

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3839215号
(P3839215)

(45) 発行日 平成18年11月1日(2006.11.1)

(24) 登録日 平成18年8月11日(2006.8.11)

(51) Int.C1.

F 1

G06F	11/10	(2006.01)	G06F	11/10	330D
G06F	3/06	(2006.01)	G06F	3/06	305A
G06F	12/16	(2006.01)	G06F	12/16	320F
H03M	13/01	(2006.01)	H03M	13/01	

請求項の数 7 (全 16 頁)

(21) 出願番号 特願2000-70824 (P2000-70824)
 (22) 出願日 平成12年3月14日 (2000.3.14)
 (65) 公開番号 特開2001-256068 (P2001-256068A)
 (43) 公開日 平成13年9月21日 (2001.9.21)
 審査請求日 平成16年3月30日 (2004.3.30)

(73) 特許権者 000005108
 株式会社日立製作所
 東京都千代田区丸の内一丁目6番6号
 (74) 代理人 100080001
 弁理士 筒井 大和
 (72) 発明者 田中 剛
 東京都国分寺市東恋ヶ窪一丁目280番地
 株式会社日立製作所 中央研究所内
 審査官 久保 正典
 (56) 参考文献 特開平07-129466 (JP, A)
 特開平02-239729 (JP, A)
 特開平08-154084 (JP, A)
 特開平06-161793 (JP, A)
 最終頁に続く

(54) 【発明の名称】誤り検出・訂正方法、計算機システムの主記憶制御装置、及び計算機システム

(57) 【特許請求の範囲】

【請求項1】

c (c > 0) ビット誤り訂正・d (d > c) ビット誤り検出符号を使用した情報通信システム、あるいはメモリシステムの誤り検出・訂正方法において、

送信対象データを符号化する以前に発生した障害の有無、及びこの障害の発生原因を示す障害情報と、前記送信対象データとから検査ビットを生成して前記障害情報、送信対象データ並びに検査ビットから成る符号語を生成するステップと、

前記符号語から前記障害情報を除いた符号列を通信路もしくはメモリに送出するステップと、

前記通信路もしくはメモリから受信した符号語について、除かれた前記障害情報に当たる部分を固定値であると仮定した符号列からシンドロームを生成し、該シンドロームのパターンに基づき、前記符号列の誤り検出・訂正を行い、もって前記障害情報を再現するステップとを有することを特徴とする誤り検出・訂正方法。

【請求項2】

プロセッサと、主記憶と、誤り制御符号化されたデータを格納する主記憶制御装置と、I/O装置と、複数の前記プロセッサと前記主記憶制御装置を接続するプロセッサバスと、前記主記憶と前記主記憶制御装置を接続するメモリバスと、前記I/O装置と前記主記憶制御装置を接続するI/Oバスを備える計算機システムにおいて、前記主記憶制御装置は、

前記プロセッサバスから転送された前記主記憶への書き込み対象データの誤りを検出する第1の誤り検出回路と、

前記 I O バスから転送された前記主記憶への書き込み対象データの誤りを検出する第 2 の誤り検出回路と、

前記書き込み対象データについて前記第 1 もしくは第 2 の誤り検出回路で訂正不能な誤りを検出したか否かをそれぞれの符号パタンで示す障害情報と、該書き込み対象データとから成る符号列から検査ビットを生成し、該障害情報を除いて該書き込み対象データと該検査ビットとから成る符号列を前記主記憶に書き込む符号語として出力する符号化回路と、

前記主記憶から読み出した符号語について、除かれた前記障害情報に当たる部分が固定のパタンであると仮定した符号列からシンドロームを生成し、該シンドロームのパタンに基づき、前記符号列の誤り検出・訂正を行い、もって前記障害情報を再現する復号回路と、

前記復号回路で再現された障害情報の符号パタンを検出して前記符号化回路による符号化以前に発生した訂正不能な誤りの発生源を識別する障害情報検出テーブルとを備えることを特徴とする計算機システムの主記憶制御装置。

【請求項 3】

請求項 2 記載の計算機システムの主記憶制御装置において、前記主記憶制御装置を他の主記憶制御装置と接続する場合には、前記主記憶制御装置とクロスバスイッチを接続するためのクロスバスイッチ入出力制御と、

前記クロスバスイッチから転送されたデータの誤りを検出する回路とを備えることを特徴とする計算機システムの主記憶制御装置。 20

【請求項 4】

請求項 2 または 3 記載の計算機システムの主記憶制御装置において、
前記誤り制御符号化に SEC - D ED 符号の短縮化符号を使用し、この SEC - D ED 符号のパリティ検査行列から削除した列に相当する複数のビットのうちの特定のビットを、前記プロセッサバスまたは前記 I O バスまたは前記クロスバスイッチからの転送データの訂正不能誤りを検出したことを意味する障害情報ビットに割り当て、前記符号化回路で前記主記憶に書き込むデータと共に符号化し、前記主記憶には前記プロセッサバスまたは前記 I O バスから転送された前記主記憶への書き込みデータと検査ビットのみを書き込み、復号時には前記プロセッサバスの障害情報ビットを 0 として復号し、前記プロセッサバスあるいは前記 I O バスあるいは前記クロスバスイッチからの転送データに障害情報が発生したデータの場合は障害情報ビットの 1 ビットエラーとして検出し、障害情報ビットが 1 である符号語に、更にもう 1 ビットのエラーが発生した場合も検出することを特徴とする計算機システムの主記憶制御装置。 30

【請求項 5】

請求項 2 または 3 記載の計算機システムの主記憶制御装置において、
前記誤り制御符号化に b ビット ($b > 2$) 単位のバイトで誤り制御をする S b E C - D b E D 符号の短縮化符号を使用し、この S b E C - D b E D 符号のパリティ検査行列から削除した列に相当する複数のビットのうちの特定のビットを、前記プロセッサバスまたは前記 I O バスまたは前記クロスバスイッチからの転送データの訂正不能誤りを検出したことを意味する障害情報ビットに割り当て、前記符号化回路で前記主記憶に書き込むデータと共に符号化し、前記主記憶には前記プロセッサバスまたは前記 I O バスから転送された前記主記憶への書き込みデータと検査ビットのみを書き込み、復号時には前記プロセッサバスの障害情報ビットを 0 として復号し、前記プロセッサバスあるいは前記 I O バスあるいは前記クロスバスイッチからの転送データに障害情報が発生したデータの場合は障害情報ビットの 1 バイトエラーとして検出し、障害情報ビットが 1 である符号語に、更にもう 1 バイトのエラーが発生した場合も検出することを特徴とする計算機システムの主記憶制御装置。 40

【請求項 6】

O S を搭載した計算機システムにおいて、主記憶に記憶されているデータに対して請求項 1 記載の処理をソフトウェアで行い、障害箇所の特定を行うことを特徴とする計算機シス 50

テム。

【請求項 7】

主記憶の内容の採取やエラーログの採取が可能なモニタ装置を持つ計算機システムにおいて、請求項 1 記載の処理をソフトウェアで行い、障害箇所の特定を行うことを特徴とする計算機システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、計算機システムの障害検出方式及びその主記憶制御装置に係り、特にエラー発生時のシステムダウンを回避し、エラー発生源の特定に好適な誤り検出・訂正方法、計算機システムの主記憶制御装置、及び計算機システムに適用して有効な技術に関する。 10

【0002】

【従来の技術】

従来、CPUバスやI/Oバスから主記憶に書き込むデータに訂正不能なエラーを検出した場合においてシステムダウンをさせない方法として、例えば特開平6-89196号公報では、主記憶制御装置がCPUバスやI/Oバスから転送されたデータに訂正不能なエラーを検出した場合、ある受信データを特定のパターンのデータに置き換え、このデータから生成した検査ビットを全て反転させた特定の誤り訂正符号付きデータを主記憶へ書き込んでいる。そして、主記憶からこのデータを読み出したときシンドロームが全1で、特定パターンのデータになっていれば、CPUバスやI/Oバスで訂正不能なエラーとなったデータであると判別する。これにより、障害情報を記憶するために必要な、主記憶へのインターフェース信号数、主記憶のメモリ容量を増加させないで障害情報を記録できる。 20

【0003】

また、CPUバスやI/Oバスでリトライなどの障害回復手段を持たない場合は、障害検出時にリブートさせず、当該データをCPUが読み出そうとしたときだけCPUに割り込みをかけて障害通知をすればよいことになり、障害のあるデータを主記憶に書き込んでもCPUから読み出しがなければデータの障害がシステムの障害（停止やリブート等、ユーザに直接見える障害）を避けることができ、システムのアベイラビリティの向上に有効である。 30

【0004】

【発明が解決しようとする課題】

しかし、前記のような従来方式で提案している符号の構成方法について、本発明者が検討した結果、以下に挙げる3つの問題点が明らかとなった。この問題点とは、（1）検査ビットを反転させれば、シンドロームが全1のパターンになり発生頻度の低い多重エラーのパターンとなるとしていること、（2）この特定の誤り訂正符号付きデータが主記憶でさらに1ビットエラーが発生すると誤訂正する場合があること、（3）特定パターンにデータを書き換えているため元のデータパターンを参照できないことの3点が挙げられる。これらの問題点について具体例を挙げながら説明する。

【0005】

まず、（1）、（2）について凡例を挙げて説明する。ここでは、説明を簡単にするため図16に示す符号長8、検査ビット長4のSEC-DED符号（単一ビット誤り・二重ビット誤り検出符号）を例に挙げる。この符号についての説明は、文献T.R.N.Rao, E.Fujiiwara著、"Error-Control Coding For Computer Systems"のP140に記載されている。 40

【0006】

図16には、パリティ検査行列H（以下、H行列と呼ぶ）と、情報ビット及び検査ビットの配置の例が示してある。H行列の各列ベクトルをh0, h1, ..., h7と呼ぶことにする。図17は（1）の問題点の具体例を示している。図16のd0とc3の位置に合計2ビットの誤りが生じたとする。このときシンドロームSとして全1のパターンが生じる。つまり、符号の作り方によってはシンドロームが全1になるケースが発生頻度の低い多 50

重誤りにはならないということである。

【0007】

次に図18を使って(2)の問題点を説明する。図18(1)に示すように、符号語が[0, 0, 0, 0, 0, 0, 0]のとき、従来方式のやり方で検査ビットを全て反転させ、データ $d = [0, 0, 0, 0, 1, 1, 1]$ を得る。次に図18(2)に示すように1ビットエラーが起き、データが[0, 0, 0, 0, 1, 1, 1, 0]になる。そして、図18(3)に示すように図16のH行列からシンドロームを生成すると、図16のH行列の h_0 という列ベクトルに一致し、 d_0 に1ビット誤りが発生したと判定することになり、誤って[1, 0, 0, 0, 1, 1, 1, 0]と誤訂正してしまう。

【0008】

従来方式を計算機システムで一般的に用いられている誤り制御符号に適用しようとする、(1), (2)の問題が発生する可能性がある。したがって、従来方式は全ての誤り制御符号には適用できず、ある特定のパターンのH行列の誤り制御符号にしか適用できない。しかし、従来例では適用した符号の種類について何の言及もなされていない。

【0009】

次に(3)の問題点について説明する。エラーを検出した時のパターンを幾つか調査すると、例えば特定のビットが0固定故障を起こしているようなパターンを発生する場合もある。このように障害発生時のデータのパターンを残していくことは、エラー原因解析の糸口となる場合もありうるので、エラー発生時のパターンを棄ててしまうのは障害原因の解析時間を延ばしてMTTR(Mean Time To Repair)の増大を引き起す。

【0010】

上記(1), (2), (3)の問題点を解決することが課題となる。さらに、エラー訂正不能な障害を発生した箇所がどこであるか特定できれば、障害回復処理の大幅な削減が見込まれる為、符号のビット数を増加させずにエラー訂正不能な箇所の特定も同時に可能とすることが課題となる。また、この提案方式を実現する装置では既存の符号化、復号化回路に大規模な修正を加えずに実現することも課題として挙げられる。

【0011】

そこで、本発明の目的は、符号化するデータが障害情報である情報をデータの大きさを変えずに符号化することができ、また復号においても誤訂正することなく障害データであると識別することができる誤り検出・訂正方法、計算機システムの主記憶制御装置、及び計算機システムを提供するものである。

【0012】

本発明の前記ならびにその他の目的と新規な特徴は、本明細書の記述および添付図面から明らかになるであろう。

【0013】

【課題を解決するための手段】

本願において開示される発明のうち、代表的なものの概要を簡単に説明すれば、次のとおりである。

【0014】

始めに、前記図16で示した誤り制御符号を用いて本発明の概要を説明する。図16のSEC-DED符号は符号長8ビット、情報ビット長4ビットであり、以降では(8, 4) SEC-DED符号と呼ぶことにする。検査ビット数が4のSEC-DED符号の最大符号長は前記文献のP139から8ビットであることがわかる。誤り制御符号で保護したい情報ビット数が2ビットの場合、図1に示すように情報ビットとして使用しないビット数分だけ列ベクトルを削除し、(6, 2) SEC-DED符号として使用する。このように必要な情報ビット数分だけH行列から切り出した符号を短縮化符号という。本発明では、この削除した列ベクトルに対応する情報ビットに障害情報を配置することが基本的な考え方である。

【0015】

図2の例では、削除した部分に障害情報 e_0 と e_1 を割り当てている。符号語でどのよう

10

20

30

40

50

にビットを配置しているかを図3に示す。通常、障害情報なしで符号化する場合は図3(1)のように障害情報に0を配置する。この場合の符号化、復号の手順を図4に示す。符号化するデータ70とデータ70に障害が発生しているかどうかを示す障害情報71を図2の(8,4)SEC-DED符号のH行列を使って符号化する。符号化とは、具体的には検査ビットc0,c1,c2,c3を生成することである。ここで生成した符号語73のうち、障害情報を除いた[d0,d1,c0,c1,c2,c3]のみを通信路または記憶装置75に送信する。通信路または記憶装置75から受信したデータ76に、固定値[0,0]を障害情報として仮定して図2の(8,4)SEC-DED符号のH行列を使い復号78し、データ79を得る。ここで復号とは、受信した符号語からシンドロームを生成し、そのシンドロームをデコードしてエラーの発生したビット位置を反転することである。ここで重要な点は、障害情報を通信路や記憶装置に送信しないことである。受信側では送信元のデータ70には、いつも障害がないと仮定して復号している点である。

【0016】

障害情報がある場合は、図3(2)の様にe0またはe1に障害情報を配置する。図5に示すように障害情報80を与えて符号化72する。そして、復号78をするときに障害情報として[0,0]を仮定しているため先頭の1ビットが誤っていることになる。したがって、復号78をすることで、データ81が得られ、障害情報が再現されることになる。ただし、この例で使用している符号を使っている限りe0,e1の両方が1となることはできない。なぜならば、ここで使用している符号の誤り訂正能力は1ビットであるから最大1ビットの情報までしか再現できないからである。SEC-DED符号よりさらに強力な誤り訂正能力を持つ符号を使用すれば、一度に転送できる障害情報は増やすことができる。このように、短縮化符号を使用している場合、短縮した部分に障害情報を乗せれば上記(1)の問題点は解決できる。さらに、符号化をする前のデータのパターンもそのまま残っている。通常使用するデータ幅では32ビット、64ビット、128ビットが多いが、それぞれの場合でSEC-DED符号化するときに必要な検査ビット数は、それぞれ7ビット、8ビット、9ビットとなる。この検査ビット数でSEC-DED符号化できる最大のデータ幅は、それぞれ、57ビット、120ビット、247ビットとなり、どの場合も必ず短縮符号を使用しているため本発明の方式をたいていの場合は適用できる。

【0017】

さらに図6では、上記(2)の問題が解決できることを示している。図6において、データ70の障害情報80とともに符号化72をし、通信路または記憶装置75に転送したデータの一部90にエラーが生じたとする。このとき復号するデータ91には92,93の2箇所でエラーが発生していることになる。これは2ビットエラーと同じであるからエラー検出が可能である。

【0018】

上記の例を一般化した場合を図7を使って説明する。図7において、Heは($m - k \times r$)の行列、Hdは($k \times r$)の行列で、Irは($r \times r$)の単位行列である。(He, Hd)は情報ビット部分を、Irは検査ビット部分を表す。図7(a)に示す最大情報ビット長mビット、検査ビット長rビット、最大符号長 $m + r$ ビットの($m + r, m$)t(t>0)ビットエラー訂正、u(u>t)ビットエラー検出符号のH行列を使用すると仮定する(なお、特許請求の範囲ではtをc、uをdとしている)。このとき、使用する情報ビット長をk($m > k > 0$)とする。この符号を使用する場合、最大 $\min(t, m - k)$ ビットの障害情報を符号化することができる。ここで $\min(a, b)$ は $a < b$ で $\min(a, b) = a$ 、 $a > b$ で $\min(a, b) = b$ を意味する。さらに、障害情報をのせた符号に $u - \min(t, m - k)$ ビットのエラーが通信路や記憶装置内で付加されても誤訂正することはない。図7(b)には、(a)のH行列を使用した場合の符号語内のデータ配置について説明している。e0, e1, ..., em-k-1のm-kビットに障害情報を配置し、d0, d1, ..., dk-1のkビットに情報ビットを配置し、c0, c1, ..., cr-1のrビットに検査ビットを配置する。ただし、これらの配置は符号の性質を変えない範囲で場所を入れ替えることが可能である。本発明では、図7の説

10

20

30

40

50

明で示した符号化方式を特徴とする誤り検出・訂正手段を提供するものである。

【0019】

また、符号語と符号語の和は符号語であることを利用して、通信路や記憶装置に転送するデータを誤り制御符号化する場合、符号化する対象であるCPUでの処理に用いるデータとこのデータの障害情報をまとめて検査ビットを生成せず、既存にあるデータ符号化手段と障害情報の符号化手段を別にし、後に線形和をとり符号化することを特徴とする誤り検出・訂正手段を提供する。

【0020】

また、本発明の課題は以下の装置で解決できる。プロセッサと、主記憶と、誤り制御符号化されたデータを格納する主記憶制御装置と、I/O装置と、複数のプロセッサと主記憶制御装置を接続するプロセッサバスと、主記憶と主記憶制御装置を接続するメモリバスと、I/O装置と主記憶制御装置を接続するI/Oバスと、特に他の主記憶制御装置と接続する場合には、主記憶制御装置とクロスバスイッチを接続する為のクロスバスイッチ入出力制御を備える計算機システムの主記憶制御装置において、プロセッサバス上のデータの誤りを検出する回路と、I/Oバス上のデータの誤りを検出する回路と、クロスバスイッチから転送されたデータの誤りを検出する回路と、主記憶へ書き込むデータと、プロセッサバスから転送された主記憶への書き込みデータに訂正不能な誤りを検出した情報、あるいはI/Oバスから転送された主記憶への書き込みデータに訂正不能な誤りを検出した情報から検査ビットを生成する符号化回路と、主記憶から読み出したデータのシンドロームを生成し、このシンドロームのパターンから誤りの検出・訂正をする復号回路と、符号化回路による符号化以前に発生した訂正不能な誤りの発生源を識別するために復号回路で生成されたシンドロームが特定のパターンであるかを検出する障害情報検出テーブルとを備えることを特徴とする主記憶制御装置を設けることで課題を解決できる。10

【0021】

【発明の実施の形態】

以下、本発明の実施の形態を図面に基づいて詳細に説明する。なお、実施の形態を説明するための全図において、同一部材には同一の符号を付し、その繰り返しの説明は省略する。20

【0022】

本発明の方式の一実施の形態として図8の(112, 104)SEC-DED-S4ED(単一4ビットのブロックを検出可能なSEC-DED符号)を使用したシステムを図面を参照しながら説明する。SEC-DED-S4EDは主記憶用メモリ素子の読み出し幅が4ビットであるようなシステムに多く使用され、单一のメモリ素子故障を検出できる。この符号については前記文献のP280-283を参照することができる。本実施の形態では、システムで使用しているデータの幅(CPUなどが実際に処理に使用するデータの幅)を64ビットと仮定する。このとき、本発明の方式ではなく通常のエラー制御方式では(72, 64)SEC-DED-S4ED符号を用いる。本発明の方式では、単一ビット誤り訂正の能力の符号を使用するので、短縮した40ビットの部分に1ビットづつ排他的に使用して、最大40種類の障害情報を符号化することができる。ここでは、先頭の4ビットe0, e1, e2, e3に障害情報をのせる。30

【0023】

図9に障害情報とシンドロームの組み合わせ表を示す。図9では、H行列のe0の位置にCPUバスの訂正不能エラーという障害情報を配置することを意味する。本発明で使用する誤り制御符号は、SEC-DED符号(単一ビット誤り・二重ビット誤り検出符号)、あるいはSEC-DED符号に機能を追加した符号のみならず、多重ビット誤り訂正の機能を持つ符号や、b(b>1)ビットのブロック(=バイト)の誤り訂正をするSbED-D_bED符号(单一バイト誤り訂正・二重バイト誤り訂正符号)といった誤り制御符号を使用するほとんどの場合に適用することが可能である。以下に示す符号化、復号の手順、及び装置例もほとんど同じ方法で実施することができる。

【0024】

10

20

30

40

50

また、本発明の方式の処理の流れを図10を使用して説明する。図10の入力データ1と障害情報入力2を入力とする符号化回路5と、通信路または記憶装置7から受信した符号語8を入力とし、誤りを訂正したデータ15と誤りを検出したことを示す誤り検出出力16と入力データ障害情報出力17を出力とする復号回路9を設け、復号回路9には、通信路または記憶装置7から受信した符号語8からシンドロームを生成するシンドローム生成回路10と、シンドローム11から誤りの発生した位置や訂正不能な誤りの検出、及び入力データ1とともに符号化された障害情報入力2を識別するシンドロームデコード回路12と誤りの位置情報13から通信路または記憶装置7から受信した符号語8の誤りを訂正し、訂正したデータ15を出力する訂正回路14を設けることで実現される。

【0025】

符号化の手順を図11に示す。(a)方法1と(b)方法2の手順があり、ここでは図7のH行列、データ構造を使うとする。(a)の手順を説明する。まず、符号化するデータに付随する障害情報をエンコードし、ビット列に変換する(S1)。次に、符号化する入力データと障害情報を一つのデータとして纏め(S2)、H行列に基づき検査ビットを生成して(S4)、符号化が完了する。次に(b)の手順を説明する、障害情報のエンコードと、入力データを全て0としたデータを作成し(S4)、H行列に基づき検査ビットを生成する(S5)。手順S4、S5と平行して、障害情報が全て0と仮定して入力データだけのデータを作成し(S6)、H行列に基づき検査ビットを生成する(S7)。次に、手順S5と手順S7で作成した検査ビットの線形和を行う(S8)ことで符号化が完了する。(b)方法2は符号語と符号語の線形和は符号語である性質を利用している。(b)方法2では、既存の符号化手段に相当するS6、S7を変更せず、手順S4、S5を追加する方法である。

【0026】

復号の手順を図12に示す。復号する符号語とH行列の転置との積をとりシンドロームSを生成する(S11)。次に、シンドロームSが全0であればエラーは存在しないので復号は終了で、シンドロームが非零であれば(S12)、H行列のどこかの列に一致するか(多重誤り訂正の場合複数の列ベクトルの線形和のパタンとなる)調べ、どの列にも一致しない(多重誤り訂正の場合複数の列ベクトルの線形和のパタンと一致しない)場合はエラー検出をして終了となる(S13)。一致した列があれば、その列が障害情報を配置した列となっているかどうか調べる(S14)。障害情報がなければエラー訂正をして(S15)、処理は終了となる。また、障害情報がある場合は、障害原因をシンドロームのパタンからデコード(S16)して、障害情報を報告して終了となる。さらに、障害情報を含んだ符号語にさらにエラーが発生した場合、シンドロームのパタンは手順S13の処理を行い、エラー検出まで可能である。SEC-DED符号、あるいはそれに準ずる機能を持つ符号では、障害情報とさらにもう1ビットのエラーまで同時に発生しても検出できる。

【0027】

本発明を実現する装置の例を図を参照しながら説明する。図13は、本発明の実施の形態に係る計算機システムの全体構成図である。この計算機システムは、ノードN100、N101をクロスバスイッチX100で信号線X101、X102、X103、X104を介して互いのノードの資源をアクセスする。ノードの数は本実施の形態では2個であるが、2個以上の構成であってもよく、複数のノードはクロスバスイッチX100で相互結合されている。

【0028】

ノードN100は、主記憶制御装置C100と、プロセッサB100、B101、プロセッサバスB102、I/O装置A100、A101、A102、A103、I/OバスA104、A105、主記憶m100から構成される。プロセッサやI/O装置やプロセッサバスやI/Oバスの個数は本実施の形態で示した個数に特定されず、任意の数の組み合わせがあってもよい。このプロセッサB100、B101は、プロセッサバスB102を介して主記憶制御装置C100と結合している。主記憶m100は信号線C113、C114で主

記憶制御装置 C 100 と結合している。

【0029】

主記憶制御装置 C 100 は、プロセッサバス制御部 B 103、I/O バス制御部 A 106、クロスバスイッチ出入力制御部 X 105、主記憶制御部 C 110、ログレジスタ C 116 から構成されている。プロセッサバス制御部 B 103 は、信号線 B 105, B 107 で主記憶制御部 C 110 と結合している。またプロセッサバス制御部 B 103 には、プロセッサバス B 102 から転送されてくる主記憶への書き込みデータの誤り検出を行うエラー検出回路 B 104 がある。エラー検出回路 B 104 で誤り訂正不能なエラー（パリティ符号を使用している場合はパリティエラー、SEC-DED 符号を使っている場合は 1 ビットを超えるエラー等）を検出した場合、信号線 B 106 で訂正不能なエラーがあったことを信号線 B 105 を渡る主記憶への書き込みデータと同期して主記憶制御部 C 110 に通知する。

10

【0030】

I/O バス制御部 A 106 には、I/O バス A 104, A 105 から転送されてくる主記憶への書き込みデータのエラー検出をするエラー検出回路 A 107 がある。エラー検出回路 A 107 で主記憶への書き込みデータに訂正不能なエラーがあった場合、信号線 A 108 で転送される主記憶への書き込みデータと同期して訂正不能エラーの存在を信号線 A 109 で主記憶制御部 C 110 へ通知する。

【0031】

主記憶制御部 C 110 では、プロセッサバス制御部 B 103 や I/O バス制御部 A 106 からそれぞれ信号線 B 105, A 108 を通して転送される書き込みデータの書き込みアドレスを解読して、自ノード N 100 か他ノード N 101 宛てのデータか判別して、他ノード N 101 宛ての場合、信号線 X 107 を通してクロスバスイッチ出入力制御部 X 105 へ、クロスバスイッチ出入力制御部 X 105 から信号線 X 102 とクロスバスイッチ X 100 と信号線 X 104 を経由して他ノード N 101 へ転送する。

20

【0032】

クロスバスイッチ出入力制御部 X 105 には、他ノード N 101 からの転送されたデータのエラー検出をするエラー検出回路 X 108 があり、訂正不能なエラーがある場合は、データを転送する信号線 X 106 と同期して信号線 X 109 で訂正不能エラーがあったことを主記憶制御部 C 110 に通知する。

30

【0033】

主記憶制御部 C 110 には、主記憶 m 100 に書き込むデータの誤り制御符号化をする符号化回路 C 111 と、主記憶 m 100 から読み出されたデータの誤りを検出・訂正する復号回路 C 112 がある。復号回路 C 112 で検出された誤りの情報や誤ったデータそのものをログとして記憶するログレジスタ C 116 があり、信号線 C 115 で復号回路 C 112 からエラー情報を受け取る。ログレジスタ C 116 の内容は、アドレス空間にマッピングされたレジスタで、プロセッサ B 100, B 101 で障害処理ルーチンが起動したときプロセッサバス B 102、プロセッサバス制御部 B 103、信号線 B 109 を経由してアクセスされ、信号線 B 108、プロセッサバス制御部 B 103、プロセッサバス B 102 を経由してプロセッサ B 100, B 101 に転送される。

40

【0034】

また、ノード N 100, N 101 の外部に設けられたモニタ D 100 は、信号線 D 101 を経由してログレジスタ C 116 にアクセスし、ログレジスタ C 116 の内容を信号線 D 102 を通してモニタ D 100 に転送され、モニタ D 100 の表示画面にログの内容が表示される。また、モニタ D 100 は信号線 D 102, D 103, D 104, D 105 を通して各ノードの主記憶 0 の内容をダンプもする。

【0035】

図 14 に示す符号化回路 C 111 について説明する。符号化回路 C 111 は、主記憶へのライトデータを格納するレジスタ C 202 と、検査ビット生成回路 C 204 と、主記憶へのライトデータにプロセッサバス制御部や I/O バス制御部やクロスバスイッチ出入力制御

50

部で検出した訂正不能エラーがあることの情報やエラー検出場所等の障害情報を格納する障害情報レジスタ C 206 と、障害情報から検査ビットのパタンを生成する障害情報照合テーブル C 208 と、検査ビットどうしの線形和を生成する EXOR 回路 C 210 と、主記憶へ書き込む符号語のデータビットを格納するレジスタ C 211 と、検査ビットを格納するレジスタ C 212 から構成される。なお、C 250 は従来の検査ビット生成回路と同一の回路であり、C 251 の部分を新たに追加することで本発明で提案した符号化方法を実現可能となる。これは既に説明した図 11 の (b) 方法 2 に相当する回路であり、手順 S6、S7 が図 14 の C 250 の部分に、図 11 の (b) 方法 2 の手順 S4、S5 が図 14 の C 251 の部分に相当する。

【0036】

符号化回路 C 111 の動作について説明する。符号化回路 C 111 の動作は既に説明した図 11 の (b) 方法 2 の手順を実行する。図 13において、プロセッサバス制御部 B 103、または I/O バス制御部 A 106、またはクロスバス入出力制御部 X 105 から主記憶制御部 C 110 に転送されたライトデータと訂正不能エラーが発生した情報は、図示はしていないが主記憶制御部 C 110 内のセレクタで選択され、それぞれ図 14 の信号線 C 200、C 201 を通し符号化回路 C 111 に転送される。ライトデータは、レジスタ C 202 から信号線 C 203 を通し、検査ビット生成回路 C 204 で検査ビットを生成し、信号線 C 205 で EXOR 回路 C 210 に転送される。一方、訂正不能エラー情報とエラーの検出元を示した信号は信号線 C 201 で障害情報レジスタ C 206 に格納され、信号線 202 07 を通し、障害情報照合テーブル C 208 に転送され、図 9 のテーブルに相当する連想回路が搭載されていて、障害発生内容からシンドロームのパタンを生成する。このシンドロームのパタンは符号化時には検査ビットのパタンになる。障害情報照合テーブル C 208 で生成された検査ビットは、信号線 C 209 を通し EXOR 回路 C 210 では方法 2 の手順 8 の処理を行う。完成した符号語はレジスタ C 211 と C 212 に格納されて、信号線 C 113 を介し主記憶へ転送される。

【0037】

次に、図 15 に示す復号回路 C 112 を説明する。復号回路 C 112 には、データビットレジスタ C 300 と、検査ビットレジスタ C 301 と、シンドローム生成回路 C 304 と、シンドロームデコード回路 C 306 と、ビット反転回路 C 308 と、障害情報照合テーブル C 351 と、データビットレジスタ C 310 と、検査ビットレジスタ C 311 がある。

【0038】

復号回路 C 112 の動作について説明する。復号回路 C 112 の動作は図 12 の手順を実行する。図 15において、主記憶から読み出されたデータは信号線 C 114 を通してデータビットレジスタ C 300 と検査ビットレジスタ C 301 に格納される。信号線 C 303 を通してシンドローム生成回路 C 304 にレジスタ C 300、C 301 のデータは転送され、シンドロームが生成される。シンドロームを生成する手順は、従来の誤り制御方式と同様に図 12 の手順 S11 である。生成したシンドロームは、信号線 C 305 を通しシンドロームデコード回路 C 306 に転送され、シンドロームが 0 ならばエラーは存在しないとし、そうでない場合は、エラー訂正あるいはエラー検出を図 12 の手順 S13、S15 の処理を行う。この処理は従来の誤り検出・訂正の手順に準ずる。シンドロームをデコードし、エラーのあったポインタを示す信号を信号線 C 307 を通し反転回路 C 308 へ転送し、信号線 C 303 で転送される主記憶からの読み出しデータのエラーポインタで示された位置をビット反転する。この誤りを訂正されたデータは、レジスタ C 310 と C 311 に格納され信号線 C 312 へ転送される。この信号線 C 312 のデータは、図示はしていないが図 13 の主記憶制御部 C 110 で主記憶の読み出しを要求した部分（プロセッサバス制御部または I/O バス制御部またはクロスバスイッチ入出力制御部）に転送する。

【0039】

ここで説明した図 15 の C 350 部分の動作は従来の誤り検出・訂正の回路と全く同一の回路である。本発明で追加された部分が C 351 の障害情報照合テーブルである。障害情

10

20

30

40

50

報照合テーブル C 351 は信号線 305 で転送されるシンドロームのパタンからデータの障害情報を解読する回路であり、動作は、既に説明したように図 12 の手順 S16 を行い、図 9 の表でシンドロームから障害発生原因を連想する処理を行う。解読した障害情報は図 15 の信号線 C115 を通し、図 13 のログレジスタ C116 へ転送される。

【0040】

さらに、本実施の形態でのプロセッサバスと主記憶や主記憶制御回路内のデータバスでは同じ誤り制御符号で保護されているとする。特に、プロセッサにはエラー検出・訂正回路が存在する。この回路は、図 14 の C250、図 15 の C350 の回路と同一の回路が存在する。

【0041】

誤り制御符号の訂正機能以上の多重誤りが起きた場合（障害情報を含んでいる場合を除く）は、図示していないがさらに、復号回路 C112 の出力 C312 を符号化回路 C111 の入力 C200 へ再入力して、障害情報を追加して符号化を行い、プロセッサバス制御回路にデータを転送する。また、障害情報が含まれている主記憶の読み出しデータの場合は、そのまま、プロセッサ制御回路にデータを転送する。

【0042】

プロセッサがプロセッサバスから障害情報を一緒に符号化した主記憶データを受け取り、プロセッサの復号回路で処理をすると、障害情報を配置した H 行列の列ベクトルは、プロセッサの復号回路では短縮された列ベクトルなので、訂正対象の誤りではなく、かつシンドロームが非零なので、訂正不能エラーとして処理される。本発明の方式は、本発明の方式を実現する回路（図 14 の C251、図 15 の C351）を追加していない従来の誤り検出・訂正回路が混在していても障害情報を含んだデータは、誤り訂正不能な障害データとして扱われるため、適切なエラー処理が行われる下位互換方式であるといえる。プロセッサで障害情報を含む主記憶の読み出しデータを復号すると訂正不能エラーが発生し、障害割り込みが発生し、障害処理ルーチンがプロセッサで処理される。この障害処理ルーチンに図 13 のログレジスタ C116 を読み出す処理を追加すれば、障害の発生した I/O バスを切り離し、漸次縮退させる方法も可能である。また、ログレジスタの内容を画面にエラー発生個所として表示することにより障害個所を直ちに確認可能となる。障害処理ルーチンで主記憶のダンプをとり、画面に表示する場合もある。この場合、主記憶の内容を BIOS 等のソフトウェア上で本発明の提案方式の処理を行い、画面に障害発生個所を表示することも可能である。

【0043】

主記憶に書き込むデータに訂正不能なエラーがあっても、そのデータがプロセッサの処理で使用されなければプロセッサの障害は発生せず、システムの稼働率を向上させることができる。また、上記の例のようにプロセッサで訂正不能エラーを検出したとしても、その障害データを読み出したプロセスのみを停止させることができ、システム全体がダウンする確率を減らすことができる。

【0044】

さらに、図 13 のモニタ D100 をシステム動作中、あるいは、停止中に動作させ、主記憶のダンプを採取し、ソフトウェア上の処理で障害情報をデコードし表示することも可能である。また、ログレジスタの内容も表示することも可能である。これにより、障害発生源の迅速な発見をすることが可能である。

【0045】

以上、本発明者によってなされた発明をその実施の形態に基づき具体的に説明したが、本発明は前記実施の形態に限定されるものではなく、その要旨を逸脱しない範囲で種々変更可能であることはいうまでもない。

【0046】

【発明の効果】

本願において開示される発明のうち、代表的なものによって得られる効果を簡単に説明すれば、以下のとおりである。

【0047】

本発明によれば、誤り検出符号を使用した情報通信システム、あるいはメモリシステムの誤り検出・訂正方法に関して、符号化するデータが障害情報である情報をデータの大きさを変えずに符号化可能な方式を与え、また本方式を追加していない復号方法でも誤訂正することなく障害データであると識別できる方式を実現することができる。この方式を実現した装置では、障害データを読み出したプロセスだけが障害処理をし、可能な限りシステムをダウンさせず、システムの稼働率を向上させることができる。また、障害発生源を特定する速度を向上でき、システムのダウントIMEの削減が可能となる。

【図面の簡単な説明】

- 【図1】本発明で提案する方法の考え方を示すための符号の例を示す説明図である。 10
- 【図2】本発明で提案する方法の考え方を示す説明図である。
- 【図3】本発明で提案する方法のデータ構造を示す説明図である。
- 【図4】本発明で提案する方法の処理の流れの例（エラーの無い場合）を示すフロー図である。
- 【図5】本発明で提案する方法の処理の流れの例（障害情報が存在する場合）を示すフロー図である。
- 【図6】本発明で提案する方法の処理の流れの例（障害情報が存在し、かつ通信路あるいは記憶装置にエラーが発生する場合）を示すフロー図である。
- 【図7】(a), (b)は本発明で提案する符号化方式を示す説明図である。
- 【図8】本発明の一実施の形態に係る誤り制御符号の例を示す説明図である。 20
- 【図9】本発明の一実施の形態に係る障害情報とシンドロームのパタンの関係表を示す説明図である。
- 【図10】本発明の一実施の形態に係る符号化方式の処理の流れを示すフロー図である。
- 【図11】(a), (b)は本発明の一実施の形態に係る符号化方式の符号化手順を示すフロー図である。
- 【図12】本発明の一実施の形態に係る符号化方式の復号手順を示すフロー図である。
- 【図13】本発明の一実施の形態に係る計算機システムの全体を示すブロック図である。
- 【図14】本発明の一実施の形態に係る装置の符号化回路を示すブロック図である。
- 【図15】本発明の一実施の形態に係る装置の復号回路を示すブロック図である。
- 【図16】従来技術の問題点を説明するための符号の例を示す説明図である。 30
- 【図17】従来技術の問題点を説明するための例（2bitエラー）を示す説明図である。
- 【図18】従来技術の問題点を説明するための例（誤訂正）を示す説明図である。

【符号の説明】

- N 1 0 0 , N 1 0 1 ノード
 X 1 0 0 クロスバスイッチ
 C 1 0 0 主記憶制御装置
 A 1 0 0 , A 1 0 1 , A 1 0 2 , A 1 0 3 I/O装置
 A 1 0 4 , A 1 0 5 I/Oバス
 A 1 0 6 I/Oバス制御部 40
 A 1 0 7 , B 1 0 4 , X 1 0 8 エラー検出回路
 B 1 0 0 , B 1 0 1 プロセッサ
 B 1 0 2 プロセッサバス
 B 1 0 3 プロセッサバス制御部
 X 1 0 5 クロスバスイッチ入出力制御部
 C 1 1 0 主記憶制御部
 C 1 1 1 符号化回路
 C 1 1 2 復号回路
 m 1 0 0 主記憶
 C 1 1 6 ログレジスタ 50

D 1 0 0 モニタ

C 2 0 2 , C 2 1 1 , C 3 0 0 , C 3 1 0 データビットレジスタ

C 2 0 4 検査ビット生成回路

C 2 0 6 障害情報レジスタ

C 2 0 8 , C 3 5 1 障害情報照合テーブル

C 2 1 0 E X O R 回路

C 2 1 2 , C 3 0 1 , C 3 1 1 検査ビットレジスタ

C 3 0 4 シンドローム生成回路

C 3 0 6 シンドロームデコード回路

C 3 0 8 ビット反転回路

10

【図 1】

図 1

$$(8,4)\text{SEC-DED符号} \quad H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

列の削除

↓

情報ビット 検査ビット

d0 d1 c0 c1 c2 c3

$$(6,2)\text{SEC-DED符号} \quad H' = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

【図 2】

図 2

$$(6,2)\text{SEC-DED符号} \quad H' = \begin{bmatrix} \text{情報} & \text{検査} \\ \text{ビット} & \text{ビット} \\ \hline d0 & d1 & c0 & c1 & c2 & c3 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

↓

↓

[障害情報] 情報 検査ビット

c0 c1 d0 d1 c0 c1 c2 c3

$$(8,4)\text{SEC-DED符号} \quad H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

【図 3】

図 3

障害 情報 ビット 検査ビット

$$D = [e0 \ e1 \ d0 \ d1 \ c0 \ c1 \ c2 \ c3]$$

(1) 障害情報無し

$$D = [\underline{0} \ 0 \ d0 \ d1 \ c0 \ c1 \ c2 \ c3]$$

(2) 障害情報あり

$$D = [\underline{1} \ 0 \ d0 \ d1 \ c0 \ c1 \ c2 \ c3]$$

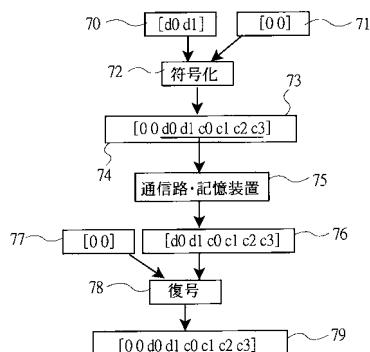
または

$$D = [\underline{0} \ 1 \ d0 \ d1 \ c0 \ c1 \ c2 \ c3]$$

【図 4】

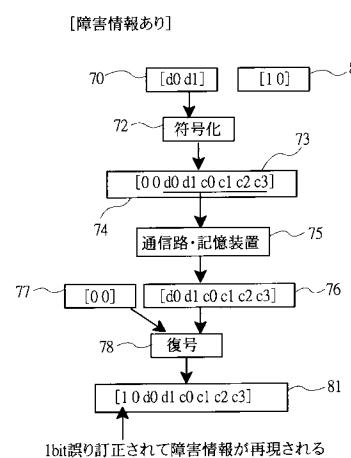
図 4

[障害情報無し]



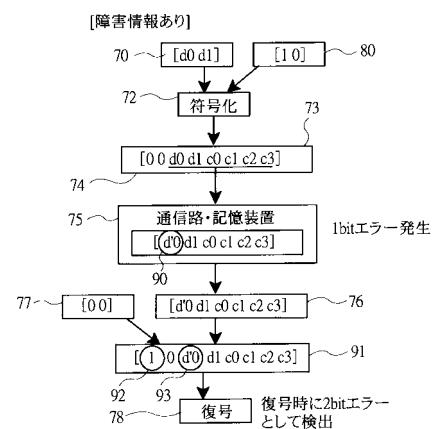
【図5】

図5



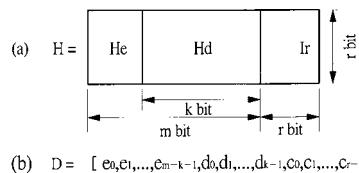
【図6】

図6



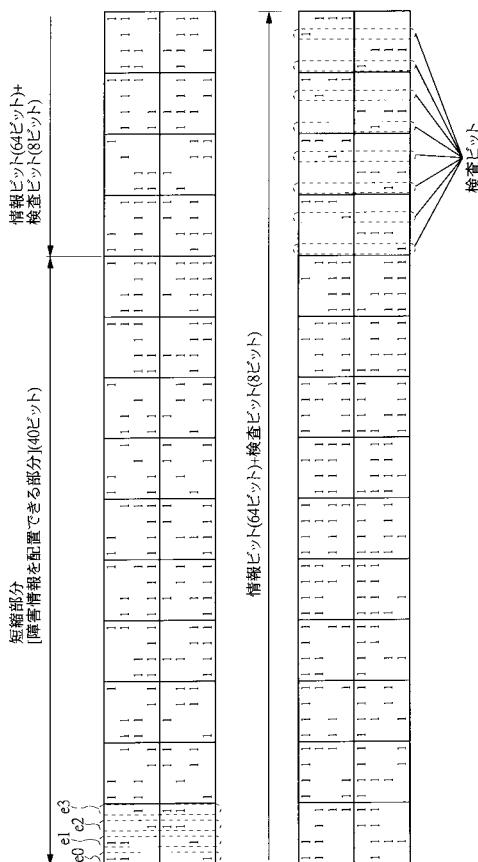
【図7】

図7



【図8】

図8



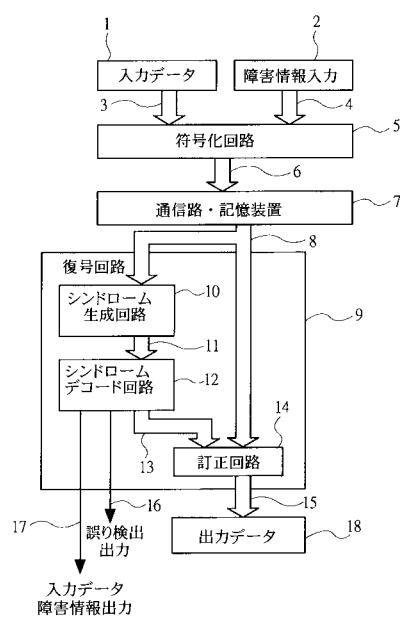
【図9】

図9

H行列での位置	シンドロームのパターン	障害発生内容
e0	[11000001]	CPUバスの訂正不能エラー
e1	[11000010]	I0バス0の訂正不能エラー
e2	[00011100]	I0バス1の訂正不能エラー
e3	[00101100]	クロスバスイッチからのデータの訂正不能エラー

【図 10】

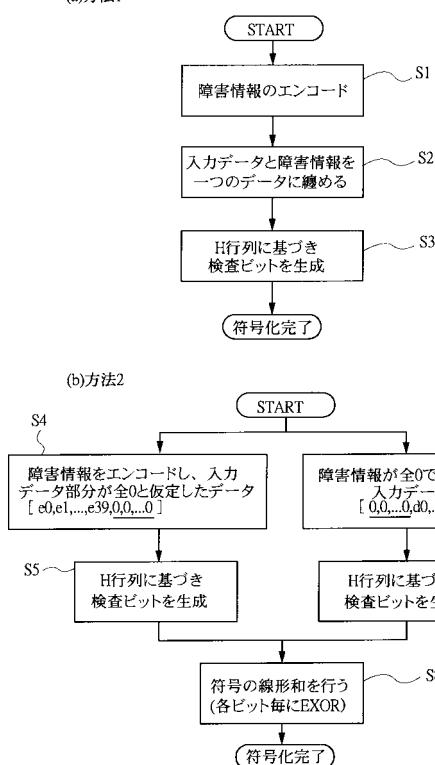
図 10



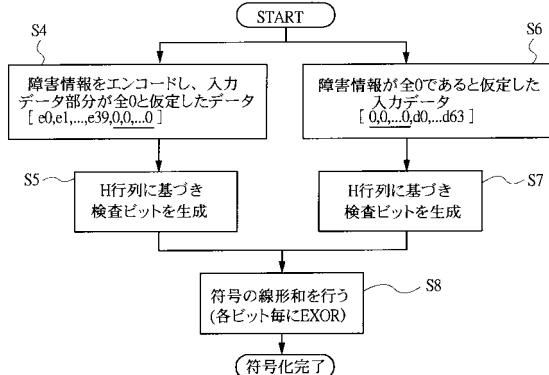
【図 11】

図 11

(a)方法1

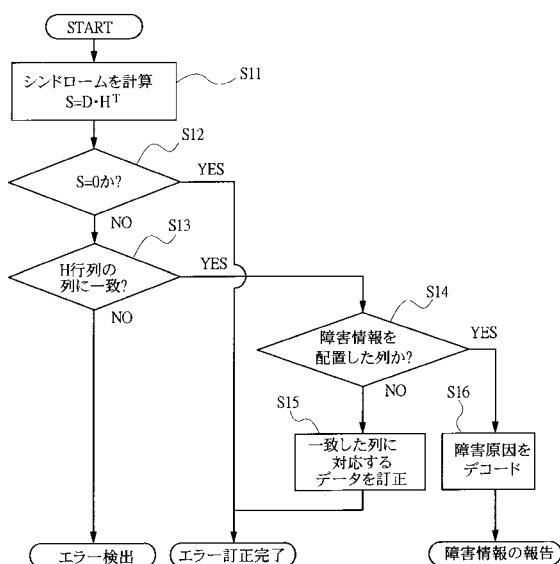


(b)方法2



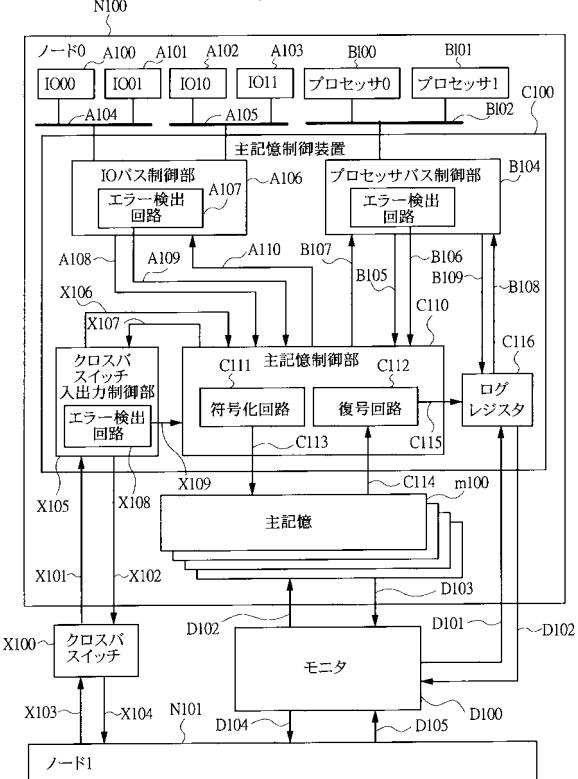
【図 12】

図 12



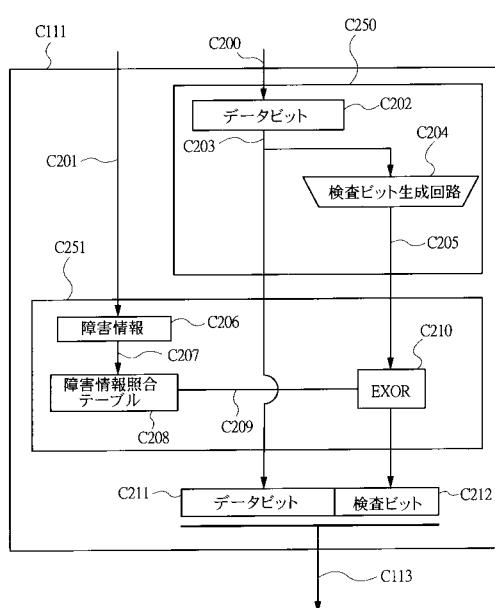
【図 13】

図 13



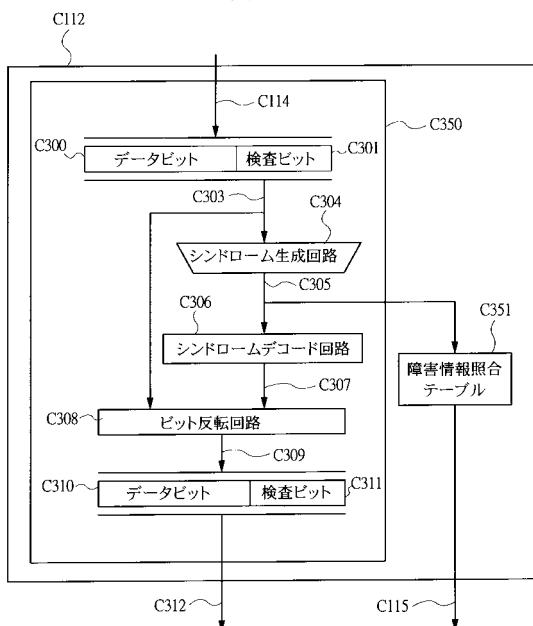
【図14】

図 14



【図15】

図 15



【図16】

図 16

$$(8,4) \text{ SEC-DED 符号の例}$$

情報ビット	検査ビット
$d_0 \ d_1 \ d_2 \ d_3$	$c_0 \ c_1 \ c_2 \ c_3$

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & | & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & | & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & | & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & | & 0 & 0 & 0 & 1 \end{bmatrix} = [h_0 \ h_1 \ h_2 \ h_3 \ h_4 \ h_5 \ h_6 \ h_7]$$

【図17】

図 17

2bitエラーの例: 図16のd0とc3にエラー発生

$$S = h_0 + h_7 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

【図18】

図 18

誤訂正の例: 検査ビットが全1の符号語にさらに1bitエラーが発生

(1) シンドロームが全1となるデータD

$$\begin{array}{c} \text{情報} \quad \text{検査} \\ \text{ビット} \quad \text{ビット} \\ \hline D = [0000 \ 1111] \end{array}$$

(2) Dにさらに1bitエラーが起きた場合のデータD'

$$D' = D + [0000 \ 0001] = [0000 \ 1110]$$

(3) D'から生成したシンドロームS'

$$S' = D' H^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = h_0$$

検査ビットd0に発生した1bitエラーと同じシンドロームとなる為、d0が誤って訂正されてデータD''となってしまう

$$D'' = [10001110]$$

フロントページの続き

(58)調査した分野(Int.Cl. , DB名)

G06F11/08-11/10

G06F12/16

G06F3/06-3/08

H03M13/00-13/53