ブロックも含んでいる。従来技術の復号器では、これらのブロックにより、ある1つのブロックへの入力は前のブロックの出力によりのみ決まる「

10

20

30

40

50

「伝送パイプライン」が形成される。即ち、1つのブロックから前のブロックへのフィードバックはない。リード・ソロモン復号器は、(図4および図5の遅延ブロックにより示す)個々の符号語に対する復号プロセスを実施する間に、受信した符号語を記憶するためのメモリまたはバッファ・ブロックを実現することも必要である。誤り位置/度合ブロックは、復号プロセスの完了時に誤りベクトルを生成し、この誤りベクトルとメモリ中に記憶させた受信済み符号語のXOR演算を行って復号器出力を形成する。一般に、使用するメモリのサイズは1符号語に等しく、そのため、この復号器は、一度に1つの符号語しか処理することができない。

#### 【0043】

図6は、本発明の原理による復号器内ハンドシェイク・プロトコルの実施形態の例を示すブロック図である。復号器内ブロックA 610および復号器内ブロックB 620は、(上記で説明した機能ブロックから選択した)リード・ソロモン復号器内の任意の2つのブロックを表し、これらのブロック間でフィードバックまたは通信を利用する。復号処理用「伝送パイプライン(delivery pipeline)」はデータ・チャネル630であり、これは、リード・ソロモン復号器の機能ブロック間の従来技術による任意の一方向チャネルを表す。それに加えて、本発明によれば、フィードバック・チャネルも示されている。このフィードバック・チャネルは、ある1つの機能ブロックが、その機能ブロックの現在または今後の非活動状態を上流側または下流側の機能ブロックに伝えるための手段とみなすことができる。例えば、復号器内ブロックA 610が、ある特定の符号語に対してその処理機能を完了したと仮定する。復号器内ブロックA 610は、指定された符号語に対するその計算機能の結果を復号器内ブロックB 620に転送する準備ができていることを示す非活動メッセージを復号器内ブロックB 620に送る。そのため、このタイプの非活動メッセージをRTS(Ready-To-Send:送信準備完了)メッセージ650と呼ぶ。逆に、復号器内ブロックB 620が、ある特定の符号語に対してその処理機能を完了したと仮定する。復号器内ブロックB 620は、それが受信する準備ができていることを示す非活動メッセージを復号器内ブロックA 610に送る。そのため、このタイプの非活動メッセージをRTT(Ready-To-Receive:受信準備完了)メッセージ640と呼ぶ。

#### 【0044】

機能ブロック間のフィードバックまたはハンドシェイク処理を伴う復号器の実施形態である本発明のこの実施形態は、こうすると、ユーザが構成可能な、IP(知的財産)コアに適当なアーキテクチャが可能になり、それにより、実現して使用したいメモリ・サイズに基づいて、ユーザが復号器の性能をカスタマイズすることができるという点で有利である。

#### 【0045】

図7は、RS(リード・ソロモン)復号器の実施形態の例の機能ブロック間のハンドシェイク・プロトコルを示すブロック図である。シンドローム生成ブロック410、誤り多項式ブロック420、誤り位置ブロック430、誤り度合ブロック440、(ガロア体加算器452およびLIFOブロック454を含む)誤り訂正ブロック450および遅延ブロック460の機能、目的および動作は、図4に関して十分に説明されており、従ってここでは繰り返さない。図7に新たに含めた上記非活動メッセージ(またはハンドシェイク信号)が示されている。これらの処理ブロックは、ハンドシェイク信号(例えば、RTSおよびRTTメッセージ)を用いて、これらの処理ブロックが送信データをもっている時点、および/またはそれらがデータを受信する準備が整っている時点を示す。こうすると、シンドローム生成ブロック410と誤り多項式ブロック420は、RTSチャネル710を介してRTSメッセージを、或いは、RTTチャネル712を介してRTTメッセージを交換する。誤り多項式ブロック420と誤り位置ブロック430は、RTSチャネル718を介してRTSメッセージを、或いは、RTTチャネル720を介してRTTメッセージを交換する。誤り多項式ブロック420と誤り度合ブロック440は、RTSチャネル724を介してRTSメッセージを、或いは、RTTチャネル726を介してRTT

10

20

30

40

50

メッセージを交換する。誤り度合ブロック 440 と誤り訂正ブロック 450 は、RTS チャネル 730 を介して RTS メッセージを、或いは、RTT チャネル 732 を介して RTT メッセージを交換する。シンドローム生成ブロック 410 は、RTT チャネル 740 を介して RTT メッセージを送信することにより次に受信する符号語を要求するようにも適合される。誤り訂正ブロック 450 は、RTS チャネル 760 を介して出力処理ステージに RTS メッセージを送信するようにも適合される。更に、本発明の選択した実施形態により、遅延を適切に調整して、誤り訂正ブロック 450 において受信した符号語が誤りベクトルに同期して検査され、正しく訂正されるように、(メッセージ・チャネル 750 を介して) 遅延ブロック 460 に RTT または RTS メッセージを送信することが必要になることがある。

10

#### 【0046】

本発明から得られる有利な有用性は、復号器内機能ブロック間のハンドシェイク処理を可能にすることにより、機能ブロックが非活動状態になったときに(即ち、機能ブロックがその最後の手順を完了したときに)、機能ブロックがより多くの処理データの受信を要求できることである。同様に、機能ブロックは、その機能ブロックがその手順を完了しており、且つ下流側の機能ブロックがその結果を受信することができるときにはいつでもその結果を転送できることを下流側の機能ブロックに示すことができる。例えば、一般に、シンドローム生成ブロック 410 は、特定の長さの受信した符号語に対して固定された(既知の)数のクロック・サイクルを用いてシンドロームを生成することになるが、誤り多項式ブロック 420 は、その実行時間が変化し得る反復プロセスである。更に、一般に、誤り位置ブロック 430 および誤り度合ブロック 440 は、固定された(既知の)数のクロック・サイクルの時間が掛かることになるが、符号語中に存在する誤りの数が多すぎて符号語が訂正不可能と判定された場合には、その符号語に対する更なる処理を中断することができ、誤り位置ブロック 430 および誤り度合ブロック 440 は、適切な RTT チャネル 720、726 を介して誤り多項式ブロック 420 に RTT メッセージを送信することにより、次の符号語に対する処理を開始するように要求する。従って、誤り位置ブロック 430 および誤り度合ブロック 440 も、その処理を完了するのにかけるクロック・サイクルを可変にする。

20

#### 【0047】

ただし、図 7 に示す本発明の実施形態は、本発明の趣旨および範囲から逸脱することなく実施することができる幾つかの実施形態の 1 つであることに留意されたい。例えば、復号器の各機能ブロックが RTS および RTT メッセージを送信するように構成される必要はない。そうではなくて、設計上の選択事項として、シンドローム生成ブロック 410 と誤り多項式ブロック 420 の間だけでハンドシェイク処理を行うことができることが望ましいことがある。多くの他の変形形態も可能である。更に、RTS および RTT チャネルを介して通信されるように RTS および RTT メッセージを示し説明するが、このような説明は単なる論理上の説明であり、メッセージ・チャネルは、データ・チャネルとは別の 1 つの共通物理層メッセージ・チャネル上に組み込むこともできるし、或いは、データ・チャネル自体と同じ物理層上に組み込むこともできる。

30

#### 【0048】

本発明の原理によれば、ユーザは、メモリ割当てサイズを変更することにより復号器の性能をカスタマイズする。従って、1 符号語長に等しいメモリ・サイズを割り当てた実施形態は、ハードウェアおよび電力が効率的に使用される設計になるが、一度に 1 つの符号語しか処理することができなくなる。1 符号語長よりも長いメモリ・サイズを割り当てられた実施形態は、1 つの符号語を処理しながら、第 2 の符号語の取込みを開始する。2 つの符号語のメモリ・サイズを割り当てられた実施形態は、2 つの符号語を同時に処理する。以下同様である。もちろん、組み込まれるメモリのサイズを増やすと、それに対応して、ハードウェアおよび電力の点でより高価になるが、所定の時間内により多くの数の符号語を処理することができるので、それに対応して復号プロセスのスピードが速くなり、且つ待ち時間が少なくなるという利益が得られる。ハンドシェイク信号によりプロセスが完

40

50

全く自動化され、それにより、ユーザは、復号器が使用するメモリ・サイズを入力するだけでよい。この単一のパラメータを制御することにより、ユーザは、スピード、電力およびサイズ（ゲート数）の点で復号器の性能をカスタマイズする。

#### 【0049】

図8は、本発明の原理によるブロック内ハンドシェイク処理を用いた復号器に関する効率を示す、RS（リード・ソロモン）復号器の例のタイミング図である。タイミング図の上側部分810に、ブロック内ハンドシェイク処理を用いない典型的な復号器における処理に必要な時間を示す。この方式では、CW1（Code Word 1：符号語1）の処理は時間t<sub>0</sub>から始まる。CW1のシンドロームは時間t<sub>1</sub>で生成され、その結果は誤り多项式ブロックに渡される。誤り多项式ブロックは時間t<sub>2</sub>でCW1の処理を完了し、以下同様に行われ、最後に時間t<sub>4</sub>でCW1の処理が完了する。従って、CW2（Code Word 2：符号語2）の処理は時間t<sub>4</sub>から始まる。CW2に対するこの単一機能逐次処理は、CW2に対する誤り位置および誤り度合計算が完了する時間t<sub>8</sub>まで続く。ブロック内ハンドシェイク処理を用いないと、2つの符号語を処理するのに、全部で(t<sub>8</sub>-t<sub>0</sub>)単位の時間が掛かることに留意されたい。

#### 【0050】

タイミング図の下側部分820に、ブロック内ハンドシェイク処理を用いた復号器における処理時間を示す。3つの符号語に用いるのに十分な大きさのメモリを実現することにより、この復号器は、一度に3つの符号語を同時に処理できることに留意されたい。各ブロックは、準備が整っていることを前のブロックに伝えるので、各ブロックは処理を終了した時点より多くのデータを受信することができるため、このように効率的に行われる。図に示した例では、CW3（Code Word 3：符号語3）の処理は、t<sub>6</sub>～t<sub>7</sub>の間のどこかの時点で完了する。

#### 【0051】

##### 誤り訂正復号器内の二重チェン・サーチ・ブロック

本発明の実施形態の一例では、二重チェン・サーチ・ブロックを用いて誤り訂正プロセスを実施する。本発明のこの実施形態の原理を利用する復号器により、復号器の待ち時間および復号器の記憶／メモリ要件が共に低減する。

#### 【0052】

図9は、本発明の原理による二重チェン・サーチ・ブロックを使用するRS（リード・ソロモン）復号器の実施形態の例を示すブロック図である。入力は受信した符号語であり、これをシンドローム生成ブロック910に転送する。シンドローム生成ブロック910によりシンドロームS(x)が計算されると、このシンドロームは、ユークリッド・アルゴリズム・ブロック920に転送される。ユークリッド・アルゴリズムを用いて、シンドロームS(x)を処理して、誤り位置多项式(x)および誤り度合多项式(x)を生成する。

#### 【0053】

本発明のこの実施形態では、符号語中の誤りの数を決定するために新しいチェン・ブロック930が組み込まれる。誤り位置多项式(x)が計算されると、チェン・ブロック930は、チェン・サーチ・アルゴリズムを適用して根の値を求める。チェン・サーチは、全ての可能な入力値に対して多项式の値を求める総当たり式アルゴリズムであり、次いで、どの出力がゼロに等しいかを決める。位置iで誤りが生じる場合、次式がゼロに等しくなる。

#### 【0054】

#### 【数6】

$$\sum_{j=0}^i A_j \alpha^{-j} = 0, \quad \text{ただし、 } i = 0 .. (n-1) \quad \text{式6}$$

チェン・ブロック930により、全てのiおよびjの値について上式の値が求まり、こ

10

20

30

40

50

の式がゼロに等しくなる回数が数えられる。得られた数は、検出された誤りの数である。

#### 【0055】

前に説明したように、誤り位置および誤り度合の両方についてこの誤り多項式の値を求めるなければならない。チェン・サーチおよびフォーニー・アルゴリズムと共に実施するには同じタイプのハードウェアが必要なので、チェン／フォーニー・ブロック940として示すように、これら2つの機能を同じブロック内で組み合わせることができる。この実施形態では、チェン・サーチの出力のところで2つの加算器を用いる。第1加算器は偶数ステージの値を合計し、他方の加算器は奇数ステージの値を合計する。最終的なチェン・サーチ出力を形成するには、次いで、これら2つの加算器の出力を合計し、ゼロ検出ブロックが根の位置を検出する。フォーニー・アルゴリズムでは、奇数ステージ用加算器の出力も用いる。奇数ステージの合計は、フォーニー方程式の分母を表す。この合計値の逆数をとり、次いで、誤り度合多項式の値を求ることにより形成された分子の値を掛ける。この出力とゼロ検出出力のアンド(AND)演算を行う。というのは、これらの誤りの値は、実際の誤り位置でしか有効でないからである(そうでない場合には、誤りの値はゼロにすべきである)。

#### 【0056】

このように、チェン／フォーニー・ブロック940は、誤り位置多項式および誤り度合多項式を用いて誤りベクトルを生成し、それを誤り訂正ブロック950に転送する。この誤りベクトルは、符号語と同じサイズであり、誤りに対応する位置で非ゼロ値を含んでいる。他の全ての位置にはゼロが入る。誤り訂正ブロック950への別の入力は遅延ブロック960の出力である。遅延ブロック960は、受信した符号語を取り込み、この受信した同じ符号語を遅延させて出力する。この実施形態では、誤り訂正ブロック950は、LIFOブロック952と共にガロア体加算器954を使用して実施する。受信した符号語中の誤りは、ガロア体加算器954を使用して、この受信した符号語に誤りベクトルを加えることにより訂正される。LIFOブロック952を使用するのは、誤りベクトルが受信した符号語とは逆の順序で生成され、従って、受信した符号語または誤りベクトルの何れかにLIFO演算を適用して、両方のベクトル中のバイトの順序を整合させなければならないからである。誤り訂正ブロック950の出力は、復号器による元の符号語の推定値である。

#### 【0057】

リード・ソロモン復号器は、 $t$ 個の誤りまでしか訂正することができない。ただし、含まれるパリティ・バイトの数は $2t$ である。検出された誤りの数が $t$ 個よりも多い場合には、一般に、この受信した符号語を変更せず且つ訂正せずに復号器の出力に転送することが望ましい。というのは、検出された誤りの数が $t$ 個よりも多い場合、この符号語は訂正不可能だからである。従来技術のチェン／フォーニーの実施形態では、誤りの数を決定することが可能になる前に、符号語全体を処理しなければならない。そのため、この操作にはNクロック・サイクルが掛かる。ただし、Nは符号語の長さである。チェン／フォーニーの計算が完了すると、誤りの数が決定され、値 $t$ と比較される。誤りの数が $t$ 以下の場合、チェン／フォーニー・ブロックからの誤りの値は、受信した符号語とのXOR(排他的論理和)演算にかけられて誤りが訂正され、それにより最終的な復号器出力が生成される。しかし、誤りの数が $t$ よりも多い場合、受信した符号語は、変更されずに復号器出力として転送される。従って、従来技術の方法では、チェン／フォーニー・ブロックによる処理が完了するまで、受信した符号語全体を記憶する必要がある。このような方式では、待ち時間による遅延も生じる。というのは、このブロックのNクロック・サイクル全体が完了するまで、復号器の出力を転送することができないからである。

#### 【0058】

対照的に、本発明のこの実施形態では、符号語をチェン・アルゴリズムに2回かける。1回目で誤りの数を決定し、2回目で誤りの位置を決定する。従って、チェン・ブロック930が、誤りの数が閾値 $t$ よりも多いことを検出すると、この受信した符号語は復号器出力として渡され、メモリが解放される。逆に、誤りの数が閾値 $t$ 以下であるときには、

このデータは、チェン／フォーニー・ブロックに渡されて、実際の誤り位置および度合が決定される。このような実施形態は、図6～図8に示す本発明のハンドシェイク処理の実施形態と共に用いると最適であろう。このような組合せにより、チェン／フォーニー・ブロック940が誤りを計算しながら、チェン・ブロック930が次の符号語の誤りの数を決定する。従って、チェン／フォーニー・ブロック940の出力を直ちに送出し、それに対応するメモリを解放する。

#### 【0059】

誤り訂正復号器用に強化されたチェン・サーチ・セル

前に説明したように、チェン／フォーニー・ブロックは、入力として、誤り位置および誤り度合多項式という2つの多項式を受信し、出力として誤りベクトルを生成する。この誤りベクトルは、受信した符号語中の誤りの復号器の推定値を表すNバイトのベクトルである。この誤りベクトルを、受信した符号語とのXOR（排他的論理和）演算にかけて誤りを訂正し、それにより復号器による元の符号語の推定値が形成される。誤りベクトルを決定するためにチェン／フォーニー・ブロックを使用する従来型復号器は、符号語ベクトルに対して逆の順序で誤りベクトルを生成する。即ち、符号語および誤りベクトルは、誤りベクトルに（または逆に符号語に）更なる処理をかけずにXOR演算にかけることができない。一般に、この更なる処理は、誤りベクトルまたは受信した符号語の何れかに対する（最後に入力されたものが最初に出力される）LIFO操作の形式をとる。これは、入力の順序を逆にして、符号語および誤りベクトルをXOR演算にかけることができるようになるためである。残念ながら、このために、Nクロック・サイクルの遅延が生じる。ただし、Nは符号語のバイト数である。更に、一部の従来方式の実施形態では、1符号語長よりも大きいメモリ・サイズを用いるので、このメモリを2つ以上の別々のLIFOブロックに分割する必要がある。そうすると、アドレス指定が複雑になり得て、メモリ・サイズも符号語長の整数倍に制限される。

#### 【0060】

本発明の実施形態の一例は、符号語に対して正しく同期がとれている出力を生成する新しいチェン・サーチ・セルである。即ち、誤りベクトルも符号語ベクトルもXOR演算にかける前に逆にする必要がなく、そのため、LIFOブロックその他の反転手段が不要になる。従って、Nクロック・サイクルの遅延の必要がないので、待ち時間がより短くなる。更に、本発明のこの実施形態では、（従来技術に比べて）アドレス指定方式が簡略化され、符号語長の単なる整数倍以外のメモリ・サイズがサポートされる。例えば、符号語長の2.5倍のメモリを使用することができ、それにより、復号器が2つの符号語を処理する間に、次の符号語の一部を取り込むことができる。

#### 【0061】

図10は、チェン・サーチおよびチェン／フォーニー・ブロックの基本的な構成単位である典型的な従来技術のチェン・サーチ・セル1000を示すブロック図である。このチェン・サーチ・ブロックを用いて、誤り位置多項式 の値を求め、それによりその根を見つける。この根の位置は、受信した符号語中の誤りのある位置に対応する。図10のチェン・サーチ・セルを用いて次式を実施する。

#### 【0062】

#### 【数7】

$$x_i = \sum_{j=0}^t A_j \alpha^{-ij}, \quad \text{ただし, } i = 0 .. (N-1) \quad \text{式7}$$

式7を計算すると、符号語中の誤りに対応するバイト位置でヌル（ゼロ）値が得られる。チェン・サーチ・セル1000は、上記の式の1つのステージを実施するのに用いるハードウエア・アーキテクチャである。各ステージでは、誤り位置多項式 の1つの係数を処理するので、全部で( t + 1 )個のステージがある（というは、誤り位置多項式の最大次数はtであり得、次数tの多項式は( t + 1 )個の係数を有するからである）。

10

20

30

40

50

## 【0063】

チェン・サーチ・セル1000は、誤り位置多項式の係数を反復して処理する。最初の反復で、マルチプレクサ1020は、処理中の符号語に対応する適切な多項式係数1010を受信し、この多項式の係数をレジスタ1030に送る。後続のクロック・サイクルで、このレジスタの出力は、まず乗算器1040に転送され、そこで $\cdot^j$ が掛けられ、次いで、マルチプレクサ1020によりレジスタに戻されて記憶される。このプロセスを全部でNクロック・サイクルに亘って実施する。再度、式7を参照すると、指数iは反復数を示し、指数jはチェン・サーチ・セルのステージを示す。即ち、jの値はゼロからtまで増え、そのため、ハードウェア内で実施されるチェン・サーチ・セル・ステージは全部で(t+1)個ある。

10

## 【0064】

例として、このセルは、第2ステージ(j=1)を表すものと仮定する。即ち、各クロック・サイクルで、レジスタ出力には $\cdot^1$ が掛けられ、その結果をレジスタに戻して記憶する。こうすると、以下の数列が生成される。

$$+ \cdot^1 X + (-^1) \cdot^1 X^2 + ((-^1) \cdot^1) \cdot^1 X^3 + \dots$$

ただし、 $X^n$ はnクロック・サイクルの遅延を表す。

## 【0065】

各項を合成することにより、最終的な数列は次式のようになる。

$$+ \cdot^1 X + \cdot^2 X^2 + \cdot^3 X^3 + \dots + \cdot(N-1) X^{N-1}$$

20

## 【0066】

この実施形態に関連する問題は、誤り位置が、対応する符号語のバイトと逆の順序で生成され、そのため、誤り位置を符号語に加えて訂正が行われる前に、LIFOブロックにより逆にする必要があることである。LIFOブロックは、複数のレジスタを含むハードウェア記憶要素である。全てのレジスタが一杯になると、LIFOブロックはその出力を送出する。入力時の最後の要素が出力時には最初の要素になり、以下同様に続く。LIFOブロックを用いると2つの問題が生じる。1つは、LIFOブロックが大規模な記憶/メモリ素子であり、そのため、ICのゲート数および電力消費量が増えることである。別の問題は、LIFOブロックによりNクロック・サイクルの待ち時間が生じることである。この待ち時間が生じるのは、LIFOブロックが最初に一杯になるのにNクロック・サイクル掛かり、最後の要素がLIFOに読み込まれるまで出力を生成することができないからである。

30

## 【0067】

図11は、本発明の原理に従ってメモリ要件および遅延が共に低減された改良型チェン・サーチ・セル1100を示すブロック図である。本発明のこの実施形態の基礎となる原理は、チェン・サーチ・ブロックに、係数が逆の順序(以下では、「正規化された順序」と称する)で生成されるにも関わらず、標準チェン・サーチにより生成されるものと数学的に等価な数列を生成させることができることである。この実施形態では、正規化された順序で誤り位置を生成して符号語を訂正するのに用いる以下の2つの式(式8および式9)を実施する。

40

## 【0068】

## 【数8】

$$x_i = \sum_{j=0}^t A_j \alpha^{-j(N-1)} \quad \text{ただし, } i=0 \quad \text{式8}$$

## 【0069】

## 【数9】

$$x_i = \sum_{j=0}^t A_j \alpha^j \quad \text{ただし, } i = 1 .. (N-1) \quad \text{式9}$$

## 【0070】

図10の従来技術のチェン・サーチ・セル1000では、誤り位置多項式の係数を、各クロック・サイクル毎に減らす のベキ指数を「ゼロ」から始め、- j ( N - 1 ) の値まで減らして反復的に処理することを想起されたい。図11のチェン・サーチ・セル1100の実施形態では、 のベキ指数の値を - j ( N - 1 ) から始めて、ベキ指数がゼロに達するまで各クロック・サイクル毎に増やす。

## 【0071】

従って、最初のクロック・サイクルの際に、前段乗算器1150により多項式の係数1110に - j ( N - 1 ) が掛けられ(ただし、jはステージ番号)、それがマルチプレクサ1120に転送され、レジスタ1130に送られて記憶される。後続のクロック・サイクルで、レジスタ1130の出力は乗算器1140に転送され、そこで  $\alpha^j$  が掛けられ、次いで、マルチプレクサ1120によりレジスタに戻される。

## 【0072】

例として、このセルは、第2ステージ(  $j = 1$  )を表すものと仮定する。こうすると、以下の数列が生成される。

$$\begin{aligned} & - (N - 1) X + ( \dots - (N - 1) ) X^2 + ( ( \dots - (N - 1) ) ) \\ & X^3 + \dots \end{aligned}$$

各項を合成すると、以下の数列が生成される。

$$\begin{aligned} & - (N - 1) + - (N - 2) X + - (N - 3) X^2 + \dots + - 1 \\ & X^{N-2} + X^{N-1} \end{aligned}$$

## 【0073】

この数列中で生成される係数は、図10の標準チェン・サーチ・セル1000により生成されるものと、それらがこの時点では(遅延係数が逆の順序で示されるように)逆の順序である点を除いて同じであることに留意されたい。図11のチェン・サーチ・セル1100は、追加のガロア体乗算器(1つではなく2つ)を用いるが、これらは、ハードウェアで実施するのに小規模且つ安価なものである。反転させるのにLIFOブロックがもはや不要であることと、LIFOを一杯にするのに必要とされる遅延が回避されることから得られる利点および有用性は、各チェン・サーチ・セル毎に追加のガロア体乗算器を供給することの欠点を補って余りあるものである。

## 【0074】

单一ブロック内でチェン・サーチをフォーニー・アルゴリズムと組み合わせることは、これらのアルゴリズムが共に類似の機能の実行を必要とすることから、一般的の設計実務慣行となっている。図11に即して示し説明される本発明のこの実施形態の原理は、フォーニー・ブロック・セルにも同様に適用される。というのは、フォーニー・アルゴリズムが極めて類似したハードウェアを使用して誤り度合多項式を処理するからである。本発明の原理を、フォーニー・ブロック・セルおよびチェン/フォーニー・ブロック・セルの設計に拡張することができることは当業者には周知のことであろう。

## 【0075】

多くの既存のシステムは、リード・ソロモン符号を符号化し復号する「既成」の集積回路を使用する。これらのICは、ある分量のプログラム可能性(例えば、RS(255, k))。ただし、tは1~16個のシンボルを表す)をサポートする傾向がある。最近の傾向は、VHDL即ちベリログ(Verilog)設計(論理コアまたは知的財産コア)に向かいつつある。こうすると、標準ICに比べて多くの利点がある。論理コアを、他のVHDL即ちベリログ・コンポーネントと一体化し、FPGA(現場で書き込み可能なゲート・アレイ)またはASIC(特定用途向け集積回路)に統合することができ、これにより

10

20

30

40

50

、複数のモジュールを単一 I C に組み合わせることができるいわゆる「システム・オン・チップ」設計が可能になる。生産量によりは、論理コアにより、「標準」I C よりもシステム・コストをしばしば大きく下げることができる。

#### 【0076】

ハードウェアによる実施形態の状況で本発明を説明しているが、本発明の原理をそのように限定されると解釈すべきではない。最近まで、「リアルタイム」でのソフトウェアによる実施形態は、最も簡単なリード・ソロモン符号（即ち、 $t$  の値が小さい符号）以外は全てあまりにも多くの計算能力を必要とした。ソフトウェアでリード・ソロモン符号を実施する際の主要な難点は、汎用プロセッサが、ガロア体算術演算をサポートしていないことである。例えば、ソフトウェアでガロア体乗算を実施するには、ゼロの検査、2つの対数テーブル索引、モジュロ加算および真数テーブル索引を必要とする。しかし、注意深く設計するとプロセッサ性能の増加とあいまって、ソフトウェアによる実施形態は、比較的高速なデータ・レートで動作し得る。

#### 【0077】

例示の実施形態および／または構成に関して本発明を説明してきたが、本発明は、この開示の趣旨および範囲内で更に改変することができる。従って、本出願は、本発明の一般原理を用いた本発明の任意の変形形態、使用形態または適応形態を包含するためのものである。更に、本出願は、本発明に関連し、の特許請求の範囲の制限に含まれる、当技術分野において周知の、または実務上の慣行に収まるようなこの開示からの派生物を包含するためのものである。

#### 【図面の簡単な説明】

#### 【0078】

【図1】誤り訂正方式を組み込んだディジタル・データ伝送システムを示すブロック図である。

【図2】典型的な誤り訂正方式の方法を示す流れ図である。

【図3】様々な誤り訂正方式を示す階層図である。

【図4】RS（リード・ソロモン）復号器を示すブロック図である。

【図5】本発明の原理に従って用いられるRS（リード・ソロモン）復号器の実施形態の例を示すブロック図である。

【図6】本発明の原理による復号器内ハンドシェイク・プロトコルの例を示すブロック図である。

【図7】本発明の原理によるRS（リード・ソロモン）復号器の実施形態の例の機能ブロック間のハンドシェイク・プロトコルを示すブロック図である。

【図8】本発明の原理によるブロック内ハンドシェイク処理を用いた復号器に関する効率を示す、RS（リード・ソロモン）復号器の例のタイミング図である。

【図9】本発明の原理によるチェン／フォーニー・ブロックに加えて、チェン・ブロックを使用するRS（リード・ソロモン）復号器の実施形態の例を示すブロック図である。

【図10】RS（リード・ソロモン）復号器で実施する典型的な従来技術のチェン・サーチ・セルを示すブロック図である。

【図11】本発明の原理に従ってメモリ要件および遅延が共に低減されたRS（リード・ソロモン）復号器で実施する改良型チェン・サーチ・セルを示すブロック図である。

10

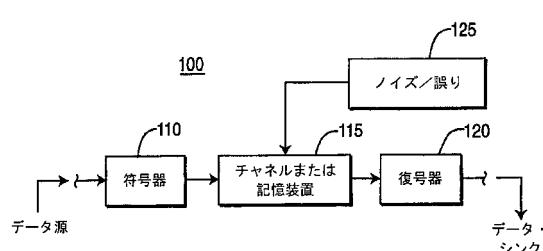
20

20

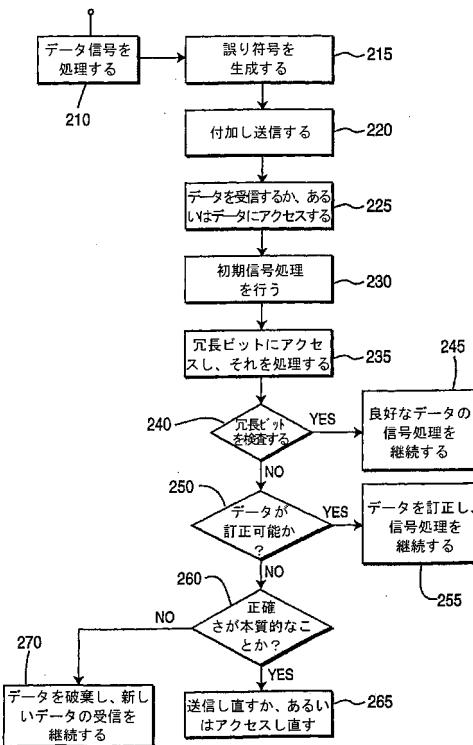
30

40

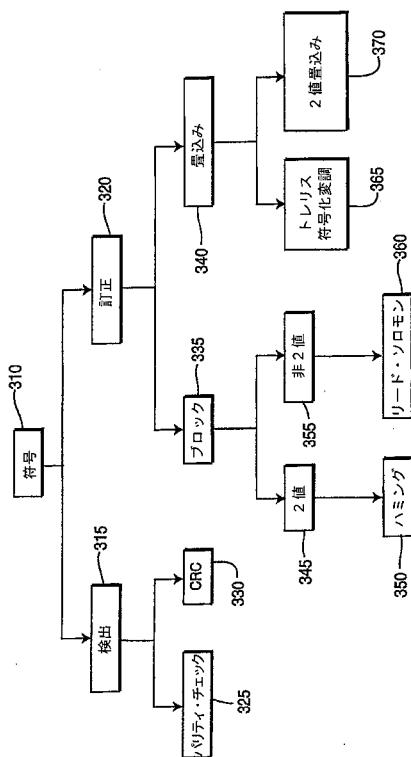
【図1】



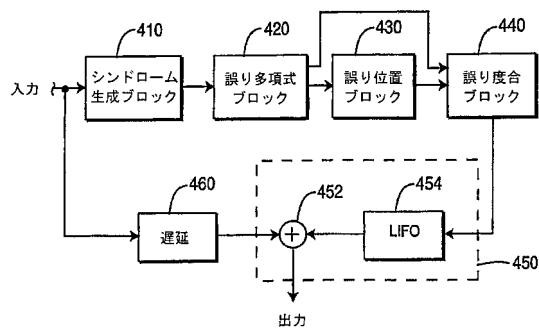
【図2】



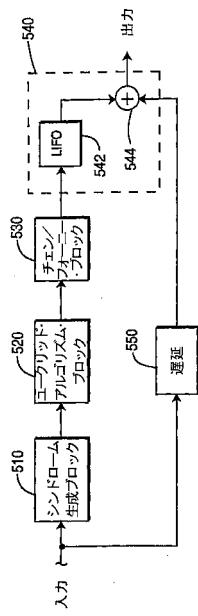
【図3】



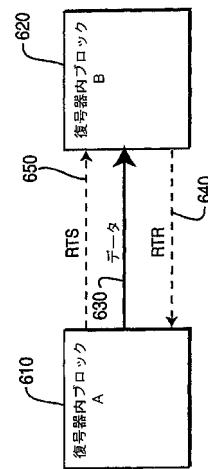
【図4】



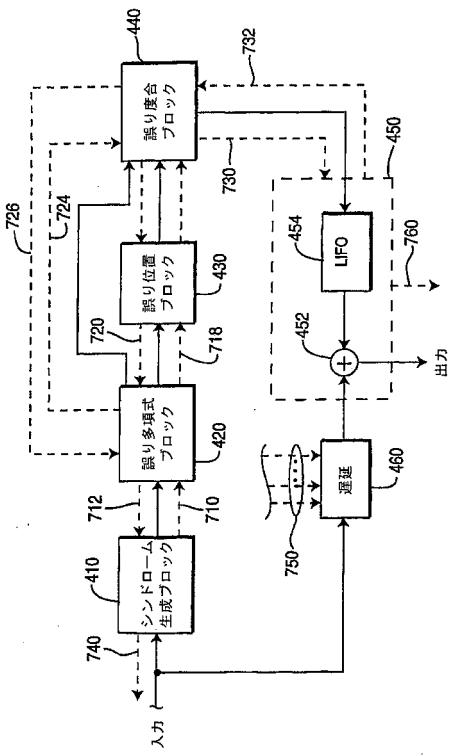
【図5】



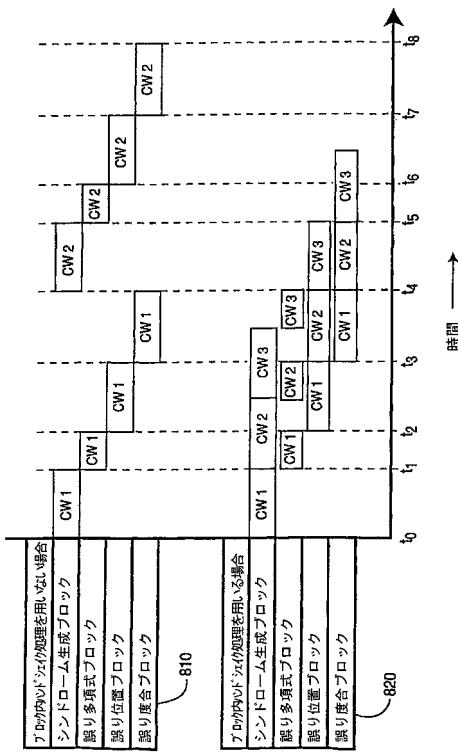
【図6】



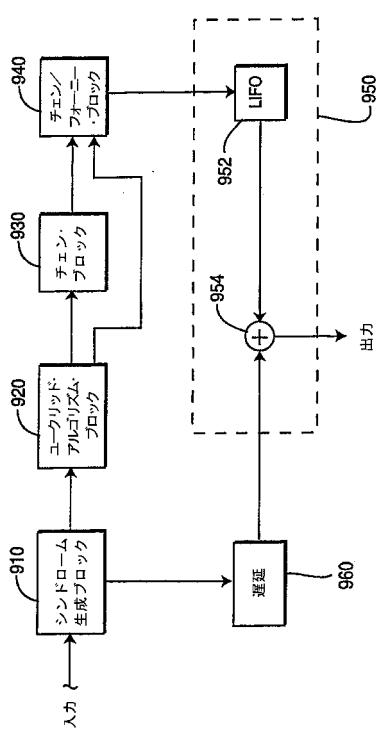
【図7】



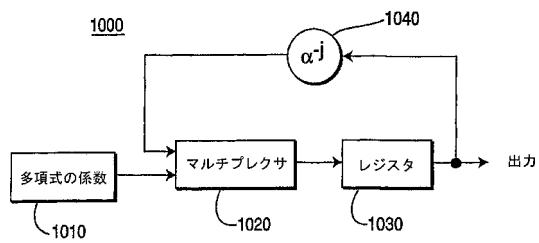
【図8】



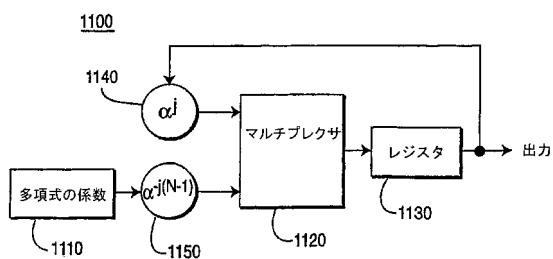
【図9】



【図10】



【図11】



## 【国際調査報告】

WO 03/063363  
INTERNATIONAL SEARCH REPORT

PCT/US03/01913

International application No.

PCT/US03/01913

<b>A. CLASSIFICATION OF SUBJECT MATTER</b>		
IPC(7) :H03M 13/00 US CL :714/781, 784 According to International Patent Classification (IPC) or to both national classification and IPC		
<b>B. FIELDS SEARCHED</b>		
Minimum documentation searched (classification system followed by classification symbols) U.S. : 714/781, 782, 783, 784, 785, 793		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) WEST search terms: error polynomial, error locator, error magnitude, euclid algorithm, berlekamp-massey algorithm		
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P	US 6,415,413 B1 (PAN et al.) 02 July 2002, col. 4, line 46 - col. 5, line 6.	1-11
A,P	US 6,374,383 B1 (WENG) Figure 1 and col. 3, lines 1-59.	1-11

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

* Special categories of cited documents:	"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	
"A"	document defining the general state of the art which is not considered to be of particular relevance	"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
"E"	earlier document published on or after the international filing date	"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
"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)	"&"	document member of the same patent family
"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		

Date of the actual completion of the international search  23 MARCH 2003	Date of mailing of the international search report  15 APR 2003
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer  EMMANUEL J. MOISE Telephone No. (703) 305-9763

---

**フロントページの続き**

(81)指定国 AP(GH,GM,KE,LS,MW,MZ,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AT, BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HU,IE,IT,LU,MC,NL,PT,SE,SI,SK,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN, GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AT,AU,AZ,BA,BB,BG,BR,BY,BZ,CA,CH,CN,CO,CR,CU,CZ,DE,DK,DM,DZ,EC, EE,ES,FI,GB,GD,GE,GH,GM,HR,HU,ID,IL,IN,IS,JP,KE,KG,KP,KR,KZ,LC,LK,LR,LS,LT,LU,LV,MA,MD,MG,MK,MN,MW, M X,MZ,NO,NZ,OM,PH,PL,PT,RO,RU,SC,SD,SE,SG,SK,SL,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,YU,ZA,ZM,ZW

(72)発明者 ベレツ , デイデイエ

アメリカ合衆国 インディアナ州 カーメル ナンスモンド・ドライブ 13970

F ターム(参考) 5B001 AA08 AA11 AB02 AB05 AC01

5J065 AC02 AD11 AE06 AF02 AG01 AG03 AG04 AH09 AH15 AH17

5K014 AA01 BA08

**【要約の続き】**

誤り訂正ブロック(950)に伴う遅延およびコストを低減する、強化型チェン・サーチ・セル・アーキテクチャを説明する。