

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2013-153394  
(P2013-153394A)

(43) 公開日 平成25年8月8日(2013.8.8)

(51) Int.Cl.	F I			テーマコード (参考)		
<b>H04L 1/00 (2006.01)</b>	H04L 1/00	B	5K014			
<b>H03M 7/14 (2006.01)</b>	H03M 7/14	B				

審査請求 未請求 請求項の数 7 O L (全 10 頁)

(21) 出願番号	特願2012-14181 (P2012-14181)	(71) 出願人	899000079
(22) 出願日	平成24年1月26日 (2012.1.26)		学校法人慶應義塾
			東京都港区三田2丁目15番45号
特許法第30条第1項適用申請有り 一般社団法人情報処理学会が2011年(平成23年)11月21日に発行した、情報処理学会研究報告. [ハイパフォーマンスコンピューティング] 2011-HPC-132 (8)にて発表		(74) 代理人	100098545
			弁理士 阿部 伸一
		(74) 代理人	100087745
			弁理士 清水 善廣
		(74) 代理人	100106611
			弁理士 辻田 幸史
		(74) 代理人	100111006
			弁理士 藤江 和典
		(74) 代理人	100116241
			弁理士 金子 一郎

最終頁に続く

(54) 【発明の名称】 デジタル通信の伝送路符号の変換方法及びこの変換方法を行う符号の生成方法

(57) 【要約】

【課題】 エラー検出及びエラー訂正機能付きのエンベデッドクロック方式の伝送路符号化を実現することで、信頼性が高くレイテンシも短いシリアルデータ通信を実現する。

【解決手段】 シリアルデータの中にクロックを埋め込み、かつ、その際DCバランシング(0と1の割合を同じにする)を行い、伝送中に発生するビット誤りの検出を行う符号化と伝送中に発生するビット誤りの訂正を行う符号化を同時に備えるような符号を発見し、その符号を用いて通信を行う。実施例としては、4ビットのデータを10ビットのビット列へ符号化し、DCバランシングを確保して通信用クロックを通信データに埋め込むことを実現すると同時に、符号化した各ビット列同士のハミング距離を4以上に保つことで、伝送中に発生する2ビット以内のビット誤りの検出、1ビットのビット誤りの訂正を可能にする。

【選択図】 図1

4ビット	10ビット	4ビット	10ビット
0000	1100101100	1000	1001110001
0001	1011001100	1001	0111000110
0010	1100110010	1010	1010110100
0011	0110011100	1011	1101001010
0100	0111010001	1100	1011010010
0101	1100011001	1101	1001100110
0110	0101110100	1110	1010101001
0111	1101000101	1111	0110101010

## 【特許請求の範囲】

## 【請求項 1】

デジタル通信の伝送路符号の変換方法であって、  
シリアル通信のデータの中にクロックを埋め込み、前記クロックの埋め込みの際に DC バランシングを行い、前記データと前記クロックの転送を同じ配線で行うことが可能な符号化手段と、  
前記符号化手段で符号化された符号語について、伝送中に発生するビット誤りの検出を行う誤り検出手段と、  
前記誤り検出手段で検出された前記ビット誤りの訂正を行う誤り訂正手段と  
を全て同時に備えたことを特徴とするデジタル通信の伝送路符号の変換方法。

10

## 【請求項 2】

前記符号化手段では、変換テーブルを用いて 4 ビットの前記データを 10 ビットのビット列の前記符号語に変換することを特徴とする請求項 1 に記載のデジタル通信の伝送路符号の変換方法。

## 【請求項 3】

それぞれの前記符号語は、連続する 0 または 1 が 5 ビット以下であることを特徴とする請求項 2 に記載のデジタル通信の伝送路符号の変換方法。

## 【請求項 4】

それぞれの前記符号語は、符号化した前記各ビット列同士のハミング距離が 4 以上であり、前記誤り検出手段では、10 ビット中 2 ビット以内のビット誤りの検出を行うことを特徴とする請求項 3 に記載のデジタル通信の伝送路符号の変換方法。

20

## 【請求項 5】

前記誤り訂正手段では、10 ビット中 1 ビットのビット誤りの訂正を行うことを特徴とする請求項 4 に記載のデジタル通信の伝送路符号の変換方法。

## 【請求項 6】

請求項 1 から請求項 5 のいずれかに記載のデジタル通信の伝送路符号の変換方法を行う符号の生成方法であって、

それぞれの前記符号語を、

10 ビット内で 0 と 1 とを同数とし、

前記符号語内の任意の 1 ビットを反転させて得られるビット列において、連続する 0 または 1 を 5 ビット以下とし、

30

2 つの前記符号語を連結して得られる 20 ビットのビット列内の任意の 1 ビットを反転させて得られるビット列において、連続する 0 または 1 を 5 ビット以下とし、

ハミング距離を 4 以上とし、

2 つの前記符号語を連結して得られる 20 ビットの前記ビット列内の任意の 2 ビットを反転させて得られるビット列において、連続する 0 または 1 を 5 ビット以下とし、

2 つの前記符号語を連結して得られる 20 ビットの前記ビット列内の一方の前記符号語内の任意の 1 ビットを反転させるとともに、他方の前記符号語内の任意の 1 ビットを反転させたときに、2 つの反転ビット同士を 4 ビット以上

40

として生成することを特徴とする符号の生成方法。

## 【請求項 7】

請求項 1 に記載のデジタル通信の伝送路符号の変換方法を行う符号の生成方法であって、

C ビットの送信データを D ビットの符号語に変化する場合において、

それぞれの前記符号語を、

符号語 D ビット内で 0 と 1 とを同数とし、

前記符号語内の任意の 1 ビットを反転させて得られるビット列において、連続する 0 または 1 を指定ビット (E ビット) 以下とし、

2 つの前記符号語を連結して得られる 2 倍の D ビットのビット列内の任意の 1 ビットを反転させて得られるビット列において、連続する 0 または 1 を E ビット以下とし、

50

ハミング距離を指定する数F以上とし(F-E)、  
 2つの前記符号語を連結して得られる2倍のDビットの前記ビット列内の任意の2ビットを反転させて得られるビット列において、連続する0または1をEビット以下とし、  
 2つの前記符号語を連結して得られる2倍のDビットの前記ビット列内の一方の前記符号語内の任意の1ビットを反転させるとともに、他方の前記符号語内の任意の1ビットを反転させたときに、2つの反転ビット同士をFビット以上として生成することを特徴とする符号の生成方法。

【発明の詳細な説明】

【技術分野】

10

【0001】

本発明は、特に、シリアル通信においてデータにクロックを埋め込んで送信するエンベデッドクロック方式に関し、さらに、デジタル通信の符号化に関わり、より詳細には、伝送路符号、誤り検出符号、誤り訂正符号に関するものである。

【背景技術】

【0002】

デジタル通信では、伝送周波数の高速化に伴い、データにクロックを埋め込んで送信するエンベデッドクロック方式を用いた高速シリアル通信への要求が高まっている。エンベデッドクロック方式を用いた実用的な高速シリアル通信を実現するためには、送信信号のDCバランシング(0と1を同程度にする方法)を確保するための伝送路符号が必要となる。伝送路符号の実例として、「Byte Oriented DC Balanced(0,4)8B10B Partition Block Transmission Code」と題する、FranaszekおよびWidmerによる米国特許第4,486,739号に記載の8B10B伝送コードがPCI ExpressやInfiniBandなどの通信規格で採用されている。

20

【0003】

さらに、現代のデジタル通信においては、伝送路でのエラー検出やエラー訂正は必須の機能となっている。例えば、PCI Expressでは、CRCを用いてデータリンクレイヤでエラー検出を行っている。受信側でエラーが検出された場合、送信側にシーケンスナンバと共にNAK(Not Acknowledge)を返すことで、送信側が再送を行う。

30

【0004】

信頼性のない伝送路では、伝送中に発生するビット誤りを検出、および訂正する必要がある。誤り検出符号、および誤り訂正符号を用いることで、伝送中に発生するビット誤りを検出、および訂正が可能である。誤り検出符号、および誤り訂正符号の実例として、ハミング符号(非特許文献1)や、BCH符号(非特許文献2)などがある。

【先行技術文献】

【非特許文献】

【0005】

【非特許文献1】R.W.Hamming; Error Detecting and Error Correcting Codes, Bell System Technical Journal, Vol.29, pp.147-160(1950)

40

【非特許文献2】R.C.Bose and D.K.Ray-Chaudhuri; On A Class of Error Correcting Binary Group Codes, Information and Control 3, Vol.1, pp.68-79(1960)

【発明の開示】

【発明が解決しようとする課題】

【0006】

エンベデッドクロック方式で用いられる伝送路符号では、DCバランシングを確保する

50

ために伝送データをより長いビット列に符号化するという特性上、伝送中に発生するビット誤りが、復号後により大きな誤りとなる場合がある。(以下、元データを符号化したデータ(ビット列)を符号語と呼称する。)例えば伝送中にビット誤りが発生し、正しい符号語からハミング距離が1離れたとすると、誤った符号語を復号した結果、復号されたデータは正しいデータからハミング距離が1以上離れる可能性がある。この結果、復号後の誤り検出、および訂正のコストが非常に高くなる。8B10B伝送コードのようなエンベデッドクロック方式でエラー訂正を同時に行いたい場合には、ビット単位でのエラー訂正は用いることが困難であり、リードソロモン符号のようなブロックエラー訂正符号をパケット単位(大きなブロック単位)で後段に行う必要がある。この場合には、パケットをすべて受信しきらないとエラー訂正を行うことができず、レイテンシが長くなり、リアルタイム通信等には不向きとなる。

ここで、近年、ヒューマノイドロボット等の複雑なリアルタイムシステムの出現により、単一プロセッサで行う処理の限界や、システムの耐故障性等を向上するために、プロセッサを複数用いた分散リアルタイムシステムへの要求が高まっている。

分散リアルタイムシステムでは、ノード内の処理においてリアルタイム性を満たすだけでなく、ノード間の通信においてもリアルタイム性を保証しなければならない。このような背景から、分散リアルタイムシステムのための通信機構としてレスポンシブリンク(Responsive Link)が国内外で標準化されている。具体的には、レスポンシブリンクは、国内ではIPJS-TS 0006として、国際的にはISO/IEC 24740として標準化されている。

#### 【0007】

従って、本発明の目的は、エンベデッドクロック方式の伝送路符号に、DCバランシング、誤り検出および誤り訂正機能を内包することで、誤り検出および誤り訂正を同時に実現することにある。具体的には、従来手法では不可能であった、データにクロックを埋め込んで通信、DCバランシング、エラー検出機能、エラー訂正機能、を一つの符号で同時に行う。エラー検出及びエラー訂正機能付きのエンベデッドクロック方式の伝送路符号化を実現することで、信頼性が高くレイテンシも短いシリアルデータ通信を実現する。

#### 【課題を解決するための手段】

#### 【0008】

本発明によるデジタル通信の伝送路符号の変換方法は、デジタル通信の伝送路符号の変換方法であって、シリアル通信のデータの中にクロックを埋め込み、前記クロックの埋め込みの際にDCバランシングを行い、前記データと前記クロックの転送を同じ配線で行うことが可能な符号化手段と、前記符号化手段で符号化された符号語について、伝送中に発生するビット誤りの検出を行う誤り検出手段と、前記誤り検出手段で検出された前記ビット誤りの訂正を行う誤り訂正手段とを備えたことを特徴とする。

また、本発明によるデジタル通信の伝送路符号の一例(以下、4B10B符号と称す)を以下に示す。前記符号化手段では、変換テーブルを用いて4ビットの送信データを10ビットの前記符号語に変換し、DCバランシングを実現しつつ、データ内にクロックを埋め込む。

また、本発明によるデジタル通信の伝送路符号の一つである4B10B符号では、それぞれの前記符号語は、連続する0または1が5ビット以下であることを特徴とし、伝送中に発生する10ビット中1ビット以内のビット誤りが発生してもエンベデッドクロックを維持できる。

また、本発明によるデジタル通信の伝送路符号の一つである4B10B符号では、それぞれの前記符号語は、前記符号化した各ビット列同士のハミング距離が4以上であり、前記誤り検出手段では、10ビット中2ビット以内のビット誤りの検出を行うことを特徴とする。

また、本発明によるデジタル通信の伝送路符号の一つである4B10B符号では、前記誤り訂正手段では、10ビット中1ビットのビット誤りの訂正を行うことを特徴とする。

また、本発明によるデジタル通信の伝送路符号の一つである4B10B符号化を行う符号

の生成方法は、それぞれの前記符号語を、10ビット内で0と1とを同数とし、前記符号語内の任意の1ビットを反転させて得られるビット列において、連続する0または1を5ビット以下とし、2つの前記符号語を連結して得られる20ビットのビット列内の任意の1ビットを反転させて得られるビット列において、連続する0または1を5ビット以下とし、ハミング距離を4以上とし、2つの前記符号語を連結して得られる20ビットの前記ビット列内の任意の2ビットを反転させて得られるビット列において、連続する0または1を5ビット以下とし、2つの前記符号語を連結して得られる20ビットの前記ビット列内の一方の前記符号語内の任意の1ビットを反転させるとともに他方の前記符号語内の任意の1ビットを反転させたときに、2つの反転ビット同士を4ビット以上として生成することを特徴とする。

10

また、本発明によるデジタル通信の伝送路符号の変換方法を行う符号の生成方法は、それぞれの前記符号語を、10ビット内で0と1とを同数とし、前記符号語内の任意の1ビットを反転させて得られるビット列において、連続する0または1を5ビット以下とし、

2つの前記符号語を連結して得られる20ビットのビット列内の任意の1ビットを反転させて得られるビット列において、連続する0または1を5ビット以下とし、ハミング距離を4以上とし、2つの前記符号語を連結して得られる20ビットの前記ビット列内の任意の2ビットを反転させて得られるビット列において、連続する0または1を5ビット以下とし、2つの前記符号語を連結して得られる20ビットの前記ビット列内の一方の前記符号語内の任意の1ビットを反転させるとともに他方の前記符号語内の任意の1ビットを反転させたときに、2つの反転ビット同士を4ビット以上として生成することを特徴とする。

20

#### 【発明の効果】

#### 【0009】

本発明によれば、ラインコード自体にエラー検出機能とエラー訂正機能の両方を持たせて信頼性の高い通信を実現することができる。

#### 【図面の簡単な説明】

#### 【0010】

【図1】本発明による4B10B符号を示す変換テーブル

【図2】本発明によるビット誤りによるハミング距離と、誤り検出、および誤り訂正の概要を示す図

30

【図3】本発明によるデコードのフローチャート

#### 【発明を実施するための形態】

#### 【0011】

本発明に従った符号化(4B10B符号化)を、図1に示す変換テーブルを用いて例示する。図1では、4ビットの元データを10ビットのデータ(符号語)へ変換することで、元データ内にクロックを埋め込み、DCバランス(0と1の割合をまったく同じにする)を実現し、10ビットのデータ中の2ビットの誤りを検出し、10ビットのデータ中の1ビットの誤り訂正をするという各機能を同時に実現する。

#### 【0012】

40

図1に示す変換テーブルによれば、符号化した10ビットの各ビット列において連続する0または1は必ず5ビット以下になるようにしつつ、クロックを埋め込んでいる。さらには、この符号語内で任意の1ビットが反転しても連続する0または1は5ビット以下になるようにしている。例えば、符号語が「1100110010」であったとした場合、符号語内の一つの任意の「1」に誤りが生じて「0」に反転した場合であっても、連続するビットは5ビット以下となる。

同様に、任意の符号化した10ビットと別の符号化した10ビットの符号語間においても連続する0または1は必ず5ビット以下になるようにしている。この結合部分においても任意の1ビットが反転しても連続する0または1は5ビット以下になるようにしている。したがって、各符号語で誤りがある場合は誤りビット同士が4ビット以上離れているとき

50

に不連続性を保証する。例えば、2つの符号語が、「1100110010」と「1001110001」であった場合、「1100110010」が「1100110000」となり、「1001110001」が「1000110001」となっても、2つの誤りビットが4ビット以上離れていれば、連続するビットは5ビット以下となる。

#### 【0013】

図1に示す変換テーブルによれば、符号化した10ビットの符号語内の0と1の数を必ず同数になるようにし、DCバランシングを実現する。

#### 【0014】

図1に示す変換テーブルによれば、符号化した10ビットの各ビット列同士のハミング距離は4以上離れており、伝送中に発生する10ビット中1ビットの誤りは、正しいビット列が一意に決まるため、誤りの訂正が可能である。ビット誤りによるハミング距離と、誤り検出、および誤り訂正の概要を図2に示す。

10

#### 【0015】

本発明に従えば、符号化を行う符号器は図1に示した変換テーブルに基づき、4ビットのデータから10ビットの符号語を生成する。復号化を行う復号器は、入力された10ビットの符号語を図1に示した変換テーブルの全てのビット列とのハミング距離を求め、変換テーブル内でハミング距離が最も小さいものが選択され、復号される。最も小さいハミング距離が0だった場合、伝送中のビット誤りは存在せず、正しく復号化される。ハミング距離が1だった場合、伝送中に1ビットの誤りが発生したことになり、誤りは訂正されて正しく復号化される。ハミング距離が2以上だった場合、伝送中に2ビット以上の誤りが発生したことになり、誤りは検出できるが、訂正はできない。10ビット中2ビットまでの誤りを検出することができる。図3に復号のフローを示す。

20

#### 【0016】

図1に示す4B10B符号化変換テーブルを生成するアルゴリズムを以下に示す。

操作1．10ビットの符号語から以下の条件1、2、3をすべて満たすものだけを抽出し、この集合をS0とする。

条件1．10ビット内で0と1が同数である。

条件2．符号語内の任意のビットを1ビット反転させて得られるビット列において、連続する0または1が5ビット以下である。

条件3．符号自身を2つ連結して得られる20ビットのビット列内の任意のビットを1ビット反転させて得られるビット列において、連続する0または1が5ビット以下である。

30

操作2．S0の各符号語について、S0中の他の符号語すべてに対して、それぞれ、以下の条件4、5のすべてを満たす符号数をカウントする。

条件4．ハミング距離が4以上である。

条件5．2つの符号語AとBを連結して得られる20ビットの2つのビット列(A BとB A)の任意のビットを最大2ビット反転させて得られるビット列において、連続する0または1が5ビット以下である。ただし、2ビット反転する際は反転ビットは符号語Aに1ビット、Bに1ビットかつ2つの反転ビット同士が4ビット以上離れているものとする。

40

操作3．S0の中で操作2によって得られたカウント数Nが16以上のものだけを抽出し、この集合をS1とする。

操作4．同様に操作2と操作3を集合の要素数が収束するまで繰り返す。最終的に得られた集合をScとする。

操作5．Scを操作2によって得られたカウント数について昇順にソートし、順に配列Sに格納する。

操作6．配列の最初の要素S[0]から順に一つずつ符号語を取り出し、集合Stの要素として加える。ただし、Stは初め空集合であり、また加える符号語はStの全符号語に対して、条件4と条件5すべてを満たす必要があり、条件を満たさない符号語はStに加えない。

50

操作 7 . 操作 6 によって得られた  $S_t$  を  $S_{max}$  とし、その要素数を  $M$  とする。

操作 8 . 同様に  $S[1]$  から順に操作 6 を行い、 $S_t$  の要素数が  $M$  を上回れば、 $S_{max}$  および  $M$  を更新する。

操作 9 . 操作 8 を配列  $S$  の最後まで同様に繰り返す。

操作 10 .  $S_{max}$  が 4 B 10 B 符号化の符号空間となる ( $M$  が  $N (= 16)$  以上となっている)。

#### 【0017】

このアルゴリズムにより、データ内にクロックを埋め込み (連続する 0 または 1 は 5 ビット以下)、DC バランシング (符号化後の 10 ビット中の 0 と 1 が同数)、10 ビット中 2 ビットの誤り検出、10 ビット中 1 ビットの誤り訂正の各機能を同時に実現するように、4 ビットのデータを 10 ビットのデータ (符号語) へ変換する符号を見つけることができる。

10

#### 【0018】

伝送路符号としてよく使用される 8 B 10 B や NRZI (Non Return to Zero Inverted) + Bit Staffing 等の符号では、伝送路上のデータ 1 ビットが化けた場合でも、復号後に複数ビットのエラーとして検出される場合があるが、本発明によれば、復号時には伝送されてきた 10 ビットをテーブル内から検索し、ハミング距離が 1 以内のものはエラーが訂正され正しく復号される。

#### 【0019】

より一般的に  $C$  ビットの元データを  $D$  ビットの符号語に変換する変換テーブルを生成するアルゴリズムを以下に示す ( $C < D$ )。

20

操作 1 .  $D$  ビットの符号語から以下の条件 1、2、3 をすべて満たすものだけを抽出し、この集合を  $S_0$  とする。

条件 1 .  $D$  ビット内で 0 と 1 が同数である。

条件 2 . 符号語内の任意のビットを 1 ビット反転させて得られるビット列において、連続する 0 または 1 が指定ビット ( $E$  ビット) 以下である。

条件 3 . 符号自身を 2 つ連結して得られる 2 倍の  $D$  ビットのビット列内の任意のビットを 1 ビット反転させて得られるビット列において、連続する 0 または 1 が  $E$  ビット以下である。

操作 2 .  $S_0$  の各符号語について、 $S_0$  中の他の符号語すべてに対して、それぞれ、以下の条件 4、5 のすべてを満たす符号数をカウントする。

30

条件 4 . ハミング距離が指定した  $F$  以上である ( $F \geq E$ )。

条件 5 . 2 つの符号語  $A$  と  $B$  を連結して得られる 2 倍の  $D$  ビットの 2 つのビット列 ( $AB$  と  $BA$ ) の任意のビットを最大 2 ビット反転させて得られるビット列において、連続する 0 または 1 が  $E$  ビット以下である。ただし、2 ビット反転する際は反転ビットは符号語  $A$  に 1 ビット、 $B$  に 1 ビットかつ 2 つの反転ビット同士が  $F$  ビット以上離れているものとする。

操作 3 .  $S_0$  の中で操作 2 によって得られたカウント数  $N$  が 2 の  $C$  乗以上のものだけを抽出し、この集合を  $S_1$  とする。

操作 4 . 同様に操作 2 と操作 3 を集合の要素数が収束するまで繰り返す。最終的に得られた集合を  $S_c$  とする。

40

操作 5 .  $S_c$  を操作 2 によって得られたカウント数について昇順にソートし、順に配列  $S$  に格納する。

操作 6 . 配列の最初の要素  $S[0]$  から順に一つずつ符号語を取り出し、集合  $S_t$  の要素として加える。ただし、 $S_t$  は初め空集合であり、また加える符号語は  $S_t$  の全符号語に対して、条件 4 と条件 5 をすべて満たす必要があり、条件を満たさない符号語は  $S_t$  に加えない。

操作 7 . 操作 6 によって得られた  $S_t$  を  $S_{max}$  とし、その要素数を  $M$  とする。

操作 8 . 同様に  $S[1]$  から順に操作 6 を行い、 $S_t$  の要素数が  $M$  を上回れば、 $S_{max}$  および  $M$  を更新する。

50

操作 9 . 操作 8 を配列 S の最後まで同様に繰り返す。

操作 10 . S m a x が C B D B 符号化の符号空間となる ( M が N 以上となっている ) 。

【 0 0 2 0 】

このアルゴリズムにより、データ内にクロックを埋め込み ( 連続する 0 または 1 は E ビット以下 )、D C バランシング ( 符号化後の D ビット中の 0 と 1 が同数 )、D ビット中 2 ビットの誤り検出、D ビット中 1 ビットの誤り訂正の各機能を同時に実現するように、C ビットのデータを D ビットのデータへ変換する符号を見つけることができる。

【 0 0 2 1 】

伝送路符号としてよく使用される 8 B 1 0 B や N R Z I ( N o n R e t u r n t o Z e r o I n v e r t e d ) + B i t S t a f f i n g 等の符号では、伝送路上のデータ 1 ビットが化けた場合でも、復号後に複数ビットのエラーとして検出される場合があるが、本発明によれば、復号時には伝送されてきた D ビットをテーブル内から検索し、ハミング距離が 1 以内のものはエラーが訂正され正しく復号される。

【 0 0 2 2 】

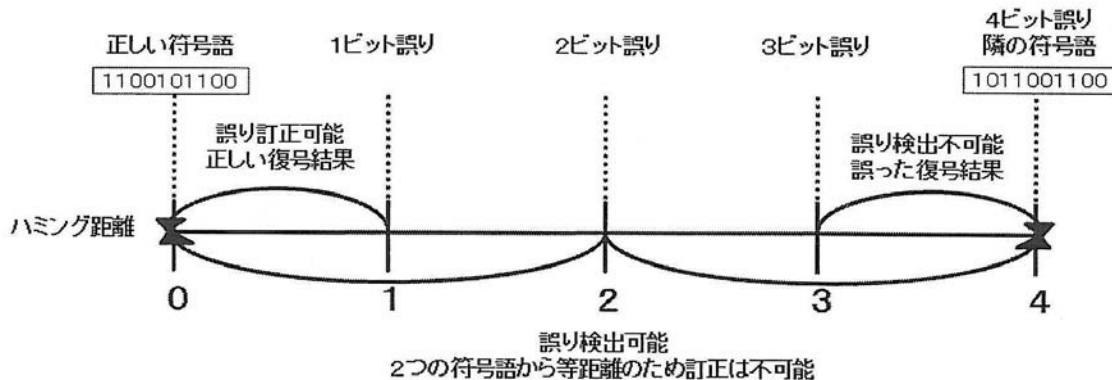
以上、本発明の実施形態を説明したが、この実施形態は本発明の説明のための例示にすぎず、本発明の範囲をこの実施形態にのみ限定する趣旨ではない。例えば、エラー検出及びエラー訂正の強度の異なる符号化 ( 何ビットエラー検出及び何ビットエラー訂正 ) は容易に実現できる。本発明は、その要旨を逸脱することなく、その他の様々な態様でも実施することができる。

10

【 図 1 】

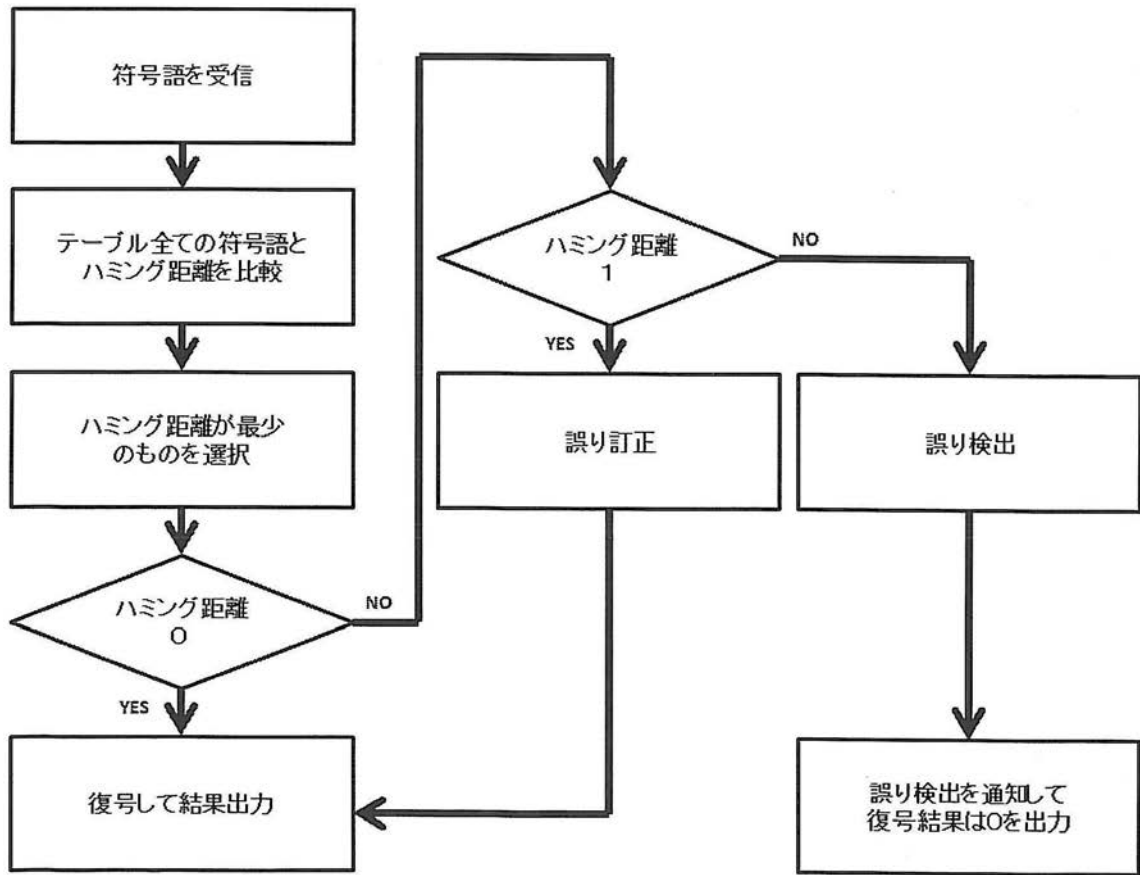
4ビット	10ビット	4ビット	10ビット
0000	1100101100	1000	1001110001
0001	1011001100	1001	0111000110
0010	1100110010	1010	1010110100
0011	0110011100	1011	1101001010
0100	0111010001	1100	1011010010
0101	1100011001	1101	1001100110
0110	0101110100	1110	1010101001
0111	1101000101	1111	0110101010

【 図 2 】





【 図 3 】



フロントページの続き

(72)発明者 山 崎 信行

神奈川県横浜市港北区日吉3丁目14番1号 慶應義塾大学 理工学部内

Fターム(参考) 5K014 BA05