

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6542132号
(P6542132)

(45) 発行日 令和1年7月10日(2019.7.10)

(24) 登録日 令和1年6月21日(2019.6.21)

(51) Int.Cl.

HO3M 13/19 (2006.01)

F 1

HO3M 13/19

請求項の数 12 (全 21 頁)

(21) 出願番号 特願2015-557231 (P2015-557231)
 (86) (22) 出願日 平成26年2月13日 (2014.2.13)
 (65) 公表番号 特表2016-510185 (P2016-510185A)
 (43) 公表日 平成28年4月4日 (2016.4.4)
 (86) 國際出願番号 PCT/US2014/016261
 (87) 國際公開番号 WO2014/127129
 (87) 國際公開日 平成26年8月21日 (2014.8.21)
 審査請求日 平成29年1月17日 (2017.1.17)
 (31) 優先権主張番号 61/764,476
 (32) 優先日 平成25年2月13日 (2013.2.13)
 (33) 優先権主張国 米国(US)
 (31) 優先権主張番号 14/179,871
 (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)

(73) 特許権者 595020643
 クアアルコム・インコーポレイテッド
 QUALCOMM INCORPORATED
 アメリカ合衆国、カリフォルニア州 92
 121-1714、サン・ディエゴ、モア
 ハウス・ドライブ 5775
 (74) 代理人 100108855
 弁理士 蔵田 昌俊
 (74) 代理人 100109830
 弁理士 福原 淑弘
 (74) 代理人 100158805
 弁理士 井関 守三
 (74) 代理人 100194814
 弁理士 奥村 元宏

最終頁に続く

(54) 【発明の名称】高レート、高並列性、および低エラーフロアのために、疑似巡回構成を使用し、パンクチャする L D P C 設計

(57) 【特許請求の範囲】

【請求項 1】

データ符号化の方法であって、
 1組の情報ビットを受信することと、
 コードワードを生成するために、リフトされた低密度パリティ検査(L D P C)符号化演算を前記1組の情報ビットに実行することと、
 前記L D P C符号化演算で使用される基礎L D P C符号の1つまたは複数のパンクチャされた基礎ビットに対応する前記コードワードのすべてのリフトされたビットをパンクチャすることと、ここにおいて、

前記1つまたは複数のパンクチャされた基礎ビットが、それぞれ前記基礎L D P C符号の1つまたは複数のパンクチャされた基礎ノードに対応するものであり、

前記基礎L D P C符号がノード対間の多重エッジを有しておらず、

ここにおいて、前記1つまたは複数のパンクチャされた基礎ノードが、前記基礎L D P C符号の少なくとも1つの最高次数の変数ノード、および、1つまたは複数の次数2の変数ノードを備え、前記1つまたは複数の次数2の変数ノードは、基礎L D P C符号から前記多重エッジを除去するために使用される、

を備える、方法。

【請求項 2】

前記1つまたは複数のパンクチャされた基礎ノードが、前記基礎L D P C符号の検査ノードの数に等しいかまたはそれより1少ない次数を有する1つまたは複数の変数ノードに

10

20

対応する、請求項 1 に記載の方法。

【請求項 3】

前記 1 つまたは複数のパンクチャされた基礎ノードが、各々別の変数ノードに接続されている 1 つまたは複数のそれぞれの検査ノードを分割し、その他の変数ノードの各々が、エッジによって、前記対応する分割された検査ノードの両方の要素に接続される、請求項 1 に記載の方法。

【請求項 4】

疑似巡回リフティングが前記基礎 L D P C 符号に適用され、前記疑似巡回リフティングのエッジクラスタの置換が巡回置換である、請求項 1 に記載の方法。

【請求項 5】

先行する請求項 1 ~ 4 のいずれか一項にしたがった方法をコンピュータに実施させるための命令を記録した、コンピュータ読み取り可能な記録媒体。

【請求項 6】

データ復号化の方法であって、

L D P C コードワードを受信することと、

基礎 L D P C 符号の 1 つまたは複数のパンクチャされたベースドビットに対応する前記 L D P C コードワードのすべてのリフトされたビットを識別することと、ここにおいて、前記 1 つまたは複数のパンクチャされた基礎ビットが、それぞれ前記基礎 L D P C 符号の 1 つまたは複数のパンクチャされた基礎ノードに対応するものであり、

前記基礎 L D P C 符号がノード対間の多重エッジを有していない、

1 組の情報ビットを回復するために前記受信されたコードワードに L D P C 復号化演算を実行することと、ここにおいて、前記識別されたリフトされたビットが、復号化の目的では消去されているとして扱われ、

ここにおいて、前記 1 つまたは複数のパンクチャされた基礎ノードが、前記基礎 L D P C 符号の少なくとも 1 つの最高次数の変数ノード、および、1 つまたは複数の次数 2 の変数ノードを備え、前記 1 つまたは複数の次数 2 の変数ノードは、基礎 L D P C 符号から前記多重エッジを除去するために使用される、

を備える、方法。

【請求項 7】

前記 1 つまたは複数のパンクチャされた基礎ノードが、前記基礎 L D P C 符号の検査ノードの数に等しいかまたはそれより 1 少ない次数を有する 1 つまたは複数の変数ノードに対応する、請求項 6 に記載の方法。

【請求項 8】

前記 1 つまたは複数のパンクチャされた基礎ノードが、各々別の変数ノードに接続されている 1 つまたは複数のそれぞれの検査ノードを分割し、その他の変数ノードの各々が、エッジによって、前記対応する分割された検査ノードの両方の要素に接続される、請求項 6 に記載の方法。

【請求項 9】

疑似巡回リフティングが前記基礎 L D P C 符号に適用され、前記疑似巡回リフティングのエッジクラスタの置換が巡回置換である、請求項 6 に記載の方法。

【請求項 10】

先行する請求項 6 ~ 9 のいずれか一項にしたがった方法をコンピュータに実施させるための命令を記録した、コンピュータ読み取り可能な記録媒体。

【請求項 11】

1 組の情報ビットを受信するための手段と、

コードワードを生成するために、前記 1 組の情報ビットに L D P C 符号化演算を実行するための手段と、

前記 L D P C 符号化演算で使用される基礎 L D P C 符号の 1 つまたは複数のパンクチャされた基礎ビットに対応する前記コードワードのすべてのリフトされたビットをパンクチャするための手段と、ここにおいて、

10

20

30

40

50

前記 1 つまたは複数のパンクチャされた基礎ビットが、それぞれ前記基礎 L D P C 符号の 1 つまたは複数のパンクチャされた基礎ノードに対応するものであり、

前記基礎 L D P C 符号がノード対間の多重エッジを有しておらず、

ここにおいて、前記 1 つまたは複数のパンクチャされた基礎ノードが、前記基礎 L D P C 符号の少なくとも 1 つの最高次数の変数ノード、および、1 つまたは複数の次数 2 の変数ノードを備え、前記 1 つまたは複数の次数 2 の変数ノードは、基礎 L D P C 符号から前記多重エッジを除去するために使用される、

を備える、符号器。

【請求項 1 2】

L D P C コードワードを受信するための手段と、

10

基礎 L D P C 符号の 1 つまたは複数のパンクチャされたベースドビットに対応する前記 L D P C コードワードのすべてのリフトされたビットを識別するための手段と、ここにおいて、

前記 1 つまたは複数のパンクチャされた基礎ビットが、それぞれ前記基礎 L D P C 符号の 1 つまたは複数のパンクチャされた基礎ノードに対応するものであり、

前記基礎 L D P C 符号がノード対間の多重エッジを有していない、

1 組の情報ビットを回復するために前記受信されたコードワードに L D P C 復号化演算を実行するための手段と、ここにおいて、前記識別されたリフトされたビットが、復号化の目的では消去されているとして扱われ、

ここにおいて、前記 1 つまたは複数のパンクチャされた基礎ノードが、前記基礎 L D P C 符号の少なくとも 1 つの最高次数の変数ノード、および、1 つまたは複数の次数 2 の変数ノードを備え、前記 1 つまたは複数の次数 2 の変数ノードは、基礎 L D P C 符号から前記多重エッジを除去するために使用される、

20

を備える、復号器。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

[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、L D P 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】

30

【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 (L D P C) 符号は、反復符号化システムを使用する特定のタイプの誤り訂正符号である L D P C 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。

40

【0 0 0 3】

[0003] たとえば、図 1 A は、例示的な L D P C 符号を表す二部グラフ 1 0 0 を示す。二部グラフ 1 0 0 は、4 つの検査ノード 1 2 0 (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 1 1 0 (円で表された) を含む。グラフ 1 0 0 のエッジは、変数ノード 1 1 0 を検査ノード 1 2 0 に接続する。図 1 B は、二部グラフ 1 0 0 の行列表現 1 5 0 を示す。行列表現 1 5 0 は、パリティ検査行列 H と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

50

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 $Z = 3$ による疑似巡回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。

【0004】

[0004]受信されたLDPCコードワードを復号して、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ことができる。誤りがない場合、または訂正可能な誤りの場合、復号化を使用して、符号化されたオリジナルのデータユニットを回復することができる。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。

【0005】

[0005]高速用途のために設計されたLDPC符号は、多くの場合、符号化および復号化演算における高並列性をサポートするために、大きいリフティング係数および比較的小さい基礎グラフをもつ疑似巡回構成を使用する。より高い符号化率(たとえば、コードワード長Nに対するメッセージ長Kの比)をもつLDPC符号は、比較的少ないパリティ検査を有する傾向がある。基礎パリティ検査の数が変数ノードの次数(たとえば、変数ノードに接続されたエッジの数)よりも少ない場合、基礎グラフにおいて、その変数ノードは、2つ以上のエッジによって基礎パリティ検査のうちの少なくとも1つに接続される(たとえば、変数ノードは「二重エッジ」を有することがある)。ベースド変数ノードおよび基礎検査ノードを2つ以上のエッジによって接続させるのは、一般に、並列ハードウェア実施目的では望ましくない。たとえば、そのような二重エッジは、同じメモリ場所に多数の同時読み取りおよび書き込み演算をもたらすことがあり、その結果として、データコヒーレンシ問題を引き起こすことがある。並列メッセージ更新のパイプライン処理は、二重エッジの存在によって悪影響を受けることがある。

【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]低密度パリティ検査(LDPC)コードワードの符号化および/または復号化を支援することができるデバイスおよび演算の方法が開示される。パンクチャされた変数ノード(状態変数ノードとしても知られている)の基礎グラフ設計への追加は、符号のレートパラメータ(k および n)を変更することなく、グラフにおける検査の数を1つだけ効果的に増加させることができることに留意されたい。いくつかの実施形態では、符号器は、1組の情報ビットを受信し、コードワードを生成するために、LDPC符号化演算を情報ビットに実行することができる。次に、デバイスは、LDPC符号化演算に使用されるリフトされたLDPC符号に基づいて、1つまたは複数の基礎変数ノードに対応する1組のリフトされたコードワードビットをパンクチャすることができ、パンクチャされたビッ

10

20

30

40

50

トは、それぞれ基礎 L D P C グラフの 1 つまたは複数のパンクチャされた基礎変数ノードに対応する。符号のグラフ式記述におけるパンクチャされた変数ノードは、検査ノードが、リフトされたパリティ検査行列に演算するプロセスを組み合わせることによって記述から除去され得ることが理解される。したがって、1 つまたは複数のパンクチャされた基礎ノードのうちの少なくとも 1 つは、パンクチャされた変数ノードの除去が多重エッジをもたらす場合、リフトされた L D P C 符号の基礎グラフのノード対間の多重エッジを除去することが理解される。

【 0 0 0 8 】

[0008] いくつかの実施形態では、1 つまたは複数のパンクチャされたノードは、L D P C 符号の検査ノードの数に等しいかまたはそれよりも 1 少ない次数を有する変数ノードを含むことができる。たとえば、パンクチャされたノードのうちの少なくとも 1 つは、L D P C 符号の最高次数の変数ノードとすることができる。そのような実施形態では、ノードの高い次数は、多くの場合、符号の性能の強化にとって望ましい。たとえば、パンクチュアリングにより、基礎グラフの二重エッジを避けながらより高い変数ノード次数が可能になる。グラフにパンクチャされた変数ノードが存在すると、普通なら、同じサイズおよびレートの符号の基礎グラフに存在することになる検査ノードの数が効果的に増加する。他の実施形態では、パンクチャされたノードのうちの少なくとも 1 つは、普通なら 2 つ以上のエッジによって L D P C 符号の変数ノードに接続されることになる検査ノードを分割するのに使用される次数 2 の変数ノードとすることができる。パンクチャされた次数 2 のノードは、それが接続される 2 つのパリティ検査を追加することによって記述から除去され得る。それにより、少なくとも 1 つのパンクチャされた基礎の次数 2 の変数ノードは、基礎 L D P C グラフの二重エッジを除去するために使用され得る。同様に、高い次数のパンクチャされたノードは、変数ノードの次数を 1 まで効果的に減少させるために、除去プロセスが制約ノードを合計することによってパリティ検査行列表現から除去され得る。次数 1 のパンクチャされたノードは、符号を変えることなく、その隣接する検査ノードとともにグラフから除去され得る。そのような除去プロセスは、表現に二重エッジまたは多重エッジを導入する可能性があり、それは、復号化の並列実施にとって望ましくない。

【 0 0 0 9 】

[0009] 基礎 L D P C グラフから二重（または多重）エッジを除去するかまたは減少させることによって、本実施形態は、L D P C 復号化演算を並列に実行するハードウェアの複雑さを減少させ、それによって、リフトされた L D P C 符号を実施する L D P C 復号器の処理効率を向上させることができる。これは、さらに、メモリで実行される読み出しおよび／または書き込み操作を簡単化し、読み出しおよび書き込み操作が、順序が乱れて、実行されないことを保証する。二重エッジを避けながら、より大きい変数ノード次数を可能にすることによって、本実施形態は、さらに、L D P C 符号化システムの誤り訂正性能を改善することができる。

【 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] いくつかの実施形態による L D P C 符号化演算を示す例示的な流れ図。

【 図 6 】 [0016] いくつかの実施形態による L D P C 復号化演算を示す例示的な流れ図。

【 図 7 】 [0017] レート $r = 2/3$ をもつ L D P C 符号に関する例示のパリティ検査行列を示す図。

10

20

30

40

50

【図8】[0018]レート $r = 13 / 15$ をもつLDPC符号に関連する例示のパリティ検査行列を示す図。

【図9】[0019]レート $r = 21 / 28$ をもつLDPC符号に関連する例示のパリティ検査行列を示す図。

【図10】[0020]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】

【0012】

[0021]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実践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態で示される。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ます、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。10

【0013】

[0022]図3は、いくつかの実施形態による通信システム300を示す。送信機310はチャネル320に信号を送信し、受信機330はチャネル320から信号を受信する。送信機310および受信機330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、チャネル320はワイヤレスである。他の実施形態では、チャネル320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。20

【0014】

[0023]通信システム300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、チャネル320の欠陥はチャネル歪みを導入することがあり、チャネル歪みは、直線歪み、マルチバス効果、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310および受信機330は、LDPC符号器と復号器とを含むことができる。詳細には、送信機310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330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310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。30

【0015】

[0024]「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複雑さを低減して、実施できるようになる。より詳細には、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、リフトされたLDPC符号は、基礎グラフのいくつか（Z個）の並列コピーを生成し、次に、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40

て発生され得る。リフトされた L D P C 符号のより詳細な議論は、たとえば、2008年3月17日に出版された、Tom Richardson および Ruediger Urbankel 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。

【0016】

[0025]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、L D P C 復号器は、 Z 個の処理要素を利用して、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ことができる。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ことができる。基礎グラフ中の二重エッジは、同じソフトビット値の並列読出しを、メモリ場所において、単一の並列パリティ検査更新の間に2回トリガがある。それにより、付加回路は、両方の更新を適切に組み込むために、メモリに戻して書き込まれるソフトビット値を組み合わせることが必要であることがある。基礎グラフ中の二重エッジを除去するのは、この余分な複雑さを避けるのに役立つ。10

【0017】

[0026]基礎 L D P C 符号から二重（および／または多重）エッジを除去するかまたは減少させることによって、パンクチュアリングは、並列検査ノードまたは変数ノード演算を実行するハードウェアの複雑さを減少させ、それによって、対応する L D P C 復号器の並列処理効率を向上させることができる。これは、さらに、メモリで実行される読出しあり／または書き込み操作を簡単化し、読出しあり／書き込み操作が、順序が乱れて、実行されないことを保証する。20

【0018】

[0027]パンクチュアリングは、コードワードからビットを除去して、より短いコードワードをもたらすための動作である。それにより、パンクチャされた変数ノードは、実際に送信されないコードワードビットに対応する。L D P C 符号の変数ノードをパンクチャすると、短縮された符号（たとえば、ビットの除去に起因して）が造り出されるとともに、さらに、検査ノードが効率的に除去される。詳細には、パンクチャされ得るビットを含む L D P C 符号の行列表現では、パンクチャされ得る変数ノードは1の次数を有している（30 符号が適切ならば、そのような表現が列組合せにより可能であり得る）場合、変数ノードのパンクチャリングは、符号から関連するビットを除去し、その単一の隣接する検査ノードをグラフから効率よく除去する。その結果として、グラフの検査ノードの数は1つだけ減少する。基礎送信ブロック長が $n - p$ であり、ここで、 p がパンクチャされた列の数であり、基礎パリティ検査の数が m である場合、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。 n および p を1だけ増加させた場合、 m を1だけ増加させ、レートおよびブロックサイズを変更しないまとまることでできるように留意されたい。

【0019】

[0028]一例として、基礎ブロック長が30であるレート0.9符号を考える。パンクチュアリングなしでは、基礎符号を定義するために使用されることになる検査ノードの数は3であり、これは、(27, 30)符号（たとえば、 $K = 27$ メッセージビット、 $N = 30$ コードワードビット）をもたらす。そのような符号は、少なくとも1つの検査ノードを変数ノードに接続する二重（またはより多くの）エッジを有する可能性がある（たとえば、すべての変数ノードが3の最大次数を有していない限り）。しかしながら、たとえば、深いエラーフロアを保証するにはより大きい次数の変数ノード（たとえば、次数4）を有することが望ましいことがある。パンクチャされた変数ノードが、L D P C 符号に導入され、それによって、変数ノードの総数を31に増加させる場合、基礎検査ノードの数は4に増加する。今では、基礎グラフに二重エッジなしに、次数4の基礎変数ノードを有することができる。しかしながら、そのような L D P C 符号は依然として(27, 30)4050

) 符号であることに留意されたい。

【0020】

[0029] L D P C 符号の二部グラフ表現において、パンクチャされた次数 2 の変数ノードは、その 2 つの隣接する検査ノードを単一の検査ノードに効率よくマージする。パンクチャされた次数 2 の変数ノードは、事実上、その 2 つの隣接する検査ノードが、次数 2 のノードがないので、同じパリティを有することを示す。それゆえに、パンクチャされた次数 2 の変数ノードを使用して、検査ノードを「分割する」ことができ、それによって、検査ノードの総数を増加させたように見える。したがって、この機構を使用して、L D P C 符号から多重エッジを除去することができる。一般に、変数ノードの次数が基礎グラフの検査ノードの総数 (N) よりも多い場合、変数ノードは 2 つ以上のエッジによって少なくとも 1 つの検査ノードに接続される。それにより、基礎グラフの多重エッジは、1 つまたは複数の次数 2 の変数ノードを導入することによって避けられ得るおよび / または除去され得る（すなわち、基礎グラフの少なくとも 1 つの変数ノードが N よりも大きい次数を有すると仮定して）。

【0021】

[0030] L D P C 符号の高い次数の基礎変数ノードをパンクチャすると、さらに、検査ノードの数を増加させることができる。加えて、高い次数の検査ノードが、高性能 L D P C 設計では望ましいことがある。たとえば、最高次数の変数ノードは、基礎グラフの検査ノードの総数に等しい（または 1 少ない）次数を有する変数ノードに対応することができる。そのような高い次数の変数ノードは、いかなる二重エッジもなしに基礎グラフに確かに存在することができる。以下でさらに詳細に説明するように、パンクチャされた変数ノードは、復号化において「消去されている」として扱われる。それにより、低い誤り率を目標とする符号では、そのようなノードが、エラーフロア事象を引き起こす組合せ構造（たとえば、トラッピングセット、またはニアコードワード）に参加しないようにすることが望ましいことがある。高い次数を有すると、一般に、ノードがエラーフロア事象の一因になる可能性が少なくなる。

【0022】

[0031] さらに、グラフ中の高い次数のパンクチャされた変数ノードは、符号の性能を改善することができる。グラフ中のパンクチャされたノードは、符号構造のいわゆる反復閾値を改善できることが知られている。標準の非正則 L D P C 設計（すなわち、パンクチャされた変数ノードのない）では、二部グラフの平均次数を増加させ、それにより、変数ノードおよび検査ノードの次数を増加させることによって、閾値は改善され得る。パンクチャされた変数ノードを用いて、同じ効果が、より低い平均次数で達成され、それによって、L D P C 符号の複雑さを低減することができる。さらに、より低い平均次数を有する L D P C 符号構造は、より小さいグラフでより良好に機能することができる。それにより、高次数の変数ノードをパンクチャすると、検査ノードの数を増加させる（それにより、より高い次数を可能にする）ことと、限られた最大変数ノード次数で符号の性能を改善することの両方が可能になる。

【0023】

[0032] 図 4 は、いくつかの実施形態による通信デバイス 400 のブロック図である。通信デバイス 400 は、符号器 410 と、復号器 420 と、通信チャネルを介して L D P C 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 430 とを含む。符号器 410 は、メモリ 412 と L D P C 符号器 414 とを含む。メモリ 412 は、L D P C 符号器 414 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。L D P C 符号器 414 は、別のデバイスに送信されるべき、L D P C 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 412 に記憶された情報ビットを処理する。

【0024】

[0033] いくつかの実施形態では、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ます。さらに、いくつかの実施形態では、基礎 L D P C 符号は、1 つまたは複数の

10

20

30

40

50

パンクチャされたノードを含むことができる。それにより、L D P C 符号器 4 1 4 は、基礎 L D P C 符号のそれぞれのパンクチャされたノードに対応するコードワードの 1 つまたは複数のビットをパンクチャすることができる。これらのパンクチャされたコードワード ビットはトランシーバ 4 3 0 によって送信されない。いくつかの実施形態では、パンクチャされたノードは、L D P C 符号の検査ノードの数に等しいかまたはそれよりも 1 少ない次数を有する基礎変数ノードを含むことができる。たとえば、パンクチャされたノードのうちの少なくとも 1 つは、L D P C 符号の最高次数の変数ノードとすることができます。他の実施形態では、パンクチャされたノードのうちの少なくとも 1 つは、2 つ以上のエッジによって L D P C 符号の変数ノードに接続されている検査ノードを分割するのに使用され得る。そのようなパンクチャされたノードは、リフトされた L D P C 符号に対して基礎グラフ中の二重エッジを除去するのに使用され得る。10

【 0 0 2 5 】

[0034]復号器 4 2 0 は、メモリ 4 2 2 と L D P C 復号器 4 2 4 とを含む。メモリ 4 2 2 は、トランシーバ 4 3 0 を介して受信され、L D P C 復号器 4 2 4 によって復号されるべきコードワードを記憶する。L D P C 復号器 4 2 4 は、メモリ 4 2 4 に記憶されたコードワードを、L D P C 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、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ます。さらに、いくつかの実施形態では、受信されたコードワードは、たとえば、対応する L D P C 符号の 1 組のパンクチャされたノードに基づいて決定されたような 1 つまたは複数のパンクチャビットを含むことができる。図 3 を参照しながら上述したように、パンクチャされたノードは、L D P C 符号の変数ノードの次数に基づいて決定され得る。それにより、L D P C 復号器 4 2 4 は、これらのパンクチャされたノードを復号化の目的では消去されてい20るとして扱うことができる。たとえば、L D P C 復号器 4 2 4 は、パンクチャされたノードの対数尤度比 (L L R) を初期化でゼロに設定することができる。

【 0 0 2 6 】

[0035]いくつかの実施形態では、L D P C 復号器 4 2 4 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、L D P C 復号器 4 2 4 は、いくつか (Z 個) の処理要素を利用して、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ことができる。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 4 2 2 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 4 2 2 に戻して書き込むことを含むことができる。基礎 L D P C 中の二重エッジは、同じソフトビット値メモリ場所の並列読出しを、単一の並列パリティ検査更新の間に 2 回トリガがある。それにより、付加回路は、一般に、両方の更新を適切に組み込むために、メモリに戻して書き込まれるソフトビット値を組み合わせることが必要である。しかしながら、たとえば図 3 に関して上述したように、L D P C 符号中の二重エッジを除去するのは、この余分な複雑さを避けるのに役立つ。30

【 0 0 2 7 】

[0036]図 5 は、いくつかの実施形態による L D P C 符号化演算 5 0 0 を示す例示的な流れ図である。たとえば図 4 を参照すると、符号器 4 1 0 は、最初に、符号化されるべき 1 組の情報ビットを受信する (5 1 0)。情報ビットは、通信チャネルまたはネットワークを通して別のデバイス (たとえば、受信デバイス) に送信されるように意図されたデータに対応することができる。たとえば、情報ビットは、中央処理装置 (C P U) から受信され、メモリ 4 1 2 に記憶され得る。40

【 0 0 2 8 】

[0037]次に、符号器 4 1 0 は、L D P C コードワードを生成するために情報ビットに L D P C 符号化演算を実行することができる (5 2 0)。いくつかの実施形態では、L D P C 符号器 4 1 4 は、符号器 4 1 0 および対応する復号器 (たとえば、受信デバイスの) に50

よって共有される L D P C 符号に基づいて、情報ビットを L D P C コードワードに符号化することができる。各コードワードは、オリジナルの情報ビットまたはその一部分ならびにオリジナル情報ビットにパリティ検査演算を実行するおよび / 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（たとえば、復号器によって）1組のパリティビットを含むことができる。

【0029】

符号器 410 は、L D P C 符号の基礎被パンクチャ変数ノード（base punctured variable node）に基づいて L D P C コードワードの 1つまたは複数のビットをさらにパンクチャすることができる（530）。たとえば、1つまたは複数のパンクチャされたコードワードビットは、それぞれ基礎 L D P C 符号の 1つまたは複数の基礎変数被パンクチャノードに対応することができる。詳細には、パンクチャされたノードのうちの少なくともいくつかは、リフトされた L D P C 符号の基礎グラフ中のノード対間の多重エッジを除去するように設定される。いくつかの実施形態では、パンクチャされたノードは、L D P C 符号の検査ノードの数に等しいかまたはそれよりも 1 少ない次数を有する変数ノードを含むことができる。他の実施形態では、パンクチャされたノードのうちの少なくとも 1 つは、次数 2 の変数ノードとすることができる。たとえば、次数 2 の変数ノードは、普通なら 2 つ以上のエッジによって L D P C 符号の別の変数ノードに接続されることになる検査ノードを分割するのに使用され得る。いくつかの実施形態では、両方が生じることがあり、詳細には、高い次数のパンクチャされた変数ノードと次数 2 のパンクチャされたノードの両方が基礎グラフに生じことがある。いくつかの実施形態では、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ます。その上さらに、L D P C 符号は、疑似巡回リフティングに基づくことができ、エッジクラスタの置換は巡回置換である。

【0030】

[0038] 図 6 は、いくつかの実施形態による L D P C 復号化演算 600 を示す例示的な流れ図である。たとえば図 4 を参照すると、復号器 420 は、最初に、復号されるべき L D P C コードワードを受信する（610）。L D P C コードワードは、送信デバイスから、たとえば、直交振幅変調（Q A M）データ信号の形態で受信され得る。それゆえに、L D P C コードワードは、デマップされた Q A M データ信号のラベル付けするビットのサブセットに対応することができる。

【0031】

[0039] 復号器 420 は、L D P C コードワードの 1 つまたは複数のパンクチャされたビットを、L D P C 符号の基礎被パンクチャノードに基づいて、識別することができる（620）。たとえば、1 つまたは複数のパンクチャされたコードワードビットは、それぞれ L D P C 符号の 1 つまたは複数の基礎被パンクチャノードに対応することができる。上述のように、基礎被パンクチャノードのうちの少なくともいくつかは、リフトされた L D P C 符号の基礎グラフ中のノード対間の多重エッジを除去するように設定される。いくつかの実施形態では、パンクチャされたノードは、L D P C 符号の検査ノードの数に等しいかまたはそれよりも 1 少ない次数を有する変数ノードを含むことができる。他の実施形態では、パンクチャされたノードのうちの少なくとも 1 つは、次数 2 の変数ノードとすることができる。上述のように、次数 2 の変数ノードは、普通なら 2 つ以上のエッジによって L D P C 符号の別の変数ノードに接続されることになる検査ノードを分割するのに使用され得る。いくつかの実施形態では、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ます（たとえば、疑似巡回リフティングに基づいて）。

【0032】

[0040] 次に、復号器 420 は、受信されたコードワードに L D P C 復号化演算を実行して、オリジナルの情報ビットを回復することができる（630）。たとえば、L D P C 復号器 424 は、L D P C 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するように試みることによってコードワードを処理することができる。いくつかの実施形態では、L D P C 復号器 424 は、パン

10

20

30

40

50

クチャされたコードワードビットを、たとえばパンクチャされたノードの L L R を初期化でゼロに設定することによって、復号化演算の間消去されているとして扱うことができる。

【 0 0 3 3 】

[0041]本実施形態では、L D P C 符号の各々は、サイズ $Z \times n$ の 2 次元 2 値アレイと見られ得る、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、提案す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、1024 Q A M では、 $k = 5$ ）。さらに、 k は 360 の約数（factor）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。10

【 0 0 3 4 】

[0042]図 7、図 8、および図 9 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 700、800、および 900 を示す。パリティ検査行列 700、800、および 900 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 3 5 】

[0043]それぞれ、レート $r = 27 / 30$ および $r = 13 / 15$ をもつ L D P C 符号に関連するパリティ検査行列 700 および 800 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 L D P C 符号に関連するパリティ検査行列 900 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。20

【 0 0 3 6 】

[0044]さらに、パリティ検査行列 700 は、パンクチャされた次数 2 の変数ノード（指標 0）を有する。そのようなノードは、単一のパリティ検査を 2 つに分割することができる。これは、基礎行列が二重エッジを有しておらず、本明細書で説明する実施形態のうちのいくつかを容易にすることを保証する。等価符号表示は、2 つのパリティ検査をマージし、パンクチャされた次数 2 の変数ノードを除去することによって構築され得る。その上、そのような等価表示は、基礎グラフに二重エッジまたは多重エッジを有するであろう。30

【 0 0 3 7 】

[0045]図 10 は、いくつかの実施形態による通信デバイス 1000 のブロック図である。通信デバイス 1000 は、トランシーバ 1010 と、プロセッサ 1020 と、メモリ 1030 とを含む。トランシーバ 1010 は、通信デバイス 1000 へのおよび / または通信デバイス 1000 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 1010 は、通信デバイス 1000 と C P U 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 1010 は、通信デバイス 1000 とネットワーク中の別の通信デバイスとの間で L D P C 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。40

【 0 0 3 8 】

[0046]メモリ 1030 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ローカルキャッシュとして使用され得るデータストア 1032 を含むことができる。さらに、メモリ 1030 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な記憶媒体（たとえば、E P R O M、E E P R O M（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 3 9 】

・コードワードを生成するために、L D P C 符号を使用して、1 組の情報ビットを符号化するための L D P C 符号化モジュール 1034、50

・ 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の L D P C 復号化モジュール 1 0 3 6。

【 0 0 4 0 】

各ソフトウェアモジュールには、プロセッサ 1 0 2 0 によって実行されると、対応する機能を符号器 1 0 0 0 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 1 0 3 0 の非一時的コンピュータ可読記憶媒体は、図 5 ~ 図 6 に関して上述した演算のすべてまたは一部を実行するための命令を含むことができる。モジュール 1 0 3 4 ~ 1 0 3 6 がメモリ 1 0 3 0 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

【 0 0 4 1 】

[0047] 符号器インターフェース 1 0 1 0 とメモリ 1 0 3 0 との間に結合されるプロセッサ 1 0 2 0 は、復号器 1 0 0 0 に(たとえば、メモリ 1 0 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ます。たとえば、プロセッサ 1 0 2 0 は、L D P C 符号化モジュール 1 0 3 4 および / または L D P C 復号化モジュール 1 0 3 6 を実行することができます。

【 0 0 4 2 】

[0048] L D P C 符号化モジュール 1 0 3 4 は、プロセッサ 1 0 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ます。たとえば、プロセッサ 1 0 2 0 は、L D P C 符号化モジュール 1 0 3 4 を実行する際に、L D P C 符号化モジュール 1 0 3 4 、および対応する受信デバイスの復号化モジュール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ます。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。いくつかの実施形態では、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ます(たとえば、疑似巡回リフティングに基づいて)。

20

【 0 0 4 3 】

[0049] プロセッサ 1 0 2 0 は、L D P C 符号化モジュール 1 0 3 4 を実行する際に、コードワードの 1 つまたは複数のビットを、対応する L D P C 符号に基づいてさらにパンクチャすることができます。たとえば、1 つまたは複数のパンクチャされたコードワードビットは、それぞれ L D P C 符号の 1 つまたは複数のパンクチャされたノードに対応することができます。上述のように、パンクチャされたノードのうちの少なくともいくつかは、リフトされた L D P C 符号の基礎グラフ中のノード対間の多重エッジを除去するように設定される。いくつかの実施形態では、パンクチャされたノードは、L D P C 符号の検査ノードの数に等しいかまたはそれよりも 1 少ない次数を有する変数ノードを含むことができる。他の実施形態では、パンクチャされたノードのうちの少なくとも 1 つは、次数 2 の変数ノード(たとえば、普通なら 2 つ以上のエッジによって L D P C 符号の別の変数ノードに接続されることになる検査ノードを分割するのに使用される)とすることができます。

30

【 0 0 4 4 】

[0050] L D P C 復号化モジュール 1 0 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 0 2 0 によって実行され得る。いくつかの実施形態では、プロセッサ 1 0 2 0 は、L D P C 復号化モジュール 1 0 3 6 を実行する際に、最初に、受信されたコードワードの 1 つまたは複数のパンクチャされたビットを L D P C 符号に基づいて識別することができます。次に、プロセッサ 1 0 2 0 は、受信されたコードワードに L D P C 復号化演算を実行するとともに、パンクチャされたコードワードビットを消去されているとして扱うことができる。たとえば、L D P C 復号化モジュール 1 0 3 6 は、プロセッサ 1 0 2 0 によって実行されるとき、パンクチャされたノードの L L R を初期化でゼロに設定することができる。いくつかの実施形態では、L D P C 符号は、リフトさ

40

50

れた L D P C 符号とすることができる（たとえば、疑似巡回リフティングに基づいて）。

【 0 0 4 5 】

[0051] 上述のように、パンクチャされたコードワードビットは、L D P C 符号のそれぞれのパンクチャされたノードに対応することができ、パンクチャされたノードのうちの少なくともいくつかは、リフトされた L D P C 符号の基礎グラフ中のノード対間の多重エッジを除去するように設定される。いくつかの実施形態では、パンクチャされたノードは、L D P C 符号の検査ノードの数に等しいかまたはそれよりも 1 少ない次数を有する変数ノードを含むことができる。他の実施形態では、パンクチャされたノードのうちの少なくとも 1 つは、次数 2 の変数ノード（たとえば、普通なら 2 つ以上のエッジによって L D P C 符号の別の変数ノードに接続されることになる検査ノードを分割するのに使用される）と 10 することができる。

【 0 0 4 6 】

[0052] 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。それゆえに、明細書および図面は、限定の意味ではなく例示の意味と見なされるべきである。たとえば、図 5～図 6 の流れ図に示された方法ステップは、他の好適な順序で実行されてもよく、多数のステップは単一のステップに組み合わされてもよく、および / またはいくつかのステップは省略されてもよい。

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。

20

【 C 1 】

データ符号化の方法であって、

1 組の情報ビットを受信することと、

コードワードを生成するために、リフトされた低密度パリティ検査（L D P C）符号化演算を前記 1 組の情報ビットに実行することと、

前記 L D P C 符号化演算で使用される基礎 L D P C 符号の 1 つまたは複数のパンクチャされた基礎ビットに対応する前記コードワードのすべてのリフトされたビットをパンクチャすることと、ここにおいて、

前記 1 つまたは複数のパンクチャされた基礎ビットが、それぞれ前記基礎 L D P C 符号の 1 つまたは複数のパンクチャされた基礎ノードに対応するものであり、

前記基礎 L D P C 符号が多重エッジを有していない、
を備える、方法。

30

【 C 2 】

前記 1 つまたは複数のパンクチャされた基礎ノードが、前記基礎 L D P C 符号の検査ノードの数に等しいかまたはそれより 1 少ない次数を有する 1 つまたは複数の変数ノードに對応する、C 1 に記載の方法。

【 C 3 】

前記 1 つまたは複数のパンクチャされた基礎ノードのうちの少なくとも 1 つが、前記基礎 L D P C 符号の最高次数の変数ノードに對応する、C 1 に記載の方法。

40

【 C 4 】

前記 1 つまたは複数のパンクチャされた基礎ノードが、1 つまたは複数の次数 2 の変数ノードに對応する、C 1 に記載の方法。

【 C 5 】

前記 1 つまたは複数のパンクチャされた基礎ノードが、各々別の変数ノードに接続されている 1 つまたは複数のそれぞれの検査ノードを分割し、他の変数ノードの各々が、エッジによって、前記対応する分割された検査ノードの両方の要素に接続される、C 4 に記載の方法。

【 C 6 】

前記 1 つまたは複数のパンクチャされた基礎ノードが、前記基礎 L D P C 符号における二重エッジを除去する、C 4 に記載の方法。

50

[C 7]

疑似巡回リフティングが前記基礎 L D P C 符号に適用され、前記疑似巡回リフティングのエッジクラスタの置換が巡回置換である、C 1 に記載の方法。

[C 8]

データ復号化の方法であって、

L D P C コードワードを受信することと、

基礎 L D P C 符号の 1 つまたは複数のパンクチャされたベースドビットに対応する前記 L D P C コードワードのすべてのリフトされたビットを識別することと、ここにおいて、

前記 1 つまたは複数のパンクチャされた基礎ビットが、それぞれ前記基礎 L D P C 符号の 1 つまたは複数のパンクチャされた基礎ノードに対応するものであり、

前記基礎 L D P C 符号が多重エッジを有していない、

1 組の情報ビットを回復するために前記受信されたコードワードに L D P C 復号化演算を実行することと、ここにおいて、前記識別されたリフト済みビットが、復号化の目的では消去されているとして扱われる、

を備える、方法。

[C 9]

前記 1 つまたは複数のパンクチャされた基礎ノードが、前記基礎 L D P C 符号の検査ノードの数に等しいかまたはそれより 1 少ない次数を有する 1 つまたは複数の変数ノードに対応する、C 8 に記載の方法。

[C 10]

前記 1 つまたは複数のパンクチャされた基礎ノードのうちの少なくとも 1 つが、前記基礎 L D P C 符号の最高次数の変数ノードに対応する、C 8 に記載の方法。

[C 11]

前記 1 つまたは複数のパンクチャされた基礎ノードが、1 つまたは複数の次数 2 の変数ノードに対応する、C 8 に記載の方法。

[C 12]

前記 1 つまたは複数のパンクチャされた基礎ノードが、各々別の変数ノードに接続されている 1 つまたは複数のそれぞれの検査ノードを分割し、他の変数ノードの各々が、エッジによって、前記対応する分割された検査ノードの両方の要素に接続される、C 11 に記載の方法。

[C 13]

前記 1 つまたは複数のパンクチャされた基礎ノードが、前記基礎 L D P C 符号における二重エッジを除去する、C 11 に記載の方法。

[C 14]

疑似巡回リフティングが前記基礎 L D P C 符号に適用され、前記疑似巡回リフティングのエッジクラスタの置換が巡回置換である、C 8 に記載の方法。

[C 15]

プログラム命令を含む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、

コードワードを生成するために、リフトされた L D P C 符号化演算を前記 1 組の情報ビットに実行し、

前記 L D P C 符号化演算で使用される基礎 L D P C 符号の 1 つまたは複数のパンクチャされた基礎ビットに対応する前記コードワードのすべてのリフトされたビットをパンクチャする、ここにおいて、

前記 1 つまたは複数のパンクチャされた基礎ビットが、それぞれ前記基礎 L D P C 符号の 1 つまたは複数のパンクチャされた基礎ノードに対応するものであり、

前記基礎 L D P C 符号が多重エッジを有していない、コンピュータ可読記憶媒体。

[C 16]

10

20

30

40

50

前記 1 つまたは複数のパンクチャされた基礎ノードが、前記基礎 L D P C 符号の検査ノードの数に等しいかまたはそれより 1 少ない次数を有する 1 つまたは複数の変数ノードに対応する、C 1 5 に記載のコンピュータ可読記憶媒体。

[C 1 7]

前記 1 つまたは複数のパンクチャされた基礎ノードのうちの少なくとも 1 つが、前記基礎 L D P C 符号の最高次数の変数ノードに対応する、C 1 5 に記載のコンピュータ可読記憶媒体。

[C 1 8]

前記 1 つまたは複数のパンクチャされた基礎ノードが、1 つまたは複数の次数 2 の変数ノードに対応する、C 1 5 に記載のコンピュータ可読記憶媒体。

10

[C 1 9]

前記 1 つまたは複数のパンクチャされた基礎ノードが、各々別の変数ノードに接続されている 1 つまたは複数のそれぞれの検査ノードを分割し、他の変数ノードの各々が、エッジによって、前記対応する分割された検査ノードの両方の要素に接続される、C 1 8 に記載のコンピュータ可読記憶媒体。

[C 2 0]

前記 1 つまたは複数のパンクチャされた基礎ノードが、前記基礎 L D P C 符号における二重エッジを除去する、C 1 8 に記載のコンピュータ可読記憶媒体。

[C 2 1]

疑似巡回リフティングが前記基礎 L D P C 符号に適用され、前記疑似巡回リフティングのエッジクラスタの置換が巡回置換である、C 1 5 に記載のコンピュータ可読記憶媒体。

20

[C 2 2]

プログラム命令を含む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、

L D P C コードワードを受信し、

基礎 L D P C 符号の 1 つまたは複数のパンクチャされたベースドビットに対応する前記 L D P C コードワードのすべてのリフトされたビットを識別し、ここにおいて、

前記 1 つまたは複数のパンクチャされた基礎ビットが、それぞれ前記基礎 L D P C 符号の 1 つまたは複数のパンクチャされた基礎ノードに対応するものであり、

30

前記基礎 L D P C 符号が多重エッジを有していない、

1 組の情報ビットを回復するために前記受信されたコードワードに L D P C 復号化演算を実行する、ここにおいて、前記識別されたリフト済みビットが、復号化の目的では消去されているとして扱われる、コンピュータ可読記憶媒体。

[C 2 3]

前記 1 つまたは複数のパンクチャされた基礎ノードが、前記基礎 L D P C 符号の検査ノードの数に等しいかまたはそれより 1 少ない次数を有する 1 つまたは複数の変数ノードに対応する、C 2 2 に記載のコンピュータ可読記憶媒体。

[C 2 4]

前記 1 つまたは複数のパンクチャされた基礎ノードのうちの少なくとも 1 つが、前記基礎 L D P C 符号の最高次数の変数ノードに対応する、C 2 2 に記載のコンピュータ可読記憶媒体。

40

[C 2 5]

前記 1 つまたは複数のパンクチャされた基礎ノードが、1 つまたは複数の次数 2 の変数ノードに対応する、C 2 2 に記載のコンピュータ可読記憶媒体。

[C 2 6]

前記 1 つまたは複数のパンクチャされた基礎ノードが、各々別の変数ノードに接続されている 1 つまたは複数のそれぞれの検査ノードを分割し、他の変数ノードの各々が、エッジによって、前記対応する分割された検査ノードの両方の要素に接続される、C 2 5 に記載のコンピュータ可読記憶媒体。

50

[C 27]

前記 1つまたは複数のパンクチャされた基礎ノードが、前記基礎 L D P C 符号における二重エッジを除去する、C 25 に記載のコンピュータ可読記憶媒体。

[C 28]

疑似巡回リフティングが前記基礎 L D P C 符号に適用され、前記疑似巡回リフティングのエッジクラスタの置換が巡回置換である、C 22 に記載のコンピュータ可読記憶媒体。

[C 29]

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、
コードワードを生成するために、リフトされた L D P C 符号化演算を前記 1組の情報
ビットに実行し、 10

前記 L D P C 符号化演算で使用される基礎 L D P C 符号の 1つまたは複数のパンクチャされた基礎ビットに対応する前記コードワードのすべてのリフトされたビットをパンクチャする、ここにおいて、

前記 1つまたは複数のパンクチャされた基礎ビットが、それぞれ前記基礎 L D P C 符号の 1つまたは複数のパンクチャされた基礎ノードに対応するものであり、

前記基礎 L D P C 符号が多重エッジを有していない、
を備える通信デバイス。

[C 30]

前記 1つまたは複数のパンクチャされた基礎ノードが、前記基礎 L D P C 符号の検査ノードの数に等しいかまたはそれより 1 少ない次数を有する 1つまたは複数の変数ノードに対応する、C 29 に記載のデバイス。 20

[C 31]

前記 1つまたは複数のパンクチャされた基礎ノードのうちの少なくとも 1つが、前記基礎 L D P C 符号の最高次数の変数ノードに対応する、C 29 に記載のデバイス。

[C 32]

前記 1つまたは複数のパンクチャされた基礎ノードが、1つまたは複数の次数 2 の変数ノードに対応する、C 29 に記載のデバイス。

[C 33]

前記 1つまたは複数のパンクチャされた基礎ノードが、各々別の変数ノードに接続されている 1つまたは複数のそれぞれの検査ノードを分割し、他の変数ノードの各々が、エッジによって、前記対応する分割された検査ノードの両方の要素に接続される、C 32 に記載のデバイス。 30

[C 34]

前記 1つまたは複数のパンクチャされた基礎ノードが、前記基礎 L D P C 符号における二重エッジを除去する、C 32 に記載のデバイス。

[C 35]

疑似巡回リフティングが前記基礎 L D P C 符号に適用され、前記疑似巡回リフティングのエッジクラスタの置換が巡回置換である、C 29 に記載のデバイス。

[C 36]

L D P C コードワードを記憶するメモリと、
復号器と、ここにおいて、前記復号器が、
基礎 L D P C 符号の 1つまたは複数のパンクチャされたベースドビットに対応する前記 L D P C コードワードのすべてのリフトされたビットを識別し、ここにおいて、 40

前記 1つまたは複数のパンクチャされた基礎ビットが、それぞれ前記基礎 L D P C 符号の 1つまたは複数のパンクチャされた基礎ノードに対応するものであり、

前記基礎 L D P C 符号が多重エッジを有していない、

1組の情報ビットを回復するために前記受信されたコードワードに L D P C 復号化演算を実行する、ここにおいて、前記識別されたリフト済みビットが、復号化の目的では消去されているとして扱われる、 50

を備える通信デバイス。

[C 3 7]

前記 1 つまたは複数のパンクチャされた基礎ノードが、前記基礎 L D P C 符号の検査ノードの数に等しいかまたはそれより 1 少ない次数を有する 1 つまたは複数の変数ノードに対応する、C 3 6 に記載のデバイス。

[C 3 8]

前記 1 つまたは複数のパンクチャされた基礎ノードのうちの少なくとも 1 つが、前記基礎 L D P C 符号の最高次数の変数ノードに対応する、C 3 6 に記載のデバイス。

[C 3 9]

前記 1 つまたは複数のパンクチャされた基礎ノードが、1 つまたは複数の次数 2 の変数ノードに対応する、C 3 6 に記載のデバイス。

[C 4 0]

前記 1 つまたは複数のパンクチャされた基礎ノードが、各々別の変数ノードに接続されている 1 つまたは複数のそれぞれの検査ノードを分割し、他の変数ノードの各々が、エッジによって、前記対応する分割された検査ノードの両方の要素に接続される、C 3 9 に記載のデバイス。

[C 4 1]

前記 1 つまたは複数のパンクチャされた基礎ノードが、前記基礎 L D P C 符号における二重エッジを除去する、C 3 9 に記載のデバイス。

[C 4 2]

疑似巡回リフティングが前記基礎 L D P C 符号に適用され、前記疑似巡回リフティングのエッジクラスタの置換が巡回置換である、C 3 6 に記載のデバイス。

[C 4 3]

1 組の情報ビットを受信するための手段と、
コードワードを生成するために、前記 1 組の情報ビットに L D P C 符号化演算を実行するための手段と、

前記 L D P C 符号化演算で使用される基礎 L D P C 符号の 1 つまたは複数のパンクチャされた基礎ビットに対応する前記コードワードのすべてのリフトされたビットをパンクチャするための手段と、ここにおいて、

前記 1 つまたは複数のパンクチャされた基礎ビットが、それぞれ前記基礎 L D P C 符号の 1 つまたは複数のパンクチャされた基礎ノードに対応するものであり、

前記基礎 L D P C 符号が多重エッジを有していない、
を備える符号器。

[C 4 4]

L D P C コードワードを受信するための手段と、
基礎 L D P C 符号の 1 つまたは複数のパンクチャされたベースドビットに対応する前記 L D P C コードワードのすべてのリフトされたビットを識別するための手段と、ここにおいて、

前記 1 つまたは複数のパンクチャされた基礎ビットが、それぞれ前記基礎 L D P C 符号の 1 つまたは複数のパンクチャされた基礎ノードに対応するものであり、

前記基礎 L D P C 符号が多重エッジを有していない、
1 組の情報ビットを回復するために前記受信されたコードワードに L D P C 復号化演算を実行するための手段と、ここにおいて、前記識別されたリフト済みビットが、復号化の目的では消去されているとして扱われる、
を備える復号器。

10

20

30

40

【図 1 A】

図 1A

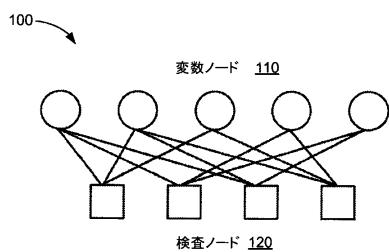


FIG. 1A

【図 2】

図 2

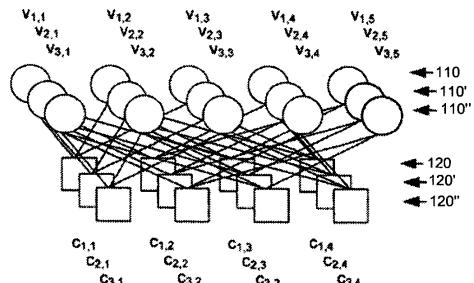


FIG. 2

【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 3】

図 3

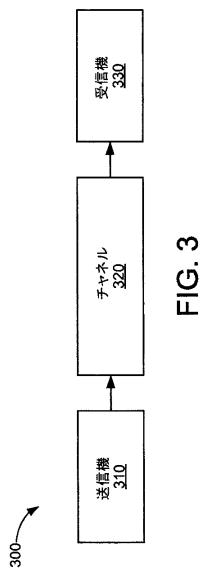


FIG. 3

【図 4】

図 4

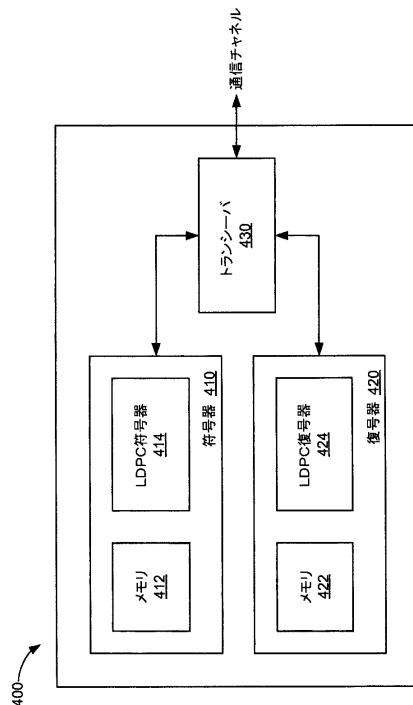


FIG. 4

【図5】

図5

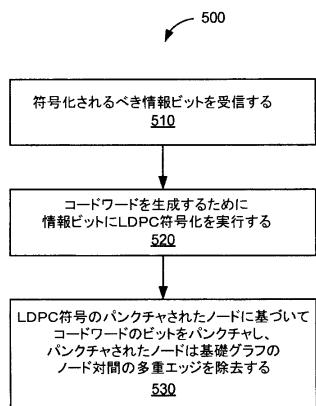


FIG. 5

【図6】

図6



FIG. 6

【図7】

図7

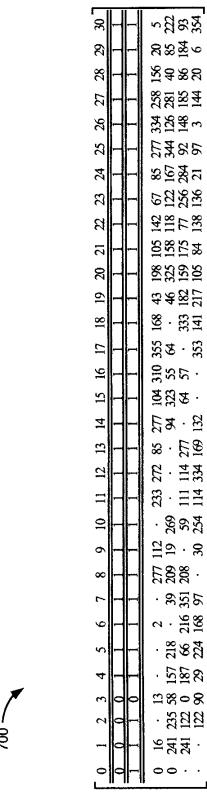


FIG. 7

【図8】

図8

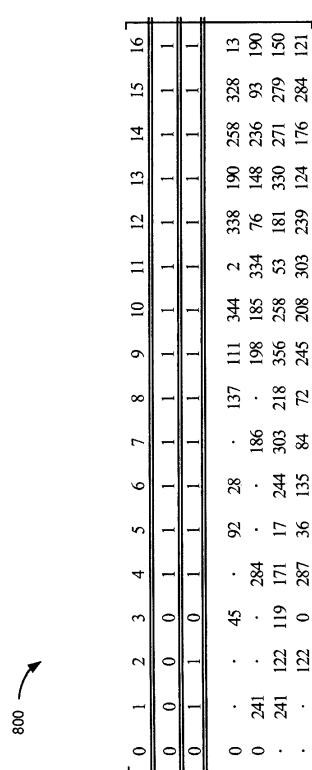


FIG. 8

【図9】

图 9

FIG. 9

【図10】

図 10

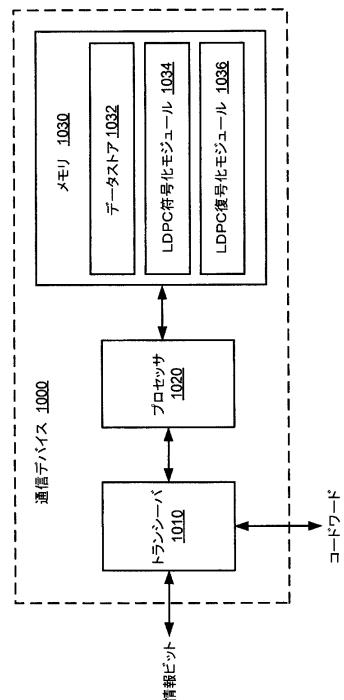


FIG. 10

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)

(72)発明者 リチャードソン、トマス・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 大野 友輝

(56)参考文献 特表2004-531972(JP,A)

米国特許出願公開第2008/0270872(US,A1)

米国特許出願公開第2008/0155385(US,A1)

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