

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5219552号
(P5219552)

(45) 発行日 平成25年6月26日(2013.6.26)

(24) 登録日 平成25年3月15日(2013.3.15)

(51) Int.Cl. F I
HO 3M 13/19 (2006.01) HO 3M 13/19
HO 3M 13/27 (2006.01) HO 3M 13/27

請求項の数 8 (全 35 頁)

(21) 出願番号	特願2008-44898 (P2008-44898)	(73) 特許権者	503447036
(22) 出願日	平成20年2月26日(2008.2.26)		サムスン エレクトロニクス カンパニー
(62) 分割の表示	特願2006-524576 (P2006-524576)		リミテッド
原出願日	平成16年8月26日(2004.8.26)		大韓民国・443-742・キョンギード
(65) 公開番号	特開2008-172824 (P2008-172824A)		・スウォン・シ・ヨントン・ク・サムスン
(43) 公開日	平成20年7月24日(2008.7.24)	(74) 代理人	100089037
審査請求日	平成20年2月26日(2008.2.26)		弁理士 渡邊 隆
(31) 優先権主張番号	10-2003-0059206	(74) 代理人	100110364
(32) 優先日	平成15年8月26日(2003.8.26)		弁理士 実広 信哉
(33) 優先権主張国	韓国 (KR)	(72) 発明者	ギューブン・キュン
前置審査			大韓民国・ソウル・135-930・ガン
			ナム・グ・ヨクサン・1ードン・794-
			14・ウォンヒョン・ヴィラ・#302
			最終頁に続く

(54) 【発明の名称】 移動通信システムにおけるブロック低密度パリティ検査符号の符号化／復号化装置及び方法

(57) 【特許請求の範囲】

【請求項 1】

復号装置がブロック低密度パリティ検査(LDPC)符号を復号する方法であって、
 情報部分とパリティ部分を含むパリティ検査行列を使用してブロックLDPC符号を復
 号する過程を含み、

前記パリティ部分は、

複数の第1の順列行列を含み、部分行列を構成する部分ブロックの中に2個の部分ブ
 ロックに0でない順列行列 P_y と、

【数 1】

$$P^{a_i}$$

と、を含む第1のセクション(B)と、

順列行列 P_x を含む第2のセクション(D)と、

第3のセクション(T)内に対角で配列される、複数の恒等行列(I)と前記複数の恒等
 行列の下に配列される複数の第2の順列行列を含む前記第3のセクション(T)と、

最後の部分ブロックのみに

【数 2】

$$P^{a_m}$$

を含む第 4 のセクション(E)と

を含むことを特徴とする L D P C 符号復号方法。

【請求項 2】

前記第 1 のセクション(B)と、第 2 のセクション(D)と、第 3 のセクション(T)と、
第 4 のセクション(E)のそれぞれに対応する順列行列は、 $(E)(T^{-1})(B)+D$ に対応す
る行列が恒等行列になるように形成されることを特徴とする請求項 1 に記載の L D P C 符
号復号方法。

10

【請求項 3】

前記第 1 の順列行列のうちの一つは、前記第 1 のセクション(B)の最初のブロックに
配列されることを特徴とする請求項 1 に記載の L D P C 符号復号方法。

【請求項 4】

前記第 1 のセクション(B)と、第 2 のセクション(D)と、第 3 のセクション(T)と、
第 4 のセクション(E)のそれぞれに対応する順列行列は、前記ブロック L D P C 符号の
因子グラフ上の最小サイクル長さが最大になり、ウェイト値が不均一になるように形成さ
れることを特徴とする請求項 1 に記載の L D P C 符号復号方法。

20

【請求項 5】

ブロック低密度パリティ検査(L D P C)符号を処理するシステムであって、
情報部分とパリティ部分を含むパリティ検査行列を使用してブロック L D P C 符号を復
号する復号装置を含み、

前記パリティ部分は、

複数の第 1 の順列行列を含み、部分行列を構成する部分ブロックの中に 2 個の部分ブ
ロックに 0 でない順列行列 P_y と、

【数 3】

$$P^{a_1}$$

30

と、を含む第 1 のセクション(B)と、

順列行列を P_x 含む第 2 のセクション(D)と、

第 3 のセクション(T)内に対角で配列される、複数の恒等行列(I)と前記複数の恒等
行列の下に配列される複数の第 2 の順列行列を含む前記第 3 のセクション(T)と、

最後の部分ブロックのみに

【数 4】

$$P^{a_m}$$

40

を含む第 4 のセクション(E)とを含むことを特徴とする L D P C 符号処理システム。

【請求項 6】

前記第 1 のセクション(B)と、第 2 のセクション(D)と、第 3 のセクション(T)と、
第 4 のセクション(E)のそれぞれに対応する順列行列は、 $(E)(T^{-1})(B)+D$ に対応す
る行列が恒等行列になるように形成されることを特徴とする請求項 5 に記載の L D P C 符
号処理システム。

【請求項 7】

前記第 1 の順列行列のうちの一つは、前記第 1 のセクション(B)の最初のブロックに

50

配列されることを特徴とする請求項5に記載のLDPC符号処理システム。

【請求項8】

前記第1のセクション(B)と、第2のセクション(D)と、第3のセクション(T)と、第4のセクション(E)のそれぞれに対応する順列行列は、前記ブロックLDPC符号の因子グラフ上の最小サイクル長さが最大になり、ウェイト値が不均一になるように形成されることを特徴とする請求項5に記載のLDPC符号処理システム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は移動通信システムに関するもので、特に、ブロック低密度パリティ検査符号を符号化/復号化する装置及び方法に関するものである。 10

【背景技術】

【0002】

1970年代末の米国で、セルラー(cellular)方式の無線移動通信システム(Mobile Telecommunication System)が開発された以来、韓国ではアナログ方式の1世代(1st Generation)移動通信システムと呼ばれるAMPS(Advanced Mobile Phone Service)方式で音声通信サービスを提供し始めた。以後、1990年代中盤から韓国では、2世代移動通信システムとして符号分割多重接続(Code Division Multiple Access:以下、“CDMA”とする)方式のシステムを商用化して音声及び低速データサービスを提供した。 20

【0003】

1990年代末から向上した無線マルチメディアサービス、全世界的ローミング(roaming)、高速データサービスなどを目標で始まった3世代移動通信システムであるIMT(International Mobile Telecommunication)-2000は、現在一部商用化されてサービスが提供されている。特に、3世代移動通信システムは、移動通信システムでサービスするデータ量が急速に増加するに従って、より高速のデータを伝送するために開発された。すなわち、3世代移動通信システムは、パケットサービス通信システム形態で発展してきている。パケットサービス通信システムは、バースト(burst)パケットデータを複数の移動局に伝送するシステムとして、大容量データ伝送に適合するように設計されている。その結果、パケットサービス通信システムは高速パケットサービスのために発展している。 30

【0004】

一方、現在は3世代移動通信システムから4世代移動通信システムに発展している状態である。4世代移動通信システムは、以前世代の移動通信システムのように単純な無線通信サービスに限定されず、有線通信ネットワークと無線通信ネットワークとの効率的連動及び統合サービスを目標として標準化されている。したがって、無線通信ネットワークで有線通信ネットワークの容量(capacity)に近接する大容量データを伝送できる技術開発が要求されている。

【0005】

このように、音声サービスデータだけでなく、映像、無線データなどの多様な情報を処理して伝送できる高速大容量通信システムが要求されることによって、適正なチャンネル符号化方式を用いてシステム伝送効率を高くすることが、システム性能向上に必須的な要素で作用するようになる。しかしながら、移動通信システムは、移動通信システムの特性上、データを伝送するときに、チャンネルの状況により雑音と、干渉(interference)及びフェージング(fading)などによって不回避に誤り(error)が発生する。したがって、誤り発生は、情報データの損失をもたらす。 40

【0006】

このような誤り発生による情報データ損失を減少させるために、チャンネルの性格によって多様な誤り制御技術を使用することによって、移動通信システムの信頼度を向上させ 50

ることができる。誤り制御技術の中で、誤り訂正符号(error-correcting code)を使用する技術が一番普遍的である。ここで、誤り訂正符号の代表的な符号であるターボ符号(turbo code)と、低密度パリティ検査(Low Density Parity Check:以下、“LDPC”とする)符号について説明する。

【0007】

ターボ符号

ターボ符号は、最近第3世代移動通信システムで注目されている同期方式と非同期方式ですべて使用されている誤り訂正符号である。従来から順方向誤り訂正のために主に利用された畳み込み符号(convolutional code)に比べて、このターボ符号は、高速データ伝送時に性能利得が優れると知られている。また、ターボ符号は、伝送チャネルで発生する雑音による誤りを効果的に訂正してデータ伝送の信頼度を向上するという長所を有する。

10

【0008】

LDPC符号

LDPC符号は、因子(factor)グラフで和積(sum-product)アルゴリズムに基づいた反復復号アルゴリズムを用いて復号することができる。LDPC符号の復号器(decoder)は、和積アルゴリズムに基づいた反復復号アルゴリズムを使用するため、ターボ符号の復号器に比べて低い複雑度を有するだけでなく、並列処理復号器で実現することが容易である。LDPC符号を因子グラフで表現すると、LDPC符号の因子グラフ上にサイクルが存在するようになる。このサイクルが存在するLDPC符号の因子グラフ上の反復復号は、準最適(sub-optimal)とは、既によく知られている事実である。また、LDPC符号は、反復復号を通じて優れた性能を有することも実験的に立証された事実である。しかしながら、LDPC符号の因子グラフで短い長さのサイクルが多く存在する場合には、LDPC符号の性能劣化が発生される。そのため、LDPC符号の因子グラフ上に短い長さのサイクルが存在しないようにLDPC符号を設計するための研究が持続的に遂行されている。

20

【0009】

LDPC符号の符号化過程は、一般的に高いウェイト(weight)密度を有する生成行列(generating matrix)の特性により低いウェイト密度を有するパリティ検査行列(parity check matrix)を用いる形態で発展されてきた。ここで、“ウェイト”とは、生成行列及びパリティ検査行列を構成する要素(element)のうち0でない値を有する要素の個数を示す。特に、パリティ検査行列でパリティに該当する部分行列(partial matrix)の形態が規則的な形態を有すると、より効率的な符号化が可能である。

30

【0010】

一方、LDPC符号は0でない値を有する多様な符号を含んでいるため、LDPC符号の実用化問題において、多様な形態を有するLDPC符号の効率的な符号化アルゴリズムと効率的な復号アルゴリズムを開発することが非常に重要である。また、LDPC符号のパリティ検査行列は、LDPC符号の性能を決定するため、優れた性能を有するパリティ検査行列を設計することも非常に重要である。すなわち、優れた性能を有する効率的なパリティ検査行列と、効率的な符号化アルゴリズム及び復号アルゴリズムを同時に考慮しなければ、高性能のLDPC符号を生成することが可能になる。

40

【0011】

また、LDPC符号は、大部分の要素が0の値を有し、0の値を有する要素以外の極少数の要素が1の値を有するパリティ検査行列によって定義される。例えば、 (N, j, k) LDPC符号は、ブロック長 N である線形ブロック符号(linear block code)で、各列(column)ごとに j 個の1の値を有する要素と、各行(row)ごとに k 個の1の値を有する要素を有し、1の値を有する要素を除いた要素は、すべて0の値を有する要素で構成された疎(sparse)構造のパリティ検査行列によって定義される。

【0012】

50

上記に説明したように、パリティ検査行列内の各列のウェイト値が j 個に一定で、パリティ検査行列内の各行のウェイト値が k 個に一定の LDPC 符号を“均一 LDPC 符号”と称する。一方、パリティ検査行列内の各列のウェイト値と各行のウェイト値が一定しない LDPC 符号を“不均一 LDPC 符号”と称する。一般的に、均一 LDPC 符号の性能に比べて、不均一 LDPC 符号の性能がさらに優れると知られている。しかしながら、不均一 LDPC 符号の場合に、パリティ検査行列内の各列のウェイト値と各行のウェイト値が一定しないため、すなわち、不均一なため、パリティ検査行列内の各列のウェイトの個数と各行のウェイトの個数を適切に調節しなければ、優れた性能の保障を受けることができない。

【0013】

10

ここで、図1を参照して、 (N, j, k) LDPC 符号、一例として $(8, 2, 4)$ LDPC 符号のパリティ検査行列を説明する。

【0014】

図1は、一般的な $(8, 2, 4)$ LDPC 符号のパリティ検査行列を示す図である。図1を参照すると、 $(8, 2, 4)$ LDPC 符号のパリティ検査行列 H は、8 個の列と 4 個の行で構成されており、各列のウェイトの個数は 2 として均一で、各行のウェイトの個数は 4 として均一である。このように、パリティ検査行列内の各列のウェイトの個数と各行のウェイトの個数が均一である。したがって、図1に示している $(8, 2, 4)$ LDPC 符号は均一 LDPC 符号となる。

【0015】

20

図1では、 $(8, 2, 4)$ LDPC 符号のパリティ検査行列について説明する。その次に、図2を参照して、図1に示した $(8, 2, 4)$ LDPC 符号の因子グラフを説明する。

【0016】

図2は、図1の $(8, 2, 4)$ LDPC 符号の因子グラフを示す図である。図2を参照すると、 $(8, 2, 4)$ LDPC 符号の因子グラフは 8 個の変数ノード(variable node)、すなわち、 x_1 2 1 1 と、 x_2 2 1 3 と、 x_3 2 1 5、 x_4 2 1 7 と、 x_5 2 1 9 と、 x_6 2 2 1 と、 x_7 2 2 3 と、 x_8 2 2 5 と、4 個の検査ノード 2 2 7, 2 2 9, 2 3 1, 2 3 3 で構成される。 $(8, 2, 4)$ LDPC 符号のパリティ検査行列の i 番目の行と j 番目の列が交差する地点にウェイト、すなわち、1 の値を有する要素が存在する場合に、変数ノード x_j と i 番目の検査ノードとの間にブランチ(branch)が形成される。

30

【0017】

上述したように、LDPC 符号のパリティ検査行列は、非常に少ない個数のウェイトを有するため、比較的長い長さを有するブロック符号でも反復復号を通じて復号が可能で、ブロック符号のブロック長さを継続して増加させると、ターボ符号のように Shannon のチャンネル容量限界に近接する形態の性能を示す。流れ伝送方式を使用する LDPC 符号の反復復号過程がターボ符号の反復復号過程にほとんど近接する性能を有する。

一方、高性能の LDPC 符号を生成するための条件を説明すると、次のようである。

【0018】

(1) LDPC 符号の因子グラフ上のサイクルを考慮すべきである。

40

サイクルとは、LDPC 符号の因子グラフで変数ノードと検査ノードを接続するエッジが構成するループ(loop)を示すのに、サイクルの長さはループを構成するエッジの個数に定義される。サイクルの長さが長いということは、LDPC 符号の因子グラフでループを構成する変数ノードと検査ノードを接続するエッジの個数が多いという意味である。その反対に、サイクルの長さが短いということは、LDPC 符号の因子グラフでループを構成する変数ノードと検査ノードを接続するエッジの個数が少ないということを意味する。

【0019】

LDPC 符号の因子グラフ上のサイクルを長く生成するほど、LDPC 符号の性能が増加するようになる。その理由は、次のようである。LDPC 符号の因子グラフ上のサイク

50

ルを長く生成する場合に、LDPC符号の因子グラフ上に短い長さのサイクルが多く存在するときに発生する誤りフロア(error floor)のような性能劣化が発生しないためである。

【0020】

(2) LDPC符号の効率的な符号化を考慮すべきである。

LDPC符号は、LDPC符号の特性上畳み込み符号やターボ符号に比べて符号化の複雑度が高くリアルタイム符号化が容易でない。LDPC符号の符号化複雑度を減少するために、反復累積(Repeat Accumulate: RA)符号などが提案された。反復累積符号もLDPC符号の符号化複雑度を低下させるのに限界を有する。したがって、LDPC符号の効率的な符号化を考慮しなければならない。

10

【0021】

(3) LDPC符号の因子グラフ上の次数分布を考慮すべきである。

一般的に、不均一LDPC符号が均一LDPC符号より性能が優れる。その理由は、不均一LDPC符号の因子グラフが多様な次数を有するためである。ここで、“次数(degree)”とは、LDPC符号の因子グラフ上で各ノード、すなわち変数ノードと検査ノードに接続されているエッジの個数を示す。また、LDPC符号の因子グラフ上の“次数分布”とは、特定次数を有するノードが全体ノードの中にどのくらい存在するかを示す。特定の次数分布を有するLDPC符号の性能が優れるということは、既に証明したところである。

【0022】

20

図3は、一般的なブロックLDPC符号のパリティ検査行列を概略的に示す図である。図3の説明に先立って、ブロックLDPC符号は、効率的な符号化だけでなく効率的なパリティ検査行列の貯蔵及び性能改善をすべて考慮した新たなLDPC符号である。このブロックLDPC符号は、均一LDPC符号の構造を一般化させて拡張した概念のLDPC符号である。図3を参照すると、ブロックLDPC符号のパリティ検査行列は、全体パリティ検査行列を複数の部分ブロックに分割し、部分ブロックの各々に順列行列(permutation matrix))を対応させる形態を有する。図3に示すように、“P”は $N_s \times N_s$ サイズを有する順列行列を示し、この順列行列Pの上付き添え字 a_{ij} は $0 \leq a_{ij} \leq N_s - 1$ 又は $a_{ij} = \infty$ を有する。図3において、“P”は部分ブロックの行の個数を、“q”は部分ブロックの列の個数を、それぞれ示す。“i”は、対応する順列行列がパリティ検査行列の部分ブロックのi番目の行に位置することを意味し、“j”は対応する順列行列がパリティ検査行列の部分ブロックのj番目の列に位置することを意味する。すなわち、

30

【0023】

【数1】

$$P^{a_{ij}}$$

【0024】

は、i番目の行とj番目の列で交差する部分ブロックに位置する順列行列である。

40

【0025】

ここで、図4を参照して、上記の順列行列について説明する。

図4に示すように、循環行列Pは $N_s \times N_s$ サイズを有する正方形行列として、循環行列Pを構成する N_s 個の行のウェイトがそれぞれ1で、循環行列Pを構成する N_s 個の行のウェイトもそれぞれ1である行列を示す。

【0026】

一方、図3で、順列行列の上付き添え字 $a_{ij} = 0$ である場合、すなわち、順列行列 P^0 は単位行列

【0027】

【数 2】

(identity matrix) $I_{N_s \times N_s}$

【0028】

を示し、順列行列 P の上付き添え字 $a_{ij} = \infty$ である場合、すなわち順列行列 P^∞ はゼロ行列を示す。

【0029】

図 3 で、ブロック LDPC 符号の全体パリティ検査行列は、全体列の個数が $N_s \times q$ ($p \leq q$) で、全体行の個数が $N_s \times p$ であるため、ブロック LDPC 符号の全体パリティ検査行列が最大ランク (full rank) を有する場合に、部分ブロックのサイズに関係なく符号化率 (coding rate) は、下記の〈式 1〉のようである。

【0030】

【数 3】

$$R = \frac{N_s \times q - N_s \times p}{N_s \times q} = \frac{q - p}{q} = 1 - \frac{p}{q} \quad (\text{式 1})$$

【0031】

すべての i, j に対して $a_{ij} \neq \infty$ である場合に、部分ブロックの各々に対応する順列行列は、各々ゼロ行列でないことを示し、部分ブロックの各々に対応する順列行列の各行のウェイトは p 、各列のウェイトは q である均一 LDPC 符号となる。ここで、部分ブロックに対応する順列行列を“部分行列”と称する。

【0032】

また、全体パリティ検査行列には $p - 1$ 個の従属的な (dependent) 行が存在するため、符号化率は〈式 1〉で計算した符号化率より高い値を有する。このブロック LDPC 符号は、全体パリティ検査行列を構成する部分行列それぞれの 1 番目の行のウェイト位置が決定されると、残りの $N_s - 1$ 個の行のウェイト位置が決定される。したがって、全体パリティ検査行列の情報を貯蔵するために、不規則にウェイトを選択する場合に比べて必要とするメモリのサイズが $1/N_s$ に縮小される。

【0033】

図 5 は、一般的な均一ブロック LDPC 符号のパリティ検査行列を示す図である。

【0034】

図 5 に示すようにパリティ検査行列は、均一ブロック LDPC 符号、すなわち (s, r) 配列 (array) 符号のパリティ検査行列である。 (s, r) 配列符号は代表的な均一ブロック LDPC 符号として、この (s, r) アレイ符号は、図 3 で $N_s = s$ で、 $q = s$ で、 $p = r$ であるブロック LDPC 符号に該当する。ここで、 s は奇数である素数であり、 r は常に $r \leq s$ の条件を満足する。

【0035】

(s, r) 配列符号のパリティ検査行列は、 s^2 個の列と $r \times s$ 個の行を有し、ランクは $r \times (s - 1)$ となる。ここで、 (s, r) 配列符号のパリティ検査行列のランクが $r(s - 1)$ となる理由は、 (s, r) 配列符号のパリティ検査行列の行方向への r 個の部分行列は、この部分行列内の各々の p 個の行をすべて加算すると、すべての要素が 1 の値を有する行が生成されるためである。すなわち、すべての要素が 1 の値を有する r 個の行が生成されるため、 r 個の従属である行が存在することが分かる。したがって、 (s, r) 配列符号の符号化率 R_{array} は、下記の〈式 2〉のように示す。

【0036】

10

20

30

40

【数 4】

$$R_{array} = \frac{s^2 - r \times (s-1)}{s^2} = 1 - \frac{r \times (s-1)}{s^2} > 1 - \frac{r}{s} \quad (\text{式 2})$$

【0037】

上記に説明したように、(s, r)配列符号は、その代数的(algebraic)な特性から長さが4のサイクルが因子グラフ上に存在しないことがわかり、またメモリ容量も減少させることができる。

【0038】

しかしながら、(s, r)配列符号は、均一LDPCブロック符号であるため、不均一LDPC符号に比べては性能劣化が発生する。また、ブロックLDPC符号の場合に、ブロックLDPC符号自体のランダムな性質(randomness)が低いため、優れた性能を保障することができない。つまり、(s, r)配列符号は、効率的な符号化を考慮したものであるが、符号化において複雑度は相変わらず高く、長さ4のサイクルが存在しないが、長さ6のサイクルは存在する。次数分布を考慮しないため、性能面で劣化が発生する。

【0039】

図6は、一般的な不均一ブロックLDPC符号のパリティ検査行列を示す図である。図6の説明に先立って、不均一ブロックLDPC符号は、図5に説明したように、配列符号を効率的な符号化を考慮して変形した形態のブロックLDPC符号である。図6において、不均一ブロックLDPC符号のパリティ検査行列でk、rはk、 $r \leq s$ の条件を満たす整数で(但し、sは素数)、Iは $s \times s$ サイズの単位行列を示し、0は $s \times s$ サイズのゼロ行列を示す。図6に示すように、不均一ブロックLDPC符号のパリティ検査行列は、図3で $N_s = s$ で、 $q = k$ で、 $p = r$ のブロックLDPC符号のパリティ検査行列に該当する。

【0040】

一方、LDPC符号の効率的な符号化のために、図6のように全体パリティ検査行列の中でパリティに該当する部分行列を完全下三角行列で構成して線形時間内で符号化が可能にした。全体パリティ検査行列の構造、すなわち情報語に該当する部分行列と、パリティに該当する部分行列構造は下記で説明するため、ここではその詳細な説明を省略する。このようにパリティに該当する部分行列を完全下三角行列で構成する場合に、パリティ検査行列の構造的な特性によってパリティ検査行列は常に最大ランクを有するようになる。そのため、変形された配列符号、すなわち不均一LDPC符号のブロック長さはksとなり、符号化率Rは下記の<式3>のように示す。

【0041】

【数 5】

$$R = \frac{k-r}{k} = 1 - \frac{r}{k} \quad (\text{式 3})$$

【0042】

しかしながら、図6のようにパリティに該当する部分行列が完全下三角行列形態を有するパリティ検査行列を有する不均一LDPC符号は、配列符号に比べては符号化面で一層効率的であるが、LDPC符号の生成時に考慮すべき因子グラフ上の次数分布を考慮せず、また長さの短いサイクルに対する除去もやはり考慮しなかった。そのため、ランダムな特性を有する不均一LDPC符号に比べては低い誤り訂正能力を有する。したがって、誤り訂正能力を最大化する不均一LDPC符号に対する必要性が増大されている。

【発明の開示】

【発明が解決しようとする課題】

10

20

30

40

50

【0043】

したがって、本発明の目的は、移動通信システムで誤り訂正能力を最大化するLDPC符号を符号化/復号化する装置及び方法を提供することにある。

【0044】

本発明の他の目的は、移動通信システムで最小サイクル長さが最大になるLDPC符号を符号化/復号化する装置及び方法を提供することにある。

【0045】

また本発明の目的は、移動通信システムで符号化の複雑度が最小化されたLDPC符号を符号化/復号化する装置及び方法を提供することにある。

【課題を解決するための手段】

10

【0046】

上記した目的を達成するための方法において、本発明は、ブロック低密度パリティ検査(Low Density Parity Check: LDPC)符号のパリティ検査行列が情報語に対応する情報部分と、パリティに対応する第1のパリティ部分及び第2のパリティ部分で構成され、誤り訂正性能を向上させるための前記パリティ検査行列を生成する方法であって、前記情報語を前記ブロックLDPC符号で符号化するとき適用される符号化率と、符号語長に基づいて前記パリティ検査行列のサイズを決定する段階と、前記決定されたサイズのパリティ検査行列を予め定められた個数のブロックに分割する段階と、前記ブロックを前記情報部分に対応するブロックと、前記第1のパリティ部分に対応するブロックと、前記第2のパリティ部分に対応するブロックに分類する段階と、前記第1のパリティ部分に分類されたブロックの中に予め定められたブロックに順列行列を配列し、前記第2のパリティ部分に分類されたブロックの中に予め定められたブロックに完全下三角形態で順列行列を配列する段階と、前記情報部分に分類されたブロックに前記ブロックLDPC符号の因子グラフの最小サイクル長さが最大になり、ウェイト値が不均一になるように前記順列行列を配列する段階と、を有することを特徴とする。

20

【0047】

また、本発明は、ブロック低密度パリティ検査(LDPC)符号の復号方法であって、情報語に対応する情報部分と、パリティに対応する第1のパリティ部分及び第2のパリティ部分で構成される前記パリティ検査行列を生成し、前記パリティ検査行列に対応してデインタリービング方式及びインタリービング方式を決定する段階と、受信信号の確率値を検出する段階と、前記受信信号の確率値から以前復号時に生成された信号を減算して第1の信号を生成する段階と、前記第1の信号を入力して前記デインタリービング方式でデインタリービングする段階と、前記デインタリービングされた信号を入力して確率値を検出する段階と、前記デインタリービングされた信号の確率値で前記デインタリービングされた信号を減算して第2の信号を生成する段階と、前記第2の信号を前記インタリービング方式でインタリービングし、前記インタリービングされた信号を反復復号する段階と、を含むことを特徴とする。

30

【0048】

なお、本発明は、ブロック低密度パリティ検査(LDPC)符号の符号化方法であって、情報語と予め生成されている、情報語に対応する情報部分と、パリティに対応する第1のパリティ部分及び第2のパリティ部分で構成される前記パリティ検査行列の第1の部分行列と乗算して第1の信号を生成する段階と、前記情報語と前記パリティ検査行列の第2の部分行列と乗算して第2の信号を生成する段階と、前記第1の信号と、前記パリティ検査行列の第3の部分行列と第4の部分行列の逆行列の行列積を乗算して第3の信号を生成する段階と、前記第2の信号と第3の信号を加算して第4の信号を生成する段階と、前記第4の信号と前記パリティ検査行列の第5の部分行列を乗算して第5の信号を生成する段階と、前記第2の信号と前記第5の信号を加算して第6の信号を生成する段階と、前記第6の信号と前記パリティ検査行列の第3の部分行列と第4の部分行列の逆行列の行列積を乗算して第7の信号を生成する段階と、前記情報語と、前記第4の信号を第1のパリティで、前記第7の信号を第2のパリティとし、前記ブロックLDPC符号フォーマットに対応

40

50

するように多重化して出力する段階と、を有することを特徴とする。

【0049】

本発明は、パリティ検査行列は、複数の情報部分ブロックと複数のパリティ部分ブロックの行と列の行列形態で配列され、前記パリティ検査行列は前記情報部分ブロックの行列で構成された情報部分と前記パリティ部分ブロックの行列で構成されたパリティ部分に区分され、前記情報部分ブロックの各々は複数の情報ビットを示す行列で構成され、前記パリティ部分ブロックの各々は、複数のパリティビットを示す行列で構成され、前記パリティ検査行列は複数の行に存在する情報部分ブロックとパリティ部分ブロックが各々第1の情報行列と第1のパリティ行列及び第2のパリティ行列に分割され、前記複数の行を除いた残りの行に存在する情報部分ブロックとパリティ部分ブロックが各々第2の情報行列と第3のパリティ行列及び第4のパリティ行列に分割され、前記第1及び第2の情報行列と、第1及び第3のパリティ行列と、第2及び第4のパリティ行列は各々同一の列に配列され、誤り訂正性能を向上させるための前記パリティ検査行列を生成する方法であって、前記第4のパリティ行列と第2のパリティ行列の逆行列と前記第1のパリティ行列の積と前記第3のパリティ行列の和が単位行列になるように設定する段階と、前記第1のパリティ行列と前記第3のパリティ行列に対応する第1のパリティベクトルの転置ベクトルは、前記第4のパリティ行列と前記第2のパリティ行列の逆行列と前記第1の情報行列の積と前記第2の情報行列の和を前記第1の情報行列と前記第2の情報行列に対応する情報ベクトルを乗算した値になるように決定する段階と、前記第2のパリティ行列と第4のパリティ行列に対応する第2のパリティベクトルの転置ベクトルは、前記第2のパリティ行列の逆行列と前記第1の情報行列と前記情報ベクトルの転置ベクトルを乗算した値と前記第1のパリティ行列と前記第1のパリティベクトルの転置ベクトルを乗算した値を加算した値を乗算した値になるように決定する段階と、を有することを特徴とする。

【0050】

さらに、本発明は、ブロック低密度パリティ検査(LDPC)符号のパリティ検査行列は、複数の部分ブロックの行と列の行列形態で配列され、前記複数の部分ブロック各々には $N_s \times N_s$ サイズを有する行列が前記複数の部分ブロックの各々に対応して予め定められた指数だけシフトして生成された順列行列が配列され、誤り訂正性能を向上させるための前記パリティ検査行列を生成する方法であって、前記ブロックLDPC符号のブロックサイクルを任意の第1の値に決定する段階と、前記ブロックサイクルを決定した後に、前記部分ブロックの各々に配列された順列行列の中に、その指数が奇数である順列行列の指数の和から前記部分ブロックの各々に配列された順列行列のうち、その指数が偶数である順列行列の指数の和を減算した値と任意の第2の値を乗算した値になるように前記第2の値を決定し、前記部分ブロックの各々が前記第1の値と第2の値を乗算した値に対応するサイクルを有するように制御する段階と、を有することを特徴とする。

【0051】

上記の目的を達成するための装置において、本発明は、ブロック低密度パリティ検査(LDPC)符号の復号装置であって、所定の制御により、情報語に対応する情報部分と、パリティに対応する第1のパリティ部分及び第2のパリティ部分で構成される前記パリティ検査行列を構成する列のそれぞれのウェイトにより変数ノードを接続して受信信号の確率値を検出して出力する変数ノード復号器と、前記変数ノード復号器から出力された信号から以前復号時に生成された信号を減算して出力する第1の加算器と、前記第1の加算器から出力された信号を入力して前記パリティ検査行列に対応して設定されたデインタリービング方式でデインタリービングして出力するデインタリーバと、所定の制御信号により、前記パリティ検査行列を構成する行のそれぞれのウェイトに対応して検査ノードを接続し、前記デインタリーバからの出力信号の確率値を検出して出力する検査ノード復号器と、前記検査ノード復号器の出力信号から、前記デインタリーバから出力された信号を減算する第2の加算器と、前記第2の加算器から出力された信号を前記パリティ検査行列により設定されたインタリービング方式でインタリービングして前記変数ノード復号器及び前記第1の加算器に出力するインタリーバと、前記パリティ検査行列を生成し、前記デイン

タリーピング方式及びインタリーピング方式を前記パリティ検査行列に対応して制御する制御器と、を含むことを特徴とする。

【0052】

また、本発明は、ブロック低密度パリティ検査(LDPC)符号の符号化装置であって、情報語を入力して予め生成されている、情報語に対応する情報部分と、パリティに対応する第1のパリティ部分及び第2のパリティ部分で構成される前記パリティ検査行列の第1の部分行列と乗算する第1の行列乗算器と、前記情報語を入力して前記パリティ検査行列の第2の部分行列と乗算する第2の行列乗算器と、前記第1の行列乗算器から出力された信号と、前記パリティ検査行列の第3の部分行列と第4の部分行列の逆行列の行列積を乗算する第3の行列乗算器と、前記第2の行列乗算器から出力された信号と第3の行列乗算器から出力された信号とを加算する第1の加算器と、前記第1の加算器から出力された信号と前記パリティ検査行列の第5の部分行列とを乗算する第4の行列乗算器と、前記第2の行列乗算器から出力された信号と前記第4の行列乗算器から出力された信号とを加算する第2の加算器と、前記第2の加算器から出力された信号と前記パリティ検査行列の第3の部分行列と第4の部分行列の逆行列の行列積を乗算する第5の行列乗算器と、前記情報語と、前記第1の加算器の出力信号を第1のパリティとし、前記第5の行列乗算器の出力信号を第2のパリティとして前記ブロックLDPC符号フォーマットに相応するように多重化して出力するスイッチと、を含むことを特徴とする。

10

【発明の効果】

【0053】

20

本発明は、移動通信システムで最小サイクル長さが最大になるブロックLDPC符号を提案することによって、誤り訂正能力を最大化させてシステム性能を向上させる効果を有する。また、本発明は、効率的なパリティ検査行列を生成することによって、ブロックLDPC符号の符号化複雑度を最小化させる効果も有する。

【発明を実施するための最良の形態】

【0054】

以下、本発明の望ましい実施形態を添付の図面を参照して詳細に説明する。

下記の説明で、本発明に関連した公知の機能又は構成に関する説明が本発明の要旨を不明にすると判断された場合に、その詳細な説明を省略する。

【0055】

30

本発明は、優秀な性能の不均一低密度パリティ検査(Low Density Parity Check:以下、“LDPC”とする)符号を符号化及び復号化する方案を提示する。すなわち、本発明は、因子(factor)グラフ上の最小サイクルの長さが最大になり、符号化複雑度が最小になり、因子グラフ上の次数分布が最適化される分布を有する不均一LDPC符号の符号化及び復号化方案を提示する。

【0056】

LDPC符号の因子グラフ上のサイクルとは、因子グラフで変数ノードと検査ノードを接続するエッジ(edge)によって構成されるループを示すが、サイクルの長さはループを構成するエッジの個数に定義される。サイクルの長さが長いということは、因子グラフでループを構成する変数ノードと検査ノードを接続するエッジの個数が多いことを示す。因子グラフ上のサイクルの長さを長く生成するほど、LDPC符号の性能が向上するようになる。その反面、因子グラフ上に長さの短いサイクルが多く存在するほどLDPC符号は誤りフロア(error floor)の性能劣化が示すため、誤り訂正能力が低下する。すなわち、因子グラフ上に長さの短いサイクルが多く存在する場合に、長さの短いサイクルに属している任意のノードから出発した自分の情報が少ない反復回数を経てまた帰ってくるようになり、反復回数が増加するほど、その情報が継続して自分に帰って来るようになる。したがって、情報アップデートがよくなされなくて、結局、LDPC符号の誤り訂正能力が低下される。

40

【0057】

図7は、パリティ検査行列が4個の部分行列で構成されたブロックLDPC符号のサイ

50

クル構造を概略的に示す図である。

【0058】

図7を説明するに先立ち、ブロックLDPC符号は効率的な符号化だけでなく効率的なパリティ検査行列の貯蔵及び性能の改善をすべて考慮した新たなLDPC符号である。このブロックLDPC符号は、均一LDPC符号の構造を一般化させて拡張した概念のLDPC符号である。図7に示すように、ブロックLDPC符号のパリティ検査行列は、4個のブロックで構成され、斜線は1の値を有する要素(element)が存在する位置を意味し、死線部分以外の部分はすべて0の値を有する要素が存在する位置を意味する。また、Pは、従来技術の図4で説明した順列行列(permutation matrix)と同一の順列行列を示す。ここで、順列行列Pは、図4で説明したように、 $N_s \times N_s$ サイズを有する正方形行列で、この順列行列Pを構成する N_s 個の行のそれぞれのウェイトが1であり、 N_s 個の行のそれぞれのウェイトも1である行列を示す。ここで、“ウェイト”とは、パリティ検査行列を構成する要素のうち、0でない値を有する要素の個数を示す。

10

【0059】

図7に示すブロックLDPC符号のサイクル構造を分析するために、部分行列 P^a の i 番目の行に位置する1の値を有する要素を基準として定め、 i 番目の行に位置する1の値を有する要素を“0-ポイント”と称する。ここで、“部分行列”は、部分ブロックに対応する行列を示す。すると、0-ポイントは部分行列 P^a の $(i+a)$ 番目の列に位置される。

20

【0060】

0-ポイントと同一の行に位置した部分行列 P^b での1の値を有する要素を“1-ポイント”と称する。この0-ポイントと同じ理由で、1-ポイントは部分行列 P^b の $(i+b)$ 番目の列に位置する。

【0061】

次に、1-ポイントと同一の列に位置した部分行列 P^c での1の値を有する要素を“2-ポイント”と称する。部分行列 P^c が単位行列Iの列の各々を右にモジュロ(modulo) N_s に対して c だけ移動して得られた行列であるため、2-ポイントはこの部分行列 P^c の $(i+b-c)$ 番目の行に位置するようになる。

【0062】

また、2-ポイントと同じ行に位置した部分行列 P^d での1の値を有する要素を“3-ポイント”と称する。3-ポイントは、部分行列 P^d での $(i+b-c+d)$ 番目の列に位置するようになる。

30

【0063】

最後に、3-ポイントと同一の列に位置した部分行列 P^a での1の値を有する要素を“4-ポイント”と称する。この4-ポイントは、部分行列 P^a の $(i+b-c+d-a)$ 番目の行に位置するようになる。

【0064】

図7に示すLDPC符号のサイクル構造で、長さが4であるサイクルが存在すると、0-ポイントと4-ポイントは相互に同一の位置となる。すなわち、0-ポイントと4-ポイントとの間には、下記の<式4>のような関係が成立するようになる。

40

【0065】

【数6】

$$\begin{aligned} i &\equiv i + b - c + d - a \pmod{N_s} \quad \text{or} \\ i + a &\equiv i + b - c + d \pmod{N_s} \end{aligned} \quad (\text{式4})$$

【0066】

そして、上記の<式4>をさらに整理すると、下記の<式5>のようである。

【0067】

50

【数 7】

$$a + c \equiv b + d \pmod{N_s} \quad (\text{式 5})$$

【0068】

その結果、＜式 5＞のような関係が成立するときに、長さ 4 のサイクルが生成される。一般的に、0-ポイントと 4m-ポイントが最初に同一になる場合は、 $i \equiv i + m(b - c + d - a) \pmod{N_s}$ 関係が成立され、下記の＜式 6＞のような関係が成立される。

【0069】

【数 8】

10

$$m(a - b + c - d) \equiv 0 \pmod{N_s} \quad (\text{式 6})$$

【0070】

さらに説明すると、与えられた a、b、c、d に対して＜式 6＞を満たす正の整数の中で最小値を有する正の整数を m とする場合に、図 7 に示すようなブロック LDPC 符号のサイクル構造では長さが 4m であるサイクルが最小長さを有するサイクルとなる。

【0071】

結果的に、上記したように、 $(a - b + c - d) \neq 0$ である場合に $\gcd(N_s, a - b + c - d) = 1$ を満たすと、 $m = N_s$ になる。ここで、 $\gcd(N_s, a - b + c - d)$ は、整数 N_s と $a - b + c - d$ の最大公約数の計算のための機能をする。したがって、長さが $4N_s$ であるサイクルが最小長さを有するサイクルとなる。

20

【0072】

図 7 に説明したように、ブロック LDPC 符号のサイクルに関する分析は、ブロック LDPC 符号のパリティ検査行列を構成するブロックの個数が 4 を超える場合、すなわちパリティ検査行列を構成する部分行列の個数が 4 を超える場合にも適用可能である。ここで、図 8 を参照して、パリティ検査行列を構成する部分行列の個数が 4 を超える場合の LDPC 符号のサイクル構造について説明する。

【0073】

図 8 は、パリティ検査行列が 6 個の部分行列で構成されたブロック LDPC 符号のサイクル構造を概略的に示す図である。

30

【0074】

図 8 に示すブロック LDPC 符号のパリティ検査行列は、6 個のブロックで構成されており、図 7 で説明したように、斜線は 1 の値を有する要素が存在する位置を示し、この斜線部分以外の部分はすべて 0 の値を有する要素が存在する位置を示す。また、P も従来技術である図 4 で説明した順列行列と同一の順列行列を示す。図 7 で説明したような方法で、図 8 におけるブロック LDPC 符号のサイクル構造を分析すると、長さが 6m であるサイクルが最小長さを有するサイクルとなる。

【0075】

一般的に、0-ポイントと 6m-ポイントが最初に同一になる場合は、 $i \equiv i + m(b - c + d - e + f - a) \pmod{N_s}$ の関係が成立され、下記の＜式 7＞のような関係を満たすようになる。

40

【0076】

【数 9】

$$m(b - c + d - e + f - a) \equiv 0 \pmod{N_s} \quad (\text{式 7})$$

【0077】

これら与えられた a、b、c、d、e、f に対して＜式 7＞を満たす正の整数の中で、最小値を有する正の整数を “m” とすると、図 8 に示すブロック LDPC 符号のサイクル

50

構造では長さが6mであるサイクルが最小長さを有するサイクルになる。

【0078】

上記に述べたように、 $(a-b+c-d+e-f) \neq 0$ である場合に $\gcd(N_s, a-b+c-d+e-f) = 1$ を満たすと、 $m = N_s$ になる。したがって、長さが $6N_s$ であるサイクルが最小長さを有するサイクルになる。

【0079】

上記に説明したように、ブロックLDPC符号について、次のような規則が推定可能である。

【0080】

<規則1>

ブロックLDPC符号で長さが2lであるサイクルが存在すると、下記の<式8>の条件を満たすべきである。

【0081】

【数10】

$$a_1 + a_3 + a_5 + \dots + a_{2l-1} \equiv a_2 + a_4 + a_6 + \dots + a_{2l} \pmod{N_s} \quad (\text{式8})$$

【0082】

<式8> $d, a_i (i = 1, 2, \dots, 2l)$ は長さが2lであるサイクルが順次的に通過する順列行列の指数を示す。すなわち、ブロックLDPC符号のパリティ検査行列を構成する部分ブロックに対して長さが2lであるサイクルが

【0083】

【数11】

$$P^{a_1} \rightarrow P^{a_2} \rightarrow \dots \rightarrow P^{a_{2l}}$$

【0084】

の順に通過することを示す。ここで、すべての a_i が相互に異なる必要はなく、重複的に通過する部分ブロックが存在することも可能であることはもちろんである。

【0085】

<規則2>

mは、下記の<式9>を満たす最小の正の整数であると定義する。

【0086】

【数12】

$$m \left(\sum_{i: \text{odd}} a_i - \sum_{j: \text{even}} a_j \right) \equiv 0 \pmod{N_s} \quad (1 \leq i, j \leq 2l) \quad (\text{式9})$$

【0087】

<式9>で、 a_i は、全体パリティ検査行列でブロック単位のサイクルが形成されるように適切に選択された順列行列の指数である。そして、 a_i は<規則1>の説明と同様に、すべて相互に異なる必要はなく、重複的に通過する部分ブロックが存在することも可能であることはもちろんである。つまり、部分行列

【0088】

【数13】

$$P^{a_i}$$

【0089】

は最小長さが2lmであるサイクル構造を有する。

【0090】

10

20

30

40

50

＜規則 1＞と＜規則 2＞を用いると、ブロック LDPC 符号のサイクル構造の特性を分析することが容易になる。一例として、＜規則 1＞と＜規則 2＞を用いると、配列符号で最小長さが 6 であるサイクルが正確にどのくらい分布しているかが分かるだけでなく、下記で説明するブロック LDPC 符号のブロック単位のサイクル(以下、“ブロックサイクル”と称する)構造の特性分析も容易になる。ここで、ブロックサイクルは、パリティ検査行列の構成において、サイクル長さを調整するのに重要な要素であり、ブロックサイクルを図 9 及び＜規則 1＞と＜規則 2＞を使用して説明する。

【0091】

図 9 は、ブロック LDPC 符号のブロックサイクル構造を概略的に示す図である。図 9 を参照すると、ブロック LDPC 符号を構成するブロックの各々がウェイト 1 を有すると仮定し、このブロックがサイクルを構成する場合に“ブロックサイクルを構成する”と定義する。図 9 には、左側から 4 個のブロックで構成されたブロックサイクルと、6 個のブロックで構成されたブロックサイクルと、8 個のブロックで構成されたブロックサイクルが示している。そして、＜規則 1＞と＜規則 2＞で説明したように、短い長さのブロックサイクルが構成されるとしてもブロックサイクルを構成するブロックの各々に該当する部分行列を適切に選択するようになると、実際のパリティ検査行列では短い長さのサイクルが生成されないように制御できる。しかしながら、ブロック LDPC 符号で複数のブロックサイクルが重複される場合は、ブロックサイクル内の実際のサイクルの最小長さが減少される。その結果、実際のパリティ検査行列で短い長さのサイクルが生成されるという問題点が発生する。

【0092】

ここで、図 10 及び＜規則 1＞と＜規則 2＞を用いてブロック LDPC 符号で複数のブロックサイクルが重複される問題点について説明する。また、ブロック LDPC 符号のパリティ検査行列を生成するときに、重複されたブロックサイクルを避けるべき理由についても説明する。

【0093】

図 10 は、パリティ検査行列の 6 個の部分行列が重複されたブロック LDPC 符号のブロックサイクル構造を概略的に示す図である。図 10 に示している矢印に従って、次のような順次的なブロック順序を考慮することができる。

【0094】

【数 14】

$$P^{a_1} \rightarrow P^{a_2} \rightarrow P^{a_4} \rightarrow P^{a_3} \rightarrow P^{a_5} \rightarrow P^{a_6} \rightarrow P^{a_2} \rightarrow P^{a_1} \rightarrow P^{a_3} \rightarrow P^{a_4} \rightarrow P^{a_6} \rightarrow P^{a_5} \rightarrow P^{a_1}$$

【0095】

順次的なブロック順序による部分行列の指数は、 N_s の値に関係なく下記の＜式 10＞の条件を常に満たす。

【0096】

【数 15】

$$a_1 - a_2 + a_4 - a_3 + a_5 - a_6 + a_2 - a_1 + a_3 - a_4 + a_6 - a_5 \equiv 0 \pmod{N_s} \quad (\text{式 10})$$

【0097】

＜式 10＞を＜規則 2＞で説明した＜式 9＞に適用させると、 $m=1$ となる。したがって、図 10 に示すような 6 個の部分行列が重複されたブロックサイクルが存在するブロック LDPC 符号の場合に、全体パリティ検査行列を構成するいずれの部分行列を選択しても、常にその長さが 12 であるサイクル構造を含むようになる。すなわち、図 10 に示すような 6 個の部分行列が重複されたブロックサイクルが存在するブロック LDPC 符号の場合に、パリティ検査行列の最小サイクル長さは最大 12 に制限されるものである。

【0098】

図11は、パリティ検査行列の7個の部分ブロックが重複されたブロックLDPC符号のブロックサイクル構造を概略的に示す図である。図11に示している矢印に従って、次のような順次的なブロック順序を考慮することができる。

【0099】

【数16】

$$P^{a_1} \rightarrow P^{a_2} \rightarrow P^{a_4} \rightarrow P^{a_5} \rightarrow P^{a_7} \rightarrow P^{a_6} \rightarrow P^{a_2} \rightarrow P^{a_1} \rightarrow P^{a_3} \rightarrow P^{a_4} \rightarrow P^{a_6} \rightarrow P^{a_7} \\ \rightarrow P^{a_5} \rightarrow P^{a_3} \rightarrow P^{a_1}$$

【0100】

順次的なブロック順序による部分行列の指数は、 N_s の値に関係なく下記の<式11>の条件を常に満たす。

【0101】

【数17】

$$a_1 - a_2 + a_4 - a_5 + a_7 - a_6 + a_2 - a_1 + a_3 - a_4 + a_6 - a_7 + a_5 - a_3 \equiv 0 \pmod{N_s} \quad (\text{式11})$$

【0102】

<式11>を<規則2>で説明した<式9>に適用すると、 $m=1$ となる。したがって、図11に示すようなパリティ検査行列の7個の部分行列が重複されたブロックサイクルが存在するブロックLDPC符号の場合に、全体パリティ検査行列を構成するいずれの部分行列を選択しても、常にその長さが14であるサイクル構造を含む。すなわち、図11に示すように、パリティ検査行列の7個の部分行列が重複されたブロックサイクルが存在するブロックLDPC符号の場合に、パリティ検査行列の最小サイクル長さは最大14に制限される。

【0103】

上述したように、ブロックLDPC符号でパリティ検査行列を構成するブロックの間にブロックサイクルが多く重複されている場合に、パリティ検査行列の部分行列を選択する方法に関係なく、サイクルの最小長さを最大化するのに制限が発生してその性能が劣化することが分かる。したがって、ブロックLDPC符号でパリティ検査行列を生成する場合に、できるだけブロックサイクル自体を少なく生成して重複ブロックサイクルが発生しないように考慮しなければならない。

【0104】

次に、ブロックサイクル以外に効率的な符号化を考慮してブロックLDPC符号のパリティ検査行列を生成する方法について説明する。

【0105】

本発明では、ブロックLDPC符号の符号化方式としてRichardson-Urbanke方式を使用することにする。このRichardson-Urbanke方式を符号化方式で使用するため、パリティ検査行列は完全下三角行列と類似した形態を有するほど符号化の複雑度も最小化させることができる。

【0106】

図12は、完全下三角行列形態を有するパリティ検査行列を示す図である。

【0107】

図12に示すように、パリティ検査行列は、完全下三角行列形態を有し、情報部分(information part)とパリティ部分(parities part)で構成される。ここで、情報部分は、ブロックLDPC符号を符号化する過程で、実際の情報語にマッピングされるパリティ検査行列の部分を示し、パリティ部分はブロックLDPC符号を符号化する過程で実際パリティにマッピングされるパリティ検査行列の部分を示す。パリティ部分は、図12に示すように、単位行列Iに基づいて0行列と部分行列が存在し、部分行列が完全下三角形態を有する。

10

20

30

40

50

【0108】

図13は、完全下三角行列形態と類似した形態を有するパリティ検査行列を示す図である。図13に示すように、パリティ検査行列が、図12に示した完全下三角行列形態のパリティ検査行列に比べてはパリティ部分の形態が完全下三角行列形態から外れる。図13において、情報部分の順列行列Pの上付き添え字 a_{ij} は $0 \leq a_{ij} \leq N_s - 1$ 又は $a_{ij} = \infty$ である。この順列行列Pの上付き添え字 $a_{ij} = 0$ である場合、すなわち順列行列 P^0 は単位行列

【0109】

【数18】

$$I_{N_s \times N_s}$$

10

【0110】

を示し、順列行列Pの上付き添え字 $a_{ij} = \infty$ である場合、すなわち順列行列 P^∞ はゼロ行列を示す。図13で、“m”は情報部分にマッピングされる部分ブロックの行の個数を、“q”はパリティ部分にマッピングされる部分ブロックの列の個数を、それぞれ示す。“i”は、対応する順列行列がパリティ検査行列の部分ブロックのi番目の行に位置することを意味し、“j”は対応する順列行列がパリティ検査行列の部分ブロックのj番目の列に位置することを意味する。すなわち、

20

【0111】

【数19】

$$P^{a_{ij}}$$

【0112】

は、i番目の行とj番目の列で交差する部分ブロックに位置する順列行列である。

また、上記のパリティ部分の順列行列の上付き添え字 a_{ij} 、x、yは、順列行列Pの指数(上付き添え字)を示し、但し、説明の便宜上、この a_{ij} 、x、yは情報部分との区分のため相互に異なるように設定しただけである。すなわち、図13で、

30

【0113】

【数20】

$$P^{a_1} \sim P^{a_m}$$

【0114】

も順列行列で、パリティ部分の対角部分に位置する。上付き添え字 $a_1 \sim a_m$ は部分行列に順次にインデックスが付けられることを示す。図13で、 p^x と p^y も順列行列で、説明の便宜上、情報部分との区分のため相互に異なるように設定して示す。

40

【0115】

図13に示すようなパリティ検査行列を有するブロックLDPC符号のブロック長さをNであると仮定すると、ブロックLDPC符号の符号化複雑度はブロック長さNに対して線形的に増加する。

【0116】

図13のパリティ検査行列を有するLDPC符号の一番大きい問題点は、部分ブロックの長さが N_s であるときに、ブロックLDPC符号の因子グラフ上に常に次数(degree)が1である N_s 個の検査ノードが生成されることである。ここで、次数が1である検査ノードは、反復復号による性能改善に影響を与えることができない。したがって、R

50

richardson-Urbanke方式に基づいた標準LDPC符号は、次数が1である検査ノードを含んでいない。したがって、次数が1である検査ノードを含まずに効率的な符号化が可能ないようにパリティ検査行列を設計するために、図13のパリティ検査行列を基本的なパリティ検査行列として仮定する。図13のように、部分行列で構成されたパリティ検査行列で部分行列の選択は、ブロックLDPC符号の性能改善において非常に重要な要素で、したがって部分行列の適切な選択基準を探すことも非常に重要な要素となる。

【0117】

したがって、ブロックLDPC符号の生成において、次のような設計基準を考慮してパリティ検査行列を構成する。

10

【0118】

＜ブロックLDPC符号のパリティ検査行列の設計基準＞

【0119】

(1)パリティ部分は固定された形態を有するように構成される。

パリティ部分が固定された形態を有するということは、後述する図16に示すように単位行列が位置する形態を有することを意味する。

【0120】

(2)優先的に次数の低い部分行列から順次的に選択する。

本発明で、部分行列の“次数”とは、3と5との間の次数と呼ばれる。また、部分行列は、次数の低い部分行列から順次に選択するときに、ブロックサイクルができるだけ少なく生成されるように配列し、次数の低い部分行列間の最小長さを有するサイクルをできるだけ長く構成する。

20

【0121】

(3)次数の低い部分行列をすべて構成した後に、次数の高い部分行列を順次に構成する。次数の高い部分行列を配列するときに、できるだけ全体的に最小長さのサイクルを長く構成する。

【0122】

上記したブロックLDPC符号のパリティ検査行列設計基準に基づいて、ブロックLDPC符号のパリティ検査行列の設計方法を説明する。

【0123】

ここで、ブロックLDPC符号のパリティ検査行列の設計方法とブロックLDPC符号の符号化方法を容易にするために、図13に示すようなパリティ検査行列を図14のように6個の部分行列で構成された形態であると仮定する。

30

【0124】

図14は、図13のパリティ検査行列を6個の部分ブロックに分割した図である。図14を参照すると、図13に示したように、ブロックLDPC符号のパリティ検査行列を情報部分sと、第1のパリティ部分 p_1 と、第2のパリティ部分 p_2 の部分ブロックに分割する。ここで、情報部分は、図12及び図13で説明した情報部分のようにブロックLDPC符号を符号化する過程で、実際の情報語にマッピングされるパリティ検査行列の部分を示し、但し、説明の便宜上、情報部分sは参照符号を異にして表示しただけである。また、第1のパリティ部分 p_1 と第2のパリティ部分 p_2 は、図12及び図13に説明したパリティ部分のようにブロックLDPC符号を符号化する過程で、実際のパリティにマッピングされるパリティ検査行列の部分を示し、パリティ部分を2個の部分に分割した。

40

【0125】

部分行列A、Cは情報部分sの部分ブロック、すなわち部分ブロックA、Cに対応し、部分行列B、Dは第1のパリティ部分 p_1 の部分ブロックB、Dに対応し、部分行列T、Eは第2のパリティ部分 p_2 の部分ブロックT、Eに対応する。ここで、図14にパリティ検査行列が7個の部分ブロックに分割されるように示しているが、‘0’は別途の部分ブロックでなく、部分ブロックTに対応する部分行列Tが完全下三角形を有するため、対角線を中心としてゼロ行列が配置された領域を‘0’で表示する。情報部分sと、第1

50

のパリティ部分 p_1 と、第 2 のパリティ部分 p_2 の部分行列を用いて符号化方法を簡略にする過程は、下記の図 17 で説明する。

【0 1 2 6】

以下に、図 14 の部分行列は、図 15 を参照して説明する。

【0 1 2 7】

図 15 は、図 14 の部分行列 B の転置行列と、部分行列 E と、部分行列 T と、部分行列 T の逆行列を示す図である。図 15 を参照すると、

【0 1 2 8】

【数 2 1】

$$P^{(k_1 \sim k_2)} \text{ は } \prod_{i=k_1}^{k_2} P^{a_i} = P^{\sum_{i=k_1}^{k_2} a_i}$$

10

【0 1 2 9】

を示す。また、図 15 に示す順列行列、例えば、

【0 1 3 0】

【数 2 2】

P^{a_1}

20

【0 1 3 1】

は単位行列となりうる。上記したように、順列行列の指数、すなわち a_1 が 0 である場合には、順列行列

【0 1 3 2】

【数 2 3】

P^{a_1}

【0 1 3 3】

が単位行列となる。また、順列行列の指数、すなわち、 a_1 が予め設定された値により増加する場合には、順列行列が増加した設定値により循環シフトされ、つまり、順列行列

30

【0 1 3 4】

【数 2 4】

P^{a_1}

【0 1 3 5】

が単位行列になる。

【0 1 3 6】

図 17 は、本発明の実施形態によるブロック LDPC 符号のパリティ検査行列の生成過程を示すフローチャートである。図 17 の説明に先立って、ブロック LDPC 符号を生成するためには、生成しようとするブロック LDPC 符号の符号語サイズと符号化率を決定し、この決定された符号語サイズと符号化率によりパリティ検査行列のサイズを決定すべきである。ブロック LDPC 符号の符号語サイズが N で、符号化率を R であると仮定すると、パリティ検査行列のサイズは $N(1-R) \times N$ となる。また、図 17 に示すようにブロック LDPC 符号のパリティ検査行列生成過程は、最初に通信システムのシステム状態に応じて生成され、この生成されたパリティ検査行列を用いることによって、実質的にパリティ検査行列の生成過程は 1 回のみを遂行すると良い。

40

【0 1 3 7】

図 17 を参照すると、制御器は、ステップ 1711 で、サイズ $N(1-R) \times N$ のパリティ

50

検査行列を横軸に p 個のブロックに分割し、縦軸で q 個のブロックに分割して総 $p \times q$ ブロックに分割した後に、ステップ 1713 に進行する。ここで、ブロックそれぞれのサイズは $N_s \times N_s$ であるため、パリティ検査行列は $N_s \times q$ 個の行と、 $N_s \times p$ 個の列で構成される。ステップ 1713 で、制御器は、 $p \times q$ 個のブロックに分割したパリティ検査行列を情報部分 s とパリティ部分、すなわち第 1 のパリティ部分 p_1 と第 2 のパリティ部分 p_2 に分類し、ステップ 1715 及びステップ 1721 に進行する。

【0138】

ステップ 1715 で、制御器は、情報部分 s をブロック LDPC 符号の高性能を保障する次数分布による '0' でないブロック又は非ゼロ行列、'0' であるブロック又はゼロ行列であるブロックを決定し、ステップ 1717 に進行する。ここで、ブロック LDPC 符号の高性能を保障する次数分布は上記したようであるため、その詳細な説明を省略する。ステップ 1717 で、制御器は、ブロック LDPC 符号の高性能を保障する次数分布により決定したブロックの中に低い次数を有するブロックのうち、非ゼロ行列部分に上記したようにブロックサイクルの最小サイクル長さが最大になるように順列行列

10

【0139】

【数 25】

$$P^{a_y}$$

【0140】

を決定し、ステップ 1719 に進行する。ここで、順列行列

20

【0141】

【数 26】

$$P^{a_y}$$

【0142】

を決定するとき、情報部分 s だけでなく第 1 及び第 2 のパリティ部分 P_1 、 P_2 のブロックサイクルも考慮して決定すべきである。

【0143】

30

ステップ 1719 で、制御器は、ブロック LDPC 符号の優れた性能を保障する次数分布により決定されたブロックの中に、高い次数を有するブロックのうちの非ゼロ行列部分にランダムに順列行列

【0144】

【数 27】

$$P^{a_y}$$

【0145】

を決定して終了する。ここで、高い次数を有するブロックのうちの非ゼロ行列部分に適用する順列行列

40

【0146】

【数 28】

$$P^{a_y}$$

【0147】

を決定するときにも、ブロックサイクルの最小サイクルサイズが最大になるように順列行列

【0148】

50

【数 2 9】

$$P^{a_y}$$

【0 1 4 9】

を決定すべきである。また、情報部分 s だけでなく第 1 のパリティ部分 p_1 と第 2 のパリティ部分 p_2 のブロックサイクルも考慮して決定すべきである。上記のように、パリティ検査行列の情報部分 s に順列行列

【0 1 5 0】

【数 3 0】

10

$$P^{a_y}$$

【0 1 5 1】

を配列した形態を図 1 6 に示す。

【0 1 5 2】

ステップ 1 7 2 1 で、制御器は、第 1 及び第 2 のパリティ部分 p_1 、 p_2 を 4 個の部分行列 B、T、D、E に分割した後に、ステップ 1 7 2 3 に進行する。ステップ 1 7 2 3 で、制御器は、部分行列 B を構成する部分ブロックの中に 2 個の部分ブロックに 0 でない順列行列 P^y と

20

【0 1 5 3】

【数 3 1】

$$P^{a_i}$$

【0 1 5 4】

を入力し、ステップ 1 7 2 5 に進行する。ここで、部分行列 B を構成する部分ブロックの中に 2 個の部分ブロックに 0 でない順列行列 P^y と

【0 1 5 5】

【数 3 2】

30

$$P^{a_i}$$

【0 1 5 6】

を入力する構造は、既に図 1 5 を参照して説明した。

【0 1 5 7】

ステップ 1 7 2 5 で、制御器は、部分行列 T の対角部分ブロックには単位行列 I を入力し、部分行列 T の対角成分の下に (i, i + 1) 番目の部分ブロックには任意の順列行列

【0 1 5 8】

【数 3 3】

40

$$P^{a_2}, P^{a_3}, \dots, P^{a_{m-1}}$$

【0 1 5 9】

を入力し、ステップ 1 7 2 7 に進行する。ここで、部分行列 T の対角部分ブロックには単位行列 I を入力し、この部分行列 T の対角成分の下に (i, i + 1) 番目の部分ブロックには任意の順列行列

【0 1 6 0】

【数 3 4】

$$P^{a_2}, P^{a_3}, \dots, P^{a_{m-1}}$$

【0 1 6 1】

を入力する方法は、図 1 5 を参照して説明した。

【0 1 6 2】

ステップ 1 7 2 7 で、制御器は、部分行列 D に順列行列 P^x を入力した後に、ステップ 1 7 2 9 に進行する。ステップ 1 7 2 9 で、制御器は、部分行列 E には最後の部分ブロックのみに

10

【0 1 6 3】

【数 3 5】

$$P^{a_m}$$

【0 1 6 4】

を入力して終了する。ここで、部分行列 E を構成する部分ブロックの中に最後の部分ブロックに 2 個の

【0 1 6 5】

【数 3 6】

20

$$P^{a_m}$$

【0 1 6 6】

を入力する構造は、既に図 1 5 を参照して説明した。

【0 1 6 7】

ブロック L D P C 符号のパリティ検査行列で、部分行列 B と、部分行列 D、及び部分行列 E を適切に構成すると、ブロック L D P C 符号の符号化過程を容易に制御することができる。ここで、ブロック L D P C 符号の符号化過程を容易にするために、パリティ検査行列の部分行列 B と、部分行列 D、及び部分行列 E を構成する過程について説明する。

30

【0 1 6 8】

上記に説明したように、図 1 3 のパリティ検査行列を図 1 4 に説明したような部分行列に分割する場合に、図 1 5 のように示すことができる。

【0 1 6 9】

符号語ベクトル $\underline{c}$ は図 1 4 に示すように情報部分 s と、第 1 のパリティ部分 p_1 と第 2 のパリティ部分 p_2 に分割する場合に、符号語ベクトル $\underline{c}$ は、情報語ベクトル $\underline{s}$ と、第 1 のパリティベクトル $\underline{P}_1$ と、第 2 のパリティベクトル $\underline{P}_2$ に分割可能である。この場合、パリティ検査行列と符号語ベクトル $\underline{c}$ の積は、下記の＜式 1 2＞及び＜式 1 3＞のように示す。

【0 1 7 0】

【数 3 7】

$$A\underline{s}^T + B\underline{p}_1^T + T\underline{p}_2^T = 0 \quad (\text{式 1 2})$$

40

【数 3 8】

$$(ET^{-1}A + C)\underline{s}^T + (ET^{-1}B + D)\underline{p}_1^T = 0 \quad (\text{式 1 3})$$

【0 1 7 1】

＜式 1 2＞で、T は転置 (t r a n s p o s e) 演算を示し、＜式 1 3＞で第 1 のパリティ

50

ィベクトル $\underline{P}_1$ と関連した部分、すなわち

【0 1 7 2】

【数 3 9】

$$\underline{p}_1^T$$

【0 1 7 3】

は下記の<式 1 4>を用いて得られる。

【0 1 7 4】

【数 4 0】

10

$$\underline{p}_1^T = \phi^{-1}(ET^{-1}A + C)\underline{s}^T \quad (\phi \triangleq ET^{-1}B + D) \quad (\text{式 1 4})$$

【0 1 7 5】

<式 1 4>で、行列 ϕ のサイズの自乗に比例してブロック LDPC 符号の符号化複雑度が発生されるため、本発明では第 1 のパリティベクトル $\underline{P}_1$ を求めるために使用される行列 ϕ を単位行列 I となるように設定する。このように、行列 ϕ を単位行列 I になるように設定することによって、ブロック LDPC 符号の符号化複雑度が最小化される。ここで、図 1 5 を参照して、行列 ϕ を単位行列 I になるように設定する過程について説明する。

【0 1 7 6】

20

まず、順列行列

【0 1 7 7】

【数 4 1】

$$P^{a_1}$$

【0 1 7 8】

は、単位行列 I に固定可能である。図 1 5 に示した部分行列 T^{-1} の部分ブロックで、

【0 1 7 9】

【数 4 2】

30

$$P^{k_1-k_2}$$

【0 1 8 0】

部分は行列

【0 1 8 1】

【数 4 3】

$$P^{k_1}$$

40

【0 1 8 2】

から行列

【0 1 8 3】

【数 4 4】

$$P^{k_2}$$

【0 1 8 4】

までの積である

【0 1 8 5】

50

【数 4 5】

$$\prod_{i=k_1}^{k_2} P^{a_i} = P^{\sum_{i=k_1}^{k_2} a_i}$$

【0 1 8 6】

を示す。行列 ϕ は、下記の＜式 1 5＞～＜式 1 7＞を用いて求められる。

【0 1 8 7】

まず、図 1 5 で部分行列 E は、一つの部分ブロックを除いてはすべてゼロ行列であるため、部分行列 E と部分行列 T の逆行列である T^{-1} の乗算は部分行列 T の逆行列である T^{-1} の最後行と部分行列 E の最後ブロックの乗算形態で＜式 1 5＞のように示す。 10

【0 1 8 8】

【数 4 6】

$$ET^{-1} = P^{a_m} [P^{2 \sim (m-1)} : P^{3 \sim (m-1)} : \dots : P^{a(m-1)} : I] \quad (\text{式 1 5})$$

【0 1 8 9】

部分行列 E と部分行列 T の逆行列 T^{-1} の積に部分行列 B を乗算すると、＜式 1 6＞のよう示す。

【0 1 9 0】

20

【数 4 7】

$$ET^{-1}B = P^{a_m} P^{2 \sim (m-1)} + P^{a_m} P^{k \sim (m-1)} P^y \quad (\text{式 1 6})$$

【0 1 9 1】

(ここで、k は P^y の位置により決定される任意の自然数である。)

＜式 1 6＞に示すように、部分行列 E と部分行列 T の逆行列である T^{-1} の積に部分行列 B を乗算する場合に、部分行列 B が 2 個の部分ブロックを除いてはすべてゼロ行列を含むため、部分行列 B の 2 個のブロックのみに対して乗算を遂行することで、簡単に演算される。 30

【0 1 9 2】

もし、

【0 1 9 3】

【数 4 8】

$$D = P^x = P^{a_m} P^{2 \sim (m-1)}, P^{a_m} P^{k \sim (m-1)} P^y = I$$

【0 1 9 4】

に設定すると、 $\phi = ET^{-1}B + D = I$ になる。したがって、行列 ϕ は単位行列 I となる。そして、＜式 1 7＞は、行列 ϕ が単位行列 I になる条件を簡略に示すものである。 40

【0 1 9 5】

【数 4 9】

$$x \equiv a_m + \left(\sum_{i=2}^{m-1} a_i \right) (\text{mod } N_s), a_m + \left(\sum_{i=k}^{m-1} a_i \right) + y \equiv 0 (\text{mod } N_s) \quad (\text{式 1 7})$$

【0 1 9 6】

＜式 1 5＞～＜式 1 7＞に示すように、行列 ϕ が単位行列 I になるように設定すると、ブロック L D P C 符号の符号化過程は、その複雑度が最小化できる。

【0 1 9 7】

次に、図 1 8 を参照して、本発明によるパリティ検査行列を使用してブロック L D P C 50

符号を符号化する過程を説明する。

【0198】

図18は、本発明の実施形態によるブロックLDPC符号の符号化過程を示すフローチャートである。図18を参照すると、制御器は、ステップ1811で、ブロックLDPC符号で符号化するための情報語ベクトル $\underline{s}$ を受信し、ステップ1813及びステップ1815に進行する。ここで、ブロックLDPC符号で符号化するために受信された情報語ベクトル $\underline{s}$ の長さは k であると仮定する。ステップ1813で、制御器は、この情報語ベクトル $\underline{s}$ とパリティ検査行列(A_s)の部分行列 A を行列乗算を遂行した後に($\underline{s}$)、ステップ1817に進行する。ここで、部分行列 A に存在する1の値を有する要素の個数は、0の値を有する要素の個数に比べて非常に少ないため、情報語ベクトル $\underline{s}$ とパリティ検査行列の部分行列 A の行列乗算は、比較的少ない回数の和積(sum-product)演算のみで可能になる。また、部分行列 A で1の値を有する要素の位置は0でないブロック位置とそのブロックの順列行列の指数で示すことができ、それによって、所定のパリティ検査行列に比べて非常に簡単な演算のみでも行列乗算が遂行可能である。ステップ1815で、制御器は、パリティ検査行列の部分行列 C と情報語ベクトル $\underline{s}$ の行列乗算を遂行し(C_s)、ステップ1819に進行する。

10

【0199】

一方、ステップ1817で、制御器は、情報語ベクトル $\underline{s}$ とパリティ検査行列($E T^{-1} A_s$)の部分行列 A の行列乗算結果と行列 $E T^{-1}$ の行列乗算を遂行し、ステップ1819に進行する。上述したように、行列 $E T^{-1}$ で1の値を有する要素の個数は非常に少ないため、ブロックの順列行列の指数がわかるだけでも、行列乗算を容易に遂行することができる。ステップ1819で、制御器は、 $E T^{-1} A_s$ と C_s を加算して第1のパリティベクトル $\underline{p}_1$ を計算した後($\underline{p}_1 = E T^{-1} A_s + C_s$)、ステップ1821に進行する。ここで、加算演算は、排他的加算(exclusive OR)演算で同一のビットが加算されるときには0となり、相違したビットが加算されるときには1となる。その結果、ステップ1819までの過程は、＜式14＞で説明したような第1のパリティベクトル $\underline{p}_1$ を計算することである。

20

【0200】

ステップ1821で、制御器は、パリティ検査行列の部分行列 B と第1のパリティベクトル $\underline{p}_1$ を乗算し($B \underline{p}_1$)、その値に A_s を加算した後に($A_s + B \underline{p}_1$)、ステップ1823に進行する。ここで、＜式12＞に説明したように、情報語ベクトル $\underline{s}$ と第1のパリティベクトル $\underline{p}_1$ を知ると、第2のパリティベクトル $\underline{p}_2$ を求めるために、パリティ検査行列の部分行列 T の逆行列 T^{-1} を乗算しなければならない。したがって、ステップ1823で、制御器は、第2のパリティベクトル $\underline{p}_2$ を求めるために、ステップ1821で計算したベクトルに部分行列 T の逆行列 T^{-1} を乗算した後($\underline{p}_2 = T^{-1}(A_s + B \underline{p}_1)$)、ステップ1825に進行する。上記したように、符号化するためのブロックLDPC符号の情報語ベクトル $\underline{s}$ のみを知ると、第1のパリティベクトル $\underline{p}_1$ と、第2のパリティベクトル $\underline{p}_2$ を求めることができ、その結果、符号語ベクトルすべてが得られる。そして、制御器は、ステップ1825で、情報語ベクトル $\underline{s}$ と、第1のパリティベクトル $\underline{p}_1$ と、第2のパリティベクトル $\underline{p}_2$ で生成された符号語ベクトル $\underline{c}$ を生成して伝送し、上記の手順を終了する。

30

40

【0201】

図19は、本発明の実施形態での機能を遂行するためのブロックLDPC符号の符号化装置の内部構造を示すブロック構成図である。

【0202】

図19を参照すると、ブロックLDPC符号の符号化装置は、行列 A 乗算器1911と、行列 C 乗算器1913と、行列 $E T^{-1}$ 乗算器1915と、第1の加算器1917と、行列 B 乗算器1919と、第2の加算器1921と、行列 $E T^{-1}$ 乗算器1923と、スイッチ1925、1927、1929とを含む。

【0203】

まず、入力信号、すなわちブロックLDPC符号で符号化しようとする長さ k の情報語

50

ベクトル $\underline{s}$ が入力され、入力された長さ k の情報語ベクトル $\underline{s}$ はスイッチ1925、行列A乗算器1911、及び行列C乗算器1913に入力される。この行列A乗算器1911は、情報語ベクトル $\underline{s}$ と全体パリティ検査行列の部分行列Aを乗算した後に、行列 $E T^{-1}$ 乗算器1915と加算器1921に出力する。また、行列C乗算器1913は、情報語ベクトル $\underline{s}$ と全体パリティ検査行列の部分行列Cを乗算した後に、第1の加算器1917に出力する。行列 $E T^{-1}$ 乗算器1915は、行列A乗算器1911から出力した信号に全体パリティ検査行列の部分行列 $E T^{-1}$ を乗算した後に、第1の加算器1917に出力する。

【0204】

第1の加算器1917は、行列 $E T^{-1}$ 乗算器1915から出力された信号と行列C乗算器1913から出力された信号を入力して加算した後に、行列B乗算器1919及びスイッチ1927に出力する。ここで、第1の加算器1917は、ビット別に排他的論理和(XOR)演算を遂行する。例えば、長さ3であるベクトル $x = (x_1, x_2, x_3)$ と長さ3であるベクトル $y = (y_1, y_2, y_3)$ が加算器1917に入力される場合に、加算器1917は長さ3であるベクトル $x = (x_1, x_2, x_3)$ と長さ3であるベクトル $y = (y_1, y_2, y_3)$ を排他的論理和演算して長さ3であるベクトル

【0205】

【数50】

$z = (x_1 \oplus y_1, x_2 \oplus y_2, x_3 \oplus y_3)$ を出力する。ここで、 $\oplus$ 演算は

【0206】

同一のビットが演算されると0になり、相異なるビットが演算されると1になる排他的論理和演算を示す。すなわち、第1の加算器1917から出力される信号が第1のパリティベクトル p_1 になる。

【0207】

また、行列B乗算器1919は、第1の加算器1917から出力された信号、すなわち第1のパリティベクトル p_1 を入力して全体パリティ検査行列の部分行列Bを乗算した後に、加算器1921に出力する。加算器1921は、行列B乗算器1919から出力された信号と行列A乗算器1911から出力された信号を加算した後に、行列 T^{-1} 乗算器1923に出力する。ここで、加算器1921は、加算器1917で説明したように、行列B乗算器1919から出力された信号と行列A乗算器1911から出力された信号を排他的論理和演算した後に、行列 T^{-1} 乗算器1923に出力する。

【0208】

行列 T^{-1} 乗算器1923は、加算器1921から出力された信号と行列 T^{-1} を乗算した後に、スイッチ1929に出力する。ここで、行列 T^{-1} 乗算器1923の出力が、結局第2のパリティベクトル p_2 になる。一方、スイッチ1925、1927、1929は、各々自分の伝送する時点のみでスイッチオン(switch on)されて該当信号を伝送する。すなわち、情報語ベクトル $\underline{s}$ が伝送される時点で、スイッチ1925がスイッチオンされ、第1のパリティベクトル p_1 が伝送される時点でスイッチ1927がスイッチオンされ、第2のパリティベクトル p_2 が伝送される時点でスイッチ1929がスイッチオンされる。

【0209】

上記したように、全体パリティ検査行列の部分行列を適切に選択することによって、行列乗算 $E T^{-1}$ が比較的簡単になり、それによって、 $E T^{-1} A \underline{s}^T$ の計算が容易になる。なお、行列 ϕ が単位行列Iになって p_1^T を計算するための ϕ^{-1} の演算過程が省略される。

【0210】

上記では、効率的な符号化を考慮したブロックLDPC符号の生成方法について説明し

た。上記のブロックLDPC符号は、ブロックLDPC符号の構造的な特性によって、パリティ検査行列に関連した情報を貯蔵するためのメモリ効率が高いだけでなく、パリティ検査行列で部分行列を適切に選択することによって効率的な符号化が可能になる。しかしながら、ブロック単位でパリティ検査行列を生成することによって、不規則性(randomness)は減少し、それによるブロックLDPC符号の性能の劣化をもたらす。すなわち、上記のように、不均一ブロックLDPC符号が均一ブロックLDPC符号に比べて性能が良いため、ブロックLDPC符号を設計することにおいて、全体パリティ検査行列で部分行列を選択することは非常に重要な要素として作用するようになる。

【0211】

ここで、図16を参照してブロックLDPC符号のサイクル特性を考慮して効率的な符号化が可能であり、かつ優れた性能を有するブロックLDPC符号の具体的な生成方法について説明する。

10

【0212】

図16は、本発明の実施形態によるブロックLDPC符号のパリティ検査行列を示す図である。図16を参照すると、ブロックLDPC符号のパリティ検査行列は構造の単純性を考慮して、上述したように

【0213】

【数51】

$$P^{a_i} = I(i=1, \dots, m-1), P^{a_m} = P^1, \quad P^x = P^1, \quad P^y = P^{-1}$$

20

【0214】

に設定する。この場合、行列 ϕ は単位行列 I となって効率的な符号化を可能にする。ここで、パリティ検査行列の部分行列のブロック長さ $N_s = 31$ で、したがって $P^{-1} = P^{30}$ である。パリティ検査行列の全体列に対するブロックの個数は32個であるため、全体ブロック長さが $32 \times 31 = 992$ で、符号化率 $1/2$ のブロックLDPC符号のパリティ検査行列が生成される。

【0215】

その結果、図16に示すようにブロックLDPC符号は、パリティ検査行列の列を基準としてウェイト値が2である15個のブロックと、ウェイト値が3である12個のブロックと、ウェイト値が1である5個のブロックで構成された不均一ブロックLDPC符号になる。したがって、図16に示すように、ブロックLDPC符号の次数分布は、下記の<式18>のように示す。

30

【0216】

【数52】

$$f_2 = 15/32, \quad f_3 = 12/32, \quad f_{11} = 5/32, \quad f_{p_7} = 7/16, \quad f_{p_8} = 9/16 \quad (\text{式18})$$

【0217】

40

<式18>で、 f_i はブロックLDPC符号の因子グラフで全体変数ノードに対する次数が i である変数ノードの比率を示し、 f_{p_i} はブロックLDPC符号の因子グラフで全体検査ノードに対する次数が i である検査ノードの比率を示す。その一例として、ブロック長さが $N_s = 32$ であるブロックLDPC符号において、ブロックLDPC符号の因子グラフ上で全体32個の変数ノードの中に15個の変数ノードに該当するパリティ検査行列の列がウェイト値32を有し、12個の変数ノードに該当するパリティ検査行列の列がウェイト値3を有し、5個の変数ノードに該当するパリティ検査行列の列がウェイト値11を有する。これら変数ノードだけでなく、検査ノードに対応するパリティ検査行列に対しても、変数ノードに対して遂行したことと同一の形態でウェイト値を考慮することができる。一方、<式18>に示すように、次数分布は、しきい値(threshold va

50

lue)を有するLDPC符号の次数分布にほとんど近接した形態を有する。また、図16に示しているブロックLDPC符号の場合は、次数が2, 3であるノード間に存在するサイクルの最小サイズが12であり、全体ノード間の場合は最小サイズが6である。

【0218】

次に、図20を参照して、本発明の実施形態によるパリティ検査行列を用いてブロックLDPC符号を復号する段階について説明する。

【0219】

図20は、本発明の実施形態によりブロックLDPC符号の復号装置の内部構造を示す図である。図20を参照すると、ブロックLDPC符号の復号装置は、変数ノード部2000と、加算器2015と、デインターリーブ(deinterleaver)2017と、
10 インタリーブ2019と、制御器2021と、メモリ2023と、加算器2025と、検査ノード部2050と、硬判定器2029とを含んで構成される。変数ノード部2000は、変数ノード復号器2011と、スイッチ2013で構成され、検査ノード部2050は検査ノード復号器2027で構成される。

【0220】

無線チャンネルを通じて受信される受信信号は、変数ノード部2000の変数ノード復号器2011に入力される。変数ノード復号器2011は、この受信された受信信号の確率値を計算し、計算された確率値をアップデートした後にスイッチ2013及び加算器2015に出力する。ここで、変数ノード復号器2011は、ブロックLDPC符号の復号装置に予め設定されているパリティ検査行列に対応して変数ノードを接続し、変数ノード
20 に接続された‘1’の個数だけの入力値と出力値を有するアップデート演算が遂行される。変数ノードの各々に接続された‘1’の個数は、パリティ検査行列を構成する列のそれぞれのウェイト値と同一である。したがって、パリティ検査行列を構成する列のそれぞれのウェイトにより、変数ノード復号器2011の内部演算が異なるように遂行される。

【0221】

第1の加算器2015は、変数ノード復号器2011から出力される信号と以前反復復号過程でインタリーブ2019の出力信号を入力し、変数ノード復号器2011からの信号から以前反復復号過程でインタリーブ2019の出力信号を減算した後に、デインターリーブ2017に出力する。ここで、復号過程が最初の復号過程である場合に、インタリーブ2019の出力信号は‘0’であるとみなすことはもちろんである。
30

【0222】

デインターリーブ2017は、第1の加算器2015から出力された信号を入力して予め設定されている設定方式に対応してデインターリーブした後に、第2の加算器2025と検査ノード復号器2027に出力する。ここで、デインターリーブ2017の内部構造は、パリティ検査行列に対応する構造を有し、その理由は、パリティ検査行列の‘1’の値を有する要素の位置によりデインターリーブ2017に対応するインタリーブ2019の入力値に対する出力値が相互に異なるためである。

【0223】

第2の加算器2025は、以前反復復号過程での検査ノード復号器2027の出力信号とデインターリーブ2017の出力信号を受信し、以前反復復号過程での検査ノード復号器2027の出力信号からデインターリーブ2017の出力信号を減算した後に、インタリーブ2019に出力する。検査ノード復号器2027は、ブロックLDPC符号の復号装置に予め設定されているパリティ検査行列に対応して検査ノードを接続し、検査ノードに接続された‘1’の個数だけの入力値と出力値を有するアップデート演算が遂行される。検査ノードの各々に接続された‘1’の個数は、パリティ検査行列を構成する行のそれぞれのウェイト値と同一である。したがって、パリティ検査行列を構成する行のそれぞれのウェイト値により検査ノード復号器2027の内部演算が相互に異なるようになる。
40

【0224】

インタリーブ2019は、制御器2021の制御によって予め設定されている設定方式で、加算器2025から出力された信号をインタリーブした後に、加算器2015及
50

び変数ノード復号器 2011 に出力する。ここで、制御器 2021 は、メモリ 2023 に貯蔵されているインタリービング方式に関連した情報を読み出してインタリーバ 2019 のインタリービング方式を制御するようになる。また、この復号化過程が最初の復号過程である場合には、デインタリーバ 2017 の出力信号は '0' であるとみなすことはもちろんである。

【0225】

上記のような過程を反復して遂行することによって、誤りなしの高信頼度の復号化を遂行し、予め設定された設定反復回数に該当する反復復号化を遂行した後に、スイッチ 2013 は変数ノード復号器 2011 と加算器 2015 との間をスイッチオフする。その後、スイッチ 2013 は、変数ノード復号器 2011 と硬判定器 2029 との間をスイッチオンして変数ノード復号器 2011 から出力された信号を硬判定器 2029 に出力する。硬判定器 2029 は、変数ノード復号器 2011 から出力された信号を入力して硬判定した後に、その硬判定結果を出力し、この硬判定器 2029 の出力値が最終的に復号された値になる。

【図面の簡単な説明】

【0226】

【図1】 一般的な(8, 2, 4)LDPC符号のパリティ検査行列を示す図である。

【図2】 図1の(8, 2, 4)LDPC符号の因子グラフを示す図である。

【図3】 一般的なブロックLDPC符号のパリティ検査行列を概略的に示す図である。

【図4】 図3の循環行列Pを示す図である。

【図5】 一般的な均一ブロックLDPC符号のパリティ検査行列を示す図である。

【図6】 一般的な不均一ブロックLDPC符号のパリティ検査行列を示す図である。

【図7】 パリティ検査行列が4個の部分行列で構成されたブロックLDPC符号のサイクル構造を概略的に示す図である。

【図8】 パリティ検査行列が6個の部分行列で構成されたブロックLDPC符号のサイクル構造を概略的に示す図である。

【図9】 ブロックLDPC符号のブロックサイクル構造を概略的に示す図である。

【図10】 パリティ検査行列の6個の部分行列が重複されたブロックLDPC符号のブロックサイクル構造を概略的に示す図である。

【図11】 パリティ検査行列の7個の部分行列が重複されたブロックLDPC符号のブロックサイクル構造を概略的に示す図である。

【図12】 完全下三角行列形態を有するパリティ検査行列を示す図である。

【図13】 完全下三角行列形態に類似した形態を有するパリティ検査行列を示す図である。

【図14】 図13のパリティ検査行列を6個の部分ブロックに分割した図である。

【図15】 図14の部分行列Bの転置行列と、部分行列Eと、部分行列Tと、部分行列Tの逆行列を示す図である。

【図16】 本発明の実施形態によるブロックLDPC符号のパリティ検査行列を示す図である。

【図17】 本発明の実施形態によるブロックLDPC符号のパリティ検査行列の生成過程を示すフローチャートである。

【図18】 本発明の実施形態によるブロックLDPC符号の符号化過程を示すフローチャートである。

【図19】 本発明の実施形態によるブロックLDPC符号の符号化装置の内部構造を示すブロック構成図である。

【図20】 本発明の実施形態によるブロックLDPC符号の復号装置の内部構造を示す図である。

【符号の説明】

【0227】

1911 行列A乗算器

10

20

30

40

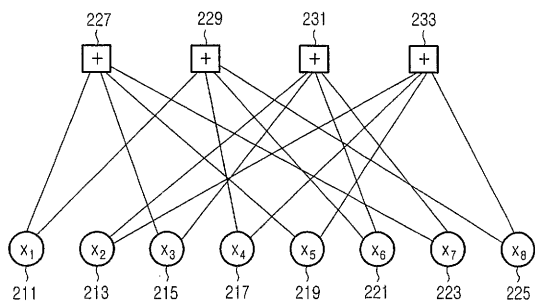
50

1 9 1 3 行列 C 乗算器
 1 9 1 5 行列 $E T^{-1}$ 乗算器
 1 9 1 7 第 1 の加算器
 1 9 1 9 行列 B 乗算器
 1 9 2 1 第 2 の加算器
 1 9 2 3 行列 $E T^{-1}$ 乗算器
 1 9 2 5, 1 9 2 7, 1 9 2 9 スイッチ

【図 1】

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

【図 2】



【図 3】

$P^{a_{11}}$	$P^{a_{12}}$	$P^{a_{13}}$	$P^{a_{14}}$	...	$P^{a_{1(q-1)}}$	$P^{a_{1q}}$
$P^{a_{21}}$	$P^{a_{22}}$	$P^{a_{23}}$	$P^{a_{24}}$	...	$P^{a_{2(q-1)}}$	$P^{a_{2q}}$
$P^{a_{31}}$	$P^{a_{32}}$	$P^{a_{33}}$	$P^{a_{34}}$	...	$P^{a_{3(q-1)}}$	$P^{a_{3q}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	$\vdots$
$P^{a_{p1}}$	$P^{a_{p2}}$	$P^{a_{p3}}$	$P^{a_{p4}}$	...	$P^{a_{p(q-1)}}$	$P^{a_{pq}}$

【図 4】

$$P = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix}$$

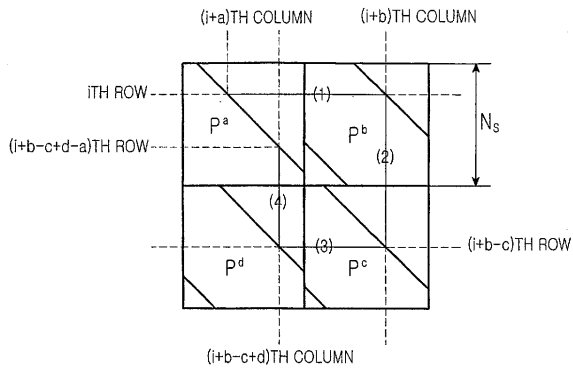
【図 5】

$$H \triangleq \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & p & p^2 & \dots & p^{p-1} \\ 1 & p^2 & p^4 & \dots & p^{2(s-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & p^{r-1} & p^{(r-1)2} & \dots & p^{(r-1)(s-1)} \end{bmatrix}$$

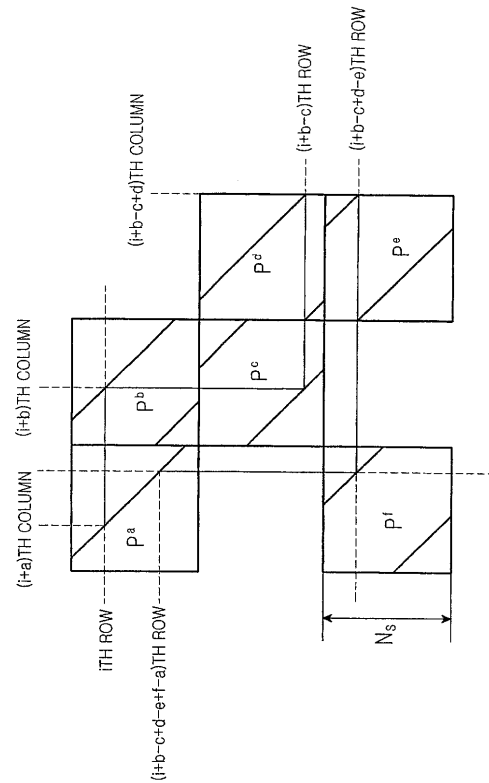
【図 6】

$$\begin{bmatrix}
 1 & 1 & 1 & \dots & 1 & 1 & \dots & 1 \\
 0 & 1 & p & \dots & p^{(r-2)} & p^{(r-1)} & \dots & p^{(k-1)} \\
 0 & 0 & 1 & \dots & p^{2(r-3)} & p^{2(r-2)} & \dots & p^{2(k-3)} \\
 \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\
 0 & 0 & \dots & 0 & 1 & p^{(r-1)} & \dots & p^{(r-1)(k-r)}
 \end{bmatrix}$$

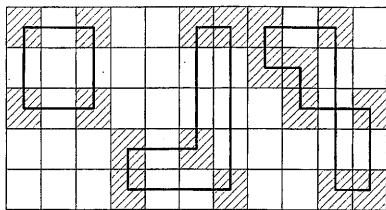
【図 7】



【図 8】

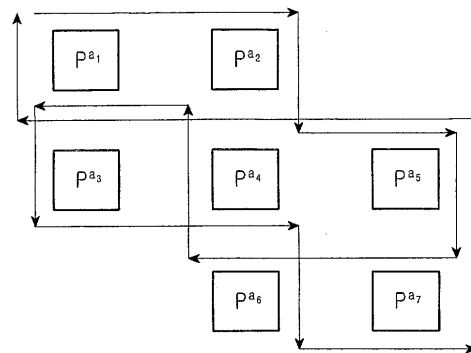


【図 9】

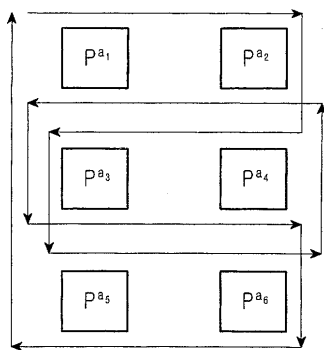


$$\equiv \begin{bmatrix}
 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1
 \end{bmatrix}$$

【図 11】



【図 10】



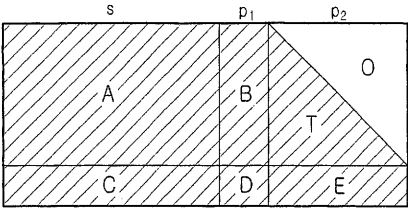
【図 1 2】

INFORMATION PART					PARITY PART				
$p^{a_{11}}$	$p^{a_{12}}$	$p^{a_{13}}$	$p^{a_{14}}$	$\dots$	$p^{a_{1(n-1)}}$	$p^{a_{1n}}$	1	0	0
$p^{a_{21}}$	$p^{a_{22}}$	$p^{a_{23}}$	$p^{a_{24}}$	$\dots$	$p^{a_{2(n-1)}}$	$p^{a_{2n}}$	...	1	0
$p^{a_{31}}$	$p^{a_{32}}$	$p^{a_{33}}$	$p^{a_{34}}$	$\dots$	$p^{a_{3(n-1)}}$	$p^{a_{3n}}$	...	...	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$	$\vdots$	...	...	...
$p^{a_{m1}}$	$p^{a_{m2}}$	$p^{a_{m3}}$	$p^{a_{m4}}$	$\dots$	$p^{a_{m(n-1)}}$	$p^{a_{mn}}$	...	...	1

【図 1 3】

INFORMATION PART					PARITY PART				
$p^{a_{11}}$	$p^{a_{12}}$	$p^{a_{13}}$	$p^{a_{14}}$	$\dots$	$p^{a_{1(n-1)}}$	$p^{a_{1n}}$	p^{a_1}	1	0
$p^{a_{21}}$	$p^{a_{22}}$	$p^{a_{23}}$	$p^{a_{24}}$	$\dots$	$p^{a_{2(n-1)}}$	$p^{a_{2n}}$	$\vdots$	p^{a_2}	1
$p^{a_{31}}$	$p^{a_{32}}$	$p^{a_{33}}$	$p^{a_{34}}$	$\dots$	$p^{a_{3(n-1)}}$	$p^{a_{3n}}$	p^y	0	p^{a_3}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$	$\vdots$	...	...	...
$p^{a_{m1}}$	$p^{a_{m2}}$	$p^{a_{m3}}$	$p^{a_{m4}}$	$\dots$	$p^{a_{m(n-1)}}$	$p^{a_{mn}}$	p^x	0	0

【図 1 4】

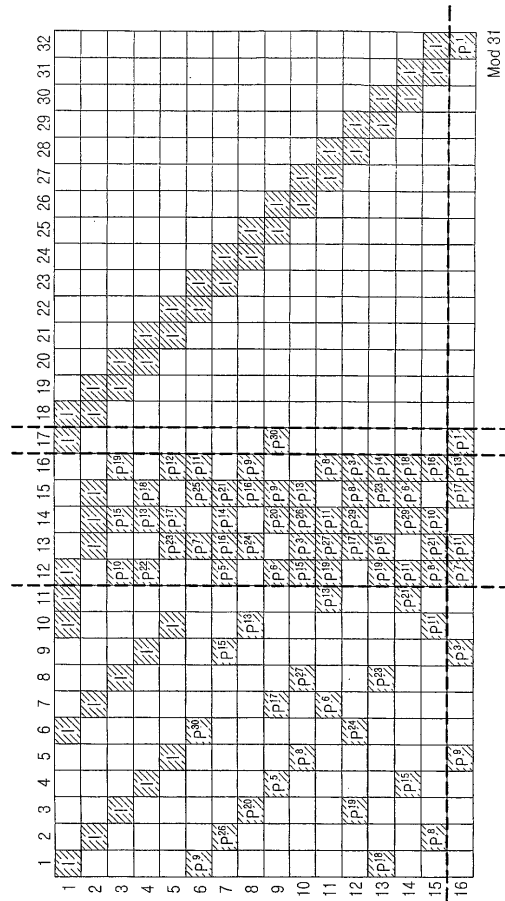


【図 1 5】

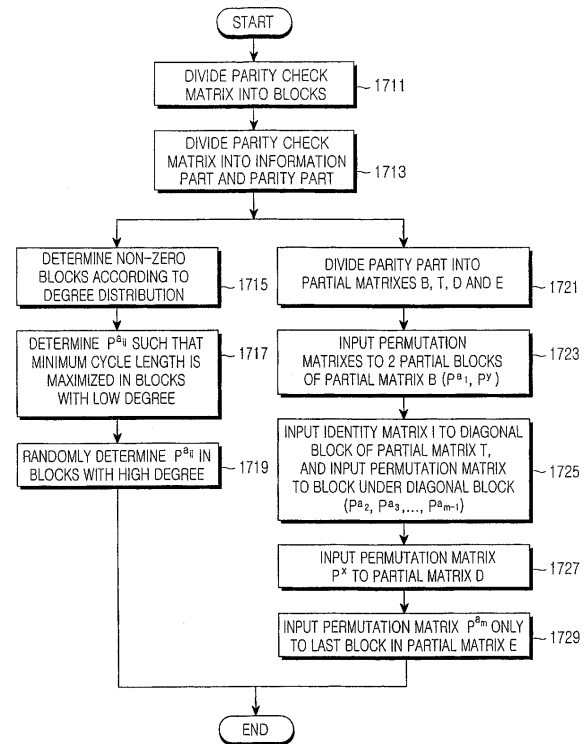
$B^T =$	p^{a_1}	0	...	p^y	...	0	0
$T =$	1	0	0	0	0	0	0
	p^{a_2}	1	0	0	0	0	0
	0	p^{a_3}	1	0	0	0	0
	...						
	0	0	0	...	$p^{a_{m-2}}$	1	0
	0	0	0	...	0	$p^{a_{m-1}}$	1

$E =$	0	0	0	...	0	0	p^{a_m}
$T^{-1} =$	1	0	0	0	0	0	0
	p^{a_2}	1	0	0	0	0	0
	p^{a_3}	p^{a_3}	1	0	0	0	0
	...						
	$p^{2-(m-2)}$	$p^{3-(m-2)}$	$p^{4-(m-2)}$	...	$p^{a_{m-2}}$	1	0
	$p^{2-(m-1)}$	$p^{3-(m-1)}$	$p^{4-(m-1)}$	...	$p^{a_{m-1}}$	$p^{a_{m-1}}$	1

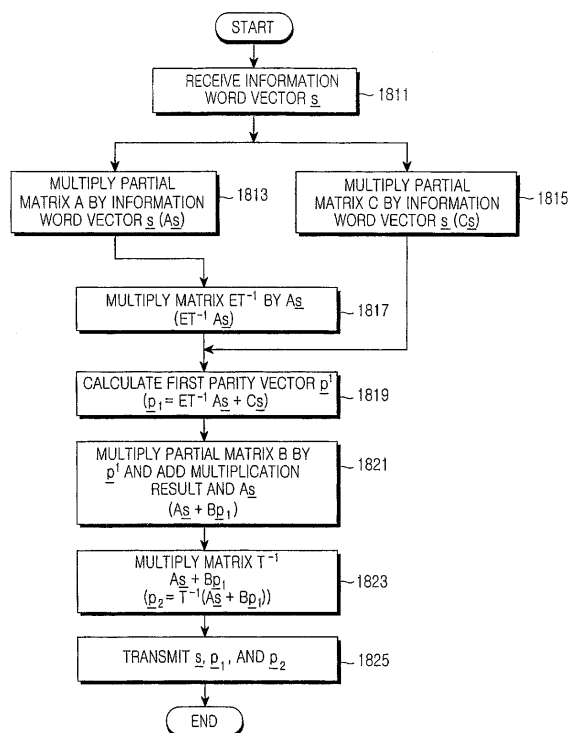
【図 16】



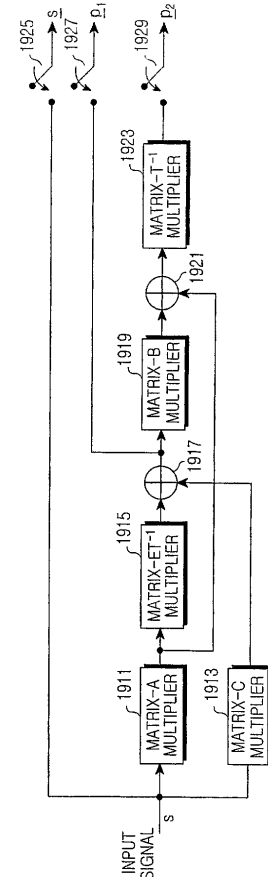
【図 17】



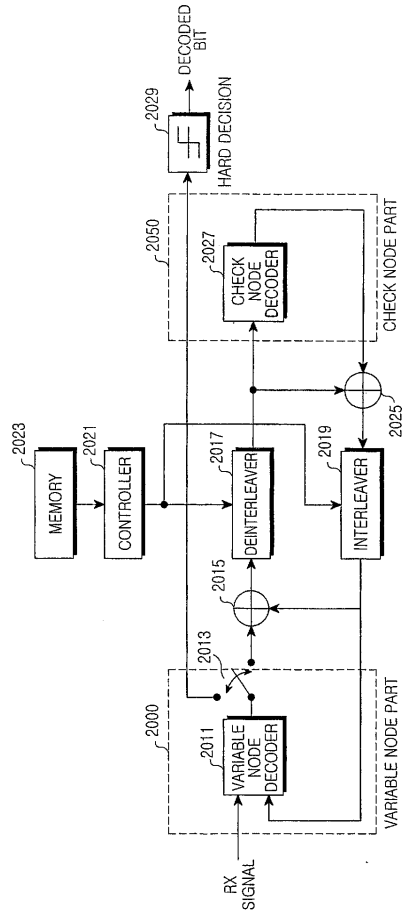
【図 18】



【図 19】



【図 20】



フロントページの続き

- (72)発明者 ホンーシル・ジェオン
大韓民国・インチョン・４０４－７８５・ソグ・シンヒョンードン・（番地なし）・シンヒョン・ジュゴン・アパート・＃２７－１０２
- (72)発明者 ジェーヨル・キム
大韓民国・キョンギード・４３５－０４２・グンポーシ・サンボン・２ードン・（番地なし）・サンボン・９－ダンジ・ペクドゥー・アパート・＃９６０－１４０１
- (72)発明者 スン－ウン・パク
大韓民国・ギョンギード・４４２－７５４・スオン－シ・バルダーグ・ウォンチョンードン・（番地なし）・ウォンチョンサムスン・アパート・＃６－６０６
- (72)発明者 キョナーチョル・ヤン
大韓民国・ギョンスンブクード・７９０－３３０・ポハン－シ・ナム－グ・ヒョジャードン・サン・３１・エルジードン・＃４１６
- (72)発明者 セーホ・ミュング
大韓民国・ギョンスンブクード・７９０－３３０・ポハン－シ・ナム－グ・ヒョジャードン・サン・３１・エルジードン・＃４１６

審査官 岡 裕之

- (56)参考文献 国際公開第０３／０３２４９９（WO, A1）
特開２００３－１１５７６８（JP, A）
特許第４１６０６１７（JP, B2）
Engling Yeo, et al., Architectures and Implementations of Low-Density Parity Check Decoding Algorithms, Proceeding of the 2002 45th Midwest Symposium on Circuits and Systems, 2002. MWSCAS-2002, 2002年 8月 4日, Vol.3, III-437 - III-440
Ivana Djurdhevic, et al., Graph-Theoretic Construction of Low-Density Parity-Check Codes, IEEE Communications Letters, 2003年 4月, Vol.7, No.4, pp.171-173
Tao Tian et al., Construction of Irregular LDPC Codes with Low Error Floors, Proceedings of the IEEE International Conference on Communications, 2003. ICC'03, 2003年 5月11日, Vol.5, pp.3125-3129
Thomas J. Richardson, et al., Efficient Encoding of Low-Density Parity-Check Codes, IEEE Transactions on Information Theory, 2001年 2月, Vol.47, No.2, pp.638-656
Kyeongcheol Yang et al., On the minimum distance of array codes as LDPC codes, Information Theory, IEEE Transactions on, 2003年12月, Vol.49, No.12, pp.3268-3271
Rich Echard et al., The π -rotation low-density parity check codes, Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE, 2001年, Vol.2, pp.980-984

- (58)調査した分野(Int.Cl., DB名)
H03M 13/19
H03M 13/27
IEEE Explore
Cinii