

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5976960号
(P5976960)

(45) 発行日 平成28年8月24日(2016. 8. 24)

(24) 登録日 平成28年7月29日(2016. 7. 29)

(51) Int.Cl.

F I

H03M 13/19 (2006.01)

H03M 13/19

請求項の数 13 (全 27 頁)

(21) 出願番号	特願2015-557232 (P2015-557232)	(73) 特許権者	595020643
(86) (22) 出願日	平成26年2月13日 (2014. 2. 13)		クァアルコム・インコーポレイテッド
(65) 公表番号	特表2016-507200 (P2016-507200A)		QUALCOMM INCORPORATED
(43) 公表日	平成28年3月7日 (2016. 3. 7)		ED
(86) 国際出願番号	PCT/US2014/016279		アメリカ合衆国、カリフォルニア州 92
(87) 国際公開番号	W02014/127140		121-1714、サン・ディエゴ、モア
(87) 国際公開日	平成26年8月21日 (2014. 8. 21)		ハウス・ドライブ 5775
審査請求日	平成28年5月25日 (2016. 5. 25)	(74) 代理人	100108855
(31) 優先権主張番号	61/764, 476		弁理士 蔵田 昌俊
(32) 優先日	平成25年2月13日 (2013. 2. 13)	(74) 代理人	100109830
(33) 優先権主張国	米国 (US)		弁理士 福原 淑弘
(31) 優先権主張番号	14/179, 871	(74) 代理人	100158805
(32) 優先日	平成26年2月13日 (2014. 2. 13)		弁理士 井関 守三
(33) 優先権主張国	米国 (US)	(74) 代理人	100194814
			弁理士 奥村 元宏

最終頁に続く

(54) 【発明の名称】 高並列性、低エラーフロア、および簡単な符号化原理を有するリフトされたLDPC符号のための設計

(57) 【特許請求の範囲】

【請求項 1】

1組の情報ビットを受信することと、

マッチトリフトLDPC符号に基づいてコードワードを生成するために、低密度パリティ検査(LDPC)符号化演算を前記1組の情報ビットに実行することと、ここにおいて、マッチトリフティングが可換リフティング群に基づく、ここにおいて、前記LDPC符号が、いくつかのパリティビットと、前記パリティビットの値を決定するための副行列とを含む、ここにおいて、

前記リフティング群の位数(Z)が前記リフティングのサイズに対応し、

前記副行列の行列式が、 $g_a + (g_0 + g_L)P$ の形式の多項式であり、ここで、 g_0 は前記群の単位元であり、

【数 1】

$$g_0 = g_L^{2^k}$$

であり、

正整数kおよびLに対して、 $Z = 2^k \times L$ であり、非負の $1 < k$ に対して、 $a = 2^{1/L} \times L$ であり、

Pは、前記リフティング群に関連付けられた2値群環の任意の非ゼロ元である、を備えるデータ符号化の方法。

【請求項 2】

前記リフティング群が巡回群であり、ここにおいて、

g_i は x^i と同一に扱うことができ、

前記副行列の前記行列式が、 $x^a + (1 + x^L) P(x)$ の形式を取り、ここで、 $P(x)$ は少なくとも 2 つの項を有し、 $2^k L = 0$ モジュロ Z である、請求項 1 に記載の方法。

【請求項 3】

前記副行列が、その行および列の置換を除いて、上三角である、請求項 1 に記載の方法。

【請求項 4】

前記副行列の第 1 の副対角線より下の元が 0 に等しい、請求項 3 に記載の方法。

【請求項 5】

プログラム命令を備える非一時的コンピュータ可読記憶媒体であって、前記プログラム命令が、通信デバイス内に設けられたプロセッサによって実行されたとき、前記プログラム命令により、前記デバイスが、

1 組の情報ビットを受信し、

マッパトリフト LDPC 符号に基づいてコードワードを生成するために、LDPC 符号化演算を前記 1 組の情報ビットに実行する、ここにおいて、マッパトリフティングが可換リフティング群に基づく、ここにおいて、前記 LDPC 符号が、いくつかのパリティビットと、前記パリティビットの値を決定するための副行列とを含む、ここにおいて、

前記リフティング群の位数 (Z) が前記リフティングのサイズに対応し、

前記副行列の行列式が、 $g_a + (g_0 + g_L) P$ の形式の多項式であり、ここで、

g_0 は前記群の単位元であり、

【数 2】

$$g_0 = g_L^{2^k}$$

であり、

正整数 k および L に対して、 $Z = 2^k \times L$ であり、

非負の $1 < k$ に対して、 $a = 2^1 \times L$ であり、

P は、前記リフティング群に関連付けられた 2 値群環の任意の非ゼロ元である、非一時的コンピュータ可読記憶媒体。

【請求項 6】

前記リフティング群が巡回群であり、ここにおいて、

g_i は x^i と同一に扱うことができ、

前記副行列の前記行列式が、 $x^a + (1 + x^L) P(x)$ の形式を取り、ここで、 $P(x)$ は少なくとも 2 つの項を有し、 $2^k L = 0$ モジュロ Z である、請求項 5 に記載の非一時的コンピュータ可読記憶媒体。

【請求項 7】

前記副行列が、その行および列の置換を除いて、上三角である、請求項 5 に記載の非一時的コンピュータ可読記憶媒体。

【請求項 8】

前記副行列の第 1 の副対角線より下の元が 0 に等しい、請求項 7 に記載の非一時的コンピュータ可読記憶媒体。

【請求項 9】

1 組の情報ビットを記憶するメモリと、

符号器と、ここにおいて、前記符号器が、

マッパトリフト LDPC 符号に基づいてコードワードを生成するために、LDPC 符号化演算を前記 1 組の情報ビットに実行する、ここにおいて、マッパトリフティングが可換リフティング群に基づく、ここにおいて、前記 LDPC 符号が、いくつかのパリティビ

10

20

30

40

50

ットと、前記パリティビットの値を決定するための副行列とを含む、ここにおいて、
 前記リフティング群の位数（ Z ）が前記リフティングのサイズに対応し、
 前記副行列の行列式が、 $g_a + (g_0 + g_L)P$ の形式の多項式であり、ここで、
 g_0 は前記群の単位元であり、
 【数 3】

$$g_0 = g_L^{2^k}$$

であり、

正整数 k および L に対して、 $Z = 2^k \times L$ であり、
非負の $1 < k$ に対して、 $a = 2^{-1} \times L$ であり、

10

P は、前記リフティング群に関連付けられた 2 値群環の任意の非ゼロ元である、
 を備える通信デバイス。

【請求項 10】

前記リフティング群が巡回群であり、ここにおいて、

g_i は x^i と同一に扱うことができ、

前記副行列の前記行列式が、 $x^a + (1 + x^L)P(x)$ の形式を取り、ここで、 $P(x)$ は少なくとも 2 つの項を有し、 $2^k L = 0$ モジユロ Z である、請求項 9 に記載のデバイス。

【請求項 11】

20

前記副行列が、その行および列の置換を除いて、上三角である、請求項 9 に記載のデバイス。

【請求項 12】

前記副行列の第 1 の副対角線より下の元が 0 に等しい、請求項 11 に記載のデバイス。

【請求項 13】

1 組の情報ビットを受信するための手段と、

マッチトリフト LDP C 符号に基づいてコードワードを生成するために、LDP C 符号化演算を前記 1 組の情報ビットに実行するための手段と、ここにおいて、マッチトリフティングが可換リフティング群に基づく、ここにおいて、前記 LDP C 符号が、いくつかのパリティビットと、前記パリティビットの値を決定するための副行列とを含む、ここにおいて、

30

前記リフティング群の位数（ Z ）が前記リフティングのサイズに対応し、

前記副行列の行列式が、 $g_a + (g_0 + g_L)P$ の形式の多項式であり、ここで、
 g_0 は前記群の単位元であり、

【数 4】

$$g_0 = g_L^{2^k}$$

であり、

40

正整数 k および L に対して、 $Z = 2^k \times L$ であり、
非負の $1 < k$ に対して、 $a = 2^{-1} \times L$ であり、

P は、前記リフティング群に関連付けられた 2 値群環の任意の非ゼロ元である、
 を備える符号器。

【発明の詳細な説明】

【技術分野】

【0001】

[0001]本実施形態は、一般に、通信およびデータ記憶システムに関し、詳細には、LDP C 符号を使用する通信およびデータ記憶システムに関する。

【背景技術】

50

【 0 0 0 2 】

[0002]多くの通信システムは誤り訂正符号を使用する。詳細には、誤り訂正符号は、データストリームに冗長性を導入することによってこれらのシステムへの情報転送の本質的な不信頼性を補償する。低密度パリティ検査 (LDPC) 符号は、反復コーディング方式を使用する特定のタイプの誤り訂正符号である。LDPC符号は、二部グラフ (しばしば「タナーグラフ」と呼ばれる) によって表すことができ、1組の変数ノードが、コードワードのビットに対応し、1組の検査ノードが、符号を定義する1組のパリティ検査制約に対応する。変数ノードおよび検査ノードがグラフのエッジで接続される場合、それらは「隣接しているもの」と見なされる。検査ノードごとに、すべての隣接する変数ノードに関連するビットが合計されて、0モジュロ2になる (すなわち、それらは偶数個の1を含む) ときおよびそのときに限って、変数ノードシーケンスと1対1に関連付けられたビットシーケンスは、有効なコードワードである。

10

【 0 0 0 3 】

[0003]たとえば、図1Aは、例示的なLDPC符号を表す二部グラフ100を示す。二部グラフ100は、4つの検査ノード120 (正方形で表された) に接続された1組の5つの変数ノード110 (円で表された) を含む。グラフ100のエッジは、変数ノード110を検査ノード120に接続する。図1Bは、二部グラフ100の行列表現150を示す。行列表現150は、パリティ検査行列Hおよびコードワードベクトルxを含み、ここで、 $x_1 \sim x_5$ はコードワードxのビットを表す。より詳細には、コードワードベクトルxは、 $Hx = 0$ のときおよびそのときに限って、有効なコードワードを表す。図2は、たとえば、本願の譲受人が所有する米国特許第7,552,097号に説明されているように、図1Aのグラフの3つのコピーを作る効果をグラフ式に示す。3つのコピーは、コピーの間の同様なエッジを置換することによって相互接続され得る。置換が巡回置換に制限される場合、結果として生じるグラフは、リフティング $Z = 3$ による擬似巡回LDPCに対応する。3つのコピーが作られたオリジナルのグラフは、本明細書では基礎グラフ (base graph) と呼ばれる。

20

【 0 0 0 4 】

[0004]受信されたLDPCコードワードは、オリジナルのコードワードの再構築されたバージョンを生成するために復号され得る。誤りがない場合、または訂正可能な誤りの場合には、復号化は、符号化されたオリジナルのデータユニットを回復するために使用され得る。LDPC復号器は、一般に、エッジに沿って、二部グラフ100内でメッセージを交換し、入来メッセージに基づいてノードで計算を実行することによりこれらのメッセージを更新することによって動作する。たとえば、グラフ100の各変数ノード110は、最初に、通信チャネルからの観察によって決定されるような関連するビット値の推定を示す「ソフトビット」 (たとえば、コードワードの受信されたビットを表す) が供給され得る。これらのソフトビットを使用して、LDPC復号器は、メモリからメッセージまたはその一部分を反復して読み出し、更新されたメッセージまたはその一部分をメモリに戻して書き込むことによってメッセージを更新することができる。更新演算は、一般に、対応するLDPC符号のパリティ検査制約に基づく。リフトされたLDPC符号のための実施態様では、同様のエッジ上のメッセージは、多くの場合、並列に処理される。

30

40

【 0 0 0 5 】

[0005]多くの実用的なLDPC符号設計は、符号化および復号化演算で高並列性をサポートするために、大きいリフティングファクタ (lifting factor) および比較的小さい基礎グラフをもつ擬似巡回構成を使用する。多くのLDPC符号は、さらに、その構成に次数2の変数ノードのチェーンを使用する。時々蓄積構造 (accumulate structure) と呼ばれるそのような次数2の変数ノードチェーンは、簡単な符号化および良好な性能をもたらす。巡回リフティングに基づくLDPC符号設計は、多項式モジュロの環の上の符号として解釈することができ、 2 値多項式モジュロ $X^Z - 1$ とすることができ、ここで、 Z はリフティングサイズ (たとえば、擬似巡回符号における巡回のサイズ) である。それにより、そのような符号の符号化は、多くの場合、この環における代数演算として解釈され得る

50

。

【発明の概要】

【0006】

[0006]本概要は、発明を実施するための形態において以下でさらに説明する概念の選択を簡略化された形で紹介するために与えるものである。本概要は、請求する主題の主要な特徴または本質的特徴を識別するものではなく、請求する主題の範囲を限定するものでもない。

【0007】

[0007]多くのLDPC符号設計は、並列性をサポートするために巡回リフティングを使用する。加えて、多くのLDPC設計は、簡単な符号化をサポートするために次数2の変数ノードの蓄積構造を使用した。その理由は、それらが良好な性能をもつ符号をもたらすからである。しかしながら、非常に低い誤り率を目標とするコーディング方式では、大きい蓄積構造は高いエラーフロアをもたらすという点で問題となることがある。したがって、いくつかのコーディング方式の性能目標は、蓄積構造を使用する巡回リフト設計を用いて達成することが困難である。エラーフロアは、累積チェーン中の変数ノードのうちのいくつかの次数を増加させるか、または、一般に、グラフ中の変数ノードの次数を増加させることによって低下され、それは、改善された性能を与えることができる。しかしながら、そのようにすることは、一般に、符号化プロセスを複雑にする。それにより、より高い次数の変数ノードをサポートしながら、符号化の簡単さを保持する、蓄積構造を変更する方法を見いだすことが望ましい。

【0008】

[0008]巡回リフティングに基づくLDPC符号設計は、多項式モジュロ $x^Z - 1$ の環上の符号として解釈することができ、ここで、 Z はリフティングサイズ（たとえば、擬似巡回符号における巡回のサイズ）である。そのような符号の符号化は、多くの場合、この環の行列方程式を解くことに還元する。蓄積構造の簡単な符号化は、符号化式に含まれる符号化行列の簡単な可逆性となって現れる。追加として、略上三角（または下三角）である符号化行列により、パリティビットの多くが、後退代入（back substitution）によって決定され得る。後退代入プロセスを開始することは、いくつかの初期ビットについて解くことを含む。この初期のステップの困難さは、符号化行列の可逆性に密接に関係付けられる。通常使用される蓄積構造は、主として符号化行列の対角線および副対角線にのみ非ゼロエントリを有する、略二重対角線である行列をもたらす。累積チェーンを使用するいくつかの符号化構造では、より高い次数、典型的には次数3である行列の1つの列が存在する。しかしながら、非常に低い誤り率を目標とするコーディング方式では、大きい蓄積構造は高いエラーフロアをもたらすという点で問題となることがある。エラーフロアは、累積チェーン中の変数ノードのうちのいくつかの次数を増加させるか、または、一般に、グラフ中の変数ノードの次数を増加させることによって低下され、それは、改善された性能を与えることができる。しかしながら、そのようにすることは、一般に、符号化プロセスを複雑にする。

【0009】

[0009]それにより、より高い次数の変数ノードをサポートしながら、符号化の簡単さを保持するように、蓄積構造を変更することが望ましい。本明細書の実施形態は、簡単な符号化を与える変更された符号化構造を提供する。いくつかの実施形態は、変更された累積チェーンと見なされ得る構造を有し、チェーン中のいくつかの変数ノードは、追加のエッジを有し、したがって、2よりも高い次数のものである。それにより、これらの符号のための符号化行列は二重対角線構造を有するが、対角線より上に追加の非ゼロ元が存在し得る。その結果として、次数が少なくとも3である追加の列が存在し得る巡回リフティング値を注意深く選択することによって、追加のエッジのうちのいくつかは、符号の基礎をなすグラフィック表示の大きいガース（girth）を維持し、それによって、低いエラーフロア性能をサポートしながら、符号化演算を簡単にするように選ばれる。説明された二重対角線および略上三角構造は、行列の列および行の適切な配列によってのみ明確に観察され

、構造は、それが目に見えるようには明白でない他の配列の場合でさえ存在することが当業者によって理解されるだろう。

【 0 0 1 0 】

【0010】本実施形態は例として示されており、添付図面の図によって限定されるものではない。

【図面の簡単な説明】

【 0 0 1 1 】

【図 1 A】 【0011】例示の L D P C 符号のグラフ表示の図。

【図 1 B】例示の L D P C 符号の行列表示の図。

【図 2】 【0012】図 1 A のグラフの 3 つのコピーを作る効果をグラフ式に示す図。

【図 3】 【0013】いくつかの実施形態による通信システムを示す図。

【図 4】 【0014】いくつかの実施形態による通信デバイスのブロック図。

【図 5】 【0015】例示の擬似巡回 L D P C 符号のリフトされたパリティ検査行列表示のパリティビット部分を示す図。

【図 6】 【0016】別の例示の擬似巡回 L D P C 符号のリフトされたパリティ検査行列表示のパリティビット部分を示す図。

【図 7】 【0017】図 1 0 に与えられたレート $r = 27 / 30$ をもつ例示的な L D P C 符号のリフトされたパリティ検査行列表現のパリティビット部分を示す図。

【図 8】 【0018】図 1 1 に与えられたレート $r = 13 / 15$ をもつ例示的な L D P C 符号のリフトされたパリティ検査行列表現のパリティビット部分を示す図。

【図 9】 【0019】図 1 2 に与えられたレート $r = 21 / 28$ をもつ例示的な L D P C 符号のリフトされたパリティ検査行列表現のパリティビット部分を示す図。

【図 1 0】 【0020】レート $r = 27 / 30$ をもつ L D P C 符号に関連する例示のパリティ検査行列を示す図。

【図 1 1】 【0021】レート $r = 13 / 15$ をもつ L D P C 符号に関連する例示のパリティ検査行列を示す図。

【図 1 2】 【0022】レート $r = 21 / 28$ をもつ L D P C 符号に関連する例示のパリティ検査行列を示す図。

【図 1 3】 【0023】いくつかの実施形態による通信デバイスのブロック図。

【発明を実施するための形態】

【 0 0 1 2 】

【0024】以下の説明では、本開示の十分な理解を提供するために、特定の構成要素、回路、およびプロセスの例などの多数の特定の細部が記載されている。本明細書で使用する「結合された」という用語は、直接接続されていること、または 1 つまたは複数の介在する構成要素もしくは回路を通して接続されていることを意味する。さらに、以下の記述においておよび説明のために、特定の命名法が、本実施形態の十分な理解を提供するために記載されている。しかしながら、これらの特定の細部は本実施形態を實踐するのに必要とされないことがあることが当業者には明白であろう。他の場合には、よく知られている回路およびデバイスは、本開示を不明瞭にしないためにブロック図の形式で示される。本明細書で説明する様々なバスにより供給される信号のいずれも他の信号と時分割多重され、1 つまたは複数の共通バスにより供給され得る。追加として、回路要素間またはソフトウェアブロック間の相互接続は、バスとして、または単一の信号ラインとして示されることがある。バスの各々は、代替として、単一の信号ラインとすることができ、単一の信号ラインの各々は、代替として、バスとすることができ、単一のラインまたはバスは、構成要素間の通信のための無数の物理的または論理的機構のうちの任意の 1 つまたは複数を表すことができる。本実施形態は、本明細書に記載された具体的な例に限定されると見なされるものではなく、むしろ、添付された特許請求の範囲によって定義されたすべての実施形態を、それらの範囲内に含むものである。

【 0 0 1 3 】

【0025】図 3 は、いくつかの実施形態による通信システム 3 0 0 を示す。送信機 3 1 0 は

チャンネル 3 2 0 に信号を送信し、受信機 3 3 0 はチャンネル 3 2 0 から信号を受信する。送信機 3 1 0 および受信機 3 3 0 は、たとえば、コンピュータ、スイッチ、ルータ、ハブ、ゲートウェイ、および/または同様のデバイスであり得る。ある実施形態では、チャンネル 3 2 0 はワイヤレスである。他の実施形態では、チャンネル 3 2 0 は、有線リンク（たとえば、同軸ケーブルまたは他の物理的接続）である。

【 0 0 1 4 】

[0026] 通信システム 3 0 0 の様々な構成要素の欠陥は、信号障害の原因になり、それにより、信号劣化を引き起こすことがある。たとえば、チャンネル 3 2 0 の欠陥はチャンネル歪みを導入することがあり、チャンネル歪みは、直線歪み、マルチパス効果、および/または付加白色ガウス雑音 (AWGN) を含むことがある。潜在的な信号劣化と戦うために、送信機 3 1 0 および受信機 3 3 0 は、LDPC 符号器および復号器を含むことができる。詳細には、送信機 3 1 0 は送出データに LDPC 符号化を実行して、オリジナルのデータの回復するために受信機 3 3 0 で後に復号され得る（たとえば、LDPC 復号化演算によって）コードワードを生成することができる。いくつかの実施形態では、送信機 3 1 0 は、1 つまたは複数の「パンクチャされた」ビットをもつ LDPC 符号化コードワードを、たとえば、1 つまたは複数のパンクチャされた変数ノードをもつ LDPC 符号に基づいて、送信することができる。

【 0 0 1 5 】

[0027] リフティングは、より小さい基礎符号の多数のコピーから比較的大きい LDPC 符号を発生させるための技法である。たとえば、リフティングは、更新された基礎グラフのいくつか (Z 個) の並列コピーを生成し、次いで、基礎グラフの各コピーのエッジクラスタの置換により並列コピーを相互接続することによって、リフトされた LDPC 符号を発生させることができる。エッジのクラスタは、基礎グラフの Z 並列コピーにおける基礎グラフ中の単一エッジの Z コピーの組を参照する。リフティングにより、LDPC 符号は、並列符号化および/または復号化実施を使用するとともに、さらに、大きい LDPC 符号に一般に関連する記述の複雑さを低減して、実施され得るようになる。リフトされた LDPC 符号のより詳細な説明は、たとえば、2008 年 3 月 17 日に出版された、Tom Richardson および Ruediger Urbanke による「Modern Coding Theory」という書名の書籍で見いだすことができ、その全体が参照により明細書に組み込まれる。

【 0 0 1 6 】

[0028] リフティングサイズ Z をもつコードワードを処理する場合、LDPC 復号器は、リフトされたグラフのすべての Z 個のエッジに対してパリティ検査演算または変数ノード演算を同時に実行するために、Z 個の処理要素を利用することができる。いくつかの実施形態では、Z の約数が使用され得る。詳細には、各パリティ検査演算は、対応するソフトビット値をメモリから読み出すことと、ソフトビット値を検査ノードに関連した他のソフトビット値と組み合わせることと、検査ノード演算に由来するソフトビットをメモリに戻して書き込むこととを含む。

【 0 0 1 7 】

[0029] 図 4 は、いくつかの実施形態による通信デバイス 4 0 0 のブロック図である。通信デバイス 4 0 0 は、符号器 4 1 0、復号器 4 2 0、および通信チャンネルを介して LDPC 符号化コードワードを送信および/または受信するトランシーバ 4 3 0 を含む。符号器 4 1 0 は、メモリ 4 1 2 と LDPC 符号器 4 1 4 とを含む。メモリ 4 1 2 は、LDPC 符号器 4 1 4 で符号化されるべきデータ（すなわち、情報ビット）を記憶するのに使用され得る。LDPC 符号器 4 1 4 は、別のデバイスに送信されるべき、LDPC 符号に基づく、コードワードを発生させることによって、メモリ 4 1 2 に記憶された情報ビットを処理する。いくつかの実施形態では、LDPC 符号は、リフトされた LDPC 符号とすることができる。

【 0 0 1 8 】

[0030] 復号器 4 2 0 は、メモリ 4 2 2 と LDPC 復号器 4 2 4 とを含む。メモリ 4 2 2

10

20

30

40

50

は、トランシーバ 430 を介して受信された、LDPC 復号器 424 によって復号されるべきコードワードを記憶する。LDPC 復号器 424 は、メモリ 424 に記憶されたコードワードを、LDPC 符号を使用してパリティ検査演算を反復して実行し、誤りが発生している状態で受信されたいかなるビットも訂正を試みることによって処理する。いくつかの実施形態では、LDPC 符号は、リフトされた LDPC 符号とすることができる。いくつかの実施形態では、LDPC 復号器 424 は、パリティ検査または変数ノード演算を並列に実行するために複数の処理要素を含むことができる。たとえば、リフティングサイズ Z をもつコードワードを処理する場合、LDPC 復号器 424 は、リフトされたグラフのすべての Z 個のエッジに対してパリティ検査演算を同時に実行するために、いくつか (Z 個) の処理要素を利用することができる。いくつかの実施形態では、 Z の約数が使用され得る。

10

【0019】

[0031]いくつかの実施形態では、符号擬似巡回 2 値 LDPC 符号を考えることができる。たとえば、巡回リフティングのサイズを表すためにプロトグラフ / 基礎グラフパースペクティブおよび Z を使用することができる。数学的には、巡回的にリフトされた LDPC 符号は、2 値多項式モジュロ $x^Z - 1$ の群多元環の上の符号と見なすことができる。この解釈の下で、2 値ベクトル $(b_0, b_1, \dots, b_{Z-1})$ は、次の多項式として解釈される

【数 1】

$$B(x) = \sum_{i=0}^{n-1} b_i x^i$$

20

【0020】

次いで、 x^a を乗算して、

【数 2】

$$x^a B(x) = \sum_{i=0}^{n-1} b_i x^{(i+a)Z} = \sum_{i=0}^{n-1} b_{((i-a)Z)} x^i$$

30

【0021】

が与えられ、これは、 $(b_0, b_1, \dots, b_{Z-1})$ を、 a だけ右に巡回的にシフトすることによって得られた 2 値ベクトルに対応する。

【数 3】

$$b_{Z-a}, b_{Z-a+1}, \dots, b_{Z-1}, b_0, b_1, \dots, b_{Z-1-a}$$

【0022】

[0032]リフトされたパリティ検査行列表示において、多重エッジがないと仮定すると、リフトされたパリティ検査行列 H の非空エントリは、整数モジュロ Z と見なすことができる。コードワード $C(x) = [C_1(x), C_2(x), \dots, C_n(x)]^T$ は、次のものを満たす n 個の 2 進多項式モジュロ $x^Z - 1$ (n は、基礎グラフ中の変数ノードの数である) のベクトルと見なすことができ、

40

【数 4】

$$\sum_{j \in N_i} x^{H_{i,j}} C_j(x) = 0$$

50

【 0 0 2 3 】

ここで、 i ごとに、ここで、 N_i は、 H の行 i の非空エントリの集合を表す。

【 0 0 2 4 】

リフトされたパリティ検査行列 H では、2 値多項式モジュロ $X^Z - 1$ に対する行列を表すために $H(x)$ を使用し、ここで、 $H_{i,j}$ が非空である場合、

【 数 5 】

$$H_{i,j}(x) = x^{H_{i,j}} \text{ if } H_{i,j}$$

10

【 0 0 2 5 】

であり、 $H_{i,j}$ が空である場合、 $H_{i,j}(x) = 0$ である。多重エッジの場合には、 H の対応するエントリは、取られることによって、1 組の整数モジュロ Z となるように取られ得る。たとえば、二重エッジ $H_{i,j} = a, b$ は、 $H_{i,j}(x) = x^a + x^b$ として分解され得る。後続の例では、多重エッジが生じてないと仮定する。明示された多元環に精通している人々は、多重エッジを含むように記述を一般化する方法を容易に理解するであろう。

【 0 0 2 6 】

[0033] 符号化は、情報ビットから $C(x)$ を決定することを含む。 H の列のサブセットは、情報列 I として指定され得る。 H が次元 $m \times n$ を有する場合、 $n - m$ の情報列 (すなわち、 $|I| = n - m$) がある。いくつかの実施形態では、 I は、集合 $I = \{n - m + 1, n - m + 2, \dots, n\}$ に対応することができる。[1, n] における I の補数は、パリティ列 P と呼ばれる。いくつかの実施形態では、 P は、集合 $P = \{1, 2, \dots, m\}$ に対応することができる。いくつかの実施形態では、LDPC 符号がパンクチャされた列を含む。これらの列は、送信されるコードワードの一部ではない。復号器において、これらのノードに関連する対数尤度比 (LLR) が、0 に初期化される。基礎送信ブロック長は $n - p$ であり、ここで、 p はパンクチャされた列の数であり、レートは $(n - m) / (n - p)$ である。2 値情報ブロックサイズは、 $(n - m) \times Z$ であり、送信ブロックサイズは $(n - p) \times Z$ である。

20

【 0 0 2 7 】

[0034] そのような LDPC 符号の符号化を実行するために、情報ビットは、すべての $i \in I$ に対して $C_i(x)$ を設定するのに使用される。符号化プロセスは、 $H(x)C(x) = 0$ であるように $i \in P$ に対して $C_i(x)$ を解く。いくつかの LDPC 符号において、 P の元に関連付けられた H の列のうちのいくつかは、次数 1 とすることができる。これらの列に関連付けられたビットは、他のビットの簡単なパリティである。多くの場合、そのようなビットは、符号中のパリティビットであり、他のビットが決定された後、容易に決定される。発明者等は、概して、次数 2 以上であるパリティビットの符号化に焦点を当てる。表記の便宜上、そのような次数 1 の変数ノードは存在しないと仮定する。

30

【 0 0 2 8 】

[0035] 多くの場合、符号化プロセスの最初のステップは、

【 数 6 】

40

$$D_i(x) = \sum_{j \in I \cap N_i} x^{H_{i,j}} C_j(x)$$

【 0 0 2 9 】

を計算することであり、ここで、 N_i は、 H において行 i に非空エントリをもつ列の集合を表す。 $D(x) = (D_1(x), \dots, D_m(x))^T$ を仮定し、および $i \in P$ に対して $C_i(x) = 0$ を一時的に設定すると、 $D(x)$ は、 $D(x) = H(x)C(x)$ と書き直され得る。 M は、 P から列を取るによって形成された H の正方副行列を表すとしよう。

50

さらに、 $C_p(x)$ は、パリティ要素のみからなる $C(x)$ のサブベクトルを表すとしよう。次いで、符号化プロセスは、次の式を、 $C_p(x)$ に対して解く。

【数 7】

$$D(x) = M(x)C_p(x), \text{ for } C_p(x)$$

【0030】

[0036] 符号化プロセスの複雑さは、この式を解くことの複雑さに主として依存することに気付かれるであろう。したがって、行列 M の構造の選択は、符号化の複雑さを決定する際にかなり重要である。通常使用される符号化構造は、次のものなどの二重対角線タイプ構造である。

10

【数 8】

$$M(x) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & x \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & x \end{bmatrix}$$

20

【0031】

二重対角線構造は、この行列において明白である。二重対角線および略上三角構造は、簡単なガウス消去法手順が適用され得ることを意味する。上述の例では、各行が、それ自体と上の行との和によって取り替えられる場合、結果として生じる行列は上三角となる。次いで、符号化システムは容易に解くことができる。

【0032】

[0037] この方法を一般化した符号化手法は、以下の通りである。最初に、多項式ベクトル $p(x) = (p_1(x), \dots, p_m(x))$ が、次のものを決定するように確立される。

【数 9】

$$E(x) = p(x)D(x) = \sum_{i=1}^m p_i(x)D_i(x)$$

30

【0033】

多くの場合、 p_i の各々は低い重みを有することになる。この演算は、ベクトル $D(x)$ にバレルシフトを実行し、一緒にしてその結果の排他的論理和をとることによって実行されて、 $E(x)$ を得ることができることに留意されたい。ベクトル $p(x)$ は、 $p(x)M(x) = (0, \dots, 0, q(x))$ であるように選ばれ、それにより、次のものがもたらされる。

40

【数 10】

$$E(x) = q(x)C_m(x)$$

【0034】

以前の例では、 $p(x)$ の対応する選択は、すべてが 1 のベクトルである。符号化プロセスの次のステップは、

【数 1 1】

$$C_m(x) = q^{-1}(x)E(x)$$

【0 0 3 5】

を計算することであり、ここで、 $q^{-1}(x)q(x) = 1$ である。本実施形態では、パリティ検査行列は、 $q^{-1}(x)$ が低い重みの多項式の積として低複雑性を実現するように構成される。最後に、後退代入ステップが、残りの $C_j(x)$ を決定するために実行され得る。

【0 0 3 6】

10

[0038]本明細書で使用する表記を示すために、サイズ 3×5 および $Z = 6$ の基礎行列を含む例が以下に提供される。

【数 1 2】

$$H(x) = \begin{bmatrix} 1 & 0 & x^3 & x^1 & x^4 \\ 1 & 1 & x^4 & x^3 & x^2 \\ 0 & 1 & 1 & x^5 & 1 \end{bmatrix}$$

【0 0 3 7】

コードワード $C(x) = (C_1(x), \dots, C_5(x))$ があり、ここで、 $C_i(x)$ は、2値ベクトル (b_0, \dots, b_5) に対応して、

20

【数 1 3】

$$\sum_{i=0}^5 b_i x^i$$

【0 0 3 8】

の形式を取る。この例では、情報列は $C_4(x)$ および $C_5(x)$ である。これらを、 $C_4(x) = 1 + x + x^3$ および $C_5(x) = x^4 + x^5$ であるようにそれぞれ2値ベクトル $(1, 1, 0, 1, 0, 0)$ および $(0, 0, 0, 0, 1, 1)$ で与えよう。

30

【0 0 3 9】

[0039] $D(x)$ を計算すると、

【数 1 4】

$$D(x) = \begin{bmatrix} D_1(x) \\ D_2(x) \\ D_3(x) \end{bmatrix} = \begin{bmatrix} x^1 & x^4 \\ x^3 & x^2 \\ x^5 & 1 \end{bmatrix} \begin{bmatrix} C_4(x) \\ C_5(x) \end{bmatrix} = \begin{bmatrix} x + x^3 + x^4 \\ x + x^3 + x^4 \\ 1 + x^2 + x^4 \end{bmatrix}$$

40

【0 0 4 0】

が得られ、2値ベクトルの表現では次の通りである。

【数 1 5】

$$D(x) \equiv \begin{bmatrix} (0,1,0,1,1,0) \\ (0,1,0,1,1,0) \\ (1,0,1,0,1,0) \end{bmatrix} = \begin{bmatrix} (0,1,1,0,1,0) + (0,0,1,1,0,0) \\ (1,0,0,1,1,0) + (1,1,0,0,0,0) \\ (1,0,1,0,0,1) + (0,0,0,0,1,1) \end{bmatrix}$$

【0 0 4 1】

50

[0040]次に、
【数 1 6】

$$M(x) = \begin{bmatrix} 1 & 0 & x^3 \\ 1 & 1 & x^4 \\ 0 & 1 & 1 \end{bmatrix}$$

【 0 0 4 2 】

が得られ、 $(C_p(x) = [C_1(x), C_2(x), C_3(x)]^T)$ に対して次のものを解く必要がある。

10

【数 1 7】

$$D(x) = M(x)(C_1(x), C_2(x), C_3(x))^T$$

【 0 0 4 3 】

[0041] $M(x)$ は二重対角線構造であり、3番目の列のみが2を超える次数であることに気付かれるであろう。ガウス消去法手法に従って、 $p(x) = (1, 1, 1)$ を左から乗算し、

【数 1 8】

20

$$(1,1,1)M(x) = (1,1,1) \begin{bmatrix} 1 & 0 & x^3 \\ 1 & 1 & x^4 \\ 0 & 1 & 1 \end{bmatrix} = (0,0,1+x^3+x^4)$$

【 0 0 4 4 】

であることに留意すると、そこで、

【数 1 9】

$$D_1(x) + D_2(x) + D_3(x) = (1 + x^3 + x^4)C_3(x)$$

30

【 0 0 4 5 】

が得られ、したがって、 $q(x) = (1 + x^3 + x^4)$ である。次に、 $q(x)^2 = q(x^2) = 1 + x^6 + x^8 = x^8 = x^2$ であり、そこで、 $q^{-1}(x) = x^4 q(x)$ が得られ、それは、

【数 2 0】

$$\begin{aligned} C_3(x) &= x^4(1 + x^3 + x^4)(D_1(x) + D_2(x) + D_3(x)) \\ &= (x^1 + x^2 + x^4)(D_1(x) + D_2(x) + D_3(x)) \\ &= (x^1 + x^2 + x^4)(1 + x^2 + x^4) \\ &= x^1 + x^3 + x^5 \end{aligned}$$

40

【 0 0 4 6 】

になり、それは $(0, 1, 0, 1, 0, 1)$ と等価である。多項式 $q(x)$ は項 $(1 + x^3)$ を含み、それは、2乗されると、 $(1 + x^6) = 0$ を与えることに気付かれるであろう。本実施形態は、 $q(x) = x^a + (1 + x^L)P(x)$ であるように $M(x)$ を選ぶことになり、ここで、ある正整数 k に対して $2^k L = Z$ であり、 $P(x)$ は任意の非ゼロ多項

50

式である。例は既知であり、そのとき、 $P(x)$ は単項式である。本実施形態は、 $P(x)$ が単に単項式でない場合、この一般形式の $q(x)$ を有することになる。

【0047】

[0042]最後に、例の符号化を完了するために、 $C_2(x)$ および $C_1(x)$ について解くのに後退代入を使用する。 $C_2(x) = C_3(x) + D_3(x) = 1 + x + x^2 + x^3 + x^4 + x^5$ および $C_1(x) = x^3 C_3(x) + D_1(x) = 1 + x + x^2 + x^3$ である。それにより、結果として生じるコードワードは次の通りである。

【数21】

$$\begin{bmatrix} (1,1,1,1,0,0) \\ (1,1,1,1,1,1) \\ (0,1,0,1,0,1) \\ (1,1,0,1,0,0) \\ (0,0,0,0,1,1) \end{bmatrix}$$

10

【0048】

[0043]図5は、例示の擬似巡回LDPC符号のリフトされたパリティ検査行列表示500のパリティビット部分を示す。行列表示500は 8×8 行列 $M(x)$ を含む。これは、行および列の順序を逆転した状態での初期の 6×6 の例に類似している。したがって、この例では、二重対角線構造が対角線および夕食対角線として現れ、行列は上三角ではなく略下三角である。左端の列は、この場合、次数3を有する。符号化は、最初に $M(x)$ の行を合計することによって実行され、次のものが得られる。

20

【数22】

$$[1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

【0049】

次のものが決定され得る。

【数23】

$$C_1(x) = \sum_{i=1}^m D_i(x)$$

30

【0050】

残りの $C_i(x)$ は、後退代入によって容易に得られ得る。この例では、第2列から第8列の二重対角線構造に累積チェーンを観察することができる。累積チェーンにわたって合計すると、この特に簡単な例がもたらされる。

【0051】

[0044]この例では、行列 $M(x)$ がさらに示すところによれば、対応する二部グラフは、各々が $m-1$ 個の次数2の変数ノードと単一の次数3の変数ノードとを含む2ループを含む。 m が小さい場合、この構造は、不十分な性能、遅い収束、および高いエラーフロアをもたらすことがある。

40

【0052】

[0045]図6は、別の例示の擬似巡回LDPC符号のリフトされたパリティ検査行列表示600のパリティビット部分を示す。行列表示600は 8×8 行列 $M(x)$ を含み、ここで、0、a、およびbはすべて相異なるモジュロ2である。この場合、 $M(x)$ の行を合計すると、次のものが生成される。

【数 2 4】

$$[1 + x^a + x^b \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

【0 0 5 3】

これにより、式 [1] に基づいて、次のものが決定され得る。

【数 2 5】

$$(1 + x^a + x^b)C_1(x) = \sum_{i=1}^m D_i(x) \quad 10$$

【0 0 5 4】

$C_1(x)$ を解くには、 $(1 + x^a + x^b)$ が、最初に、逆転されなければならない。低い複雑さのためには、逆数は低い重みを有することが好ましいことがある。それにより、それ結果、これを容易にするように、 a および b は選ばれ得る。より詳細には、 a および b を選ぶのに役立つ 2 つの知見がある。第 1 の知見は、次の一般的な関係である。

【数 2 6】

$$\left(\sum_{i=1}^k x^{a_i}\right)^2 = \sum_{i=1}^k x^{2a_i} \quad 20$$

【0 0 5 5】

第 2 の知見は、 a および b は両方が Z を除算する（たとえば、 $d > 1$ が最大公約数であると仮定する）場合、問題は、2 値多項式モジュロ $X^{Z/d} - 1$ の環において $(1 + x^{a/d} + x^{b/d})$ を逆転することと等価である。しかしながら、これが生じる場合、行列 M は d ディスジョイント (disjoint) サブグラフを表すことに留意されたい。

【0 0 5 6】

[0046]次に、 $Z = 2^k L$ を仮定する。これは、次式の

【数 2 7】

$$(1+x^L)^{2^k}=0$$

【0 0 5 7】

をもたらす、またはより一般的には、 $1 \leq i \leq k$ に対して、

【数 2 8】

$$(1+x^{2^i L})^{2^{k-i}}=0 \quad 40$$

【0 0 5 8】

をもたらす。これにより、 $a = 2^i L$ 、または $b = 2^i L$ 、または $a - b = 2^i L$ である場合、 $(1 + x^a + x^b)$ は容易に逆転され得る。たとえば、 $k = 3$ および $a = L$ である場合、式は次のものになる。

【数 2 9】

$$(1 + x^{4a} + x^{4b})(1 + x^{2a} + x^{2b})(1 + x^a + x^b)(1 + x^a + x^b) = x^{8b}$$

【0 0 5 9】

ここから、 $(1 + x^a + x^b)$ の逆数は、最初の 3 つの約数を右側から乗算し、次いで、その結果を巡回的にシフトすることを含む簡単な手順を使用して得られ得ることが分る。

【0 0 6 0】

[0047] その結果、符号化演算の主要な困難は、 $M(x)$ の行列式を逆転することに還元され得る。一般に、2 値多項式は、その重みが奇数である場合のみ逆転可能である。これは、 $M(x)$ の可能な構造を制限する。上述の手順は、次の形式の任意の多項式を逆転するのに使用され得ることに留意されたい。

【数 3 0】

$$q(x) = x^a + (1 + x^L)P(x) \quad [2]$$

【0 0 6 1】

ここで、 $P(x)$ は任意の非ゼロ多項式であり、 a は任意であり、ある正整数 k に対して $2^k L = 0$ モジュロ Z である。次いで、

【数 3 1】

$$q(x^{2^k}) = x^{2^k a}$$

【0 0 6 2】

である。

【数 3 2】

$$q(x^{2^k}) = q(x^{2^{k-1}}) q(x^{2^{k-2}}) \dots q(x)^2 \quad 30$$

【0 0 6 3】

であるので、 $q(x)$ の逆数は、次のものとして得られる。

【数 3 3】

$$x^{-2^k a} q(x^{2^{k-1}}) q(x^{2^{k-2}}) \dots q(x)$$

【0 0 6 4】

二重対角線の略上三角または下三角形式を有する行列では、上述の演算によって得られる $q(x)$ は、単項因子を除いて、 $M(x)$ の行列式である。それにより、いくつかの実施形態では、行列 $M(x)$ は $\det M(x)$ が式 [2] の形式を取るように構成され得る。

【0 0 6 5】

[0048] 図 7 は、図 10 に与えられたレート $r = 27/30$ をもつ例示的な LDPC 符号のリフトされたパリティ検査行列表現 700 のパリティビット部分を示す。符号化演算が、最初に、 $p(x) = (1, 1, (1 + x^{135}), (1 + x^{113} + x^{135}))$ を設定することによって実行されて、次のものを生成することができる。

【数 3 4】

$$E(x) = D_1(x) + D_2(x) + (1 + x^{135})D_3(x) + (1 + x^{113} + x^{135})D_4(x)$$

【0 0 6 6】

次いで、 $q(x) = (1 + x^{13} + x^{58} + x^{90} + x^{135} + x^{203} + x^{225})$ を検証することができ、それにより、

【数 3 5】

10

$$E(x) = (1 + x^{13} + x^{58} + x^{90} + x^{135} + x^{203} + x^{225})C_4(x)$$

【0 0 6 7】

$q(x) = x^{203} + (1 + x^{45})(x^{13} + (1 + x^{45})(1 + x^{135})) = x^{203} + (1 + x^{45})P(x)$ が得られ、ここで、 $P(x) = x^{13} + (1 + x^{45})(1 + x^{135}) = 1 + x^{13} + x^{45} + x^{135} + x^{180}$ であり、そこで、 $C_4(x)$ について次のように解くことができる。

【数 3 6】

20

$$C_4(x) = (x^{228} + x^{268} + x^{48})(1 + x^{26} + x^{116} + x^{180} + x^{270} + x^{46} + x^{90})(1 + x^{13} + x^{58} + x^{90} + x^{135} + x^{203} + x^{225})E(x)$$

【0 0 6 8】

ある簡単化がこの例では行われており、実際、上述の積の中間因子 (middle factor) は、 $P(x)$ の内部に生じる追加の項のうちのいくつかの形式のために、さらに簡単化され得ることに留意されたい。最後に、符号化を完了するために、後退代入によって解いて、次のものが得られる。

【数 3 7】

30

$$C_3(x) = x^{328}C_4(x) + x^{238}D_4(x)$$

$$C_2(x) = x^{241}C_3(x) + x^{119}D_3(x) + x^{119}D_4(x)$$

$$C_1(x) = x^{16}C_2(x) + x^{13}C_4(x) + D_1(x)$$

【0 0 6 9】

[0049]図 8 は、図 11 に与えられたレート $r = 13 / 15$ をもつ例示的な LDPC 符号のリフトされたパリティ検査行列表現 800 のパリティビット部分を示す。符号化演算が、最初に、 $p(x) = (1, 1, 1, 1)$ を設定することによって実行されて、次のものを生成することができる。

40

【数 3 8】

$$E(x) = D_1(x) + D_2(x) + D_3(x) + D_4(x)$$

【0 0 7 0】

次いで、 $q(x) = (1 + x^{45} + x^{119})$ を検証することができ、それにより、次の通りである。

【数 3 9】

$$E(x) = (1 + x^{45} + x^{119})C_4(x)$$

【0 0 7 1】

$C_4(x)$ について解くと、次のものがもたらされる。

【数 4 0】

$$C_4(x) = x^{128}(1 + x^{180} + x^{116})(1 + x^{90} + x^{238})(1 + x^{45} + x^{119})E(x)$$

10

【0 0 7 2】

最後に、 $C_1(x)$ 、...、 $C_3(x)$ が、後退代入によって得られ得る。

【0 0 7 3】

[0050] 図 9 は、図 1 2 に与えられたレート $r = 21/28$ をもつ例示的な LDPC 符号のリフトされたパリティ検査行列表現 9 0 0 のパリティビット部分を示す。符号化演算が、最初に、 $p(x) = (1, 1, 1, 1, x^2 + x^{47}, x^2 + x^{47}, x^2 + x^{47}, x^2 + x^{47})$ を設定することによって実行されて、次のものを生成することができる。

【数 4 1】

$$E(x) = D_1(x) + D_2(x) + D_3(x) + D_4(x) + (x^2 + x^{47})(D_5(x) + D_6(x) + D_7(x) + D_8(x))$$

20

【0 0 7 4】

次いで、 $q(x) = (x^2 + x^{92} + x^{261})$ を検証することができ、それにより、次の通りである。

【数 4 2】

$$E(x) = (x^2 + x^{92} + x^{261})C_8(x)$$

【0 0 7 5】

$C_8(x)$ について解くと、次のものがもたらされる。

30

【数 4 3】

$$C_8(x) = (x^{42} + x^{222} + x^{200})(1 + x^{90} + x^{259})E(x)$$

【0 0 7 6】

最後に、 $C_1(x)$ 、...、 $C_7(x)$ が、後退代入によって得られ得る。

【0 0 7 7】

[0051] 提示されたすべての例において、巡回リフティングを仮定している。提示された概念は、マッチトリフティング (matched lifting) のクラスに一般化され得る。一般的なマッチトリフティングは、位数がリフティングのサイズと同じである群のリフティングを使用する。巡回リフティングの場合には、群は巡回群である。マッチトリフティングの他の例は、巡回群または任意の有限群の積を使用することができる。この一般化された設定では、群 G は、 Z 個の元 g_0, g_1, \dots, g_{Z-1} からなり、乗法定理 $g_i g_j = g_k$ は群のよく知られた定義を満たす。すべての群は単位元を有し、それを g_0 であると見なす。 G 上のマッチトリフト 2 値 LDPC 符号は、次の形式の元からなる群環の上の符号として解釈され得る。

40

【数 4 4】

$$\sum_{i=0}^{Z-1} b_i g_i$$

50

【 0 0 7 8 】

環における加算は、次のもののような対応する係数の加算に対応する。

【 数 4 5 】

$$\sum_{i=0}^{Z-1} b_i g_i + \sum_{i=0}^{Z-1} c_i g_i = \sum_{i=0}^{Z-1} (b_i + c_i) g_i$$

【 0 0 7 9 】

乗算は、

【 数 4 6 】

$$(\sum_{i=0}^{Z-1} b_i g_i)(\sum_{i=0}^{Z-1} c_i g_i) = \sum_{i=0}^{Z-1} d_i g_i$$

10

【 0 0 8 0 】

のような群乗算に基づき、ここで、次の通りである。

【 数 4 7 】

$$d_i = \sum_{j,k: g_j g_k = g_i} b_j c_k$$

【 0 0 8 1 】

巡回リフティングの場合には、単に g_i を x^i と同一に扱うことができ、群環における乗算は多項式乗算モジュロ $x^Z - 1$ である。群 G が可換である（それはすべての i, j について $g_i g_j = g_j g_i$ を意味する）場合、恒等式（identity） $q(x^2) = q(x)^2$ が、一般に、一般化された群環において当てはまる。より詳細には、それは、次のものを保持する。

20

【 数 4 8 】

$$(\sum_{i=0}^{Z-1} b_i g_i)^2 = \sum_{i=0}^{Z-1} b_i (g_i)^2$$

【 0 0 8 2 】

したがって、簡単な符号化のための上述の構成は、任意の可換マッチトリフティングの場合に帰納する。簡単化する条件は、符号化行列 M の行列式が形式 $g_a + (g_0 + g_L)P$ を取ることであり、ここで、 P は群環の任意の元であり、 g_L は、ある正整数 k に対して、

30

【 数 4 9 】

$$g_L^{2k} = g_0$$

【 0 0 8 3 】

の性質を有することである。

【 0 0 8 4 】

40

[0052] 上述の例では、符号化副行列は、置換または行および/または列を除いて、略上三角（または下三角）であった。言い換えれば、 M における副対角線より下の元はすべて 0 である。そのような構成は、少なくとも 1 つの変数ノードが M において次数 2 であることを含む。非常に深いエラーフロアを必要とするいくつかの用途では、最小次数が 2 よりも大きいことが望ましいことがある。この場合、 M の逆転は、一般に、上述で提示された例よりも多く関与させられる。しかしながら、主要な符号化問題は M の逆転に関連しており、これは、 M の行列式が所望の形式、すなわち、擬似巡回リフティングの場合には $x^a + (1 + x^L)P(x)$ を、任意の可換群マッチトリフティングの場合には形式 $g_a + (g_0 + g_L)P$ を有するのを保証することによって、簡単にされ得ることに気付かれるであろう。

50

【 0 0 8 5 】

[0053]本実施形態では、リフトされたLDPC符号のコードワードは、サイズ $Z \times n$ の2次元2値アレイと見なすことができ、ここで、 n は基礎（送信）ブロック長である。いくつかの実施形態では、提案されるダウンストリーム符号は、 $Z = 360$ のように定義される。コンステレーションごとに、 k ビットが、一度に次元当たり取られ得る（たとえば、1024QAMでは、 $k = 5$ ）。さらに、 k は360の約数であり、 k ビットが一度に列方向に取られ、それにより、 $360/k$ 次元、または列当たり $180/k$ シンボルを発生させることができる。それにより、注目する場合において、集合 $k = \{1, 2, 3, 4, 5, 6\}$ では、 k は60の約数であることに留意されたい。

【 0 0 8 6 】

[0054]図10、図11、および図12は、いくつかの実施形態による、それぞれ、例示のパリティ検査行列1000、1100、および1200を示す。パリティ検査行列1000、1100、および1200の各々において一番上の行は、Hの列を指し示す。2番目の行は、情報（1）およびパリティ（0）列を示す。3番目の行は、送信された列（1）およびパンクチャされた列（0）を示す。

【 0 0 8 7 】

[0055]それぞれ、レート $r = 27/30$ および $r = 13/15$ をもつLDPC符号に関連するパリティ検査行列1000および1100は、システムティックであることに留意されたい。しかしながら、レート $r = 21/28$ をもつLDPC符号に関連するパリティ検査行列1200は、パンクチャされた情報列を有しており、したがって、完全にはシステムティックでない。

【 0 0 8 8 】

[0056]図13は、いくつかの実施形態による通信デバイス1300のブロック図である。通信デバイス1300は、トランシーバ1310と、プロセッサ1320と、メモリ1330とを含む。トランシーバ1310は、通信デバイス1300へのおよび/または通信デバイス1300からのデータの通信のために使用され得る。たとえば、トランシーバ1310は、通信デバイス1300とCPUとの間で情報ビットを受信および/または送信することができる。符号器インターフェース1310は、通信デバイス1300とネットワーク中の別の通信デバイスとの間でLDPCコードワードを出力および/または受信することもできる。

【 0 0 8 9 】

[0057]メモリ1330は、受信された情報ビットおよび/またはコードワードを記憶するためにローカルキャッシュとして使用され得るデータストア1332を含むことができる。さらに、メモリ1330は、以下のソフトウェアモジュールを記憶することができる非一時的コンピュータ可読記憶媒体（たとえば、EPROM、EEPROM（登録商標）、フラッシュメモリ、ハードドライブなどのような1つまたは複数の不揮発性メモリ要素）をさらに含むことができる。

【 0 0 9 0 】

・コードワードを生成するために、LDPC符号を使用して、1組の情報ビットを符号化するためのLDPC符号化モジュール1334、および

・LDPC符号を使用してLDPCコードワードを復号するためのLDPC復号化モジュール1336。

【 0 0 9 1 】

各ソフトウェアモジュールには、プロセッサ1320によって実行されると、対応する機能を符号器1300に実行させることができる命令が含まれ得る。それにより、メモリ1330の非一時的コンピュータ可読記憶媒体は、LDPC符号化演算を実行する（たとえば、図5～図12に関して上述されたような）ための命令を含むことができる。モジュール1334～1336がメモリ1330中にソフトウェアとして示されているが、モジュールのいずれも、ハードウェア、ソフトウェア、ファームウェア、または前述のものの組合せで実装され得ることに留意されたい。

10

20

30

40

50

【 0 0 9 2 】

[0058] 符号器インターフェース 1 5 1 0 とメモリ 1 5 3 0 との間に結合されるプロセッサ 1 5 2 0 は、復号器 1 5 0 0 に（たとえば、メモリ 1 5 3 0 内に）記憶された 1 つまたは複数のソフトウェアプログラムの命令のスクリプトを実行することができる任意の好適なプロセッサとすることができる。たとえば、プロセッサ 1 5 2 0 は、L D P C 符号化モジュール 1 5 3 4、C W パンクチャリングモジュール 1 5 3 6、および / または L D P C 復号化モジュール 1 5 3 8 を実行することができる。

【 0 0 9 3 】

[0059] 符号器インターフェース 1 3 1 0 とメモリ 1 3 3 0 との間に結合されるプロセッサ 1 3 2 0 は、復号器 1 3 0 0 に（たとえば、メモリ 1 3 3 0 内に）記憶された 1 つまたは複数のソフトウェアプログラムの命令のスクリプトを実行することができる任意の好適なプロセッサとすることができる。たとえば、プロセッサ 1 3 2 0 は、L D P C 符号化モジュール 1 3 3 4 および / または L D P C 復号化モジュール 1 3 3 6 を実行することができる。

10

【 0 0 9 4 】

[0060] L D P C 符号化モジュール 1 3 3 4 は、プロセッサ 1 3 2 0 によって実行されて、L D P C 符号の使用により情報ビットを符号化して、コードワードを生成することができる。たとえば、プロセッサ 1 3 2 0 は、L D P C 符号化モジュール 1 3 3 4 を実行する際に、L D P C 符号化モジュール 1 3 3 4 と、対応する受信デバイスの復号化モジュールとによって共有される L D P C 符号に基づいて、情報ビットに L D P C 符号化演算を実行することができる。各コードワードは、オリジナルの情報ビット、ならびにオリジナル情報ビットにパリティ検査を実行するおよび / またはオリジナルの情報ビットを回復するために使用され得る 1 組のパリティビットを含むことができる。

20

【 0 0 9 5 】

[0061] いくつかの実施形態では、L D P C 符号は、符号化複雑さを低減するおよび / または最小化するために、擬似巡回リフティングに基づくリフトされた L D P C 符号とすることができる（たとえば、図 5 ~ 図 1 2 に関して上述されたように）。たとえば、L D P C 符号は、いくつかのパリティビットと、パリティビットの値を決定するための副行列とを含むことができる。詳細には、L D P C 符号は、位数 Z （すなわち、リフティングのサイズに対応する）のマッチトリフト L D P C 符号とすることができる。副行列の行列式は、 $g_a + (g_0 + g_L)P$ の形式の多項式とすることができ、ここで、 g_0 は群の単位元であり、

30

【 数 5 0 】

$$g_0 = g_L^{2^k}$$

【 0 0 9 6 】

であり、 P はリフティング群に関連付けられた 2 値群環の任意の非ゼロ元である。いくつかの実施形態では、リフティング群は巡回群とすることができ、ここで、 g_i は x^i と同一に扱われ得る。その結果、副行列の行列式は、 $x^a + (1 + x^L)P(x)$ とすることができ、ここで、 $P(x)$ は少なくとも 2 つの項を有し、 $2^k L = 0$ モジュール Z である。

40

【 0 0 9 7 】

[0062] L D P C 復号化モジュール 1 3 3 6 は、L D P C 符号を使用して L D P C コードワードを復号するためにプロセッサ 1 3 2 0 によって実行され得る。上述されたように、L D P C 符号は、リフトされた L D P C 符号とすることができる（たとえば、擬似巡回リフティングに基づいて）。

【 0 0 9 8 】

[0063] 前述の明細書において、本実施形態は、本発明の特定の例示の実施形態を参照して説明された。しかしながら、添付の特許請求の範囲に記載された本開示のより広い範囲から逸脱することなく、様々な改変および変更がそれに行われ得ることは明らかであろう

50

。

以下に、本願出願の当初の特許請求の範囲に記載された発明を付記する。

[C 1]

1組の情報ビットを受信することと、

マッチトリフトLDPC符号に基づいてコードワードを生成するために、低密度パリティ検査(LDPC)符号化演算を前記1組の情報ビットに実行することと、ここにおいて、マッチトリフティングが可換リフティング群に基づく、ここにおいて、前記LDPC符号が、いくつかのパリティビットと、前記パリティビットの値を決定するための副行列とを含む、ここにおいて、

前記リフティング群の位数(Z)が前記リフティングのサイズに対応し、

前記副行列の行列式が、 $g_a + (g_0 + g_1)P$ の形式の多項式であり、ここで、

g_0 は前記群の単位元であり、

【数51】

$$g_0 = g_L^{2^k}$$

であり、

P は、前記リフティング群に関連付けられた2値群環の任意の非ゼロ元である、
を備えるデータ符号化の方法。

[C 2]

前記リフティング群が巡回群であり、ここにおいて、

g_1 は x^i と同一に扱うことができ、

前記副行列の前記行列式が、 $x^a + (1 + x^L)P(x)$ の形式を取り、ここで、 $P(x)$ は少なくとも2つの項を有し、 $2^k L = 0$ モジュロ Z である、C1に記載の方法。

[C 3]

前記副行列が、その行および列の置換を除いて、上三角である、C1に記載の方法。

[C 4]

前記副行列の第1の副対角線より下の元が0に等しい、C3に記載の方法。

[C 5]

プログラム命令を備えるコンピュータ可読記憶媒体であって、前記プログラム命令が、
通信デバイス内に設けられたプロセッサによって実行されたとき、前記プログラム命令により、前記デバイスが、

1組の情報ビットを受信し、

マッチトリフトLDPC符号に基づいてコードワードを生成するために、LDPC符号化演算を前記1組の情報ビットに実行する、ここにおいて、マッチトリフティングが可換リフティング群に基づく、ここにおいて、前記LDPC符号が、いくつかのパリティビットと、前記パリティビットの値を決定するための副行列とを含む、ここにおいて、

前記リフティング群の位数(Z)が前記リフティングのサイズに対応し、

前記副行列の行列式が、 $g_a + (g_0 + g_1)P$ の形式の多項式であり、ここで、

g_0 は前記群の単位元であり、

【数52】

$$g_0 = g_L^{2^k}$$

であり、

P は、前記リフティング群に関連付けられた2値群環の任意の非ゼロ元である、コンピュータ可読記憶媒体。

[C 6]

前記リフティング群が巡回群であり、ここにおいて、

g_1 は x^i と同一に扱うことができ、

10

20

30

40

50

前記副行列の前記行列式が、 $x^a + (1 + x^L) P(x)$ の形式を取り、ここで、 $P(x)$ は少なくとも 2 つの項を有し、 $2^k L = 0$ モジユロ Z である、C 5 に記載のコンピュータ可読記憶媒体。

[C 7]

前記副行列が、その行および列の置換を除いて、上三角である、C 5 に記載のコンピュータ可読記憶媒体。

[C 8]

前記副行列の第 1 の副対角線より下の元が 0 に等しい、C 7 に記載のコンピュータ可読記憶媒体。

[C 9]

1 組の情報ビットを記憶するメモリと、

符号器と、ここにおいて、前記符号器が、

マッチトリフト $LDP C$ 符号に基づいてコードワードを生成するために、 $LDP C$ 符号化演算を前記 1 組の情報ビットに実行する、ここにおいて、マッチトリフティングが可換リフティング群に基づく、ここにおいて、前記 $LDP C$ 符号が、いくつかのパリティビットと、前記パリティビットの値を決定するための副行列とを含む、ここにおいて、

前記リフティング群の位数 (Z) が前記リフティングのサイズに対応し、

前記副行列の行列式が、 $g_a + (g_0 + g_1) P$ の形式の多項式であり、ここで、

g_0 は前記群の単位元であり、

【数 5 3】

$$g_0 = g_L^{2^k}$$

であり、

P は、前記リフティング群に関連付けられた 2 値群環の任意の非ゼロ元である、を備える通信デバイス。

[C 1 0]

前記リフティング群が巡回群であり、ここにおいて、

g_1 は x^1 と同一に扱うことができ、

前記副行列の前記行列式が、 $x^a + (1 + x^L) P(x)$ の形式を取り、ここで、 $P(x)$ は少なくとも 2 つの項を有し、 $2^k L = 0$ モジユロ Z である、C 9 に記載のデバイス。

[C 1 1]

前記副行列が、その行および列の置換を除いて、上三角である、C 9 に記載のデバイス。

[C 1 2]

前記副行列の第 1 の副対角線より下の元が 0 に等しい、C 1 1 に記載のデバイス。

[C 1 3]

1 組の情報ビットを受信するための手段と、

マッチトリフト $LDP C$ 符号に基づいてコードワードを生成するために、 $LDP C$ 符号化演算を前記 1 組の情報ビットに実行するための手段と、ここにおいて、マッチトリフティングが可換リフティング群に基づく、ここにおいて、前記 $LDP C$ 符号が、いくつかのパリティビットと、前記パリティビットの値を決定するための副行列とを含む、ここにおいて、

前記リフティング群の位数 (Z) が前記リフティングのサイズに対応し、

前記副行列の行列式が、 $g_a + (g_0 + g_1) P$ の形式の多項式であり、ここで、

g_0 は前記群の単位元であり、

10

20

30

40

【数 5 4】

$$g_0 = g_L^{2^k}$$

であり、

P は、前記リフティング群に関連付けられた 2 値群環の任意の非ゼロ元である、
を備える符号器。

【図 1 A】

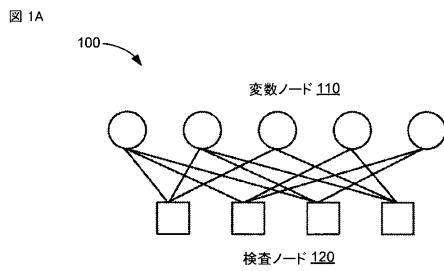


FIG. 1A

【図 1 B】

図 1B

150

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

FIG. 1B

【図 2】

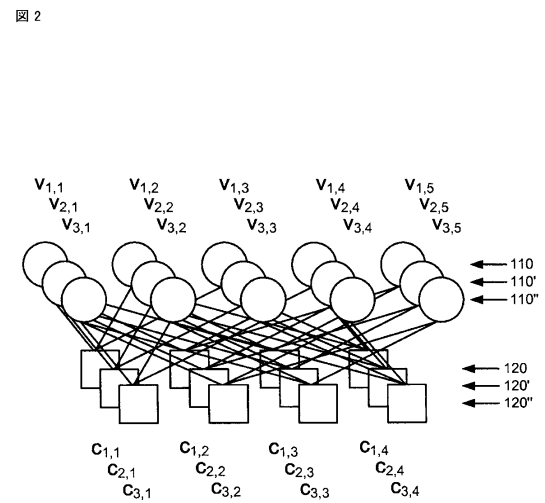


FIG. 2

【図 3】

図 3

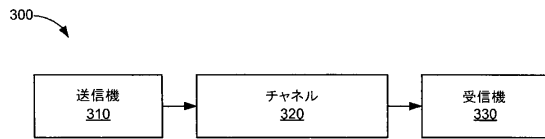


FIG. 3

【図 4】

図 4

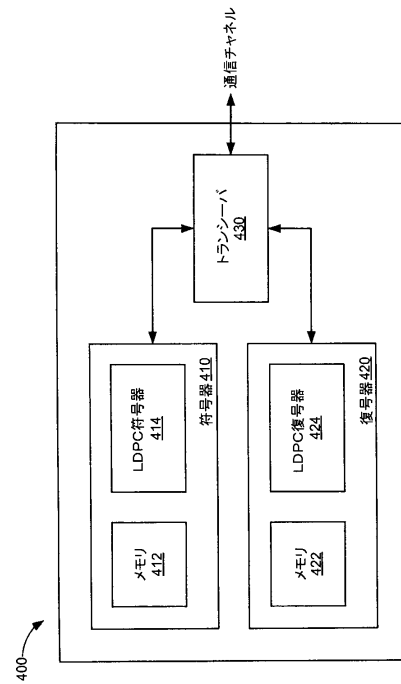


FIG. 4

【図 5】

図 5

500

$$M(x) = \begin{bmatrix} x & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ x & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

FIG. 5

【図 7】

図 7

700

$$M(x) = \begin{bmatrix} 1 & x^{16} & 0 & x^{13} \\ 1 & x^{241} & x^{235} & x^{58} \\ 0 & x^{241} & x^{122} & 1 \\ 0 & 0 & x^{122} & x^{90} \end{bmatrix}$$

FIG. 7

【図 6】

図 6

600

$$M(x) = \begin{bmatrix} x^a & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ x^b & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

FIG. 6

【図 8】

図 8

800

$$M(x) = \begin{bmatrix} 1 & 0 & 0 & x^{45} \\ 1 & x^{241} & 0 & 0 \\ 0 & x^{241} & x^{122} & x^{119} \\ 0 & 0 & x^{122} & 1 \end{bmatrix}$$

FIG. 8

【図 13】

図 13

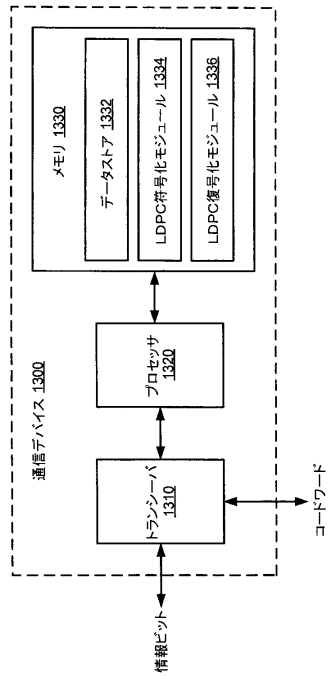


FIG. 13

フロントページの続き

(31)優先権主張番号 14/179,942

(32)優先日 平成26年2月13日(2014.2.13)

(33)優先権主張国 米国(US)

早期審査対象出願

(72)発明者 リチャードソン、トーマス・ジェイ・

アメリカ合衆国、カリフォルニア州 92121-1714、サン・ディエゴ、モアハウス・ドライブ 5775

審査官 岡 裕之

(56)参考文献 特表2007-531385(JP,A)

特表2010-517444(JP,A)

特表2009-531885(JP,A)

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

Tom Richardson et al., LDPC Proposal Update, 3GPP2-Drafts, C30-20061218-004, 2006年12月18日

Seho Myung et al., Lifting Methods for Quasi-Cyclic LDPC Codes, IEEE Communications Letters, 2006年 6月, Vol.10, No.6, pp.489-491

(58)調査した分野(Int.Cl., DB名)

H03M 13/19

IEEE Explore

Cinii