

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4602406号
(P4602406)

(45) 発行日 平成22年12月22日 (2010.12.22)

(24) 登録日 平成22年10月8日 (2010.10.8)

(51) Int. Cl. F I
HO3M 13/19 (2006.01) HO3M 13/19

請求項の数 10 (全 17 頁)

(21) 出願番号	特願2007-525667 (P2007-525667)	(73) 特許権者	390009597
(86) (22) 出願日	平成17年8月3日 (2005.8.3)		モトローラ・インコーポレイテッド
(65) 公表番号	特表2008-510379 (P2008-510379A)		MOTOROLA INCORPORATED
(43) 公表日	平成20年4月3日 (2008.4.3)		ED
(86) 国際出願番号	PCT/US2005/027686		アメリカ合衆国イリノイ州シャンバーグ、
(87) 国際公開番号	W02006/020484		イースト・アルゴンクイン・ロード1303
(87) 国際公開日	平成18年2月23日 (2006.2.23)	(74) 代理人	100116322
審査請求日	平成19年3月7日 (2007.3.7)		弁理士 桑垣 衛
(31) 優先権主張番号	60/600,953	(72) 発明者	ブランケンシップ、ユフェイ ダブリュ.
(32) 優先日	平成16年8月12日 (2004.8.12)		アメリカ合衆国 60107 イリノイ州
(33) 優先権主張国	米国 (US)		ストリームウッド エス. グリーン メ
(31) 優先権主張番号	11/070,129		ドウス ブルバード 1687
(32) 優先日	平成17年3月1日 (2005.3.1)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 データをエンコード及びデコードするための方法並びに装置

(57) 【特許請求の範囲】

【請求項1】

情報ブロックに基づきパリティ検査ビットを生成する送信機を動作させる方法であって、

同じ符号化率を有する複数の符号のうち最大の符号長さを有する符号に対する1組のシフトサイズ $p(i, j)$ を有する基本モデル行列を定義する工程と、

1組のシフトサイズ $p(i, j)$ に基づき、前記複数の符号のうち最大の符号長さを有する符号以外の各符号に対するシフトサイズ $p(f, i, j)$ を決定する工程と、

[ここで、 f は前記複数の符号のうちの第 f 番目の符号に対する指標であり、 $p(f, i, j) = F(p(i, j), z_0 / z_f)$ であり、 z_0 / z_f は第 f 番目の符号のシフトサイズに対する最大の符号長さを有する符号のシフトサイズのスケーリング比を示す係数である]

情報ブロック $s = (s_0, \dots, s_{k_f - 1})$ を受信する工程と、

$p(f, i, j)$ によって定義されるモデル行列を用いてパリティ検査ビットを決定する工程と、

情報ブロックに加えてパリティ検査ビットを送信する工程と、からなる方法。

[ここで $\{p(i, j)\}$ は次の通りである]

1に記載の方法。

【請求項6】

同じ符号化率を有する複数の符号のうち最大の符号長さを有する符号に対する1組のシフトサイズ $p(i, j)$ を有する基本モデル行列を記憶するための記憶手段と、

情報ブロック $s = (s_0, \dots, s_{k_f-1})$ 及び基本モデル行列を受信し、1組のシフトサイズ $p(i, j)$ に基づき、前記複数の符号のうち最大の符号長さを有する符号以外の各符号に対するシフトサイズ $p(f, i, j)$ を決定するマイクロプロセッサと、

[ここで、 f は前記複数の符号のうちの第 f 番目の符号に対する指標であり、 $p(f, i, j) = F(p(i, j), z_0 / z_f)$ であり、 z_0 / z_f は第 f 番目の符号のシフトサイズに対する最大の符号長さを有する符号のシフトサイズのスケーリング比を示す係数である]

10

マイクロプロセッサは $p(f, i, j)$ によって定義されるモデル行列に基づくパリティ検査ビットと、情報ブロック $s = (s_0, \dots, s_{k_f-1})$ とを出力することと、からなる装置。

[ここで $\{p(i, j)\}$ は次の通りである]

【表2】

```

-1 94 73 -1 -1 -1 -1 -1 55 83 -1 -1 7 0 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 27 -1 -1 -1 22 79 9 -1 -1 -1 12 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 24 22 81 -1 33 -1 -1 -1 0 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1
61 -1 47 -1 -1 -1 -1 -1 65 25 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1
-1 -1 39 -1 -1 -1 84 -1 -1 41 72 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1
-1 -1 -1 -1 46 40 -1 82 -1 -1 -1 79 0 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1
-1 -1 95 53 -1 -1 -1 -1 -1 14 18 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1
-1 11 73 -1 -1 -1 2 -1 -1 47 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1
12 -1 -1 -1 83 24 -1 43 -1 -1 -1 51 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1
-1 -1 -1 -1 -1 94 -1 59 -1 -1 70 72 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1
-1 -1 7 65 -1 -1 -1 -1 39 49 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0
43 -1 -1 -1 -1 66 -1 41 -1 -1 -1 26 7 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

```

20

【請求項7】

$p(f, i, j)$ は次式の通りである請求項6に記載の装置。

【数4】

$$p(f, i, j) = \begin{cases} p(i, j), & p(i, j) \leq 0 \\ \frac{p(i, j)z_f}{z_0}, & p(i, j) > 0 \end{cases}$$

30

[ここで、 x は床関数を表す]

【請求項8】

$p(f, i, j) = F(p(i, j), z_0 / z_f)$ の値を有するリードオンリメモリを含む請求項6に記載の装置。

40

【請求項9】

z_0 / z_f の値を有するリードオンリメモリを含む請求項6に記載の装置。

【請求項10】

情報ブロック $s = (s_0, \dots, s_{k_f-1})$ を推定する受信機を動作させる方法であって、

信号ベクトルを受信する工程と、

同じ符号化率を有する複数の符号のうち最大の符号長さを有する符号に対する1組のシ

50

フトサイズ $p(i, j)$ を有する基本モデル行列を定義する工程と、

1組のシフトサイズ $p(i, j)$ に基づき、前記複数の符号のうち最大の符号長さを有する符号以外の各符号に対するシフトサイズ $p(f, i, j)$ を決定する工程と、

[ここで、 f は前記複数の符号のうちの第 f 番目の符号に対する指標であり、 $p(f, i, j) = F(p(i, j), z_0 / z_f)$ であり、 z_0 / z_f は第 f 番目の符号のシフトサイズに対する最大の符号長さを有する符号のシフトサイズのスケール比を示す係数である]

$p(f, i, j)$ によって定義されるモデル行列と受信した信号ベクトルとに基づき、情報ブロック $s = (s_0, \dots, s_{k_f-1})$ を推定する工程と、からなる方法。

[ここで $\{p(i, j)\}$ は次の通りである]

【表3】

```

-1 94 73 -1 -1 -1 -1 -1 55 83 -1 -1 7 0 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 27 -1 -1 -1 22 79 9 -1 -1 -1 12 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 24 22 81 -1 33 -1 -1 -1 0 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1
61 -1 47 -1 -1 -1 -1 -1 65 25 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1
-1 -1 39 -1 -1 -1 84 -1 -1 41 72 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1
-1 -1 -1 -1 46 40 -1 82 -1 -1 -1 79 0 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1
-1 -1 95 53 -1 -1 -1 -1 -1 14 18 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1
-1 11 73 -1 -1 -1 2 -1 -1 47 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1
12 -1 -1 -1 83 24 -1 43 -1 -1 -1 51 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1
-1 -1 -1 -1 -1 94 -1 59 -1 -1 70 72 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1
-1 -1 7 65 -1 -1 -1 -1 39 49 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0
43 -1 -1 -1 -1 66 -1 41 -1 -1 -1 26 7 -1 -1 -1 -1 -1 -1 -1 -1 -1 0

```

【発明の詳細な説明】

【技術分野】

【0001】

本発明はデータのエンコード及びデコードに関する。より詳細には、本発明は低密度パリティ検査(LDPC)符号を利用するデータのエンコード及びデコードのための方法並びに装置に関する。

【背景技術】

【0002】

低密度パリティ検査(LDPC)符号は、パリティ検査行列 H によって指定される線形ブロック符号である。一般に、LDPC符号は、 $q = 2$ のガロア域 $GF(q)$ によって定義される。 $q = 2$ の場合、符号は二値符号である。全ての線形ブロック符号は、 k ビットの情報ベクトル $s_{1 \times k}$ と符号生成行列 $G_{k \times n}$ との積として記述可能であり、 n ビットの符号語 $x_{1 \times n}$ を生成する。ここで符号化率は、 $r = k/n$ である。符号語 x はノイズを有するチャンネルを通じて送信され、受信信号ベクトル y は情報ベクトル $s_{1 \times k}$ を推定するためにデコーダへ渡される。

【0003】

n 次元空間が与えられると、 G の行は k 次元の符号語部分空間 C を張り、パリティ検査行列 $H_{m \times n}$ の行は、 m 次元の双対空間 C^\perp を張る。ここで、 $m = n - k$ である。 $x = sG$ 、かつ、 $GH^T = 0$ ならば、部分空間 C における全ての符号語に対し、 $xH^T = 0$ となる。ここで、“ T ”(又は斜体の“ T ”)は行列の転置を表す。LDPC符号の説明では、これは一般に次のように記述される。

【0004】

【数1】

$$Hx^T = 0^T \quad (1)$$

10

20

30

40

50

ここで、 0 は全てゼロの行ベクトルであり、符号語 $x = [s \ p] = [s_0, s_1, \dots, s_{k-1}, p_0, p_1, \dots, p_{m-1}]$ である。ここで、 p_0, \dots, p_{m-1} はパリティ検査ビットである。また、 s_0, \dots, s_{k-1} はシステムティックビットであり、情報ブロック内の情報ビットと等しい。

【0005】

LDP C 符号においては、 H における非ゼロのエントリの密度は低く、即ち、 H にはわずかな割合の 1 しか存在せず、密な H を用いるよりも優れた誤り訂正性能と、簡単なデコードとが可能となる。また、パリティ検査行列は二分グラフによっても記述可能である。二分グラフは、符号のグラフ記述であるのみならず、デコーダのモデルでもある。二分グラフでは、符号語ビット（即ち、 H の各列）は左側の変数ノードによって表現され、各パリティ検査方程式（即ち、 H の各行）は右側の検査ノードによって表現される。各変数ノードは H の列に相当し、各検査ノードは H の行に相当するので、交換可能に「変数ノード」及び H の「列」と呼ばれ、交換可能に「検査ノード」及び H の「行」と呼ばれる。変数ノードは検査ノードにのみ接続されており、検査ノードは変数ノードにのみ接続されている。 n 個の符号語ビット及び m 個のパリティビットを有する符号では、符号語ビット j が検査方程式 i に参加する場合、エッジにて変数ノード v_j が検査ノード c_i へ接続される（ $i = 0, 1, \dots, m-1, j = 0, 1, \dots, n-1$ ）。換言すると、パリティ検査行列 H のエントリ h_{ij} が 1 である場合、変数ノード j が検査ノード i へ接続される。方程式（1）を反映すると、全ての検査ノードが偶パリティを有する場合、変数ノードは有効な符号語を表現する。

【0006】

パリティ検査行列、パリティ検査方程式及び二分グラフの間の関係を示すため、一例を以下に示す。 $n = 12$ とすると、符号化率 $1/2$ の符号は次式によって定義される。

【0007】

【数2】

$$H = \left[\begin{array}{cccccc|cccccc} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \quad (2)$$

左側部分は $k (= 6)$ 個の情報ビット s に相当し、右側部分は $m (= 6)$ 個のパリティビット p に相当する。式（1）を適用すると、式（2）における H によって次の6個のパリティ検査方程式が定義される。また、 H の対応する二分グラフを図1に示す。

【0008】

【数3】

$$\begin{cases} x_0 + x_2 + x_6 + x_7 = 0 \\ x_1 + x_4 + x_7 + x_8 = 0 \\ x_2 + x_5 + x_6 + x_8 + x_9 = 0 \\ x_0 + x_3 + x_9 + x_{10} = 0 \\ x_1 + x_4 + x_{10} + x_{11} = 0 \\ x_3 + x_5 + x_6 + x_{11} = 0 \end{cases} \quad (3)$$

十分な有限長さの LDP C 符号の二分グラフは、必然的にサイクルを有する。長さ $2d$ のサイクル（サイクル - $2d$ のように表記される）は、 d 個の変数ノード及び d 個の検査

10

20

30

40

50

ノードを通過し、いずれのエッジも繰り返すことなく各ノードをそのノード自体へ接続する、 $2d$ 個のエッジからなる経路である。短いサイクル、特にサイクル-4は、反復的デコーダ (iterative decoder) の性能を低下させるので、通常、符号設計においては回避される。

【0009】

符号サイズが大きくなると、無作為に構成されているLDPC符号のエンコード及びデコードは困難になる。構造化LDPC設計では、大きな $m \times n$ 擬似乱数 H 行列を直接的に構築する代わりに、小さな $m_b \times n_b$ の基本行列 H_b から開始し、 z 個の H_b の複製を作成して、その z 個の複製を相互に連結して大きな $m \times n$ の H 行列を形成する。ここで、 $m = m_b \times z$ 、 $n = n_b \times z$ である。行列表現を用いると、 H_b から H を構築するには、 H_b の各1は $z \times z$ 置換部分行列によって置き換えられ、 H_b の各0は $z \times z$ 全ゼロ部分行列によって置き換えられる。この置換は非常に簡単であり、性能を損なわないことが示されている。例えば、単位行列の列を所与の量だけ巡回右シフトすることによって置換部分行列が得られる簡単な巡回右シフトは、デコード性能を低下させることなく用いられ得る。 $(x \bmod z)$ 回の巡回左シフトは $((z-x) \bmod z)$ 回の巡回右シフトと等しいので、本明細書では、巡回右シフトのみを説明し、簡単のため巡回シフトと呼ぶ。この制約を用いると、 $z \times z$ の全ゼロ行列を表すように H_b の各 $h_{ij} = 0$ を $p(i, j) = -1$ によって置き換え、 H_b の各 $h_{ij} = 1$ を $p(i, j) = 0$ の巡回シフトサイズに置き換えることによって得られる $m_b \times n_b$ のモデル行列 $H_{b,m}$ により、各行列 H をユニークに表現することが可能である。

【0010】

したがって、拡張した行列 H を用いる代わりに、モデル行列 $H_{b,m}$ によって、符号がユニークに定義される。十分に小さな $m_b \times n_b$ の $H_{b,m}$ と、各々サイズ z を有するビットのベクトルとに基づき、エンコード及びデコードの両方が実行され得る。

【0011】

この手順では、本質的には、 H_b の各エッジは H のサイズ z のベクトルエッジ($H_{b,m}$ の $p(i, j)$ によって表現される)に対しマッピングされ、 H_b の各変数ノードは H の長さ z のベクトル変数ノード($H_{b,m}$ の列に相当)に対しマッピングされ、 H_b の各検査ノードは H の長さ z のベクトル検査ノード($H_{b,m}$ の行に相当)に対しマッピングされる。構造化設計では、次の2つの段階を通じて H に無作為性が構築される：(a) 偽似乱数基本行列 H_b 、(b) 各ベクトルエッジ内のエッジの偽似乱数シフト。無作為化に関する両段階は非常に簡単であるため、構造化設計の記憶及び処理の複雑性は低い。

【0012】

多くの場合、IEEE 802.16標準規格において定義されているシステムなどのシステムでは、サイズ (n_f, k_f) の符号のファミリーに対する誤り訂正符号を提供することが必要である。ここで、ファミリー内の符号は全て、同じ符号化率 $R = k_f / n_f$ を有し、符号サイズは、 $n_f = z_f \times n_b$ 、 $k_f = z_f \times k_b$ ($f = 0, 1, \dots, f_{max}$)のように、基本サイズからスケールアップされている。ここで、符号ファミリーにおけるメンバの総数は $(f_{max} + 1)$ であり、 z_f はファミリーにおける第 f 番目の符号に対する拡張係数である。これらのシステムでは、1つの基本行列 H_b 及び1組の適切な z_f から、全ての (n_f, k_f) に対する符号を導くことが可能である。 $p(f, i, j)$ を、拡張係数 z_f の第 f 番目のモデル行列 $H_{b,m}(f)$ 内の位置 (i, j) に位置するベクトルエッジのシフトサイズとする。したがって、交換可能に1組のシフトサイズ $\{p(f, i, j)\}$ 及びモデル行列 $H_{b,m}(f)$ と呼ばれる。

【0013】

しかしながら、各 $H_{b,m}(f)$ に対してシフトサイズ $p(f, i, j)$ を如何に定義するかは明らかでない。符号のファミリーを定義するための一手法は、全ての所与の f に対して独立に、基本行列 H_b 及び $p(f, i, j)$ ($0 \leq i < m-1, 0 \leq j < n-1$)のうちの一方又は両方をサーチすることである。しかしながら、この手法では、全ての f について、 H_b 及び $p(f, i, j)$ ($0 \leq i < m-1, 0 \leq j < n-1$)のうちの一方又は

10

20

30

40

50

両方が指定され、記憶される必要がある。

【0014】

H_b によってLDPCデコーダの基本構造及びメッセージの相互連結が定義されるため、ファミリの全ての符号に対し H_b を再利用することが好適であろう。ファミリの全ての符号によって同じ H_b が共有されると、

・ H_b のエントリ (i, j) が0であるとき、シフトサイズ $p(f, i, j) = -1$ である。シフトサイズ $p(f, i, j) = -1$ は、二値パリティ検査行列 $H(f)$ への拡張の際にモデル行列 $H_{b,m}(f)$ のエントリ (i, j) を置き換えるために用いられる、 $z_f \times z_f$ の全ゼロ部分行列を表すために用いられる。 H_b のエントリ (i, j) が0である場合、任意の f に対する $p(f, i, j)$ は同じである。即ち、 $p(f, i, j) = -1$ 10
 ”など、負でない整数以外の任意の他のラベルが同等に用いられ得る。

・ H_b のエントリ (i, j) が1であるとき、シフトサイズ $p(f, i, j) = 0$ である。シフトサイズ $p(f, i, j) = 0$ は、 $p(f, i, j)$ 列だけ巡回右シフトされた $z_f \times z_f$ 単位部分行列を表すために用いられる。この部分行列は、二値パリティ検査行列 $H(f)$ への拡張の際にモデル行列 $H_{b,m}(f)$ のエントリ (i, j) を置き換えるために用いられる。異なる f に対する $p(f, i, j)$ の値は異なる場合がある。即ち、異なる f に対する $H_{b,m}(f)$ のエントリ (i, j) は異なる場合がある。

【0015】

負でない $p(f, i, j)$ の値に関しては、任意の z_f に対し、 $p(f, i, j) = p(i, j) \bmod z_f$ を用いることが提案されている。ここで、シフトサイズの組 $\{p(i, j)\}$ は全ての z_f に対して同じである。したがって、 $\{p(i, j)\}$ の1つの組しか指定される必要がなく、異なる z_f の符号を実装することの複雑性は潜在的に減少される。しかしながら、モジュロ演算の影響のため、1組の $\{p(i, j)\}$ が1つの z_f に対する不良なサイクルパターンを回避するように設計されることによって、別の z_f に対するサイクルの数は大きく、符号語の重みは低くなり、一部の (n_f, k_f) に対する誤り訂正性能は低下する。 20

【発明の開示】

【発明が解決しようとする課題】

【0016】

したがって、全ての符号サイズ (n_f, k_f) に対して所望の符号特性を維持しつつ、1組の $\{p(i, j)\}$ からシフトサイズ $\{p(f, i, j)\}$ を導く方法の必要が存在する。 30

【課題を解決するための手段】

【0017】

上述の必要に対処するために、各符号化率のうちの最大の符号長さに対し、基本モデル行列が定義される。基本モデル行列におけるシフトの組 $\{p(i, j)\}$ は、同じ符号化率の他の全ての符号長さに対するシフトサイズを決定するために用いられる。拡張係数 z_f に相当する符号サイズに対するシフトサイズ $\{p(f, i, j)\}$ は、 $p(i, j)$ を比例的にスケールリングすることによって $\{p(i, j)\}$ から導かれ、 $\{p(f, i, j)\}$ によって定義されるモデル行列は、第 f 番目の符号に対するパリティ検査ビットを決定するために用いられる。 40

【0018】

本発明は、情報ブロックに基づきパリティ検査ビットを生成する送信機を動作させるための方法に関する。この方法は、最大の符号長さに対する1組のシフトサイズ $p(i, j)$ を有する基本モデル行列を定義する工程と、1組のシフトサイズ $p(i, j)$ に基づき他の全ての符号長さに対するシフトサイズ $p(f, i, j)$ を決定する工程と、を含む。ここで、 f は符号長さの指標であり、 $p(f, i, j) = F(p(i, j), z_0 / z_f)$ であり、 z_0 は最大の符号長さの拡張係数であり、 z_f は第 f 番目の符号長さの拡張係数である。情報ブロックが受信され、モデル行列を用いてパリティ検査ビットが決定され 50

る。モデル行列は $p(f, i, j)$ によって定義される。

【0019】

また、本発明は、最大の符号長さに対する1組のシフトサイズ $p(i, j)$ を有する基本モデル行列を記憶するための記憶手段を備える装置に関する。この装置は、これに加えて、情報ブロック $s = (s_0, \dots, s_{k_f - 1})$ 及び基本モデル行列を受信するマイクロプロセッサを備える。マイクロプロセッサは、1組のシフトサイズ $p(i, j)$ に基づき他の全ての符号長さに対するシフトサイズ $p(f, i, j)$ を決定する。ここで、 f は符号長さの指標であり、 $p(f, i, j) = F(p(i, j), z_0 / z_f)$ であり、 z_0 は最大の符号長さの拡張係数であり、 z_f は第 f 番目の符号長さの拡張係数である。マイクロプロセッサは $p(f, i, j)$ によって定義されるモデル行列に基づくパリティ検査ビットと、情報ブロック $s = (s_0, \dots, s_{k_f - 1})$ とを出力する。

10

【0020】

また、本発明は、情報ブロック $s = (s_0, \dots, s_{k_f - 1})$ を推定する受信機を動作させるための方法に関する。この方法は、信号ベクトルを受信する工程と、最大の符号長さに対する1組のシフトサイズ $p(i, j)$ を有する基本モデル行列を定義する工程と、1組のシフトサイズ $p(i, j)$ に基づき他の全ての符号長さに対するシフトサイズ $p(f, i, j)$ を決定する工程と、を含む。ここで、 f は符号長さの指標であり、 $p(f, i, j) = F(p(i, j), z_0 / z_f)$ であり、 z_0 は最大の符号長さの拡張係数であり、 z_f は第 f 番目の符号長さの拡張係数である。 $p(f, i, j)$ によって定義されるモデル行列と受信した信号ベクトルとに基づき、情報ブロック $s = (s_0, \dots$

20

【0021】

最後に、本発明は、最大の符号長さに対する1組のシフトサイズ $p(i, j)$ を有する基本モデル行列を記憶するための記憶手段を備える装置に関する。この装置は、これに加えて、信号ベクトルを受信し、1組のシフトサイズ $p(i, j)$ に基づき他の全ての符号長さに対するシフトサイズ $p(f, i, j)$ を決定するデコーダを備える。ここで、 f は符号長さの指標であり、 $p(f, i, j) = F(p(i, j), z_0 / z_f)$ であり、 z_0 は最大の符号長さの拡張係数であり、 z_f は第 f 番目の符号長さの拡張係数である。デコーダは $p(f, i, j)$ によって定義されるモデル行列と受信した信号ベクトルとに基づき情報ブロック $s = (s_0, \dots, s_{k_f - 1})$ に対する推定値を出力する。

30

【発明を実施するための最良の形態】

【0022】

拡張した行列 H の特性は、基本行列 H_b の特性及びシフトサイズ $p(i, j)$ に密接に関連することが示されている。符号設計の疑似巡回性のため、一定の所望されないシフトサイズ $p(i, j)$ のパターンによって H_b のサイクル及び符号語のパターンが保存され、拡張した行列 H において複数回繰り返されることによって、誤り訂正性能は許容不能なものとなる。

【0023】

低い重みの符号語は短いサイクルを含むため、 H_b に重み - 1 列が存在しない場合、十分なデコード性能を得るには、対象の符号サイズ (n_f, k_f) の全てに対し短いサイクルが壊れていることを確認することで充分である。

40

【0024】

次の条件が満たされる場合、拡張した行列において H_b のサイクルが複製されることが分かっている。

基本行列 H_b において $2c$ 個のエッジが長さ $2c$ のサイクルを形成する場合、拡張した行列 H において、対応する $2c$ 個のベクトルエッジは、次式であるとき、かつそのときに限り、長さ $2c$ の z 個のサイクルを形成する。

【0025】

【数4】

$$\sum_{\substack{i=2j \\ j=0,\dots,c-1}} p(i) = \sum_{\substack{i=2j+1 \\ j=0,\dots,c-1}} p(i) \pmod{z}, \quad p(i) \geq 0 \quad (4)$$

ここで、 z は拡張係数であり、 $p(i)$ はモデル行列 $H_{b,m}$ におけるエッジ i の巡回シフトサイズであり、エッジ $0, 1, 2, \dots, 2c-1$ は (この順序で) H_b においてサイクルを形成する。

【0026】

1つの z_f 値に対して方程式 (4) を満たすことを回避する固定組のシフトサイズ $\{p(i, j)\}$ が、別の z_f 値に対する方程式 (4) を満たす場合が実際にあるが、 $\{p(i, j)\}$ を z_f に比例してスケールリングする場合、方程式 (4) の線形性により、全ての z_f に対して方程式 (4) を満たすことは回避されることが示される。

【0027】

2つの拡張係数 z_0, z_1 に対し、1組のシフトサイズ $\{p(i, j)\}$ を用いて所与の基本行列 H_b を拡張することが想定される。ここで、 $\alpha = z_0 / z_1 > 1$ である。シフトサイズの組 $\{p(i, j)\}, \{p(0, i, j)\}$ によって、拡張係数 z_0 に対する長さ $2c$ のサイクルが回避されると仮定すると、

【0028】

【数5】

$$\sum_{\substack{i=2j \\ j=0,\dots,c-1}} p(i) \neq \sum_{\substack{i=2j+1 \\ j=0,\dots,c-1}} p(i) \pmod{z_0} \quad (5)$$

また

$$\sum_{\substack{i=2j \\ j=0,\dots,c-1}} \frac{p(i)}{\alpha} \neq \sum_{\substack{i=2j+1 \\ j=0,\dots,c-1}} \frac{p(i)}{\alpha} \pmod{z_1} \quad (6)$$

ここで、 $p(i)$ はモデル行列 $H_{b,m}(0)$ におけるエッジ i の巡回シフトサイズであり、エッジ $0, 1, 2, \dots, 2c-1$ は (この順序で) H_b においてサイクルを形成する。スケールリングされたシフトサイズ $\{p(i, j) / \alpha\}$ の組が拡張係数 z_1 に対し用いられる場合、 z_1 から拡張された H 行列によって長さ $2c$ のサイクルも回避されることが、方程式 (6) によって示される。 $2c$ は任意のサイクル長さであり得るので、スケールリングされたシフトサイズ $\{p(i, j) / \alpha\}$ を用いることにより、 z_0 に対する $\{p(i, j)\}$ の組によって回避される、 z_1 に対する全てのサイクルタイプが回避される。

【0029】

上述の説明では、スケールリングの後のシフトサイズが整数である必要があるという制限を無視した。整数を得るために、全ての $p(i, j) / \alpha$ に対し、例えば、床関数 $\lfloor x \rfloor$ (x 以下の最大の整数)、天井関数 $\lceil x \rceil$ (x 以上の最小の整数) 又は丸め関数 $[x]$ (x との差が最小の整数) が実行される必要がある。一般に、 z_0 に対するシフトサイズ $p(i, j) - p(0, i, j)$ が与えられると、 z_1 に対するシフトサイズは $p(i, j)$ 及び $p(0, i, j)$ の関数 $F(\cdot)$ として導かれ得る。

【0030】

10

20

30

40

【数 6】

$$p(1, i, j) = F(p(i, j), \alpha) = F(p(i, j), z_0/z_1) \quad (7)$$

例えば、式(6)のトップに対し丸め関数が用いられ、 z_0 に対して設計されたシフトサイズが $p(i, j)$ である場合、 z_1 に適用されるシフトサイズの組は次式となる。

【0031】

【数 7】

$$p(1, i, j) = \begin{cases} p(i, j), & p(i, j) \leq 0 \\ \left\lceil \frac{p(i, j)z_1}{z_0} \right\rceil = \left\lceil \frac{p(i, j)}{\alpha} \right\rceil, & p(i, j) > 0 \end{cases} \quad (8)$$

10

通常では全ての正の $p(i, j)$ がスケールされるが、式(8)のようなスケールは $\{p(i, j)\}$ のサブセットに対してのみ適用される場合がある。例えば、存在する場合には H_b の重み - 1 列のエッジなど、任意のサイクルに含まれるものがスケールされる必要がない。負でない全ての $p(i, j)$ に対してスケールが適用される場合には、関数 $F(\cdot)$ の定義に応じて、 $H_{b_m}(0)$ 、 $H_{b_m}(1)$ の基本行列が同じであってもよく、同じでなくてもよい。

20

【0032】

システムに 2 より多くの拡張係数が必要である場合、 $p(f, i, j)$ の発見に対して上述の解析が容易に適用される。この場合、最大の符号長さに対する 1 組のシフトサイズ $p(0, i, j)$ を有する母モデル行列(基本モデル行列とも呼ぶ) $H_{b_m}(0)$ が定義され、この行列から、第 f 番目の符号ファミリのメンバに対するシフトサイズ $p(f, i, j)$ を有するモデル行列 $H_{b_m}(f)$ が導かれる ($f = 1, \dots, f_{max}$)。 $z_0 = \max(z_f)$ 、かつ、 $p(0, i, j) = p(i, j)$ と仮定すると、 z_f の全範囲に対して同じサイクルの基本行列が回避されるように、 $p(i, j)$ から $p(f, i, j)$ を導く際、式(8)のような拡張において $\alpha = z_0/z_f$ が用いられる。詳細には、全ての $p(i, j)$ が発見されると仮定すると、一般に、次式を用いて $p(i, j)$ から $p(f, i, j)$ が導かれる。

30

【0033】

【数 8】

$$p(f, i, j) = F(p(i, j), \alpha_f) = F(p(i, j), z_0/z_f) \quad (9)$$

さらに、一例として、 $z_0 = \max(z_f)$ と仮定し、丸め関数を用いると、式(8)に対応して関数 $F(\cdot)$ は次のように定義され得る。

【0034】

【数 9】

$$p(f, i, j) = \begin{cases} p(i, j), & p(i, j) \leq 0 \\ \left\lceil \frac{p(i, j)z_f}{z_0} \right\rceil = \left\lceil \frac{p(i, j)}{\alpha_f} \right\rceil, & p(i, j) > 0 \end{cases} \quad (10)$$

40

同様に、丸め関数 $\lceil x \rceil$ の代わりに床関数 $\lfloor x \rfloor$ 又は天井関数 $\lceil x \rceil$ を用いることが可能である。

【0035】

なお、上述の設計手順は任意のタイプの基本行列 H_b に対し適用される。例えば、次の

50

ように2つの部分からなる H_b に対し適用され得る。

【 0 0 3 6 】

【 数 1 0 】

$$H_b = \left[\begin{array}{c|c} (H_{b1})_{m_b \times k_b} & (H_{b2})_{m_b \times m_b} \end{array} \right] \tag{11}$$

さらに、行列式部分 H_{b2} は次のように2つの区分へ分割され得る。ここで、ベクトル h_b は奇数の重み $w_h > 2$ を有し、 H'_{b2} は行列式の階段構造を有する。

【 0 0 3 7 】

【 数 1 1 】

10

$$H_{b2} = [h_b | H'_{b2}]$$

$$= \left[\begin{array}{c|cccc} h_b(0) & 1 & & & \\ h_b(1) & 1 & 1 & & \mathbf{0} \\ \cdot & & 1 & \ddots & \\ \cdot & & & \ddots & 1 \\ \cdot & & \mathbf{0} & & 1 & 1 \\ h_b(m_b-1) & & & & & 1 \end{array} \right] \tag{12}$$

20

換言すると、 H'_{b2} の第 i 行、第 j 列の行列要素は、次式と等しい。

【 0 0 3 8 】

【 数 1 2 】

$$H'_{b2}(i,j) = \begin{cases} 1, & i=j \text{ 又は } i=j+k; \\ 0, & \text{それ以外} \end{cases}$$

【 符号のファミリに対するエンコーダ実装 】

上述の設計によるファミリのメンバは全て母モデル行列 $H_{b_m} = H_{b_m}(0)$ から導かれ、したがって全て同じ構造を有するため、ファミリの各メンバに対するエンコード処理は同様である。モデル行列の一部分又は全体が記憶され、グループ化された情報シーケンスの巡回シフトを実行するためのパレルシフトに対する命令として解釈され得る。

30

【 0 0 3 9 】

ファミリのメンバは全て母モデル行列 $H_{b_m} = H_{b_m}(0)$ から導かれるため、ファミリのエンコーダの実装に必要なのは、母行列が記憶されることのみである。丸め関数 $[x]$ が用いられると仮定すると、ファミリの第 f 番目のメンバでは、全ての $p(i,j) > 0$ に対し、母モデル行列の巡回シフト $p(i,j)$ が巡回シフト $[p(i,j) / (z_0 / z_f)]$ に置き換えられる。ここで、 z_f は、エンコードされるファミリの第 f 番目のメンバの拡張係数を示す。この直接的な実装は、ファミリの各メンバに対する $z_f^{-1} = (z_0 / z_f)^{-1}$ (又は、 $z_f = (z_0 / z_f)$) の値をリードオンリメモリに記憶し、実行時に乗算器を用いて $[p(i,j) / (z_0 / z_f)]$ の値 ($p(i,j) > 0$) を計算することである。これに代えて、ファミリの各メンバに対するシフトサイズの組 $\{p(f, i, j)\}$ ($f = 0, 1, \dots, f_{max}$) は予め式 (8) (又は、一般的には式 (7)) を用いて計算され、リードオンリメモリに記憶されてもよい。

40

【 0 0 4 0 】

パレルシフトはファミリのメンバに相当する全ての語サイズ z_f に巡回シフトを提供するように修正され得る。パレルシフトを修正することによってパレルシフトの論理は複雑となり、より低速のクロック速度が必要となるが、余分な論理資源を必要とする代替案では、各語サイズ z_f に対して異なるパレルシフトがインスタンス化される。

50

【 0 0 4 1 】

図2は、エンコーダ200のブロック図である。示すように、エンコーダ200は、拡張係数 z_f を決定するために、マイクロプロセッサ201、参照テーブル203及び論理回路205を備える。互いの外部に存在するように示したが、論理回路205の機能がマイクロプロセッサ201内に実装されてもよいことが、当業者には認識される。マイクロプロセッサ201は、好適には、デジタル信号プロセッサ(DSP)を含む。デジタル信号プロセッサには、MSC8300及びDSP56300のDSPなどがあるが、それらに限定されない。これに加えて、参照テーブル203は行列を記憶するための記憶手段として作用し、リードオンリメモリを含む。しかしながら、他の形態のメモリ(例えば、ランダムアクセスメモリ、磁気記憶メモリなど)が同様に利用されてよいことが、当業者には認識される。第2の実施形態では、マイクロプロセッサ201、参照テーブル203及び論理回路205の機能は、特定用途向け集積回路(AASIC)又はフィールドプログラマブルゲートアレイ(FPGA)へ組み込まれ得る。詳細には、参照テーブル203は、回路における信号経路の存在又は非存在に対応するメモリの形態で実装され得る。

10

【 0 0 4 2 】

上述のように、一般に、エンコードされたデータはシステムティックビットに加えて複数のパリティ検査ビットの形態を取り、パリティ検査ビット及びシステムティックビットが共に符号語 x を形成する。本発明の第1の実施形態では、基本モデル行列 H_{b_m} は参照テーブル203に記憶され、パリティ検査ビットを発見するためにマイクロプロセッサ201によってアクセスされる。詳細には、マイクロプロセッサ201は、情報ブロック $s = (s_0, \dots, s_{k_f-1})$ 、拡張係数 z_f 及び基本モデル行列 H_{b_m} に基づき、パリティ検査ビット $p = (p_0, \dots, p_{m_f-1})$ に対する適切な値を決定する。拡張係数 z_f は、 $z_f = k_f / k_b = n_f / n_b$ を用いて論理205によって決定され、長さ z_f のベクトルや発見した $f = z_0 / z_f$ へとビットをグループ化するために用いられる。パリティ検査ビットが発見された後、パリティ検査ビット及びシステムティックビットは送信機へ渡され、受信機へ送信される。

20

【 0 0 4 3 】

図3は、本発明の一実施形態によるデコーダ300のブロック図である。示すように、デコーダ300は、拡張係数 z_f を決定するために、マイクロプロセッサ301、参照テーブル303及び論理回路305を備える。本発明の第1の実施形態では、マイクロプロセッサ301は、デジタル信号プロセッサ(DSP)を含む。デジタル信号プロセッサには、MSC8300及びDSP56300のDSPなどがあるが、それらに限定されない。これに加えて、参照テーブル303は基本モデル行列 H_{b_m} を記憶するための記憶手段として作用し、リードオンリメモリを含む。しかしながら、他の形態のメモリ(例えば、ランダムアクセスメモリ、磁気記憶メモリなど)が同様に利用されてよいことが、当業者には認識される。第2の実施形態では、マイクロプロセッサ301及び参照テーブル303の機能は、特定用途向け集積回路(AASIC)又はフィールドプログラマブルゲートアレイ(FPGA)へ組み込まれ得る。詳細には、参照テーブル303は、回路における信号経路の存在又は非存在に対応するメモリの形態で実装され得る。

30

【 0 0 4 4 】

受信信号ベクトル(受信機を介して受信される) $y = (y_0, \dots, y_{n_f-1})$ は、ノイズを有するチャネルを通じて送信される符号語 x に相当する。ここで、上述のように、エンコードされたデータ x は、符号ファミリの第 f 番目の符号語ベクトルである。本発明の第1の実施形態では、基本モデル行列 H_{b_m} は参照テーブル303に記憶され、 y をデコードして情報ブロック $s = (s_0, \dots, s_{k_f-1})$ を推定するために、マイクロプロセッサ301によってアクセスされる。詳細には、マイクロプロセッサ301は、受信信号ベクトル $y = (y_0, \dots, y_{n_f-1})$ 及び基本モデル行列 H_{b_m} に基づき、情報ブロック $s = (s_0, \dots, s_{k_f-1})$ を推定する。拡張係数 z_f は、 $z_f = k_f / k_b = n_f / n_b$ を用いて論理305によって決定され、長さ z_f のベクトルや発見した $f = z_0 / z_f$ へと受信信号及びビットをグループ化するために用いられる。

40

50

【 0 0 4 5 】

図4は、エンコーダ200の、詳細には、マイクロプロセッサ201の動作を示すフローチャートである。論理フローは工程401にて開始し、情報ブロック(s_0, \dots, s_{k_f-1})がマイクロプロセッサ201によって受信される。工程403にて、情報ブロック及び $H_{b_m}(f)$ に基づきパリティ検査ビットの値が決定される。ここで、 $H_{b_m}(f)$ によって $\{p(f, i, j)\}$ がユニークに定義される。詳細には、基本モデル行列 H_{b_m} のシフトサイズの組 $\{p(i, j)\}$ がメモリから読み出される。マイクロプロセッサは $\{p(i, j)\}$ 及び f を利用して、 $\{p(f, i, j)\}$ を決定する。パリティ検査ビット(p_0, \dots, p_{m_f-1})は、方程式(1)を解くことによって決定される。工程405にて、情報ブロック及びパリティ検査ビットがチャンネルを通じて送信される。

10

【 0 0 4 6 】

図5は、デコーダ300の、詳細には、マイクロプロセッサ301の動作を示すフローチャートである。論理フローは工程501にて開始し、受信信号ベクトル $y = (y_0, \dots, y_{n_f-1})$ が受信される。工程503にて、 $H_{b_m}(f)$ に基づき情報ブロック $s = (s_0, \dots, s_{k_f-1})$ の推定値が決定される。ここで、 $H_{b_m}(f)$ は $\{p(f, i, j)\}$ によってユニークに定義される。詳細には、基本モデル行列 H_{b_m} のシフトサイズの組 $\{p(i, j)\}$ がメモリから読み出される。マイクロプロセッサは $\{p(i, j)\}$ 及び f を利用して、 $\{p(f, i, j)\}$ を決定する。説明したように、マイクロプロセッサは、シフトサイズ $\{p(f, i, j)\}$ (又は同等に、 $H_{b_m}(f)$)によって受信信号ベクトルを処理し、情報ブロックの推定値を得る。好適な一実施形態では、マイクロプロセッサは、符号の二分グラフを用いるメッセージ通過アルゴリズムによって、処理を実行する。

20

【 0 0 4 7 】

[実施例]

一例として、576 ~ 2304ビットの範囲にある19の符号サイズの n_f に対する符号設計を記載する。各基本モデル行列は、シフトサイズ $z_0 = 96$ に対して設計されている。1組のシフトサイズ $\{p(i, j)\}$ は基本モデル行列に対して定義されており、同じ符号化率の他の符号サイズに対して用いられる。他の符号サイズでは、シフトサイズは基本モデル行列から次のように導かれる。

30

【 0 0 4 8 】

拡張係数 z_f に相当する符号サイズに対しては、そのシフトサイズ $\{p(f, i, j)\}$ は、次式のように、 $p(i, j)$ を比例的にスケールリングすることによって $\{p(i, j)\}$ から導かれる。

【 0 0 4 9 】

【 数 1 3 】

$$p(f, i, j) = \begin{cases} p(i, j), & p(i, j) \leq 0 \\ \left\lfloor \frac{p(i, j) z_f}{z_0} \right\rfloor = \left\lfloor \frac{p(i, j)}{\alpha_f} \right\rfloor, & p(i, j) > 0 \end{cases}$$

40

なお、 $\alpha_f = z_0 / z_f$ であり、 x は、 $-$ の方向の最も近い整数を与える床関数を表す。

【 0 0 5 0 】

1/2, 2/3, 3/4の3つの符号化率について、基本モデル行列を以下に示す。ここで、 f は所与の符号化率における符号長さの指標であり、 $f = 0, 1, 2, \dots, 18$ である。

【 0 0 5 1 】

[符号化率 1 / 2] :

基本モデル行列のサイズは $n_b = 24, m_b = 12$ であり、拡張係数 $z_0 = 96$ (即ち

50

、 $n_0 = 24 \times 96 = 2304$) である。他の符号サイズ n_f を得るため、拡張係数 z_f は $n_f / 24$ と等しい。

【0052】

【表1】

-1	94	73	-1	-1	-1	-1	-1	55	83	-1	-1	7	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	27	-1	-1	-1	22	79	9	-1	-1	-1	12	-1	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	24	22	81	-1	33	-1	-1	-1	0	-1	-1	0	0	-1	-1	-1	-1	-1	-1	-1	-1
61	-1	47	-1	-1	-1	-1	-1	65	25	-1	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	39	-1	-1	-1	84	-1	-1	41	72	-1	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	46	40	-1	82	-1	-1	-1	79	0	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1
-1	-1	95	53	-1	-1	-1	-1	-1	14	18	-1	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1	-1
-1	11	73	-1	-1	-1	2	-1	-1	47	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1
12	-1	-1	-1	83	24	-1	43	-1	-1	-1	51	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1	
-1	-1	-1	-1	-1	94	-1	59	-1	-1	70	72	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1
-1	-1	7	65	-1	-1	-1	-1	39	49	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0
43	-1	-1	-1	-1	66	-1	41	-1	-1	-1	26	7	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0

10

[符号化率 2 / 3] :

基本モデル行列のサイズは $n_b = 24$, $m_b = 8$ であり、拡張係数 $z_0 = 96$ (即ち、 $n_0 = 24 \times 96 = 2304$) である。他の符号サイズ n_f を得るため、拡張係数 z_f は $n_f / 24$ と等しい。

20

【0053】

【表2】

56	-1	-1	54	-1	75	-1	82	93	-1	-1	49	-1	3	83	-1	7	0	-1	-1	-1	-1	-1	-1
-1	47	36	-1	-1	4	62	-1	14	-1	-1	37	63	-1	-1	11	-1	0	0	-1	-1	-1	-1	-1
-1	61	-1	37	-1	-1	84	-1	54	-1	2	93	-1	23	-1	79	0	-1	0	0	-1	-1	-1	-1
84	-1	-1	77	-1	80	-1	31	78	-1	9	-1	65	-1	-1	58	-1	-1	-1	0	0	-1	-1	-1
-1	55	40	-1	8	-1	13	-1	-1	79	60	-1	95	-1	-1	30	-1	-1	-1	-1	0	0	-1	-1
11	-1	-1	45	0	-1	-1	10	-1	13	21	-1	-1	70	86	-1	-1	-1	-1	-1	0	0	-1	-1
35	-1	6	-1	16	40	-1	30	-1	57	-1	-1	89	-1	74	-1	-1	-1	-1	-1	-1	-1	0	0
-1	89	95	-1	77	-1	56	-1	-1	74	-1	14	-1	78	14	-1	7	-1	-1	-1	-1	-1	-1	0

30

[符号化率 3 / 4] :

基本モデル行列のサイズは $n_b = 24$, $m_b = 6$ であり、拡張係数 $z_0 = 96$ (即ち、 $n_0 = 24 \times 96 = 2304$) である。他の符号サイズ n_f を得るため、拡張係数 z_f は $n_f / 24$ と等しい。

40

【0054】

【表3】

43	90	41	40	19	-1	-1	-1	-1	86	-1	83	26	74	50	-1	-1	62	7	0	-1	-1	-1	-1
-1	-1	95	61	84	2	16	-1	-1	0	-1	-1	-1	20	30	91	18	95	-1	0	0	-1	-1	-1
-1	-1	-1	87	0	-1	58	16	-1	87	16	-1	-1	93	-1	54	24	33	0	-1	0	0	-1	-1
-1	12	-1	-1	65	48	-1	10	10	95	-1	49	-1	52	6	-1	36	57	-1	-1	-1	0	0	-1
65	-1	31	-1	15	-1	12	-1	6	57	0	89	9	29	-1	-1	-1	75	-1	-1	-1	-1	0	0
-1	65	-1	-1	48	40	-1	83	18	45	29	-1	73	84	-1	77	-1	95	7	-1	-1	-1	-1	0

50

本発明は特定の実施形態に関連して詳細に示され、説明されているが、本発明の精神及び範囲から逸脱することなく、形態及び詳細における種々の変更が行なわれ得ることが、当業者には理解される。例えば、符号サイズの範囲が非常に大きいとき、例えば、 z / 2 に近づくときには、適正なシフトサイズの組 $\{p(i, j)\}$ を発見することは非常に困難となる。したがって、符号範囲が大き過ぎる場合、即ち、 z が大きい場合、符号ファミリにおいて一定の範囲の z_f を各々覆う、 $\{p(i, j)\}$ の複数の組が用いられる。別の例として、説明では、第 0 番目の符号ファミリに対し、母モデル行列 $H_{b_m}(0)$ 、 $z_0 = \max(z_f)$ 及び $p(0, i, j) = p(i, j)$ が用いられると仮定したが、符号ファミリに存在しないが対象の符号ファミリに対するシフトサイズ $p(f, i, j)$ を導くために用いられる符号サイズに対して $H_{b_m}(0)$ 、 z_0 、 $p(i, j)$ が定義されてもよいことが、当業者には理解される。別の例として、説明では、 $z_0 = \max(z_f)$ であると仮定したが、シフトサイズ導出において $\max(z_f)$ と等しくない z_0 の値が用いられてよいことが、当業者には理解される。かかる変更は添付の特許請求の範囲内にあることが意図される。

10

【図面の簡単な説明】

【0055】

【図1】(12, 6)のH行列の二分グラフ。

【図2】エンコーダのブロック図。

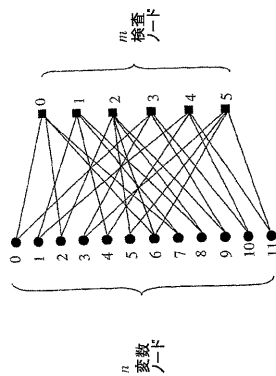
【図3】デコーダのブロック図。

【図4】図2のエンコーダの動作を示すフローチャート。

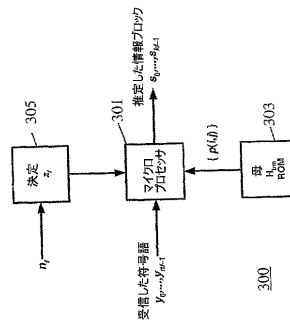
20

【図5】図3のデコーダの動作を示すフローチャート。

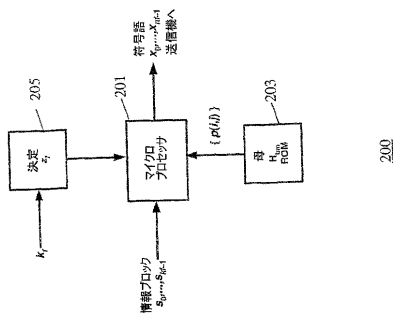
【図1】



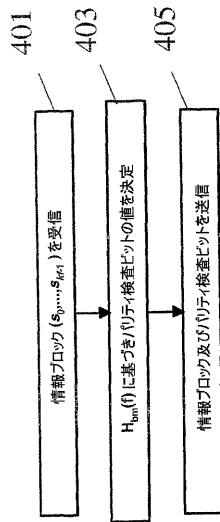
【図3】



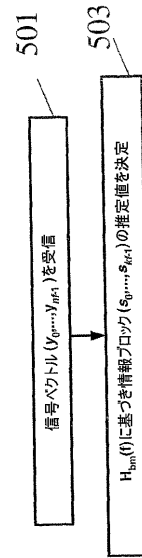
【図2】



【 図 4 】



【 図 5 】



フロントページの続き

(31)優先権主張番号 11/078,817

(32)優先日 平成17年3月11日(2005.3.11)

(33)優先権主張国 米国(US)

(72)発明者 ブランケンシップ、ティ・キース

アメリカ合衆国 60107 イリノイ州 ストリームウッド エス・グリーン メドウズ ブル
バード 1687

審査官 藤井 浩

(56)参考文献 国際公開第2004/019268(WO, A1)

特開2003-115768(JP, A)

Hao Zhong et al., Joint code-encoder-decoder design for LDPC coding system VLSI Implementation, Proceedings of the 2004 International Symposium on Circuits and Systems, 2004. (ISCAS'04), IEEE, 2004年 5月23日, Vol.2, pp.11-389-11-392

Brian Classon et al., LDPC coding for OFDMA PHY, IEEE C802.16e-05/006, IEEE 802.16 Broadband Wireless Access Working Group, 2005年 1月10日, URL, http://wirelessman.org/tge/contrib/C80216e-05_006.pdf

Brian Classon et al., LDPC coding for OFDMA PHY, IEEE C802.16e-04/278r1, IEEE 802.16 Broadband Wireless Access Working Group, 2004年 8月17日, URL, http://wirelessman.org/tge/contrib/C80216e-04_278r1.pdf

(58)調査した分野(Int.Cl., DB名)

H03M 13/00