

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第4053228号  
(P4053228)

(45) 発行日 平成20年2月27日(2008.2.27)

(24) 登録日 平成19年12月14日(2007.12.14)

(51) Int.Cl.	F 1
H03M 13/15	(2006.01)      H03M 13/15
G01R 31/02	(2006.01)      G01R 31/02
G06F 11/10	(2006.01)      G06F 11/10      330M
G06F 13/00	(2006.01)      G06F 13/00      301L
HO4B 17/00	(2006.01)      HO4B 17/00      E

請求項の数 3 (全 13 頁)

(21) 出願番号	特願2000-327211 (P2000-327211)
(22) 出願日	平成12年10月26日 (2000.10.26)
(65) 公開番号	特開2001-211085 (P2001-211085A)
(43) 公開日	平成13年8月3日 (2001.8.3)
審査請求日	平成16年10月18日 (2004.10.18)
(31) 優先権主張番号	09/438063
(32) 優先日	平成11年11月10日 (1999.11.10)
(33) 優先権主張国	米国 (US)

(73) 特許権者	398038580 ヒューレット・パッカード・カンパニー HEWLETT-PACKARD COMPANY アメリカ合衆国カリフォルニア州パロアルト ハノーバー・ストリート 3000
(74) 代理人	100087642 弁理士 古谷 聰
(74) 代理人	100063897 弁理士 古谷 韶
(74) 代理人	100076680 弁理士 溝部 孝彦

最終頁に続く

(54) 【発明の名称】配線縮退故障を検出するためのECCコードメカニズム

## (57) 【特許請求の範囲】

## 【請求項 1】

データメッセージの伝送中に配線故障を検出するための方法であって、前記データメッセージは、データ部と、前記データ部に関連づけられた第1のエラーコード(ECC)部とからなり、

前記データ部および前記第1のECC部に存在する特定ビットの数を表すカウンタ値を決定するステップであって、前記特定ビットが、2進数の1と2進数のゼロからなるグループから選択される、ステップと、

前記カウンタ値に関連づけられる第2のECC部を形成するステップと、

前記カウンタ値および前記第2のECC部とともに前記データメッセージを符号化するステップと、

前記データメッセージの伝送中に、配線故障によるエラーが前記データ部に存在するか否かを判定するために、受信体が、前記符号化されたカウンタ値を使用するステップを含む、方法。

## 【請求項 2】

前記使用するステップが、

前記メッセージを、前記データ部及び前記第1のECC部からなる第1のサブメッセージと、前記カウンタ値及び前記第2のECC部からなる第2のサブメッセージとを有する、少なくとも2つのサブメッセージに分割するステップと、

前記第1及び第2のサブメッセージを復号化するステップと、

10

20

前記第1のサブメッセージに存在する特定ビットの数を表す別のカウンタ値を決定するステップと、

前記別のカウンタ値と前記第2のサブメッセージのカウンタ値を比較するステップと、

前記比較ステップが、前記別のカウンタ値が前記第2のサブメッセージのカウンタ値と異なることを示す場合に、エラーを通知するステップを含む、請求項1の方法。

#### 【請求項3】

データメッセージの伝送中に配線故障を検出するための方法であって、前記データメッセージは、データ部分を含み、

前記データ部分に存在する特定ビットの数を表すカウンタ値を決定するステップであって、前記特定ビットが、2進数の1と2進数のゼロからなるグループから選択される、ステップと、

高次ビットを除去することにより前記カウンタ値の切り捨てを行うステップであって、これにより、残りのビットの数が、前記データ部分を伝送するのに必要なサイクル数の2倍に等しい数までカウントするのに十分なものとなるようにする、ステップと、

前記切り捨てがなされたカウンタ値とともに前記データメッセージを符号化するステップと、

前記データメッセージの伝送中に、配線故障によるエラーが前記データ部分に存在するか否かを判定するために、受信体が、前記符号化されたカウンタ値を使用するステップを含む、方法。

#### 【発明の詳細な説明】

##### 【0001】

##### 【発明の属する技術分野】

本発明は、一般に、エラー訂正コードに関し、より詳細には、配線故障、特に縮退故障(stuck-at-fault)による配線故障を検出するエラー訂正コードに関する。

##### 【0002】

##### 【従来の技術】

従来技術において、配線を介して伝送されるデータは、エラーを生じやすい。すなわち、2進数の1が歪んで2進数の0として現れ、またその逆の場合もある。エラーは、データストリームの1ビットがエラーとなるシングルビットエラーの場合もあり、データストリームの2ビットがエラーとなるダブルビットエラーの場合もある。なお、典型的には、データは、1本の配線ではなく1組の配線を介して伝送されるが、1本及び複数本の配線伝送システムのいずれにおいても、エラーが生じる可能性がある。さらに、データは、配線の数よりはるかに多いため、複数サイクルで送信され、たとえば、16本の配線が4回のサイクルで64ビットを送信する。したがって、複数サイクルでエラーが生じるおそれがある。システムの中には、データをパケット化する、すなわち、特定のサイズのデータパケットでデータを送出する、ものもある。したがって、異なるデータパケットでエラーが生じる可能性がある。典型的なエラーの原因の1つは、配線故障である。配線のショートまたは断線により、故障信号が配線で送出され得る。これらの故障は、2つのタイプのうちの一つに分類される。第1のタイプは、配線が1縮退故障(stuck-at-one)または0縮退故障(stuck-at-zero)である場合である。この場合、入力に関係なく、配線は、1(1縮退故障の場合)またはゼロ(0縮退故障の場合)のみを中継し、信号を切り換えることはない。第2のタイプは、悪質な(malicious)故障である。このタイプの故障では、入力がどうであれ配線の出力が切り替わっていく。たとえば、入力がゼロの場合、配線出力は1またはゼロのいずれかとなり、入力が1であっても、配線出力は1またはゼロのいずれかとなる。すなわち、配線の挙動は予測不能である。さらに、配線故障が正しい結果を搬送することがあるため、エラーが隠されてしまう場合がある。

##### 【0003】

かかるエラーを検出するために、ECCコードをデータとともに伝送する。巡回符号は、配線故障を検出することができるECCコードの一種である。巡回符号は、重要なコード

10

20

30

40

50

のクラスである。これらのコードのジェネレータ / パリティマトリクスは、行(row)の周期的なシフトにより形成される。これらは、多数のランダムエラー、バイトエラーおよびバーストエラーを検出 / 訂正するのに効率的な巡回符号である。巡回符号については、T. R.N., RaoおよびE. Fujiwaraによる「Error Control Coding for Computer Systems」(Prentice Hall, Englewood Cliffs, New Jersey 07632, ISBN 0-13-28395-9) に詳細が記載されており、参照によりこれを本明細書中に組み込む。巡回符号は、悪質な故障を検出することを目的としており、従って、故障は、悪質な故障であると推定する。巡回符号は後者を対象にしているため、縮体故障を検査する場合よりも多くのビットを必要とし、必要なビット数は、設計者が残しておかなければならぬビット数より多いことがある。たとえば、32バイト (= 256ビット) からなるデータメッセージを想定する。シングルビットエラー訂正およびダブルビットエラー検出を可能にするために、10ビットが余分に必要であり、シングルエラー訂正には9ビットが必要となるが、これは、 $2^9$ が(256 + 9)より大きい2のべき数のうち最も小さいべき数であるからである。10番目のビットは、ダブルビットエラーを検出するために使用され、合計266ビットになる。したがって、シングルエラー訂正およびダブルエラー検出を行うために10ビットが必要になる。これら266ビットを10本の配線で送る場合、6本の配線が27ビットを搬送し、4本の配線が26ビットを搬送する。したがって、配線故障が最大27ビットに影響を及ぼす。悪質な配線故障を検出するために、27の追加ビットが必要となるが、これについては、RaoおよびFujiwaraによる前記書籍にある、(度の)  $g(x)$  により生成された巡回符号が、長さ 以下の任意のバーストを検出できるという定理3.7を参照されたい。  
これにより、合計293ビットで、配線故障に加えて2本の連続配線にわたる長さ27以下の任意のバーストを検出する。したがって、合計37ビットがエラー検出に必要となる。これによるオーバーヘッドは大きく、伝送用のシステム資源の多くの部分が消費される。  
。

#### 【0004】

37ビットは293ビットのわずか13%にすぎないが、より高い割合のオーバヘッドは、エラー検出、特にデータをブロックまたはパケットで伝送する時に必要な余分なビットに起因することがある。たとえば、1ブロックのデータが、10本の配線にわたる7サイクルのデータで構成され、1ブロックあたり合計70ビットであるとする。このとき、256ビットでは4ブロック(3.6ブロックを切り上げる)、293ビットでは5ブロック(4.1ブロックを切り上げる)がそれぞれ必要となる。したがって、エラー検出には余分なブロック、すなわち20%の余分なビットが必要になる。

#### 【0005】

データ伝送にオーバヘッドが生じることに加え、巡回符号は、実施するのがより複雑である。受信側でデコードすることは、巡回符号を実施するための多くの異なる手段が存在するので複雑である。

#### 【0006】

##### 【発明が解決しようとする課題】

したがって、大きなオーバヘッドを生じず、かつ、実施が容易な配線縮退故障を検出するエラー検出メカニズムが当該技術において必要とされている。

#### 【0007】

##### 【課題を解決するための手段】

これら及びその他の目的、特徴および技術的利点は、エラー検出メカニズム(エラー検出手段)を用いて配線縮退故障を検出するシステムおよび方法により達成される。本メカニズムを用いて、配線縮退故障検出が可能な既存のECCコードを強化することが可能である。たとえば、既存のECCコードは、データ伝送においてランダムなシングルエラーやダブルエラー(SEC-DED)を検出できるが、一方、本発明のメカニズムは、配線故障を検出し、その故障した配線上を伝送されたデータのエラーを特定する。本発明のメカニズムは、メッセージ用の既存のECCコードを含む、メッセージ内の1(または0)の数を決定して、この情報をそのメッセージに付加する。このカウント値は、それ自体、メ

10

20

30

40

50

ツセージに使用される同一の ECC コードにより保護される。受信側でメッセージを復号化（デコード）する場合、任意の配線障害による縮退故障（縮退故障配線障害、配線縮退故障）が、付加された情報をメッセージの内容と比較することにより検出される。

#### 【 0 0 0 8 】

本メカニズムは、配線縮退故障を検出することに加え、復号後のデータ部分において、0から1への遷移数が1から0への遷移数に等しくない場合、任意の数の複数エラー検出することもできる。従来技術の巡回符号に対する本発明のメカニズムの利点は、必要なチェックビットの数が少なくて済むことと、実施が比較的簡単であること、そして、必要な追加チェックビットの数と配線故障検出をトレードオフできることである。本発明のメカニズムは、コードワードが、配線故障が生じている状態で複数サイクルにわたって伝送される際に生じる複数エラーの検出に特に有用である。10

#### 【 0 0 0 9 】

本発明のメカニズムは、縮退故障および最も悪質な配線故障を検出するものである。本発明のメカニズムは、すべての悪質な配線故障を検出するものではなく、特に、ECC復号後に0から1への遷移数が1から0への遷移数と等しい場合には、配線故障を検出しない。したがって、本発明は、データ伝送において、1または0の数を変化させる予測可能な故障、たとえば、配線が0または1に固定される縮退故障、を検出することを主な目的とする。本発明のメカニズムは、データ伝送において使用される配線数に応じてスケーリングすることが可能である。20

#### 【 0 0 1 0 】

以上、以下の発明の詳細な説明をよりよく理解できるように、本発明の特徴および技術的利点をかなり広義に概説した。本発明の特許請求の範囲の主題を構成する本発明の他の特徴および利点を以下で説明する。なお、開示された概念および特定の実施形態を、本発明と同じ目的を実施するための他の構造を修正または設計する基本として、容易に利用することができるということは当業者には明らかであろう。また、かかる等価な構成が、特許請求の範囲に記載される本発明の思想および範囲から逸脱しないことも当業者には理解されるはずである。20

#### 【 0 0 1 1 】

##### 【発明の実施の形態】

本発明およびその利点をより十分に理解できるようにするために、添付図面を参照して以下に説明する。30

#### 【 0 0 1 2 】

本発明のエラー検出メカニズムは、図1に示すようにメッセージを符号化する。本発明のメカニズムにより、エラー訂正コード（ECC）（もしあれば）が、実データに作用する（または、演算を施す）ことができる（11）。次に、本メカニズムは、データビットおよび関連するECCチェックビット（もしあれば）を、その「データ」として処理し、これを配線縮退故障から保護する。本発明のメカニズムは、データ部分、すなわち、実データおよびチェックビットにおける1の数をカウントし（12）、この数をカウンタ値として記憶する。本発明のメカニズムは、カウンタ値がデータ部分と同一のECC方式により保護されるようにECCチェックビットを追加する（13）ことによってカウンタ値ビットを保護することができる。したがって、本発明のメカニズムは、4つの部分、すなわち、図2に示すように、実データ部21と、実データを保護するECCチェックビット22と、実データとそのチェックビットにおける1の数を有するカウンタ値23と、データ部と同一のECC能力でカウンタ値を保護するチェックビット24とから構成されるデータメッセージ20を形成する（14）。したがって、伝送メッセージは、実データ、実データECC、カウンタ値およびカウンタ値ECCで符号化される。そしてメッセージは、伝送線または他の伝送システムを介して目的地に伝送される。なお、シングルエラー訂正およびダブルエラー検出（SEC -DED）コードのECC能力に関して本発明を説明したが、本発明は、パリティを含む任意のECCコードを用いて動作することもできる。さらに、1の数をカウントすることに関して本発明を説明したが、本発明は、1をカウントす4050

る代わりに 0 をカウントすることによって演算することもできる。

#### 【 0 0 1 3 】

メッセージの宛先に配置される本発明のメカニズムの第 2 の例は、図 3 に示すようにメッセージを復号化する。復号時、メッセージは 2 つのサブメッセージに分割される（31）。第 1 のサブメッセージは実データと実データ ECC からなり、第 2 のサブメッセージはカウンタ値とカウンタ値 ECC からなる。第 1 および第 2 のサブメッセージは、次に符号化に使用された ECC メカニズムにしたがって別々に復号される（32）。実データまたはカウンタ値のいずれかにシングルビットエラーがあれば、これを検出してそれぞれの ECC により訂正する（33）。実データまたはカウンタ値のいずれかにダブルエラーがあれば、それぞれの ECC によりこれを検出する（33）。ダブルエラーが発見されると、メッセージは破棄され、再送またはリトライ信号がメッセージの送信元に送信される。SEC - DED が完了すると、配線故障を検出することができる。次に、本発明のメカニズムは、復号化された実データと実データ ECC における 1 の数をカウントする（34）。復号化されたカウンタ値を、復号化された実データと実データ ECC チェックビットにおいてカウントされた 1 の数と比較（35）して、配線縮退故障が存在するか否かを判定する（36）。このチェックにおいて、悪質な配線故障を含む複数エラーが存在していれば、複数エラーが相殺されない限り、これも検出される。なお、悪質な配線故障では、故障した配線により伝送された値がランダムに変更される。複数エラーの相殺は、0 から 1 への遷移数が 1 から 0 への遷移数と等しい場合に起こる。カウンタ値が実データと実データ ECC における 1 の数と一致しない場合、エラーが通知される（または、エラーフラグが立てられる）（37）。配線エラーが検出されると、データは破損された（データに誤りがある）とみなされる。別のシステムでは別のエラー回復メカニズムを呼び出すことができる。配線エラーが検出されなかった場合は、メッセージの传送中に検出可能なエラーが発生しなかった（38）ということになる。

#### 【 0 0 1 4 】

たとえば、64 ビットの実データを有するメッセージパケットを転送する、17 ビット幅、すなわち 17 本の配線幅のチャンネルを考える。さらに、すべてのシングルエラーが訂正され、すべてのダブルエラーが検出され、配線故障が検出されると想定する。SEC - DED コードでは、(72, 64) ECC コードを形成する 8 チェックビットが必要とされる。このコードは、配線故障を検出する能力を有していないため、最大 5 つのエラーを招く可能性がある。ここで、72 ビットは 17 本の配線上で送信されるため、少なくとも 1 本の配線がメッセージの 5 ビットを搬送し、この配線が故障すると、最大 5 つのエラーが発生することに留意されたい。実データと実データ ECC とを足すと 72 ビットであるので、実データと実データ ECC とを合わせて、合計 72 個の 1 を有することがある。したがって、カウンタ値は 7 ビット値になる。カウンタ値自体は、(12, 7) SEC - DED ECC コードを形成する 5 ビットの ECC により保護される。このため、本発明のメカニズムにより形成されるメッセージは 84 ビットであり、このうち 64 ビットが実データである。メッセージは、(84, 64) コードとなる。符号化されたメッセージは、(チャンネルが 17 本の配線で構成されるため) チャンネルを介して 5 サイクルにわたって传送される。

#### 【 0 0 1 5 】

カウンタ値は、実データと同一の ECC で保護されるため、メッセージ全体は、所望の ECC（この場合、SEC - DED）で保護されることが保証される。たとえば、カウンタ値が ECC で保護されないとすると、カウンタ値のシングルビットエラーにより、実データに実データ ECC を加えたもの（以下ではこれを、実データ + 実データ ECC とも記載）に対する 1 の数に不整合が生じる。この不整合により、誤った配線故障フラグが発光される。さらに、配線エラーは、検出にかかるエラーパターンで発生する場合がある。たとえば、3 つのエラーが実データ + 実データ ECC の 72 ビットにあり、1 つのエラーがカウンタ値にある、合計 4 つのエラーをもたらす配線故障によるエラーパターンがある。（72, 64) ECC は、3 つのエラーの検出を保証しないため、復号化されたビット

パターンに2つまたは4つのエラーが生じうる。カウンタ値のビットの1つが変化する、誤ったデータが伝送される可能性がある。また、配線故障に伴う他のエラーパターンが検出されない可能性もある。このため、カウンタ値をいくつかのECCで保護することで、これらの問題を防止する。

#### 【0016】

しかしながら、カウンタ値にカウンタ値ECCを加えたもの（以下ではこれを、カウンタ値+カウンタ値ECCとも記載）が同一配線上で2ビットを越えることはない。たとえば、伝送用に17本の配線があれば、SEC-DED ECCの場合、カウンタ値+カウンタ値ECCが34ビットを越えることはない。すなわち、カウンタ値+カウンタ値ECCが、伝送チャンネルの配線幅の2倍を越えることはない。この制限により、配線故障によるカウンタ値のエラーは、ECCにより確実に検出される。このため、カウンタ値の任意のシングルエラーは訂正され、カウンタ値の任意のダブルエラーは、それが、偶発故障または配線故障によるものであっても、検出することができる。したがって、カウンタ値は、シングルエラー訂正、ダブルエラー検出および配線故障検出により保護される。なお、この制限は、使用されるECCのタイプが、メッセージのサイズならびに検出可能なエラー数にも影響を及ぼすため、使用されるECCのタイプによって異なる。たとえば、シングルエラー検出（SED）ECCを使用する場合、シングルエラーのみが検出可能であるため、カウンタ値とカウンタ値ECCを加えたものが配線サイズを超えることはないが、ECCのサイズは、SEDのビット数がSEC-DEDより少ないとから小さくなる。実データ上で使用されるECCがパリティビットの場合、カウンタ（値）自体およびパリティビットが配線故障に対して保護されるとすれば、パリティビットによりカウンタ値を保護するだけで十分である。一般に、カウンタ値とカウンタ値ECCを足したものサイズは、チャンネルのサイズに、ECCにより検出可能なエラー数を掛けたもののサイズより大きくはない。なお、異なるタイプのECCを使用することもできる。たとえば、SEC-DED ECC、SED ECCまたはダブルエラー訂正 トリプルエラー検出 ECCコード、あるいはその他任意のタイプのECCコードによりデータを保護することができる。

#### 【0017】

しかしながら、本発明のエラー検出メカニズムは、カウンタ値とカウンタ値ECCの合計が上記の制限を上回る状況に適応できるように再構成可能である。たとえば、上述の(72, 64)コードを用い、チャンネルを17ビット幅ではなく5ビット幅にする。カウンタ値およびカウンタ値ECCのサブメッセージは依然として合計12ビットであり、(12, 7)コードが形成される。このため、サブメッセージの2ビットが、3番目のサイクルで传送される（すなわち、最初の2サイクルで10ビットが送信される）。3ビットが少なくとも1本の配線で送信されると、SEC-DED ECCは、カウンタ値の精度を保証することができない。

#### 【0018】

第1の解決策は、各カウンタ値それぞれがECCチェックビットを有する複数のカウンタ値を使用することである。たとえば、ECCがSEC-DEDの場合、カウンタ値およびカウンタ値ECCチェックビットが配線幅の2倍より大きいときは、第1のカウンタ値とそのECCチェックビットにおける1の数をカウントすることによって第2のカウンタ値が形成される。第2のカウンタ値は、ECCチェックビットを添付される。このプロセスは、最後のカウンタ値とそのECCチェックビットが配線幅の2倍より小さくなるまで再帰的(recursively)に続行される。第1のカウンタ値は( $\log k$ )ビットを有する。ここで、kは情報ビットの数である。第2のカウンタ値は、必要であれば、( $\log(\log k)$ )ビットを有する。図5は、上記例による第1の解決策を示したものであり、SEC-DED ECCビットを含む、72個のデータビット51、 $d_0 \dots d_{71}$ と、SEC-DED ECCビットを含む12個の第1のカウンタ値ビット52、 $C_0 \dots C_{11}$ と、を含むデータパケットメッセージ50が形成されている。この例では第2のカウンタ値が必要とされる。第2のカウンタ値は、第1のカウンタ値における1の数を保持するものであり、12

10

20

30

40

50

ビットコードにおける 1 の数を保持するために 4 ビット幅である必要がある。4 ビットの第 2 のカウンタ値を保護するために、4 ビットの ECC が必要とされる。これにより、(8, 4) SEC -DED コード 53、すなわち AC<sub>0</sub>...AC<sub>7</sub> が形成される。これらの 8 ビットは、2 サイクルで 5 本の配線上を传送可能である。第 2 のカウンタ値および第 2 のカウンタ値 ECC を 3 サイクルより少ないサイクルで传送可能であるため、第 3 のカウンタは必要ではない。なお、このコードは、第 3 のサブメッセージを形成し、他の 2 つのサブメッセージとは別々に復号化される。したがって、合計 20 ビットのカウンタ値およびカウンタ値 ECC が実データおよび実データ ECC に添付される。

#### 【0019】

第 2 の解決策は、カウンタ値を追加せずに、第 1 のカウンタ値 ECC の選択されたビットを別々に保護するものである。カウンタ値の該ビットは、ECC 後に、任意のグループにおける合計ビット数がチャンネル幅の 2 倍を超えないように別々のグループで符号化される。すなわち、カウンタ値の該ビットは複数の ECC コードに分割される。図 6 は、この例による第 2 の解決策を示すものであって、SEC -DED ECC ビットを含む、72 個のデータビット d<sub>0</sub>...d<sub>71</sub> と、SEC -DED ECC ビットを含む 10 個の第 1 のカウンタ値ビット C<sub>0</sub>...C<sub>9</sub> と、を含むデータパケットメッセージ 60 が形成されている。第 1 の解決策の場合、カウンタ値と ECC 52 を足したもののが 12 ビット幅であったが、この例では、コードの全サイズを最大 10 ビットとすることができます。7 ビットのカウンタ値の 5 ビットは、5 ビットの ECC により保護可能であり、このため、(10, 5) SEC -DED ECC コード 62 が形成される。なお、10 ビットは 2 サイクルで传送されるため、この(10, 5) コードで十分である。カウンタ値の残りの 2 ビットは、4 ビットの ECC で保護され、このため(6, 2) コード 63 が形成される。したがって、本解決策により、合計 16 ビットのカウンタ値およびカウンタ値 ECC が実データおよび実データ ECC に添付されることになる。なお、この例では、第 2 の解決策は、第 1 の解決策より良好な結果をもたらすが、値が異なると結果も異なる。いずれの解決策においても、カウンタ値は保護される。

#### 【0020】

以下に別の例を挙げる。64 ビットが 5 本の配線を用いて传送され、所望のエラー検出により、配線縮退故障に加えて、任意のシングルエラーも検出する場合を考える。この場合、シングルエラー検出メカニズムは、65 ビット（データに 64 ビット + パリティビットに 1 ビット）を必要とする。図 7 は、第 1 の解決策における、例として 64 ビットであるメッセージデータパケット 70 を示しており、このパケットは、64 個のデータビット d<sub>0</sub>...d<sub>63</sub> と、データビット d<sub>71</sub> に対するパリティビット Pd<sub>72</sub> を含む。このため、65 ビットが 5 本の配線で传送されることにより、7 ビット幅のカウンタビット、すなわち C<sub>0</sub>~C<sub>6</sub> が必要とされる。これらの 7 ビットのカウンタビットは、別のパリティビットである PC<sub>74</sub> により保護され、従って 8 ビットが必要となる。これらの 8 ビットは配線故障を受けやすい（1 本の配線を 2 回通過するビットもある）ため、4 ビット幅、すなわち AC<sub>0</sub>~AC<sub>3</sub> 75 の第 2 のカウンタが必要であり、パリティ PAC<sub>76</sub> がこのカウンタビットを合計 5 ビットで保護する。5 本の配線があるため、さらに多くのカウンタは必要ない。第 2 の解決策は、7 ビットカウンタの 4 ビットを取り、パリティがこれを保護するものである（図 8）。残りの 3 ビットはパリティを別に保護するためのものである。図 8 は、第 2 の解決策において、例として 64 データビットを有するメッセージデータパケット 80 を示しており、このパケットは、64 個のデータビット d<sub>0</sub>...d<sub>63</sub> と、データビット d<sub>81</sub> に対するパリティビット Pd<sub>82</sub> を含む。このため、65 ビットが 5 本の配線で传送されることにより、7 ビットの幅を有するカウンタが必要とされる。第 2 の解決策は、最大サイズが 5 ビットとなるように、5 本の配線でカウンタ（値）を複数部分に分割する。1 つのパリティビットが保護用に必要とされるため、カウンタ（値）は最大 4 ビットのセグメントに分割される。このため、カウンタ（値）の 4 ビット C<sub>0</sub>~C<sub>3</sub> がパリティビット PCL<sub>84</sub> により保護され、カウンタ（値）の 3 ビット C<sub>4</sub>~C<sub>6</sub> がパリティビット PCH<sub>85</sub> により保護される。なお、カウンタ（値）は下位 4 ビット

10

20

30

40

50

と上位 3 ビットにセグメント化されるものとして図示されているが、他の分割も可能である。配線（縮退）故障検出によってシングルエラーがカバーされるため、パリティビットはシングルエラーを検出するためには不要である。しかしながら、それらは、配線故障検出がすべてのタイプの ECC (たとえば、SEC -DED) をカバーし得ないため一例として使用される。

#### 【0021】

第 3 の解決策は、カウンタ値のサイズを縮小する、すなわち、幅がより小さいカウンタ値を使用することである。カウンタ値が小さいと、サブメッセージが小さくなる。さらに、カウンタ値が小さいと、保護に必要な ECC ビットが少なくなり、この結果サブメッセージが小さくなる。上述の例において、チャンネルが 17 ビット幅であって、64 ビットの実データと 8 ビットの ECC チェックビットを有するメッセージパケット 90 を転送し、図 9 に示すような (72, 64) コードを形成する場合を考える。なお、72 ビットは 17 の配線上を 5 サイクルで送信されるため、少なくとも 1 本の配線が 5 ビットのメッセージを搬送し、この配線が故障または縮退故障した場合は、5 つのエラーが発生する。従って、配線縮退故障は、-5 (すなわち、配線は 0 に固定されているが、実際の値は 1 である場合) 乃至 +5 (すなわち、配線は 1 に固定されているが、実際の値は 0 である場合) だけカウンタ (値) に影響を及ぼすにすぎない。したがって、幅 4 ビットのカウンタ (値) は、10 の幅を許容できるため、この幅のカウンタ (値) で十分である。このため、カウンタ値を保護するために、前述の実施形態で必要される (12, 7) コードではなく、(8, 4) SEC -DED ECC コード 92、すなわち、C<sub>0</sub> ~ C<sub>7</sub> が必要とされる。一般に、カウンタ値は、全コードワードを伝送するサイクル数の 2 倍の対数 (底 2) の上限である幅を有する必要がある。したがって、本例では、16 を法とする (モジュロ 16) 1 の数がカウントされる。これは、1 を加え、モジュロ演算を行うことにより実行されるが、カウント中は、全幅を有することになるが、送信中は、余分な上部ビットを切り離すことが可能である。第 3 の解決策は好ましい実施形態である。

#### 【0022】

なお、設計者は、(チェックビットの数に換算して) 検出とオーバヘッドとをトレードオフすることにより、信頼性を低くした場合に必要とされるビット数よりカウンタ値ビットを少なくすることを決定することもできる。すなわち、カウンタビットの数を、配線縮退故障検出を行うのに必要な数より少なくすることができるが、その場合は、配線縮退故障検出を保証することはできない。

#### 【0023】

また、カウンタ値および実データビットは、チェックビットに不足がある場合、ともに ECC により保護され得る (または、結合して保護される ECC になり得る)。しかしながら、これは、配線故障または縮退故障に対する完全なカバレージを提供するものではない。たとえば、簡単なパリティ保護がなされたデータおよびカウンタ (値) があり、該データにおいて 1 ビットを変化させ、かつカウンタ (値) (最下位ビット) において 1 ビットを変化させる配線故障が発生したとする。パリティは、エラーを検出せず、カウンタ (値) もまた検出しない。従って、上記例では、カウンタ (値) に 4 ビットが、データに 64 ビットが必要とされる。これらの 68 ビットを保護するために、8 ビットの ECC データが必要とされるため、(76, 68) SEC -DED ECC コードが形成される。

#### 【0024】

さらに、単一の配線故障を検出することができるが、0 から 1 への遷移数が 1 から 0 への遷移数と等しい場合には、複数の配線故障を検出できない場合がある。このため、複数の配線故障のいくつかについて検出することができるが、すべての配線故障を検出できるわけではない。

#### 【0025】

さらに、本発明は、複数の配線を含む任意のデータ伝送環境においても動作可能であり、たとえば、LAN、WAN またはインターネットを介して接続された 2 つのコンピュータ間で ECC が使用される。また、本発明は、コンピュータシステム内部の、たとえば、R

10

20

30

40

50

AMとCPU間のデータ伝送にも使用可能である。これは、特にマルチプロセッサシステムについていえる。

#### 【0026】

図4は、本発明に使用されるように構成されたコンピュータシステム400を示す。データビットを送信するために使用される配線は、バス402、または405と406を結合するバス、あるいはネットワーク412である。システム400において、中央処理装置(CPU)401は、バス402に結合される。さらに、バス402は、ランダムアクセスメモリ(RAM)403、読み取り専用メモリ(ROM)404、入出力(I/O)カード405、通信カード411、ユーザインターフェースカード408およびディスプレイカード409に結合される。RAM403およびROM404は、当該技術において周知のように、ユーザとシステムデータとプログラムとを保持する。I/Oカード405は、ハードドライブ406等の記憶装置をコンピュータシステムに接続する。通信カード411は、コンピュータシステムをローカルネットワーク、広域ネットワークまたはインターネットネットワーク412に結合するよう構成されている。ユーザインターフェースカード408は、キーボード413、ポインティングデバイス407等のユーザ入力装置をコンピュータシステム400に結合する。最後に、ディスプレイカード409は、CPU401により駆動されて、表示装置410の表示を制御する。CPU401は、HP PA-8200等の任意の汎用CPUとすることができる。しかしながら、本発明は、CPU401が、本明細書において説明したような本発明の動作をサポートする限り、CPU401のアーキテクチャにより制限されるものではない。なお、エラーコード(EC)という用語は、訂正および/または検出を意味する。

#### 【0027】

本発明およびその利点を詳細に説明してきたが、特許請求の範囲により規定される本発明の思想および範囲から逸脱することなく、本明細書で説明した事項について種々の変更、置換および修正を行うことができる理解されよう。

以下においては、本発明の種々の構成要件の組み合わせからなる例示的な実施態様を示す。

1. データメッセージの传送中に配線故障を検出するための方法であって、前記データメッセージ(20)は、データ部(21)と、前記データ部に関連づけられた第1のエラーコード(EC)部(22)とからなり、

前記データ部および前記第1のEC部に存在する特定ビットの数を表すカウンタ値(23)を決定するステップ(12)であって、前記特定ビットが、2進数の1と2進数のゼロからなるグループから選択される、ステップと、

前記カウンタ値に関連づけられる第2のEC部(24)を形成するステップ(13)と、前記カウンタ値および前記第2のEC部を用いて前記データメッセージを符号化するステップ(14)と、

前記データメッセージの传送中に、配線故障によるエラーが前記データ部に存在するか否かを判定するために、受信体が、前記符号化されたカウンタ値を使用するステップからなる、方法。

2. 前記第1のEC部(22)および前記第2のEC部(24)が、シングルエラー訂正コードおよびダブルエラー検出コードである、上項1の方法。

3. 前記第1のEC部(22)および前記第2のEC部(24)が、シングルエラー検出コードである、上項1の方法。

4. 前記使用するステップが、

前記メッセージ(20)を、前記データ部(21)及び前記第1のEC部(22)からなる第1のサブメッセージと、前記カウンタ値(23)及び前記第2のEC部(24)からなる第2のサブメッセージとを有する、少なくとも2つのサブメッセージに分割するステップ(31)と、

前記第1及び第2のサブメッセージを復号化するステップ(32)と、

前記第1のサブメッセージに存在する特定ビットの数を表す別のカウンタ値を決定するス

10

20

30

40

50

ステップ(34)と、

前記別のカウンタ値と前記第2のサブメッセージのカウンタ値を比較するステップ(35)と、

前記比較ステップが、前記別のカウンタ値が前記第2のサブメッセージのカウンタ値と異なることを示す場合に、エラーを通知するステップ(37)  
からなる、上項1の方法。

5. 前記使用するステップが、

前記第1のEC部(22)によって前記第1のサブメッセージにおけるエラーを検出するステップと、

前記第2のEC部(24)によって前記第2のサブメッセージにおけるエラーを検出する  
ステップ

10

をさらに含み、

エラー検出する前記ステップが、別のカウンタ値を決定する前記ステップに先立って行われる、上項4の方法。

6. 前記カウンタ値(23)と前記第2のEC部(24)のサイズが、前記メッセージの伝送時に使用される伝送経路の幅と比較して、前記第2のEC部により保護可能なサイズより大きく、

先行するカウンタ値および先行するのEC部に存在する特定ビットの数を表す次のカウンタ値を生成するステップと、

前記次のカウンタ値と関連づけられた次のEC部を形成するステップと、

20

前記次のカウンタ値と前記次のEC部のサイズが、前記伝送経路の幅と比較して、前記次のEC部により保護可能なサイズ程度(または、サイズ以下)になるまで、前記生成するステップと形成するステップを繰り返すステップ

からさらになる、上項1の方法。

7. 前記カウンタ値(23)および前記第2のEC部(24)のサイズが、前記メッセージの伝送時に使用される伝送経路の幅と比較して、前記第2のEC部により保護可能なサイズより大きく、

前記カウンタ値を複数のカウンタ値部分に分割するステップと、

複数のEC部を形成するステップであって、各EC部が、前記複数のカウンタ値部分の特定の1つと関連づけられる、ステップ

30

からさらになる上項1の方法であって、前記各カウンタ値部分およびそれに関連づけられたEC部のサイズが、前記伝送経路の幅と比較して、前記関連づけられたEC部により保護可能なサイズ程度(または、サイズ以下)である、方法。

8. 高次ビットを除去することにより前記カウンタ値(23)を切り捨てるステップであって、これにより、残りのビットが、前記データ部(21)および前記第1のEC部(22)を伝送するのに必要なサイクル数の2倍を示すのに十分な幅となるようにする、ステップをさらに含み、

前記切り捨てるステップが、前記第2のEC部(24)を形成する前記ステップに先立つて行われる、上項1の方法。

【0028】

40

【発明の効果】

本発明によれば、大きなオーバヘッドを生じず、かつ、実施が容易な、配線縮退故障を検出するエラー検出メカニズムを提供することができる。

【図面の簡単な説明】

【図1】メッセージを符号化する本発明のメカニズムのフロー図を示す。

【図2】図1により符号化されたメッセージを示す。

【図3】図2のメッセージを復号化する本発明のメカニズムのフロー図を示す。

【図4】図1および図3の本発明を実施するよう構成されたコンピュータシステムのプロック図を示す。

【図5】図1の本発明の1実施態様によって形成される図2のデータメッセージの例を示

50

す。

【図6】図1の本発明の、図5とは異なる実施態様によって形成される図2のデータメッセージの例を示す。

【図7】図1の本発明の、図5及び6とは異なる実施態様によって形成される図2のデータメッセージの例を示す。

【図8】図1の本発明の、図5、6及び7とは異なる実施態様によって形成される図2のデータメッセージの例を示す。

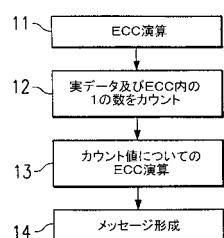
【図9】図1の本発明の、図5、6、7及び8とは異なる実施態様によって形成される図2のデータメッセージの例を示す。

【符号の説明】

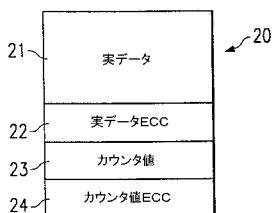
- 2 0 データメッセージ
- 2 1 実データ部
- 2 2 ECCチェックビット（第1のEC部）
- 2 3 カウンタ値
- 2 4 ECCチェックビット（第2のEC部）

10

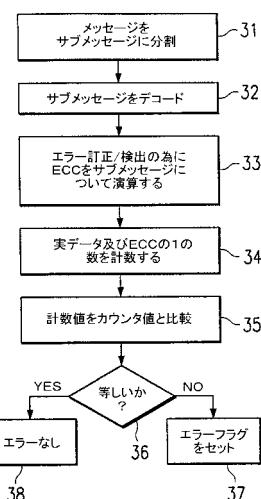
【図1】



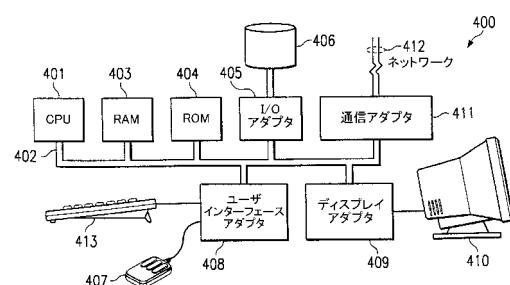
【図2】



【図3】



【図4】



【図5】

d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
d <sub>9</sub>	d <sub>8</sub>	d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>
		⋮		
d <sub>69</sub>	d <sub>68</sub>	d <sub>67</sub>	d <sub>66</sub>	d <sub>65</sub>
C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>	d <sub>71</sub>	d <sub>70</sub>
C <sub>7</sub>	C <sub>6</sub>	C <sub>5</sub>	C <sub>4</sub>	C <sub>3</sub>
AC <sub>0</sub>	C <sub>11</sub>	C <sub>10</sub>	C <sub>9</sub>	C <sub>8</sub>
AC <sub>5</sub>	AC <sub>4</sub>	AC <sub>3</sub>	AC <sub>2</sub>	AC <sub>1</sub>
		AC <sub>7</sub>	AC <sub>6</sub>	53

51  
50

d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
d <sub>9</sub>	d <sub>8</sub>	d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>
		⋮		
d <sub>59</sub>	d <sub>58</sub>	d <sub>57</sub>	d <sub>56</sub>	d <sub>55</sub>
P <sub>d</sub>	d <sub>63</sub>	d <sub>62</sub>	d <sub>61</sub>	d <sub>60</sub>
C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>
AC <sub>1</sub>	AC <sub>0</sub>	P <sub>C</sub>	C <sub>6</sub>	C <sub>5</sub>
		P <sub>AC</sub>	AC <sub>3</sub>	AC <sub>2</sub>
			74	76

71  
7072  
73  
75

【図6】

d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
d <sub>9</sub>	d <sub>8</sub>	d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>
		⋮		
d <sub>69</sub>	d <sub>68</sub>	d <sub>67</sub>	d <sub>66</sub>	d <sub>65</sub>
C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>	d <sub>71</sub>	d <sub>70</sub>
C <sub>7</sub>	C <sub>6</sub>	C <sub>5</sub>	C <sub>4</sub>	C <sub>3</sub>
AC <sub>2</sub>	AC <sub>1</sub>	AC <sub>0</sub>	C <sub>9</sub>	C <sub>8</sub>
		AC <sub>5</sub>	AC <sub>4</sub>	AC <sub>3</sub>
				63

60  
62

【図8】

d <sub>4</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
d <sub>9</sub>	d <sub>8</sub>	d <sub>7</sub>	d <sub>6</sub>	d <sub>5</sub>
		⋮		
d <sub>59</sub>	d <sub>58</sub>	d <sub>57</sub>	d <sub>56</sub>	d <sub>55</sub>
P <sub>d</sub>	d <sub>63</sub>	d <sub>62</sub>	d <sub>61</sub>	d <sub>60</sub>
P <sub>CL</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>
P <sub>CH</sub>		C <sub>6</sub>	C <sub>5</sub>	C <sub>4</sub>
				85

80  
82  
84  
8581  
83

【図9】

d <sub>16</sub>					-	·	·	·	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>
d <sub>33</sub>						·	·	·		d <sub>17</sub>	
d <sub>50</sub>						·	·	·		d <sub>34</sub>	
d <sub>67</sub>						·	·	·		d <sub>51</sub>	
	C <sub>7</sub>	C <sub>6</sub>	C <sub>5</sub>	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>	d <sub>71</sub>	d <sub>70</sub>	d <sub>69</sub>
											d <sub>68</sub>

90  
91  
92

---

フロントページの続き

(72)発明者 デベンドラ・ダス・シャーマ

アメリカ合衆国カリフォルニア州 95051 , サンタクララ , グラナダ・アベニュー・ナンバー 1  
74 , 3480

審査官 矢頭 尚之

(56)参考文献 特開昭 60 - 014347 (JP, A)

特開平 05 - 157816 (JP, A)

南谷 崇 , フォールトトレランス技術の最近の動向 (1) - セルフチェックングプロセッサ - ,  
電子情報通信学会誌 , 1990年 9月 , Vol.73 No.9 , pp.991-999

(58)調査した分野(Int.Cl. , DB名)

H03M 13/15