

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5593312号  
(P5593312)

(45) 発行日 平成26年9月24日 (2014. 9. 24)

(24) 登録日 平成26年8月8日 (2014. 8. 8)

(51) Int. Cl.

F I

H03M 13/19 (2006.01)

H03M 13/19

請求項の数 46 (全 33 頁)

(21) 出願番号	特願2011-515675 (P2011-515675)	(73) 特許権者	506054914
(86) (22) 出願日	平成21年5月26日 (2009. 5. 26)		ラマト アット テル アビブ ユニバー
(65) 公表番号	特表2011-525771 (P2011-525771A)		シティ リミテッド
(43) 公表日	平成23年9月22日 (2011. 9. 22)		イスラエル、61392 テル アビブ、
(86) 国際出願番号	PCT/IB2009/052204		ビー. オー. ボックス 39296
(87) 国際公開番号	W02009/156883	(74) 代理人	100092897
(87) 国際公開日	平成21年12月30日 (2009. 12. 30)		弁理士 大西 正悟
審査請求日	平成24年5月14日 (2012. 5. 14)	(72) 発明者	シャロン、エラン
(31) 優先権主張番号	61/074, 701		イスラエル国、75493 リション レ
(32) 優先日	平成20年6月23日 (2008. 6. 23)		ジョン、ハダガン ストリート 16/1
(33) 優先権主張国	米国 (US)	(72) 発明者	アルロド、イダン
			イスラエル国、46643 ヘルツェリヤ
			、ハーシェル ストリート 70

最終頁に続く

(54) 【発明の名称】 デコーダのリセットによるLDPCトラッピング集合の克服

(57) 【特許請求の範囲】

【請求項 1】

K 個の情報ビットを  $N (> K)$  個の符号ワードビットとして符号化する符号ワードの表現を復号化する方法であって、

(a) チャンネルから前記符号ワードの前記表現をインポートすること、

(b) N 個のビットノードおよび  $N - K$  個のチェックノードを含むグラフにおける前記ビットノードと前記チェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新することおよび、

(c) (i) 所定の失敗基準に照らして前記復号化が収束に失敗し、且つ

(ii) 前記符号ワードビットの推定値が、前記グラフにトラッピング集合が含まれている兆候を示す基準を満たしている場合、前記反復を続ける前に前記メッセージの少なくとも一部をリセットすることを含む方法。

10

【請求項 2】

(d) 前記グラフの少なくとも一部を複数の部分グラフに分割することを更に含み、

前記メッセージの交換の少なくとも一部が各部分グラフ内で別個に実行され、

前記グラフにトラッピング集合が含まれる兆候を示す前記基準が、前記部分グラフのうち 1 個だけで復号化の収束が失敗することを含む、請求項 1 に記載の方法。

【請求項 3】

前記グラフにトラッピング集合が含まれている兆候を示す前記基準が、前記推定値のシンδροームの要素の 1 パーセント以下 が非ゼロであり、2 個の連続的な反復において一定

20

であることを含む、請求項 1 に記載の方法。

【請求項 4】

前記リセットが、前記チェックノードから送られるメッセージの少なくとも一部をゼロに設定することを含む、請求項 1 に記載の方法。

【請求項 5】

前記リセットが、前記チェックノードから送られる全てのメッセージをゼロに設定することを含む、請求項 4 に記載の方法。

【請求項 6】

前記リセットが、前記ビットノードから送られるメッセージの少なくとも一部を切り捨てることを含む、請求項 1 に記載の方法。

10

【請求項 7】

前記リセットが、前記ビットノードから送られる全てのメッセージを切り捨てることを含む、請求項 6 に記載の方法。

【請求項 8】

前記メッセージが対数尤度比であり、前記切り捨てが 10 ~ 16 の間の値である、請求項 6 に記載の方法。

【請求項 9】

K 個の情報ビットを N ( $> K$ ) 個の符号ワードビットとして符号化する符号ワードの表現を復号化する方法であって、

(a) チャンネルから前記符号ワードの前記表現をインポートすること、

20

(b) N 個のビットノードおよび N - K 個のチェックノードを含むグラフにおける前記ビットノードと前記チェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新することおよび、

(c) 所定の失敗基準に照らして前記復号化が収束に失敗した場合、前記反復を続ける前に前記ビットノードから送られた前記メッセージの少なくとも一部を切り捨てることを含む方法。

【請求項 10】

前記所定の失敗基準が、前記推定値のシンδροームの要素の少なくとも所定個数が非ゼロである、請求項 9 に記載の方法。

【請求項 11】

30

前記所定個数が 1 である、請求項 10 に記載の方法。

【請求項 12】

前記所定の失敗基準が、前記反復の所定回数実行後に、前記シンδροームの要素の少なくとも所定個数が非ゼロである、請求項 10 に記載の方法。

【請求項 13】

前記所定の失敗基準が、所定時間経過後に、前記シンδροームの要素の少なくとも所定個数が非ゼロである、請求項 10 に記載の方法。

【請求項 14】

前記所定の失敗基準が、前記メッセージの所定回数の交換後に、前記シンδροームの要素の少なくとも所定個数が非ゼロである、請求項 10 に記載の方法。

40

【請求項 15】

前記所定の失敗基準が、連続的な 2 回の反復において、前記シンδροームの要素の高々所定個数が非ゼロのままである、請求項 9 に記載の方法。

【請求項 16】

前記所定の失敗基準が、連続的な 2 回の反復実行後に、非ゼロである前記シンδροームの要素の個数の差が所定の限度を下回る、請求項 9 に記載の方法。

【請求項 17】

前記所定の失敗基準が、所定回数の連続的な反復実行の前後で、前記推定値間で所定の限度を下回るハミング距離を含む、請求項 9 に記載の方法。

【請求項 18】

50

連続的な反復の前記所定回数が1である、請求項17に記載の方法。

【請求項19】

前記ビットノードから送られた全てのメッセージが切り捨てられる、請求項9に記載の方法。

【請求項20】

前記メッセージが対数尤度比であり、前記切り捨てが10～16の間の値である、請求項9に記載の方法。

【請求項21】

K個の情報ビットを $N (> K)$ 個の符号ワードビットとして符号化する符号ワードの表現を復号化するデコーダであって、

10

(a) N個のビットノードおよび $N - K$ 個のチェックノードを含むグラフにおける前記ビットノードと前記チェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新することおよび、

(b) (i) 所定の失敗基準に照らして前記復号化が収束に失敗し、且つ

(ii) 前記符号ワードビットの推定値が、前記グラフにトラッピング集合が含まれている兆候を示す基準を満たしている場合、前記反復を続ける前に前記メッセージの少なくとも一部をリセットすること、を含むステップにより、前記符号ワードの推定値を更新するアルゴリズムを実行することにより前記符号ワードの前記表現を復号化するプロセッサを含むデコーダ。

20

【請求項22】

K個の情報ビットを $N (> K)$ 個の符号ワードビットとして符号化する符号ワードの表現を復号化するデコーダであって、

(a) N個のビットノードおよび $N - K$ 個のチェックノードを含むグラフにおける前記ビットノードと前記チェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新することおよび、

(b) 所定の失敗基準に照らして前記復号化が収束に失敗した場合、前記反復を続ける前に前記ビットノードから送られた前記メッセージの少なくとも一部を切り捨てること、を含むステップにより、前記符号ワードの推定値を更新するアルゴリズムを実行することにより前記符号ワードの前記表現を復号化するプロセッサを含むデコーダ。

30

【請求項23】

(a) K個の情報ビットを $N (> K)$ 個の符号ワードビットの符号ワードとして符号化するエンコーダと、

(b) (i) N個のビットノードおよび $N - K$ 個のチェックノードを含むグラフにおける前記ビットノードと前記チェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新することおよび、

(ii) (A) 所定の失敗基準に照らして前記復号化が収束に失敗し、且つ

(B) 前記符号ワードビットの推定値が、前記グラフにトラッピング集合が含まれている兆候を示す基準を満たしている場合、前記反復を続ける前に前記メッセージの少なくとも一部をリセットすること、を含むステップにより、前記符号ワードの推定値を更新するアルゴリズムを実行することにより前記符号ワードの前記表現を復号化するプロセッサを含むデコーダとを含むメモリコントローラ。

40

【請求項24】

(c) 前記符号ワードの少なくとも一部を主メモリに保存し、且つ前記符号ワードの前記少なくとも一部の表現を前記主メモリから取得する回路を更に含む、請求項23に記載のメモリコントローラ。

【請求項25】

(a) 請求項24に記載の前記メモリコントローラと、

(b) 前記主メモリとを含むメモリ装置。

【請求項26】

(a) K個の情報ビットを $N (> K)$ 個の符号ワードビットの符号ワードとして符号化

50

するエンコーダと、

(b)(i) N個のビットノードおよびN - K個のチェックノードを含むグラフにおける前記ビットノードと前記チェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新することおよび、

(ii) 所定の失敗基準に照らして前記復号化が収束に失敗した場合、前記反復を続ける前に前記ビットノードから送られた前記メッセージの少なくとも一部を切り捨てること、を含むステップにより、前記符号ワードの推定値を更新するアルゴリズムを実行することにより前記符号ワードの前記表現を復号化するプロセッサを含むデコーダとを含んでいるメモリコントローラ。

【請求項27】

(c) 前記符号ワードの少なくとも一部を主メモリに保存し、且つ前記符号ワードの前記少なくとも一部の表現を前記主メモリから取得する回路を更に含む、請求項26に記載のメモリコントローラ。

【請求項28】

(a) 請求項27に記載の前記メモリコントローラと、

(b) 前記主メモリとを含むメモリ装置。

【請求項29】

(a) 通信チャネルから受信したメッセージを復調することによりK個の情報ビットをN (> K) 個の符号ワードビットとして符号化する符号ワードの表現を生成する復調器と、

(b)(i) N個のビットノードおよびN - K個のチェックノードを含むグラフにおける前記ビットノードと前記チェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新することおよび、

(ii) (A) 所定の失敗基準に照らして前記復号化が収束に失敗し、且つ

(B) 前記符号ワードビットの推定値が、前記グラフにトラッピング集合が含まれている兆候を示す基準を満たしている場合、前記反復を続ける前に前記メッセージの少なくとも一部をリセットすること、を含むステップにより、前記符号ワードの推定値を更新するアルゴリズムを実行することにより前記符号ワードの前記表現を復号化するプロセッサを含むデコーダとを含む受信器。

【請求項30】

(a) 通信チャネルから受信したメッセージを復調することによりK個の情報ビットをN (> K) 個の符号ワードビットとして符号化する符号ワードの表現を生成する復調器と、

(b)(i) N個のビットノードおよびN - K個のチェックノードを含むグラフにおける前記ビットノードと前記チェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新することおよび、

(ii) 所定の失敗基準に照らして前記復号化が収束に失敗した場合、前記反復を続ける前に前記ビットノードから送られた前記メッセージの少なくとも一部を切り捨てること、を含むステップにより、前記符号ワードの推定値を更新するアルゴリズムを実行することにより前記符号ワードの前記表現を復号化するプロセッサを含むデコーダとを含む受信器。

【請求項31】

メッセージを送受信する通信システムであって、

(a)(i) 前記メッセージのK個の情報ビットをN (> K) 個の符号ワードビットの符号ワードとして符号化するエンコーダ、および

(ii) 通信チャネルを介して前記符号ワードを変調信号として送信する変調器を含む送信器と、

(b)(i) 前記通信チャネルから前記変調信号を受信して前記変調信号を復調することにより、前記符号ワードの表現を生成する復調器、および

(ii) (A) N個のビットノードおよびN - K個のチェックノードを含むグラフ

10

20

30

40

50

における前記ビットノードと前記チェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新することおよび、

(B)(I)所定の失敗基準に照らして前記復号化が収束に失敗し、且つ

(II)前記符号ワードビットの推定値が、前記グラフにトラッピング集合が含まれている兆候を示す基準を満たしている場合、前記反復を続ける前に前記メッセージの少なくとも一部をリセットすること、を含むステップにより、前記符号ワードの推定値を更新するアルゴリズムを実行することにより前記符号ワードの前記表現を復号化するプロセッサを含むデコーダを含む受信器とを含む通信システム。

#### 【請求項 3 2】

メッセージを送受信する通信システムであって、

(a)(i)前記メッセージのK個の情報ビットを $N(>K)$ 個の符号ワードビットの符号ワードとして符号化するエンコーダ、および

(ii)通信チャネルを介して前記符号ワードを変調信号として送信する変調器を含む送信器と、

(b)(i)前記通信チャネルから前記変調信号を受信して前記変調信号を復調することにより、前記符号ワードの表現を生成する復調器、および

(ii)(A)N個のビットノードおよび $N-K$ 個のチェックノードを含むグラフにおける前記ビットノードと前記チェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新することおよび、

(B)所定の失敗基準に照らして前記復号化が収束に失敗した場合、前記反復を続ける前に前記ビットノードから送られた前記メッセージの少なくとも一部を切り捨てること、を含むステップにより、前記符号ワードの推定値を更新するアルゴリズムを実行することにより前記符号ワードの表現を復号化するプロセッサを含むデコーダを含む受信器とを含む通信システム。

#### 【請求項 3 3】

K個の情報ビットを $N(>K)$ 個の符号ワードビットとして符号化する符号ワードの表現を復号化する方法であって、

(a)チャネルから前記符号ワードの表現をインポートすること、

(b) $N-K$ 行およびN列を有するパリティチェック行列を提供すること、

(c)前記行列の前記行と前記列との間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新することおよび、

(d)(i)所定の失敗基準に照らして前記復号化が収束に失敗し、且つ

(ii)前記符号ワードビットの推定値が、前記パリティチェック行列にトラッピング集合が含まれている兆候を示す基準を満たしている場合、前記反復を続ける前に前記メッセージの少なくとも一部をリセットすることを含む方法。

#### 【請求項 3 4】

K個の情報ビットを $N(>K)$ 個の符号ワードビットとして符号化する符号ワードの表現を復号化する方法であって、

(a)チャネルから前記符号ワードの表現をインポートすること、

(b) $N-K$ 行およびN列を有するパリティチェック行列を提供すること、

(c)前記行と前記列との間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新することおよび、

(d)所定の失敗基準に照らして前記復号化が収束に失敗した場合、前記反復を続ける前に前記列から送られた前記メッセージの少なくとも一部を切り捨てることを含む方法。

#### 【請求項 3 5】

K個の情報ビットを $N(>K)$ 個の符号ワードビットとして符号化する符号ワードの表現を復号化するデコーダであって、

(a) $N-K$ 行およびN列を有するパリティチェック行列を提供すること、

(b)前記行と前記列との間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新すること、

(c)(i) 所定の失敗基準に照らして前記復号化が収束に失敗し、且つ

(ii) 前記符号ワードビットの推定値が、前記パリティチェック行列にトラッピング集合が含まれている兆候を示す基準を満たしている場合、前記反復を続ける前に前記メッセージの少なくとも一部をリセットすること、を含むステップにより、前記符号ワードの推定値を更新するアルゴリズムを実行することにより前記符号ワードの前記表現を復号化するプロセッサを含むデコーダ。

【請求項 36】

K 個の情報ビットを  $N (> K)$  個の符号ワードビットとして符号化する符号ワードの表現を復号化するデコーダであって、

(a)  $N - K$  行および  $N$  列を有するパリティチェック行列を提供すること、

(b) 前記行と前記列との間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新すること、

(c) 所定の失敗基準に照らして前記復号化が収束に失敗した場合、前記反復を続ける前に前記列から送られた前記メッセージの少なくとも一部を切り捨てること、を含むステップにより、前記符号ワードの推定値を更新するアルゴリズムを実行することにより前記符号ワードの前記表現を復号化するプロセッサを含むデコーダ。

【請求項 37】

(a) K 個の情報ビットを  $N (> K)$  個の符号ワードビットの符号ワードとして符号化するエンコーダと、

(b)(i)  $N - K$  行および  $N$  列を有するパリティチェック行列を提供すること、

(ii) 前記行と前記列との間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新すること、

(iii)(A) 所定の失敗基準に照らして前記復号化が収束に失敗し、且つ

(B) 前記符号ワードビットの推定値が、前記パリティチェック行列にトラッピング集合が含まれている兆候を示す基準を満たしている場合、前記反復を続ける前に前記メッセージの少なくとも一部をリセットすること、を含むステップにより、前記符号ワードの推定値を更新するアルゴリズムを実行することにより前記符号ワードの前記表現を復号化するプロセッサを含むデコーダとを含むメモリコントローラ。

【請求項 38】

(c) 前記符号ワードの少なくとも一部を主メモリに保存し、且つ前記符号ワードの前記少なくとも一部の表現を前記主メモリから取得する回路を更に含む、請求項 37 に記載のメモリコントローラ。

【請求項 39】

(a) 請求項 38 に記載の前記メモリコントローラと、

(b) 前記主メモリとを含むメモリ装置。

【請求項 40】

(a) K 個の情報ビットを  $N (> K)$  個の符号ワードビットの符号ワードとして符号化するエンコーダと、

(b)(i)  $N - K$  行および  $N$  列を有するパリティチェック行列を提供すること、

(ii) 前記行と前記列との間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新すること、

(iii) 所定の失敗基準に照らして前記復号化が収束に失敗した場合、前記反復を続ける前に前記列から送られた前記メッセージの少なくとも一部を切り捨てること、を含むステップにより、前記符号ワードの推定値を更新するアルゴリズムを実行することにより前記符号ワードの前記表現を復号化するプロセッサを含むデコーダとを含むメモリコントローラ。

【請求項 41】

(c) 前記符号ワードの少なくとも一部を主メモリに保存し、且つ前記符号ワードの前記少なくとも一部の表現を前記主メモリから取得する回路を更に含む、請求項 40 に記載のメモリコントローラ。

10

20

30

40

50

## 【請求項 4 2】

- (a) 請求項 4 1 に記載の前記メモリコントローラと、
- (b) 前記主メモリとを含むメモリ装置。

## 【請求項 4 3】

(a) 通信チャネルから受信したメッセージを復調することにより  $K$  個の情報ビットを  $N (> K)$  個の符号ワードビットとして符号化する符号ワードの表現を生成する復調器と、

(b) (i)  $N - K$  行および  $N$  列を有するパリティチェック行列を提供すること、  
(ii) 前記行と前記列との間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新すること、

(iii) (A) 所定の失敗基準に照らして前記復号化が収束に失敗し、且つ  
(B) 前記符号ワードビットの推定値が、前記パリティチェック行列にトラッピング集合が含まれている兆候を示す基準を満たしている場合、前記反復を続ける前に前記メッセージの少なくとも一部をリセットすること、を含むステップにより、前記符号ワードの推定値を更新するアルゴリズムを実行することにより前記符号ワードの前記表現を復号化するプロセッサを含むデコーダとを含む受信器。

## 【請求項 4 4】

(a) 通信チャネルから受信したメッセージを復調することにより  $K$  個の情報ビットを  $N (> K)$  個の符号ワードビットとして符号化する符号ワードの表現を生成する復調器と、

(b) (i)  $N - K$  行および  $N$  列を有するパリティチェック行列を提供すること、  
(ii) 前記行と前記列との間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新すること、

(iii) 所定の失敗基準に照らして前記復号化が収束に失敗した場合、前記反復を続ける前に前記列から送られた前記メッセージの少なくとも一部を切り捨てること、を含むステップにより、前記符号ワードの推定値を更新するアルゴリズムを実行することにより前記符号ワードの前記表現を復号化するプロセッサを含むデコーダとを含む受信器。

## 【請求項 4 5】

メッセージを送受信する通信システムであって、

(a) (i) 前記メッセージの  $K$  個の情報ビットを  $N (> K)$  個の符号ワードビットの符号ワードとして符号化するエンコーダ、および

(ii) 通信チャネルを介して前記符号ワードを変調信号として送信する変調器を含む送信器と、

(b) (i) 前記通信チャネルから前記変調信号を受信して前記変調信号を復調することにより、前記符号ワードの表現を生成する復調器、および

(ii) (A)  $N - K$  行および  $N$  列を有するパリティチェック行列を提供すること、

(B) 前記行と前記列との間でのメッセージ交換を含むステップにより複数回の復号化反復で前記符号ワードビットの推定値を更新すること、

(C) (I) 所定の失敗基準に照らして前記復号化が収束に失敗し、且つ

(II) 前記符号ワードビットの推定値が、前記パリティチェック行列にトラッピング集合が含まれている兆候を示す基準を満たしている場合、前記反復を続ける前に前記メッセージの少なくとも一部をリセットすること、を含むステップにより、前記符号ワードの推定値を更新するアルゴリズムを実行することにより前記符号ワードの前記表現を復号化するプロセッサを含むデコーダを含む受信器とを含む通信システム。

## 【請求項 4 6】

メッセージを送受信する通信システムであって、

(a) (i) 前記メッセージの  $K$  個の情報ビットを  $N (> K)$  個の符号ワードビットの符号ワードとして符号化するエンコーダ、および

(ii) 通信チャネルを介して前記符号ワードを変調信号として送信する変調器を含む送信器と、

(b)(i) 前記通信チャネルから前記変調信号を受信して前記変調信号を復調することにより、前記符号ワードの表現を生成する復調器、および

(ii) (A) N - K 行および N 列を有するパリティチェック行列を提供すること

、  
(B) 前記行と前記列との間でのメッセージ交換を含むステップにより複数の復号化反復で前記符号ワードビットの推定値を更新すること、

(C) 所定の失敗基準に照らして前記復号化が収束に失敗した場合、前記反復を続ける前に前記列から送られた前記メッセージの少なくとも一部を切り捨てること、を含むステップにより、前記符号ワードの推定値を更新するアルゴリズムを実行することにより前記符号ワードの前記表現を復号化するプロセッサを含むデコーダを含む受信器とを含む通信システム。

10

【発明の詳細な説明】

【技術分野】

【0001】

本特許出願は、2008年6月23日出願の米国仮特許出願第61/074,701号を優先権主張するものである。

【0002】

本明細書に、トラッピング集合に起因する非収束を克服する低密度パリティチェック(LDPC)復号化の方法および関連装置を開示する。

【背景技術】

20

【0003】

誤り訂正符号(ECC)は、通信システムおよび記憶装置において共通的に用いられる。通信チャネルおよび記憶装置の両方において生じる各種の物理現象の結果として通信または保存された情報を損なうノイズ効果が生じる。誤り訂正符号化スキームを用いて、通信または保存された情報を結果的に生じた誤りから保護することができる。これは、通信チャネルを介した伝送またはメモリ装置への保存の前に情報を符号化することにより行われる。符号化処理は、情報に冗長性を加えることにより情報ビットシーケンスを符号ワードに変換する。次いでこの冗長性を用いて、復号処理を介して損なわれた可能性のある符号ワードから情報を復元することができる。

【0004】

30

通信システムおよび記憶システムの両方において、情報ビットシーケンス  $\underline{i}$  は、通信チャネルにまたはメモリ装置に適合された一連のシンボル  $x$  に変調または写像された符号化ビットシーケンス  $\underline{y}$  に符号化される。通信チャネルまたはメモリ装置の出力端で一連のシンボル  $\underline{y}$  が得られる。当該システムの ECC デコーダはシーケンス  $\underline{y}$  を復号化して、ビットシーケンス

$\hat{\underline{i}}$

を復元するものであり、元の情報ビットシーケンス  $\underline{i}$  を高い確度で再構築しなければならない。

【0005】

40

一般的な ECC ファミリーは、線形 2 値ブロック符号のファミリーである。長さが N である K 次元の線形 2 値符号は、長さ K の情報ビットシーケンスを長さ N の符号ワードに線形写像したものであって、 $N > K$  である。符号レートは  $R = K / N$  で定義される。 $1 \times N$  次元の符号ワード  $\underline{v}$  の符号化処理が通常、次式に従い  $1 \times K$  次元の情報ビットシーケンス  $\underline{i}$  に  $K \times N$  次元の生成行列  $G$  を乗ずることで行われる。

【0006】

【数 1】

$$\underline{v} = \underline{i} \cdot G$$

(1)

【0007】

50



また、 $M \times N$ 次元のパリティチェック行列  $H$  を定義することも慣習的であり、 $M = N - K$  である。パリティチェック行列は、次式を介して生成行列に関連付けられる。

【 0 0 0 8 】

【数 2】

$$GH^T = 0 \quad (2)$$

【 0 0 0 9 】

パリティチェック行列を用いて、長さが  $N$  である 2 値ベクトルが有効な符号ワードであるか否かを調べることができる。  $1 \times N$  の 2 値ベクトル  $\underline{v}$  は、次式が成立する場合に限って符号に属する。

【 0 0 1 0 】

【数 3】

$$H \cdot \underline{v}' = 0 \quad (3)$$

( 式 ( 3 ) において、  $\underline{v}'$  のプライム記号は  $\underline{v}$  が列ベクトルであることを意味する。 )

【 0 0 1 1 】

近年、反復的符号化スキームが広く普及している。これらのスキームにおいて、符号はいくつかの簡単な組成符号の接続として構成され、簡単な符号の組成デコードの間で情報を交換することにより、反復的な復号化アルゴリズムを用いて復号化される。通常、符号は組成符号間の相互接続を記述する二部グラフを用いて定義することができる。この場合、復号化はグラフの辺を介してやり取りされる反復的メッセージと見なすことができる。

【 0 0 1 2 】

反復符号の伝播しているクラスは、低密度パリティチェック ( LDPC ) 符号である。LDPC 符号は、疎なパリティチェック行列  $H$  により定義される線形 2 値ブロック符号である。図 1 に示すように、当該符号を同等に、 $N$  個のビットノード ( 図 1 では  $N = 13$  ) の集合  $V$ 、 $M$  個のチェックノード ( 図 1 では  $M = 10$  ) の集合  $C$ 、およびビットノードをチェックノードに接続する辺の集合  $E$  ( 図 1 では  $E = 38$  ) を有する疎な二部グラフ  $G = ( V, C, E )$  により定義できる。ビットノードは符号ワードビットに対応し、チェックノードはビットに対するパリティチェック制約に対応する。ビットノードは、辺により当該ビットノードが関与するチェックノードに接続される。図 1 の左側の符号の行列表現において、ビットノード  $i$  をチェックノード  $j$  に接続している辺を行  $j$  と列  $i$  の交差箇所における非ゼロの行列成分で表す。

【 0 0 1 3 】

図 1 の最初および最後のチェックノードの次に、式 ( 3 ) に等価な行を示す。記号

“ $\oplus$ ”

は「XOR」を意味する。

【 0 0 1 4 】

LDPC 符号は、反復的メッセージ受け渡し復号化アルゴリズムを用いて復号化することができる。これらのアルゴリズムは、ビットノード間でメッセージを交換することにより動作し、符号を表す基礎となる二部グラフの辺に沿ってノードをチェックする。デコードには、符号ワードビットの初期推定値が与えられる ( 通信チャネル出力または読み出されたメモリ内容に基づいて )。これらの初期推定値は、有効な符号ワードとしてビットが満たすべきパリティチェック制約 ( 式 ( 3 ) による ) を課すことにより洗練および改良される。これは、グラフの辺に沿って受け渡されるメッセージを用いて、符号ワードビットを表すビットノードと符号ワードビットに対するパリティチェック制約を表すチェックノードの間で情報を交換することにより行われる。

【 0 0 1 5 】

10

20

30

40

50

反復復号化アルゴリズムにおいて、ビット推定値およびビット推定値の信頼度の両方を伝達する「弱い」ビット推定値の利用が一般的である。

【 0 0 1 6 】

グラフの辺に沿って受け渡されるメッセージにより伝達されるビット推定値はさまざまな形式で表すことができる。「弱い」ビット推定値を表す一般的な尺度は、対数尤度比 (LLR) であり、

$$\log \frac{\Pr(v=0 | \text{current constraints and observations})}{\Pr(v=1 | \text{current constraints and observations})},$$

(current constraints and observations: 現在の制約および観測値)

10

ここに、「現在の制約および観測値」が手元のメッセージを計算する際に考慮される各種のパリティチェック制約であり、観測値  $y$  はこれらのパリティチェックに関与するビットに対応する。一般性を失うことなく、簡潔のため以下において常に LLR メッセージを用いるものと仮定する。LLR の符号はビットの推定値を与える (すなわち、正の LLR は  $v=0$  に対応し、負の LLR は  $v=1$  に対応する)。LLR の大きさは、推定値の信頼度を与える (すなわち、 $|LLR|=0$  は推定値が全く信頼できないことを意味し、 $|LLR|=\pm$  は推定値が完全に信頼でき且つビット値が既知であることを意味する)。

【 0 0 1 7 】

通常、ビットノードとチェックノードの間のグラフ辺に沿った復号化を行う間に受け渡されるメッセージは外因的である。辺  $e$  上のノード  $n$  から受け渡された外因的メッセージ  $m$  は、辺  $e$  以外で  $n$  に接続している边上で受取られた全ての値を考慮に入れる (当該メッセージは新しい情報だけに基いているため、外因的と呼ばれる)。

20

【 0 0 1 8 】

メッセージ受け渡し復号化アルゴリズムの一例として、確率伝播 (BP) アルゴリズムがあり、メッセージ受け渡しアルゴリズムファミリの中で最良のアルゴリズムと考えられる。

【 0 0 1 9 】

$P_v = \log (\Pr(v=0 | y) / \Pr(v=1 | y))$  が、受信されたかまたは読み出された記号  $y$  だけに基いてビット  $v$  の初期デコード推定値を表すとする。ビットのいくつかは通信チャネルを介して送信されていないかまたはメモリ装置に保存されていないためにこれらのビットの  $y$  観測値が無い可能性もある点に注意されたい。この場合、二つの可能性がある。1) 短縮ビット、すなわちビットは先験的に知られていて  $P_v = \pm$  (ビットが 0 か 1 に応じて) である。2) パンクチャードビット、すなわちビットは先見的には知られておらず  $P_v = \log (\Pr(v=0) / \Pr(v=1))$  であり、 $\Pr(v=0)$  および  $\Pr(v=1)$  はビット  $v$  が各々 0 または 1 である先験的確率である。情報ビットが 0 または 1 である先験的確率が等しく、且つ符号が線形であると仮定すれば、 $P_v = \log ((1/2) / (1/2)) = 0$  である。

30

【 0 0 2 0 】

$Q_v = \log (\Pr(v=0 | \underline{y}, H \cdot \underline{v} = 0) / \Pr(v=1 | \underline{y}, H \cdot \underline{v} = 0))$  が、受取ったかまたは読み出された全体のシーケンス  $\underline{y}$  に基づき、且つビット  $v$  が符号ワードの一部であると仮定 (すなわち、 $H \cdot \underline{v} = 0$  と仮定) して、ビット  $v$  の最終デコード推定値を表すとする。

40

【 0 0 2 1 】

$Q_{vc}$  がビットノード  $v$  からチェックノード  $c$  へのメッセージを表すとする。 $R_{cv}$  をチェックノード  $c$  からビットノード  $v$  へのメッセージを表すとする。

【 0 0 2 2 】

BP アルゴリズムは以下の更新規則を利用してメッセージを計算する。

【 0 0 2 3 】

ビットノードからチェックノードへの計算規則は次式である。

【 0 0 2 4 】

50

【数 4】

$$Q_{vc} = P_v + \sum_{c' \in N(v, G) \setminus c} R_{c'v} \quad (4)$$

【0025】

ここに、 $N(n, G)$  はグラフ  $G$  におけるノード  $n$  の隣接ノードの集合を表し、 $c' \in N(v, G) \setminus c$  は、ノード「 $c$ 」を除く隣接ノードを指す（和は  $c$  以外の全ての隣接ノードについて求める）。

【0026】

チェックノードからビットノードへの計算規則は次式である。

10

【0027】

【数 5】

$$R_{cv} = \varphi^{-1} \left( \sum_{v' \in N(c, G) \setminus v} \varphi(Q_{v'c}) \right) \quad (5)$$

【0028】

ここに、 $\varphi(x) = \{ \text{sign}(x), -\log \tanh(|x|/2) \}$  であって、領域内の演算は群  $\{0, 1\} \times \mathbb{R}^+$  にわたり行われる（これは基本的に、ここでの和は大きさの合計と符号同士の XOR として定義されていることを意味する）。式（4）の表記と同様に、 $N(c, G)$  はグラフ  $G$  内のチェックノード  $c$  の隣接するビットノードの集合を表し、 $v' \in N(c, G) \setminus v$  はノード「 $v$ 」を除く隣接ノードを指す（和は  $v$  以外の全ての隣接ノードについて求める）。

20

【0029】

ビット  $v$  に対する最終デコード推定値は以下の通りである。

【0030】

【数 6】

$$Q_v = P_v + \sum_{c \in N(v, G)} R_{c'v} \quad (6)$$

【0031】

メッセージ受け渡し復号化を行う間におけるメッセージ受け渡しの順序を復号化スケジュールと呼ぶ。BP 復号化は特定のスケジュールを用いることは示唆せず、計算規則（式（4）、（5）および（6））を定義するだけである。復号化スケジュールは、符号の期待される誤り訂正能力に影響を及ぼさない。しかし、復号化スケジュールは、デコードの収束速度およびデコードの複雑度に大いに影響し得る。

30

【0032】

LDPCC 符号を復号化する標準メッセージ受け渡しスケジュールは、各反復において全ての可変ノード、および後続する全てのチェックノードが隣接ノードに新規メッセージを受け渡すフラッディングスケジュールである（R. G. Gallager, Low-Density Parity-Check Codes, Cambridge, MA: MIT Press 1963）。フラッディングスケジュールに基づく標準 BP アルゴリズムを図 2 に与える。

40

【0033】

フラッディングスケジュールに基づく BP アルゴリズムの標準的な実装は必要なメモリの観点から高価である。（ $P_v$ ,  $Q_v$ ,  $Q_{vc}$ 、および  $R_{cv}$  個のメッセージを保存するために）総計で  $2|V| + 2|E|$  個のメッセージを保存する必要がある。更に、フラッディングスケジュールは収束速度が遅いため、必要とされる誤り訂正能力を所与の復号化スループットで提供するためにより高度な復号化論理（例：ASIC 上により多くのプロセッサ）を必要とする。

【0034】

より効率的な、連続メッセージ受け渡し復号化スケジュールが知られている。連続メッ

50

セージ受け渡し復号化スケジュールでは、ビットまたはチェックノードを連続的に辿り、各ノードについて、当該ノードとの間で対応するメッセージがやり取りされる。例えば、連続スケジュールは、グラフ内のチェックノードを何らかの順序で連続的に辿ることにより実装でき、各チェックノード  $c \in C$  について以下のメッセージは送られる。

【0035】

1. 各  $v \in N(c)$  に対して  $Q_{vc}$  (すなわち、ノード  $c$  への全ての  $Q_{vc}$  メッセージ)
2. 各  $v \in N(c)$  に対して  $R_{cv}$  (すなわち、ノード  $c$  からの全ての  $R_{cv}$  メッセージ)

【0036】

フラッディングスケジュールとは対照的に、連続スケジュールはグラフ上で情報の即時且つより高速な伝播を可能にする結果、収束がより速く（ほぼ2倍速く）なる。更に、連続スケジュールは、必要なメモリを大幅に減らしつつ効率的に実装できる。これは、 $Q_{vc}$  メッセージをオンザフライ計算するために  $Q_v$  個のメッセージおよび  $R_{cv}$  個のメッセージを用いることにより実現でき、従って  $Q_{vc}$  個のメッセージを保存するための追加メモリを使用する必要がなくなる。これは式(4)、(6)に基づいて  $Q_{vc}$  を  $(Q_v - R_{cv})$  で表すことにより行われる。更に、先験的なメッセージ  $P_v$  により初期化されたものと同メモリを用いて、反復的に更新された  $Q_v$  個の後験的なメッセージを保存する。フラッディングスケジュールの標準的な実装では符号のグラフ構造を保存するために2倍のメモリを必要とする両方のデータ構造  $N(c) \subset C$  および  $N(v) \subset V$  を使用するのに対し、連続スケジュールでは  $N(c) \subset C$  に関する知識だけを使用すればよい。必要メモリを更に減らすことができる。連続にスケジューリングされた復号化アルゴリズムを図3に示す。

【0037】

要約すれば、連続復号化スケジュールは、フラッディングスケジュールに対して以下の利点がある。

【0038】

1) 連続復号化スケジュールは、標準フラッディングスケジュールと比較して収束速度を2倍に上げる。これは、フラッディングスケジュールに基づくデコードと比較して、所与のスループットで所与の誤り訂正能力を提供するために半分の復号化論理だけで済むことを意味する。

【0039】

2) 連続復号化スケジュールは、デコードの実装におけるメモリ効率を高める。 $|V| + |E|$  個のメッセージだけを保存するRAMが必要である（標準フラッディングスケジュールのように  $2|V| + 2|E|$  個のメッセージを保存するのではなく）。標準フラッディングスケジュールと比較して符号のグラフ構造を保存するのに半分のROMサイズで済む。

【0040】

3) 反復実行の間、「オンザフライ」収束テストを計算の一環として実施できるため、反復実行中に収束が検出でき、且つ任意の時点で復号化を終了できる。これにより復号に要する時間とエネルギー消費を節約できる。

【0041】

[定義]

本明細書に記述する方法は、少なくとも二つの異なる条件下でデータ誤りの訂正に適用できる。一つの条件、記憶媒体からどのデータを取得するかである。他の条件は、伝送媒体からどのデータを受信するかである。記憶媒体および伝送媒体は、データに誤りを入れる「チャンネル」の特別な場合である。データの「取得」および「受信」という概念を本明細書ではデータを「インポート」という概念に一般化している。データの「取得」およびデータの「受信」は、データをチャンネルから「インポートする」特別な場合である。

【0042】

本明細書に示す方法により復号化されるデータは符号ワードの表現である。これらの方

10

20

30

40

50

法の一つが復号化に適用される前にチャネル内で符号ワードがノイズにより損なわれている恐れがあるため、データは符号ワードの「表現」に過ぎず、符号ワード自体ではない。

【発明の概要】

【発明が解決しようとする課題】

【0043】

反復符号化システムでは図4で示すように、エラーフロアと呼ばれる好ましくない現象が生じ、通信チャネルまたはメモリ装置内で特定の「ノイズ」レベル以下では、ビット誤りの原因である「ノイズ」がたとえ減少するにせよ、デコーダの出力におけるブロック誤り率(BER)ははるかに遅く減少し始める。この現象は特に、必要なデコーダ出力ブロック誤り率が極めて低い( $\sim 10^{-10}$ )ことが求められる記憶システム内で問題となる。

10

図4では右へ行くほどノイズが増加する点に注意されたい。

【0044】

反復符号化システムの誤り訂正能力およびエラーフロアは、符号長が増加するにつれて向上する(これはどのECCシステムでも言えるが、特に短い符号長で誤り訂正能力が劣る反復符号化システムで顕著である)ことが良く知られている。

【0045】

しかし、反復符号化システムの従来方式の実装では、復号化ハードウェアのメモリの複雑度は符号長と比例するため、たとえ知られている最も効率的な実装(例:連続的にスケジューリングされたデコーダ)であっても長い符号を使用すれば複雑度が高まる。

【0046】

20

従って、本明細書では、複雑度の低い復号化ハードウェアを用いて極めて低いエラーフロアおよびほぼ最適な誤り訂正能力を提供する極めて長いLDPC符号を実装する方法を開示する。

【0047】

適切に設計されたLDPC符号は極めて強力であって、符号ワード内の多くの誤りを訂正できるが、たとえ不正確なビットの数が非常に少なくてもグラフ内の特定の領域に限定できるにせよ、「トラッピング集合」として知られる現象に起因してデコーダが失敗し、符号のエラーフロアが増加する恐れがある。トラッピング集合は一般的なLDPC符号に対して明確に定義されていないものの、「これらは、誘導された部分グラフがごく少数の奇数次数チェックノードしか存在しないように、比較的少ない数の可変ノードを有する集合である。」として記述されている。

30

【0048】

トラッピング集合は、LDPCグラフのトポロジおよび使用する特定の復号化アルゴリズムに関連していて、回避および解析するのが困難である。

【0049】

トラッピング集合は、歴史的に記憶装置が要求する信頼度が、例えば保存された $10^{14}$ 個のビット当たり1個のビット誤りというように比較的高いため、記憶装置分野で問題であった。その結果、フラッシュメモリ装置等のメモリ装置で用いられる符号はエラーフロアが低くなければならないが、トラッピング集合はエラーフロアを増やしてしまう。

【課題を解決するための手段】

40

【0050】

従って、本明細書に示す一実施形態は、K個の情報ビットを $N > K$ 個の符号ワードビットとして符号化する符号ワードの表現を復号化する方法であって、本方法は(a)チャネルから符号ワードの表現をインポートすること、(b)N個のビットノードおよび $N - K$ 個のチェックノードを含むグラフにおけるビットノードとチェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新することおよび、(c)(i)所定の失敗基準に照らして復号化が収束に失敗し、且つ(ii)符号ワードビットの推定値が、当該グラフにトラッピング集合が含まれている兆候を示す基準を満たしている場合、反復を続ける前にメッセージの少なくとも一部をリセットすることを含んでいる。

50

## 【0051】

本明細書に示す別の実施形態は、 $K$ 個の情報ビットを $N > K$ 個の符号ワードビットとして符号化する符号ワードの表現を復号化する方法であって、本方法は（a）チャネルから符号ワードの表現をインポートすること、（b） $N$ 個のビットノードおよび $N - K$ 個のチェックノードを含むグラフにおけるビットノードとチェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新することおよび、（c）所定の失敗基準に照らして復号化が収束に失敗した場合、反復を続ける前に当該ビットノードから送られたメッセージの少なくとも一部を切り捨てることを含んでいる。

## 【0052】

10

本明細書に示す別の実施形態は、 $K$ 個の情報ビットを $N > K$ 個の符号ワードビットとして符号化する符号ワードの表現を復号化するデコーダであって、（a） $N$ 個のビットノードおよび $N - K$ 個のチェックノードを含むグラフにおけるビットノードとチェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新することおよび、（b）（i）所定の失敗基準に照らして復号化が収束に失敗し、且つ（ii）符号ワードビットの推定値が、当該グラフにトラッピング集合が含まれている兆候を示す基準を満たしている場合、反復を続ける前にメッセージの少なくとも一部をリセットすること、を含むステップにより、符号ワードの推定値を更新するアルゴリズムを実行することにより符号ワードの表現を復号化するプロセッサを含んでいる。

## 【0053】

20

本明細書に示す別の実施形態は、 $K$ 個の情報ビットを $N > K$ 個の符号ワードビットとして符号化する符号ワードの表現を復号化するデコーダであって、（a） $N$ 個のビットノードおよび $N - K$ 個のチェックノードを含むグラフにおけるビットノードとチェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新することおよび、（b）所定の失敗基準に照らして復号化が収束に失敗した場合、反復を続ける前に当該ビットノードから送られたメッセージの少なくとも一部を切り捨てること、を含むステップにより、符号ワードの推定値を更新するアルゴリズムを実行することにより符号ワードの表現を復号化するプロセッサを含んでいる。

## 【0054】

本明細書に示す別の実施形態はメモリコントローラであって、（a） $K$ 個の情報ビットを $N > K$ 個の符号ワードビットの符号ワードとして符号化するエンコーダと、（b）（i） $N$ 個のビットノードおよび $N - K$ 個のチェックノードを含むグラフにおけるビットノードとチェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新することおよび、（ii）（A）所定の失敗基準に照らして復号化が収束に失敗し、且つ（B）符号ワードビットの推定値が、当該グラフにトラッピング集合が含まれている兆候を示す基準を満たしている場合、反復を続ける前にメッセージの少なくとも一部をリセットすること、を含むステップにより、符号ワードの推定値を更新するアルゴリズムを実行することにより符号ワードの表現を復号化するプロセッサを含むデコーダとを含んでいる。

30

## 【0055】

40

本明細書に示す別の実施形態はメモリコントローラであって、（a） $K$ 個の情報ビットを $N > K$ 個の符号ワードビットの符号ワードとして符号化するエンコーダと、（b）（i） $N$ 個のビットノードおよび $N - K$ 個のチェックノードを含むグラフにおけるビットノードとチェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新することおよび、（ii）所定の失敗基準に照らして復号化が収束に失敗した場合、反復を続ける前に当該ビットノードから送られたメッセージの少なくとも一部を切り捨てること、を含むステップにより、符号ワードの推定値を更新するアルゴリズムを実行することにより符号ワードの表現を復号化するプロセッサを含むデコーダとを含んでいる。

## 【0056】

50

本明細書に示す別の実施形態は受信器であって、(a) 通信チャネルから受信したメッセージを復調することより  $K$  個の情報ビットを  $N > K$  個の符号ワードビットとして符号化する符号ワードの表現を生成する復調器と、(b) (i)  $N$  個のビットノードおよび  $N - K$  個のチェックノードを含むグラフにおけるビットノードとチェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新することおよび、(ii) (A) 所定の失敗基準に照らして復号化が収束に失敗し、且つ (B) 符号ワードビットの推定値が、当該グラフにトラッピング集合が含まれている兆候を示す基準を満たしている場合、反復を続ける前にメッセージの少なくとも一部をリセットすること、を含むステップにより、符号ワードの推定値を更新するアルゴリズムを実行することにより符号ワードの表現を復号化するプロセッサを含むデコーダとを含んでいる。

10

## 【0057】

本明細書に示す別の実施形態は受信器であって、(a) 通信チャネルから受信したメッセージを復調することより  $K$  個の情報ビットを  $N > K$  個の符号ワードビットとして符号化する符号ワードの表現を生成する復調器と、(b) (i)  $N$  個のビットノードおよび  $N - K$  個のチェックノードを含むグラフにおけるビットノードとチェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新することおよび、(ii) 所定の失敗基準に照らして復号化が収束に失敗した場合、反復を続ける前に当該ビットノードから送られたメッセージの少なくとも一部を切り捨てること、を含むステップにより、符号ワードの推定値を更新するアルゴリズムを実行することにより符号ワードの表現を復号化するプロセッサを含むデコーダとを含んでいる。

20

## 【0058】

本明細書に示す別の実施形態はメッセージを送受信する通信システムであって、(a) (i) 当該メッセージの  $K$  個の情報ビットを  $N > K$  個の符号ワードビットの符号ワードとして符号化するエンコーダ、および (ii) 通信チャネルを介して符号ワードを変調信号として送信する変調器を含む送信器と、(b) (i) 通信チャネルから変調信号を受信して当該変調信号を復調することより、符号ワードの表現を生成する復調器、および (ii) (A)  $N$  個のビットノードおよび  $N - K$  個のチェックノードを含むグラフにおけるビットノードとチェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新することおよび、(B) (I) 所定の失敗基準に照らして復号化が収束に失敗し、且つ (II) 符号ワードビットの推定値が、当該グラフにトラッピング集合が含まれている兆候を示す基準を満たしている場合、反復を続ける前にメッセージの少なくとも一部をリセットすること、を含むステップにより、符号ワードの推定値を更新するアルゴリズムを実行することにより符号ワードの表現を復号化するプロセッサを含むデコーダを含む受信器とを含んでいる。

30

## 【0059】

本明細書に示す別の実施形態はメッセージを送受信する通信システムであって、(a) (i) 当該メッセージの  $K$  個の情報ビットを  $N > K$  個の符号ワードビットの符号ワードとして符号化するエンコーダ、および (ii) 通信チャネルを介して符号ワードを変調信号として送信する変調器を含む送信器と、(b) (i) 通信チャネルから変調信号を受信して当該変調信号を復調することにより、符号ワードの表現を生成する復調器、および (ii) (A)  $N$  個のビットノードおよび  $N - K$  個のチェックノードを含むグラフにおけるビットノードとチェックノードとの間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新することおよび、(B) (I) 所定の失敗基準に照らして復号化が収束に失敗した場合、反復を続ける前に当該ビットノードから送られたメッセージの少なくとも一部を切り捨てること、を含むステップにより、符号ワードの推定値を更新するアルゴリズムを実行することにより符号ワードの表現を復号化するプロセッサを含むデコーダを含む受信器とを含んでいる。

40

## 【0060】

本明細書に示す別の実施形態は、 $K$  個の情報ビットを  $N > K$  個の符号ワードビットとし

50

て符号化する符号ワードの表現を復号化する方法であって、本方法は (a) チャンネルから符号ワードの表現をインポートすること、(b)  $N - K$  行および  $N$  列を有するパリティチェック行列を提供すること、(c) 当該行列の行と列との間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新することおよび、(d) (i) 所定の失敗基準に照らして復号化が収束に失敗し、且つ (ii) 符号ワードビットの推定値が、当該パリティチェック行列にトラッピング集合が含まれている兆候を示す基準を満たしている場合、反復を続ける前にメッセージの少なくとも一部をリセットすることを含んでいる。

【0061】

本明細書に示す別の実施形態は、 $K$  個の情報ビットを  $N > K$  個の符号ワードビットとして符号化する符号ワードの表現を復号化する方法であって、本方法は (a) チャンネルから符号ワードの表現をインポートすること、(b)  $N - K$  行および  $N$  列を有するパリティチェック行列を提供すること、(c) 行と列との間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新することおよび、(d) 所定の失敗基準に照らして復号化が収束に失敗した場合、反復を続ける前に当該列から送られたメッセージの少なくとも一部を切り捨てることを含んでいる。

【0062】

本明細書に示す別の実施形態は、 $K$  個の情報ビットを  $N > K$  個の符号ワードビットとして符号化する符号ワードの表現を復号化するデコーダであって、(a)  $N - K$  行および  $N$  列を有するパリティチェック行列を提供すること、(b) 行と列との間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新すること、(c) (i) 所定の失敗基準に照らして復号化が収束に失敗し、且つ (ii) 符号ワードビットの推定値が、当該パリティチェック行列にトラッピング集合が含まれている兆候を示す基準を満たしている場合、反復を続ける前にメッセージの少なくとも一部をリセットすること、を含むステップにより、符号ワードの推定値を更新するアルゴリズムを実行することにより符号ワードの表現を復号化するプロセッサを含んでいる。

【0063】

本明細書に示す別の実施形態は、 $K$  個の情報ビットを  $N > K$  個の符号ワードビットとして符号化する符号ワードの表現を復号化するデコーダであって、(a)  $N - K$  行および  $N$  列を有するパリティチェック行列を提供すること、(b) 行と列との間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新すること、(c) 所定の失敗基準に照らして復号化が収束に失敗した場合、反復を続ける前に当該列から送られたメッセージの少なくとも一部を切り捨てること、を含むステップにより、符号ワードの推定値を更新するアルゴリズムを実行することにより符号ワードの表現を復号化するプロセッサを含んでいる。

【0064】

本明細書に示す別の実施形態はメモリコントローラであって、(a)  $K$  個の情報ビットを  $N > K$  個の符号ワードビットの符号ワードとして符号化するエンコーダと、(b) (i)  $N - K$  行および  $N$  列を有するパリティチェック行列を提供すること、(ii) 行と列との間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新すること、(iii) (A) 所定の失敗基準に照らして復号化が収束に失敗し、且つ (B) 符号ワードビットの推定値が、当該パリティチェック行列にトラッピング集合が含まれている兆候を示す基準を満たしている場合、反復を続ける前にメッセージの少なくとも一部をリセットすること、を含むステップにより、符号ワードの推定値を更新するアルゴリズムを実行することにより符号ワードの表現を復号化するプロセッサを含むデコーダとを含んでいる。

【0065】

本明細書に示す別の実施形態はメモリコントローラであって、(a)  $K$  個の情報ビットを  $N > K$  個の符号ワードビットの符号ワードとして符号化するエンコーダと、(b) (i)  $N - K$  行および  $N$  列を有するパリティチェック行列を提供すること、(ii) 行と列との

10

20

30

40

50



間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新すること、(iii)所定の失敗基準に照らして復号化が収束に失敗した場合、反復を続ける前に当該列から送られたメッセージの少なくとも一部を切り捨てること、を含むステップにより、符号ワードの推定値を更新するアルゴリズムを実行することにより符号ワードの表現を復号化するプロセッサを含むデコーダとを含んでいる。

【0066】

本明細書に示す別の実施形態は受信器であって、(a)通信チャネルから受信したメッセージを復調することによりK個の情報ビットを $N > K$ 個の符号ワードビットとして符号化する符号ワードの表現を生成する復調器と、(b)(i) $N - K$ 行およびN列を有するパリティチェック行列を提供すること、(ii)行と列との間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新すること、(iii)(A)所定の失敗基準に照らして復号化が収束に失敗し、且つ(B)符号ワードビットの推定値が、当該パリティチェック行列にトラッピング集合が含まれている兆候を示す基準を満たしている場合、反復を続ける前にメッセージの少なくとも一部をリセットすること、を含むステップにより、符号ワードの推定値を更新するアルゴリズムを実行することにより符号ワードの表現を復号化するプロセッサを含むデコーダとを含んでいる。

10

【0067】

本明細書に示す別の実施形態は受信器であって、(a)通信チャネルから受信したメッセージを復調することによりK個の情報ビットを $N > K$ 個の符号ワードビットとして符号化する符号ワードの表現を生成する復調器と、(b)(i) $N - K$ 行およびN列を有するパリティチェック行列を提供すること、(ii)行と列との間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新すること、(iii)所定の失敗基準に照らして復号化が収束に失敗した場合、反復を続ける前に当該列から送られたメッセージの少なくとも一部を切り捨てること、を含むステップにより、符号ワードの推定値を更新するアルゴリズムを実行することにより符号ワードの表現を復号化するプロセッサを含むデコーダとを含んでいる。

20

【0068】

本明細書に示す別の実施形態はメッセージを送受信する通信システムであって、(a)(i)当該メッセージのK個の情報ビットを $N > K$ 個の符号ワードビットの符号ワードとして符号化するエンコーダ、および(ii)通信チャネルを介して符号ワードを変調信号として送信する変調器を含む送信器と、(b)(i)通信チャネルから変調信号を受信して当該変調信号を復調することにより、符号ワードの表現を生成する復調器、および(ii)(A) $N - K$ 行およびN列を有するパリティチェック行列を提供すること、(B)行と列との間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新すること、(C)(I)所定の失敗基準に照らして復号化が収束に失敗し、且つ(II)符号ワードビットの推定値が、当該パリティチェック行列にトラッピング集合が含まれている兆候を示す基準を満たしている場合、反復を続ける前にメッセージの少なくとも一部をリセットすること、を含むステップにより、符号ワードの推定値を更新するアルゴリズムを実行することにより符号ワードの表現を復号化するプロセッサを含むデコーダを含む受信器とを含んでいる。

30

40

【0069】

本明細書に示す別の実施形態はメッセージを送受信する通信システムであって、(a)(i)当該メッセージのK個の情報ビットを $N > K$ 個の符号ワードビットの符号ワードとして符号化するエンコーダ、および(ii)通信チャネルを介して符号ワードを変調信号として送信する変調器を含む送信器と、(b)(i)通信チャネルから変調信号を受信して当該変調信号を復調することにより、符号ワードの表現を生成する復調器、および(ii)(A) $N - K$ 行およびN列を有するパリティチェック行列を提供すること、(B)行と列との間でのメッセージ交換を含むステップにより複数回の復号化反復で符号ワードビットの推定値を更新すること、(C)所定の失敗基準に照らして復号化が収束に失敗した場合、反復を続ける前に当該列から送られたメッセージの少なくとも一部を切り捨てること、

50

を含むステップにより、符号ワードの推定値を更新するアルゴリズムを実行することにより符号ワードの表現を復号化するプロセッサを含むデコーダを含む受信器とを含んでいる。

【 0 0 7 0 】

本明細書において、チャネルからインポートされた  $K$  個の情報ビットを  $N > K$  個の符号ワードビットとして符号化する符号ワードの表現を復号化する 4 種の一般的な方法を示す。

【 0 0 7 1 】

最初の 2 種の一般的な方法によれば、 $N$  個のビットノードおよび  $N - K$  個のチェックノードを含むグラフのビットノードとチェックノードとの間でメッセージを交換することにより、複数回の復号化反復で符号ワードビットの推定値を更新する。

10

【 0 0 7 2 】

第 1 の一般的な方法によれば、所定の失敗基準に照らして復号化が失敗し、且つ符号ワードビットの推定値が、当該パリティチェック行列にトラッピング集合が含まれている兆候を示す基準を満たしている場合、反復を続ける前にメッセージの少なくとも一部がリセットされる。

【 0 0 7 3 】

第 1 の一般的な方法のいくつかの実施形態において、グラフの少なくとも一部が複数の部分グラフに分割される。メッセージ交換の少なくとも一部が各部分グラフ内で別個に実行される。当該グラフにトラッピング集合が含まれるとする関連基準は、当該部分グラフのうち 1 個だけで復号化の収束が失敗することを含んでいる。

20

【 0 0 7 4 】

当該グラフにトラッピング集合が含まれる別の基準は、符号ワードビットの推定値の シンδροームの要素の高々約 1 パーセントが非ゼロ 且つ 2 個の連続的な反復において一定であることである。

【 0 0 7 5 】

メッセージの少なくとも一部をリセットすることは、好適にはチェックノードから送られるメッセージの少なくとも一部の設定、および / またはビットノードから送られるメッセージの少なくとも一部の切り捨てを含んでいる。最も好適には、リセットは、チェックノードから送られた全てのメッセージをゼロに設定する、および / またはビットノードから送られる全てのメッセージを切り捨てることを含んでいる。好適には、ビットノードから送られるメッセージは対数尤度比であり、そのうち切り捨てられるメッセージは、約 10 ~ 約 16 桁である。

30

【 0 0 7 6 】

第 2 の一般的な方法によれば、所定の失敗基準に照らして復号化が収束に失敗した場合、反復を続ける前にビットノードから送られたメッセージの少なくとも一部が切り捨てられる。

【 0 0 7 7 】

一つの好適な失敗の基準は、例えば所定回数の反復の後、または所定時間の後、あるいはビットノードとチェックノード間での所定回数のメッセージ交換の後で、符号ワードビット推定値の非ゼロであるシンδροームの要素が少なくとも所定個数であることを含んでいる。別の好適な失敗基準は、2 つの連続的な反復において符号ワードビット推定値の非ゼロであるシンδροームの要素が多くとも所定個数であることを含んでいる。好適な他の失敗基準は、連続的な 2 回の反復の後で、符号ワードビット推定値の非ゼロであるシンδροームの要素の数が所定の限度を下回ることを含んでいる。別の好適な失敗基準は、連続的な所定回数の反復の前後（例：1 回の反復の前後）で、符号ワードビット推定値間のハミング距離が所定の限度を下回ることを含んでいる。

40

【 0 0 7 8 】

好適には、ビットノードから送られた全てのメッセージが切り捨てられる。

【 0 0 7 9 】

50

好適には、メッセージは対数尤度比であり、そのうち切り捨てられるメッセージは、高々約10～約16桁が切り捨てられる。

【0080】

上述のように、LDPC復号化のグラフ図は図1に示すように行列表現と等価である。従って、第3および第4の一般的な方法によれば、符号ワードビットの推定値は、Nビットのベクトル要素を有するビットベクトルとN-K個のチェックベクトル要素を有するチェックのベクトルを接続するパリティチェック行列を用いて更新される。複数回の復号化反復において、符号ワードビットの推定値は、そのように接続されたビットベクトル要素とチェックベクトル要素との間でメッセージを交換することにより更新される。

【0081】

第3の一般的な方法によれば、所定の失敗基準に照らして復号化が失敗し、且つ符号ワードビットの推定値が、当該パリティチェック行列にトラッピング集合が含まれている兆候を示す基準を満たす場合、反復を続ける前にメッセージの少なくとも一部がリセットされる。

【0082】

第4の一般的な方法によれば、所定の失敗基準に照らして復号化が収束に失敗した場合、反復を続ける前に当該列から送られたメッセージの少なくとも一部が切り捨てられる。

【0083】

4種の一般的な方法の一つに対応するデコーダは、対応する一般的な方法により符号ワードビット推定値を更新するアルゴリズムを実行することにより符号ワードの表現を復号化する1個以上のプロセッサを含んでいる。

【0084】

4種の一般的な方法の一つに対応するメモリコントローラは、K個の情報ビットを $N > K$ ビットの符号ワードとして符号化するエンコーダと、一般的な方法に対応するデコーダとを含んでいる。通常、このようなメモリコントローラは、符号ワードの少なくとも一部をメインメモリに保存すると共に、当該メインメモリから当該符号ワードの（おそらくノイズが多い）少なくとも一部の表現を取得する回路を含んでいる。4種の一般的な方法の一つに対応するメモリ装置は、このようなメモリコントローラを含むと共に、メインメモリをも含んでいる。

【0085】

4種の一般的な方法の一つに対応する受信器は、通信チャネルから受信したメッセージを復調する復調器を含んでいる。復調器は、K個の情報ビットを $N > K$ 個の符号ワードビットとして符号化する符号ワードの表現を提供する。そのような受信器はまた、一般的な方法に対応するデコーダを含んでいる。

【0086】

4種の一般的な方法の一つに対応する通信システムは、送信器および受信器を含んでいる。送信器は、メッセージのK個の情報ビットを $N > K$ 個の符号ワードビットの符号ワードとして符号化するエンコーダと、通信チャネルを介して当該符号ワードを変調信号として送信する変調器を含んでいる。受信器は、一般的な方法に対応する受信器である。

【0087】

添付の図面を参照しながら、各種の実施形態についてあくまでも例示目的で記述する。

【図面の簡単な説明】

【0088】

【図1】LDPC符号が疎なパリティチェック行列または疎な二部グラフとしてどのように表現できるかを示す。

【図2】フラッディングスケジュール確率伝播アルゴリズムを示す。

【図3】従来の連続スケジュール確率伝播アルゴリズムを示す。

【図4】エラーフロアを示す。

【図5】部分グラフ内および部分グラフと外部チェックノードの集合との間でどのようにメッセージが交換されるかを示す。

10

20

30

40

50

【図 6】部分グラフ内および部分グラフと外部チェックノードの集合との間でメッセージが交換される確率伝播アルゴリズムを示す。

【図 7 A】図 6 のアルゴリズムを実装するデコーダの高水準モードブロック図である。

【図 7 B】図 6 のアルゴリズムを実装するデコーダの高水準モードブロック図である。

【図 8】図 1 の疎な二部グラフを部分グラフに分割する 2 通りの方法を示す。

【図 9】図 1 の疎な二部グラフを部分グラフに分割する 2 通りの方法を示す。

【図 10】コントローラが図 7 A のデコーダを含むフラッシュメモリ装置の高水準モードブロック図である。

【図 11】図 10 の詳細図である。

【図 12】受信器が図 7 A のデコーダを含む通信システムの高水準モードブロック図である

10

【発明を実施するための形態】

【0089】

複雑度が低い L P D C 復号化およびトラッピング集合に起因する非収束を克服する L P D C 復号化の原理および動作は、図面および付随する記述を参照することでよりよく理解されよう。

【0090】

L D P C 符号用の従来型デコーダにおいて、デコーダが必要とするメモリは符号長  $N$  (符号の基礎となるグラフ  $|V|$  の可変ノードの数に等しい) および符号の基礎となるグラフ  $|E|$  の辺の数に比例する。効率的な実装において (例えば、連続的にスケジュールされたデコーダに基づいて)、必要とされるメモリは  $(|V| + |E|) \cdot b \cdot p \cdot m$  ビットと同程度に小さくてよく、ここに  $|V|$  はビット推定値の数、 $|E|$  は辺メッセージの数、 $b \cdot p \cdot m$  はデコーダのメモリに保存されている 1 メッセージ当たりのビット数である (ここでは簡便のため、ビット推定値および辺メッセージの保存に同じビット数が必要とされると仮定しているが必ずしもそうでなくてもよい点に注意されたい)。十分な復号化時間可以利用できると仮定すれば、本明細書に開示するデコーダは従来型デコーダと比較して、 $|V|$  個のビット推定値および  $|E|$  個の辺メッセージの一部だけを同時に保存することで、デコーダの誤り訂正能力を全く低下させることなく、復号化の実装に用いるメモリがはるかに小さくて済む。これは、適切な復号化スケジュールを採用し、且つ本明細書に記述する復号化ハードウェアを用いることにより実現される。

20

30

【0091】

本明細書に記述する方法およびデコーダは、符号を表す基礎となるグラフをいくつかの区間に分割することにより動作し、グラフの異なる区間を一度に 1 個以上順次処理することによりメッセージ受け渡し復号化アルゴリズムを実行する。復号化実行中の各段階で、現在処理されているグラフ区間 (群) に対応するビット推定値および辺メッセージだけが保存される。このようにして、極めて長い L D P C 符号は採用するため、複雑度が低い復号化ハードウェアを利用しながら、ほぼ最適な誤り訂正能力および極めて低いエラーフロアが得られる。

【0092】

本明細書に開示するデコーダは、主に以下の三つの理由により、メモリ装置での使用に極めて適している。

40

【0093】

1. 低い E C C エラーフロアは、厳しいデコーダ出力 B E R 要件 ( $< 10^{-15}$ ) があるメモリ装置において特に重要である。短い符号を用いる場合、このような低いエラーフロアを実現することは極めて難しく、通常は符号の誤り訂正能力を犠牲にすることを必要とするが、これは符号の長さが短いために既に下げられている。従って、等価な長い符号を用いて符号の誤り訂正能力が向上し、従って保存されたデータを損なう所与のメモリ「ノイズ」から情報を保護するためにより低い E C C 冗長性が必要とされる。より大量の情報を所与の数のメモリセル (または所与のメモリシリコンサイズを用いて) に保存することができるため、これは結果的にメモリの費用対効果を向上させる。従って、メモリ装置の

50

長 ECC を使用することで大きい利点が見られるものと期待される。

【 0 0 9 4 】

2. 本明細書に開示する L D P C 方法は、一度にグラフ全体ではなく、各処理フェーズで符号の基礎となるグラフの区間を処理することを可能にする。このことは、各フェーズで「弱い」ビット推定値の全てではなく、「弱い」ビット推定値の一部だけを保存することができることを意味する。ここで、「弱い」ビット推定値という用語は、ストレージ（恐らくはフラッシュ装置）からの読み出しから推論される各々の保存されたビットの推定値「 $y$ 」の信頼度を記述する一群のビットを指す。

【 0 0 9 5 】

現在必要とされるビット観測値（ $y$ ）だけを記憶装置から読み出すことができ、従って ECC 復号化を実装するためにメモリコントローラ内に大きいバッファを必要としないため、この特徴をメモリ装置で容易に利用することができる。あるいは、全てのビット観測値（ベクトル  $\underline{v}$  で表す）を一度にメモリから読み出す場合でも、これらの保存に必要なバッファは通常、デコーダが必要とするビット観測値（ $P_v$  個のメッセージ）の保存に必要なメモリよりもはるかに小さい。このように、現在デコーダにより処理中であるグラフ区間に対応する弱いビット推定値の一部だけが毎回生成される結果、必要とされるデコーダのメモリが小さくて済む。

【 0 0 9 6 】

例えば、各セルが単一のビット  $v$  を保存し、各セルから読み出された状態  $y$  が 0 または 1 のいずれかである S L C フラッシュメモリ装置を考える（1セル当たり1ビットを保存するフラッシュメモリ装置である。「S L C」は「単一レベルセル」を意味するが、実際には各セルが2レベルをサポートするため誤った名称である。「S L C」の「S」はプログラミングされたレベルが一つしかないことを示す）。すると、読み出されたセル状態のベクトル  $\underline{v}$  の保存に必要なメモリは  $N$  ビットである。一方、全ての弱いビット推定値（ $P_v$  個のメッセージ）の保存により大きいメモリが必要となり得る（例えば、各 L L R 推定値が 6 ビットに保存される場合は  $6 N$  ビット）。従って、各デコーダ起動において必要とされる弱いビット推定値だけを生成する方がより効率的である。あるビット  $v$  に対する L L R ビット推定値  $P_v = \log ( P_r ( v = 0 | y ) / P_r ( v = 1 | y ) )$  を、メモリ「ノイズ」に関する先験的な知識に基づいてフラッシュメモリ装置から読み出された対応するビット観測値  $\underline{v}$  から生成することができる。換言すれば、メモリ「ノイズ」統計値を知ることにより、「 $y$ 」がセルから読まれる前提で、特定のメモリセルに保存されたビット  $v$  が 0 / 1 である確率を推論することができる。

【 0 0 9 7 】

例えば、特定の S L C フラッシュメモリ装置において、プログラムされたものとは異なるセルの状態を読み出す確率を  $p = 10^{-2}$  とすれば、 $y = 0$  ならば  $P_v = \log ( ( 1 - p ) / p ) = 4.6$ 、および  $y = 1$  ならば  $P_v = \log ( p / ( 1 - p ) ) = -4.6$  である。更に、セルが単一ビット（1個の「強いビット」）を保存し、当該装置は2個の「弱いビット」に等価な8個の閾値電圧レベルを読み出すように構成されているためにフラッシュ装置の各セル（「 $\underline{v}$ 」で表す）から読み出すことができる状態の数が8である場合、コントローラ内で3ビットのストレージを必要とする各要素「 $\underline{v}$ 」が、3ビットより多い、例えば6ビット（B P M = 1 メッセージ当たりのビット数 = 6）として表現することができる L L R 値  $P_v$  に変換される。これら6ビットは、フラッシュセルから読み出された2個の弱いビットとは逆に弱いビット推定値であって、この6ビットの L L R 値に対応している。

【 0 0 9 8 】

3. 本明細書に示す種類の復号化スケジュールは、（従来の復号化スケジュールと比較して）必要なメモリがより小さくて済む。しかし、本明細書に開示する復号化スケジュールでは、デコーダ収束速度が遅くなり、特にデコーダの最大誤り訂正能力付近で動作しているときに復号化時間が増大する恐れがある。このようなデコーダは、可変 ECC 復号化待ち時間を許容できるメモリ装置には極めて適している。例えば、損傷ビットの数が多

10

20

30

40

50

ために、保存された正しい符号ワードに ECC が収束するために必要とされる復号化時間が長い場合、メモリコントローラは以前に読まれた符号ワードの復号化が終了するまでメモリからの読み出しを止めることができる。フラッシュメモリ装置の寿命の大半では、メモリ「ノイズ」が小さく、且つ損傷ビットの数が少ない点に注意されたい。従って、デコーダは効率的且つ迅速に動作するため、パイプライン化された効率的なメモリ読み出しが可能になる。稀に、メモリから読み出された損傷ビットの数が多く、復号化時間が長引いた結果、読み出パイプラインがストールすることがある。従って、これらの可変的な復号化時間特徴にも拘わらず、平均的にはスループットは影響を受けないままである。

#### 【 0 0 9 9 】

実施形態の一つのクラスによれば、符号を表す二部グラフ  $G = (V, C, E)$  が以下のよういくつかの区間に分割される。すなわち、1) ビットノードの集合  $V$  を  $t$  個の互いに素な部分集合  $V_1, V_2, \dots, V_t$  (但し  $V = V_1 \cup V_2 \cup \dots \cup V_t$ ) に分割する。2) ビットノードの各部分集合  $V_i$  について、 $V_i$  内のビットノードだけに接続しているチェックノードの全てを含むチェックノードの部分集合  $C_i$  を形成する。3) これまでに形成されたチェックノードの部分集合のいずれにも存在しないチェックノードの全てを含む外部チェックノードの部分集合  $C_J$  すなわち  $C_J = C \setminus (C_1 \cup C_2 \cup \dots \cup C_t)$  を形成する。4) グラフ  $G$  を  $G_i = (V_i, C_i, E_i)$  のように  $t$  個の部分グラフ  $G_1, G_2, \dots, G_t$  に分割する。ここに  $E_i$  は  $V_i$  内のビットノードと  $C_i$  内のチェックノードとの間で接続している辺の集合である。集合  $C_J$  に接続している辺を  $E_J$  で表す ( $E_J = E \setminus (E_1 \cup E_2 \cup \dots \cup E_t$  に注意)。

#### 【 0 1 0 0 】

これらの実施形態において、復号化フェーズを反復的に実行することによりグラフ  $G$  は特別なメッセージ受け渡しスケジュールに従い処理され、各復号化フェーズは以下の順序でグラフの辺に沿ってメッセージを交換する。

#### 【 0 1 0 1 】

$i = 1$  から  $t$  まで

1. 図 5 に  $R_{C_J V_i}$  メッセージと表記された  $R_{C_J V_i}$  個のメッセージを  $E_J$  内の辺に沿ってチェックノード  $c \in C_J$  からビットノード  $v \in V_i$  へ送る。図 5 に  $R_{C_i V_i}$  メッセージと表記された、チェックノード  $c \in C_i$  からビットノード  $v \in V_i$  への  $R_{C_i V_i}$  個のメッセージをゼロに設定する。全てのビット  $v \in V_i$  について、初期ビット推定値を図 5 に  $P_{V_i}$  メッセージと表記された  $P_{V_i}$  に設定する。メッセージ  $R_{C_J V_i}$  が当該ステップより前に他の  $t - 1$  個の部分グラフ  $G_k$  ( $k \neq i$ ) のためにデコーダを起動した結果である点に注意されたい。他の部分グラフが未だ処理されていない場合、図 5 に示すそれらの対応メッセージ  $Q_{V_i C_J}$  は  $P_{V_i}$  すなわちメモリから読み出されたかまたは通信チャネルから受信された推定値に設定される。これらがパンクチャードビットである場合、その  $P_{V_i}$  はゼロである。

#### 【 0 1 0 2 】

2. 何らかのスケジュールに従い (例:  $C_i$  内のチェックノードを連続的に辿って各チェックノードについて当該チェックノードとの間でメッセージをやり取りすることにより実行される図 3 に記述する連続スケジュールに従い)、 $E_i$  内の辺に沿って  $V_i$  内のビットノードから  $C_i$  内のチェックノードへ  $Q_{V_i C_i}$  個のメッセージを、および  $C_i$  内のチェックノードから  $V_i$  内のビットノードへ  $R_{C_i V_i}$  個のメッセージを送ることにより 1 回以上の反復を実行する。これを図 5 に  $Q_{V_i C_i}$  および  $R_{C_i V_i}$  メッセージと表記する。

#### 【 0 1 0 3 】

3. 図 5 に  $Q_{V_i C_J}$  メッセージと表記された  $Q_{V_i C_J}$  個のメッセージを  $E_J$  内の辺に沿って  $V_i$  内のビットノードから  $C_J$  内のチェックノードへ送る。

#### 【 0 1 0 4 】

復号化は、デコーダが全てのパリティチェック制約を満たす有効な符号ワードに収束するまで、または許された復号化フェーズの最大数に達するまで続けられる。各部分グラフ  $i$  内でのメッセージの受け渡しの停止基準も同様である。当該部分グラフ内のパリティチェック制約が全て満たされるか、または許された最大反復回数に達するまで繰り返す。一

10

20

30

40

50

般に、許された最大反復回数は、部分グラフ毎に、またはデコーダの起動毎に変化する場合がある。

#### 【0105】

$E_j$ の辺に沿って送られたメッセージ（図5の $R_{c_j v_i}$ メッセージおよび $Q_{v_j c_j}$ メッセージ）を用いてグラフの異なる区間同士で情報を交換する。復号化を実行する間に各ステージで送られたメッセージを、メッセージ受け渡し復号化アルゴリズムの標準的な計算規則により計算することができる。例えば、BP復号化が実装されている場合、メッセージは式(4)および(5)に従い計算される。Min-Sumアルゴリズム、Gallagher-Aアルゴリズム、およびGallagher-Bアルゴリズム等、他のメッセージ受け渡し復号化アルゴリズムは自身の計算規則を有している。

10

#### 【0106】

このような復号化アルゴリズムを、BP復号化を実装している各部分グラフ内で連続的にスケジュールされたメッセージ受け渡し復号化を仮定して図6に要約する。このアルゴリズムでは、ビットノード $v \in V_i$ に対応する $Q_v$ 個のメッセージだけを復号化する間の各段階において、 $E_i$ の辺に対応する $R_{c_v}$ 個のメッセージおよび $E_j$ の辺に対応するメッセージが保存される。従って、このクラスの実施形態のデコーダは、効率的な従来型デコーダにおける $(|V| + |E|)$ 個のメッセージに対して、 $(\max\{|V_1|, |V_2|, \dots, |V_i|\} + \max\{|E_1|, |E_2|, \dots, |E_i|\} + |E_j|)$ 個のメッセージだけを同時に保存すればよい。従って、必要なメモリは、従来型デコーダが必要とするメモリの $\sim 1/t$ 分数である。長いLDPC符号を実装する場合、これはデコーダの複雑度に対して大きい利点をもたらす。

20

#### 【0107】

このクラスの実施形態による例示的なデコーダ30の高水準モードブロック図を図7Aに示す。デコーダ30は、以下のものを含んでいる。

#### 【0108】

1. メモリから読み出されたかまたは通信チャネルから受信した対応ビット観測値 $y_i = [y_v : v \in V_i]$  ( $y_v$ はビット $v$ に対応する観測値)に基づいて、現在処理中である部分グラフ $G_i = (V_i, C_i, E_i)$ 内のビット $v \in V_i$ について初期ビット推定値 $P_i = [P_v : v \in V_i]$ を計算する初期LLR計算ブロック32。

#### 【0109】

2. 現在処理中である部分グラフ( $P_v$ メッセージとして初期化された $Q_v$ メッセージ)内のビットノード $v \in V_i$ に対するビット推定値を保存するメモリ領域36を含む読み出し/書き込みメモリ34。

30

#### 【0110】

3. 以下を含む読み出し/書き込みメモリ35

3a. 現在処理中である部分グラフの辺集合 $E_i$ に対応する $R_{c_v}$ 個のメッセージを保存するメモリ領域38。

#### 【0111】

3b.  $E_j$ の辺に沿ってメッセージを保存するメモリ領域40。メモリ領域40は、i) ビットノード $v \in V_i$ ,  $i' \in \{1, \dots, n\} \setminus i$ からチェックノード $c \in C_j$ への $Q_{v c}$ 個のメッセージ( $i$ は現在処理中である部分グラフの添え字)を保存すると共に、ii) ビットノード $v \in V_i$ についてメモリ領域40は最初にチェックノード $c \in C_j$ からの $R_{c_v}$ 個のメッセージを保存し、その後で当該部分グラフを処理しているメモリ領域40は $Q_{v c}$ をチェックノード $c \in C_j$ に保存する。

40

#### 【0112】

4. (図6に示す)メッセージの更新に関わる計算を実行する処理ユニット42。

#### 【0113】

5. メモリ34と処理ユニット42との間でメッセージをルーティングするルーティング層44。例えば、このクラスの実施形態のいくつかのサブクラスにおいて、図6の部分グラフ $G_1 \sim G_t$ にわたるループ内で、ルーティング層44は各プロセッサ42に現在の

50

部分グラフ  $G_i$  の当該プロセッサ自身のチェックノードを割り当て、チェックノードの処理は  $G_i$  の全てのチェックノードについて（または、プロセッサ 42 の数と同数の、 $G_i$  のチェックノードについて）並列に実行される。

【0114】

6. 符号のグラフ構造を保存する読み出し専用メモリ (ROM) 46。メモリのアドレス指定およびルーティング層 44 によるスイッチングは ROM 46 のエントリに基づく。

【0115】

デコーダ 30 は複数の処理ユニット 42 を含んでいるため、メッセージの更新に関わる計算を並列に実行することができる。単一の処理ユニット 42 しか備えていない別の実施形態はルーティング層 44 を含んでいない。

【0116】

上述のように、連続的な受け渡しスケジュールは、チェックノードまたはビットノードのいずれかが連続的に辿る。図 7A のデコーダ 30 はチェックノードを連続的に辿る。図 7B は、ビットノードを連続的に辿る同様のデコーダ 31 の高水準モードブロック図である。

【0117】

このクラスの実施形態によるグラフ分割の一例を図 8 に示す。全てのビットノードが 2 個のチェックノードに接続され、全てのチェックノードが 4 個のビットノードに接続されるように、18 個のビットノードおよび 9 個のチェックノードを有する規則的な二部グラフにより記述される LDPC 符号を本例に用いる。これは長さが 18、レートが  $1/2$  の LDPC 符号である。元のグラフを図 8 の左側に示す。これは図 1 のグラフでもある。自身のビットノード、チェックノード、および辺を部分集合に分割した後のグラフを図 8 の右側に示す。これが同じグラフを明快さのために再構成したものに過ぎない点に注意されたい。この符号の場合、従来技術の効率的なデコーダは  $18 + 36 = 54$  個のメッセージを保存する必要がある一方、対応するデコーダ 30 は  $6 + 8 + 12 = 26$  個のメッセージだけを保存すればよい。そのため、同じ誤り訂正能力を維持しながらデコーダのメモリ複雑度が 52% 減少される。

【0118】

図 8 の例のように、全ての部分グラフがトポロジ的に同一であることが好適である。この文脈において、「トポロジ的同一性」は、全ての部分グラフが同数のビットノードおよび同数のチェックノードを有し、各々のビットノードが内部チェックノードへの接続性に関して他の全ての部分グラフに対応するビットノードを有し、各部分グラフのチェックノードは、ビットノードへの接続性に関して他の全ての部分グラフに対応するチェックノードを有することを意味する。例えば図 8 において以下が成り立つ。

【0119】

ビットノード 1 および 5 が部分グラフ 1 の両方のチェックノードに接続し、ビットノード 11 および 16 が部分グラフ 2 の両方のチェックノードに接続し、ビットノード 13 および 17 が部分グラフ 3 の両方のチェックノードに接続して、且つこれらのビットノードのいずれも外部チェックノード（集合  $C_j$  のチェックノード）に接続していないため、ビットノード 1、5、11、13、16 および 17 は対応している。

【0120】

残りのビットノードは、これらのビットノードの各々が同一部分グラフの 1 個のチェックノードに接続しているため、対応している。

【0121】

部分グラフの全てのチェックノードは、これらのチェックノードの各々が部分グラフのチェックノードだけに接続している自身の部分グラフの 2 個ビットノード、および外部チェックノードにも接続している自身の部分グラフの他の 2 個のビットに接続しているため、対応している。

【0122】

部分グラフが「トポロジ的に同一」であるために外部チェックノードに対して同一の接続性を有する必要がある点に注意されたい。例えば、同一の外部チェックノード 7 に接

10

20

30

40

50



続している部分グラフ 3 の 2 個のビットノード 15、18 はまた、部分グラフ 3 の同一チェックノード 9 にも接続しているが、同一外部チェックノード 2 に接続している部分グラフ 1 の 2 個のビットノード 4、12 は部分グラフ 1 の異なるチェックノード 3、8 に接続している。

#### 【0123】

しかし、必要ならばグリーディアルゴリズムにより任意の LDPC グラフ  $G$  を部分グラフに分割することができる。第 1 の部分グラフは、ビットノードの任意の集合を選択することにより構築される。第 1 の部分グラフのチェックノードは、それらのビットノードだけに接続するチェックノードである。第 2 の部分グラフは、残りのビットノードの中から任意のビットノードの集合を選択することにより構築される。無論、第 2 の部分グラフ内のビットノードの数が第 1 の部分グラフのビットノードの数に等しいことが好適である。ここでも、第 2 の部分グラフのチェックノードは、第 2 の部分グラフのビットノードだけに接続するチェックノードである。ビットノードの任意選択は必要な回数だけ繰り返される。最後の部分グラフは選択されなかったビットノードおよびそれらのビットノードだけに接続するチェックノードを含んでいる。残りのチェックノードが  $C_j$  を構成する。

#### 【0124】

上述の実施形態のクラスにおいて、LDPC グラフ  $G$  は、各々が自身のビットノードおよびチェックノードを有する  $t$  個の部分グラフに加え、チェックノードだけからなる別個の部分集合  $C_j$  に分割される。別のクラスの実施形態において、図 9 に示すように、 $G$  は各々が自身のビットノードおよびチェックノードを有するちょうど  $t$  個の部分グラフに分割される。例えば上述のグリーディアルゴリズムを用いて、最後の部分グラフ ( $G_t$ ) は選択されなかったビットノード、これらのビットノードだけに接続するチェックノード、および残り全てのチェックノードを含んでいる。これは、部分グラフのビットノードから分離されたビットノードの自身の部分集合に接続している第 1 のクラスの実施形態の集合  $C_j$  と同等である。このクラスの実施形態において、図 6 のアルゴリズムは、部分グラフループ内の部分グラフ  $G_1 \sim G_{t-1}$  だけを含め、且つ専ら  $G_t$  内だけに切り離されたメッセージ交換により部分グラフループを辿ることにより各復号化フェーズを終了することにより変更されている。図 9 に  $t = 4$  の場合を示す。これらの実施形態の 1 個の下位クラスにおいて、いくつかのビットはパンクチャードビットであり、 $G_t$  はこれらのビット専用である。 $G_t$  の全てのビットはパンクチャードビットであり、全てのパンクチャードビットは  $G_t$  のビットである。

#### 【0125】

図 10 は、フラッシュメモリ装置の高水準モードブロック図である。行列に配置された複数のメモリセル  $M$  を含むメモリセルアレイ 1 は、列制御回路 2、行制御回路 3、 $c$ -ソース制御回路 4、および  $c$ - $p$  ウェル制御回路 5 により制御される。列制御回路 2 はメモリセルアレイ 1 のビット線 ( $BL$ ) に接続されていて、メモリセル ( $M$ ) に保存されたデータを読み、書込み動作中のメモリセル ( $M$ ) の状態を判定し、ビット線 ( $BL$ ) の電位レベルを制御して書込みを実行するかまたは書込みを禁止する。行制御回路 3 はワード線 ( $WL$ ) に接続されていて、ワード線 ( $WL$ ) の一つを選択し、読み出し電圧を印加し、列制御回路 2 により制御されたビット線の電位レベルと組み合わせられた書込み電圧を印加し、メモリセル ( $M$ ) が形成されている  $p$  型領域の電圧に結合された消去電圧を印加する。 $c$ -ソース制御回路 4 は、メモリセル ( $M$ ) に接続している共通ソース線を制御する。 $c$ - $p$  ウェル制御回路 5 は、 $c$ - $p$  ウェル電圧を制御する。

#### 【0126】

メモリセル ( $M$ ) に保存されたデータは、列制御回路 2 により読み出されて、入出力線およびデータ入力/出力バッファ 6 を介して外部入出力線へ出力される。メモリセルに保存されるプログラムデータは、外部入出力線を介してデータ入出力バッファ 6 に入力され、列制御回路 2 へ転送される。外部入出力線は、コントローラ 20 に接続している。

#### 【0127】

フラッシュメモリ装置を制御する命令データが、コントローラ 20 に接続している外部

10

20

30

40

50

制御線に接続している命令インタフェースに入力される。命令データは、要求されている動作をフラッシュメモリに知らせる。入力命令は、列制御回路2、列制御回路3、c - ソース制御回路4、c - pウェル制御回路5、およびデータ入力/出力バッファ6を制御する状態機械8へ転送される。状態機械8は、READY / BUSYまたはPASS / FAIL等、フラッシュメモリの状態データを出力することができる。

#### 【0128】

コントローラ20は、パーソナルコンピュータ、デジタルカメラ、形態情報端末等のホストシステムに接続しているかまたは接続可能である。メモリアレイ1にデータを保存または読み出す等の命令を発行し、且つそのようなデータを提供または受け取るのはホスト側である。コントローラ20はそのような命令を、命令回路7により解釈および実行可能な命令信号に変換する。コントローラ20はまた、通常はメモリアレイに書き込まれるかまたは読み出されているユーザーデータ用のバッファメモリを含んでいる。典型的なメモリ装置は、コントローラ20、各々がメモリアレイおよび付随する制御、入出力および状態機械回路を含む1個以上の集積回路チップ22を含む1個の集積回路チップ21を含んでいる。このトレンドは無論、そのような装置のメモリアレイとコントローラ回路を1個以上の集積回路チップ上に集積することである。メモリ装置は、ホストシステムの一部として埋め込まれていても、またはホストシステムの嵌め込みソケットに着脱自在に挿入可能なメモリカードに含まれていてもよい。そのようなカードはメモリ装置全体を含んでいても、またはコントローラとメモリアレイが付随する周辺回路と共に別個のカードで提供されてもよい。

#### 【0129】

図11は図10の一部の拡大図であり、コントローラ20が、ホストから1個以上の符号ワードとして受信したユーザーデータを符号化するエンコーダ52、符号ワードを（または符号ワードのいずれかのビットがパンクチャードビットである場合その非パンクチャードビットだけを）をメモリセル配列1に保存するよう命令回路7に指示すると共に、保存された符号ワード（またはパンクチャードビットの場合はその保存された部分）をメモリセルアレイ1から取り出すよう命令回路7に指示する回路54、および回路54により取得された符号ワードの表現を復号化するデコーダ30を含んでいることを示す。あるいは、コントローラ20はデコーダ30の代わりにデコーダ31を含んでいてもよい。

#### 【0130】

本明細書に開示する方法およびデコーダは主としてデータ記憶システムで用いることを意図しているが、これらの方法およびデコーダはまた通信システム、特に高周波を大幅に減衰させる媒体を介した波伝播に依存する通信システムにも適用できる。そのような通信は本質的に遅く且つノイズが多い。そのような通信の一例として、沿岸局と水面下の潜水艦との間の無線通信がある。

#### 【0131】

図12は、送信器110、チャネル103、および受信器112を含む通信システム100の高水準モードブロック図である。送信器110はエンコーダ101および変調器102を含んでいる。受信器112は復調器104およびデコーダ30を含んでいる。エンコーダ101はメッセージを受信して対応する符号ワードを生成する。変調器102は、生成された符号ワードにBPSK、QPSKまたは多値QAM等のデジタル変調を施し、その結果得られた変調信号をチャネル103経由で受信器112に送信する。受信器112において、復調器104はチャネル103から変調信号を受信し、受信した変調信号に対しBPSK、QPSKまたは多値QAM等のデジタル復調を施す。デコーダ30は、その結果得られた上述のような元の符号ワードの表現を復号化する。あるいは、受信器112はデコーダ30の代わりにデコーダ31を含んでいてもよい。

#### 【0132】

ここでトラッピング集合の問題に注目すれば、LDPC復号化におけるトラッピング集合を克服する2種類の従来方法がある。

#### 【0133】

1. トラッピング集合無しでLDPC符号を設計することによりトラッピング集合を回避する。

【0134】

2. 復号化を行う間、アルゴリズム手段によりトラッピング集合を克服する。

【0135】

第1の種類の従来方法には以下の短所がある。

【0136】

トラッピング集合が明確に定義されておらず、且つ長いLDPC符号が極めて複雑であるため、エラーフロアが低いグラフを設計して、エラーフロアが低いことを証明するのは、広範囲なシミュレーション要する困難なタスクになる恐れがある。更に、そのようなアプローチにより、符号化/復号化スキームにおける実装の複雑度、復号化の速度および柔軟性等、他の態様に関して良好な特性を示すある種のLDPC符号の利用が除外される恐れがある。

10

【0137】

復号化を行う間にトラッピング集合を克服するアルゴリズム的方法を用いる第2の種類の従来方法に関しては以下が言える。

【0138】

文献においていくつかの方法が提案されている。

1. 平均化。

2. 通知された動的スケジューリング

20

3. トラッピング集合を識別して、それを回避すべくカスタマイズされた積和アルゴリズムを設計する。

【0139】

1. 平均化方法は、ビット値に対する更新アルゴリズムを用いる。更新は、先行する反復の結果だけでなく、いくつかの反復の結果の平均にも基づいている。算術平均、幾何平均、および重み付き算術幾何平均を含むいくつかの平均化方法が提案されている。

【0140】

2. 通知された動的スケジューリング。この方法では、毎回の反復で全てのチェックノードが更新される訳ではなく、グラフ内のメッセージの現在状態に基づいて更新される次のチェックノードを選択する。チェックノードは、当該チェックノードの更新が復号処理にとってどの程度有用であるかを測定する尺度に基づいて選択される。

30

【0141】

両方法共にエラーフロアの向上を実現することができるが、平均化のために先行メッセージの履歴を保存する必要があり、通知された動的スケジューリングは高い計算量をもたすため、アルゴリズムに付随する複雑度が高い。

【0142】

第3の種類の方法は、トラッピング集合の識別および各グラフ用に特化されたアルゴリズムを必要とするため、特に同一アプリケーションで複数のLDPC符号が考慮する場合に、特定のシナリオにこれらを使用するのは限界がある。

【0143】

40

現在述べている革新的な方法によれば、符号ワードの復号化は2段階のフェーズで実行される。第一フェーズを行う間、従来の復号化はLDPC符号により定義されるグラフに沿って実行される。

【0144】

トラッピング集合の存在が疑われ、復号処理が妥当な符号ワード（すなわち全てのパリティチェック式を満たす符号ワード）に収束することが阻害される場合、復号化の第2フェーズに移行する。このフェーズでは、当該符号のグラフのノードに関連付けられた値のいくつかが変更される。

【0145】

トラッピング集合が存在することは少数のビットが正しく収束することに失敗している

50

ことを示唆するため、トラッピング集合の存在を識別することができるのは、連続的に復号化を反復する間に少数を除く全てのビットが安定しているか、または他の全てのパリティチェック式が満たされる一方で少数のパリティチェック式が失敗する場合である。例えば、上述のように分割されたグラフの部分グラフのうち 1 個だけのパリティチェック式が失敗したならば、当該部分グラフはトラッピング集合であるかまたは含んでいることが疑われる。トラッピング集合の存在を示唆する別の兆候は、1 パーセント以下のパリティチェック式だけが一貫して失敗していることである。例えば、シンδροーム  $H \cdot v'$  ( $v'$  は推定されたビットの列ベクトル) のいくつかの要素が非ゼロであって、連続的な 2 回の反復において同一であることは、トラッピング集合の存在を示唆する。

【0146】

10

このような変更の二つの例は以下の通りである。

【0147】

1. チェックノードの値  $R_{cv}$  をゼロにリセットする。

【0148】

2. ビット確率の弱い値  $Q_v$  を切り捨てる、すなわちビット確率に対応する弱い値  $Q_v$  の大きさを所定の値、典型的には 10 ~ 16 の間の値に制限する。

【0149】

この方法論の背景にある動機は、反復処理を行う間に誤ったビットが高い確率を実現し、且つ誤った結果 (パリティチェック式に対応するノードで含まれる) の信頼度もまた高い場合に小さいトラッピング集合に起因して収束の失敗が生じることである。このような状況では、更に反復しても、誤ったビットに対してなされた強い決定 (好適には、弱い値の符号として実装される) は変わらない。

20

【0150】

しかし、デコーダが、小さいトラッピング集合に含まれない全てのビットが既にそれらの正しい値に設定されている初期状態で動作を開始していた場合、符号ワードを正しく復号化する確率は極めて高い。

【0151】

メッセージ  $R_{cv}$  の値をゼロにリセットすることにより、トラッピング集合に含まれない全てのビットが正しい状態に戻る。

【0152】

30

この状況において、正しく復号化されているビット (大部分のビットはこの段階にある) に関連するメッセージ  $Q_{vc}$  および  $R_{cv}$  は急速に高い信頼度値まで上昇する一方、トラッピング集合内のビットに関するメッセージは遅く上昇するため、正しいメッセージからのトラッピング集合内のビットに対応する値に対してより大きな影響がある。そのような手順は、トラッピング集合内のビットの値を訂正する際に有用である。

【0153】

この手順は、従来の LDPC 復号化アルゴリズムに最小限の複雑度が加わるに過ぎない。

【0154】

一実施形態において、本アルゴリズムは限られた回数の反復について復号化を実行する。収束に失敗した場合、本アルゴリズムは、 $R_{cv}$  メッセージのいくつかまたは全て等、特定の変数をゼロに設定するステップを追加し、ついで従来の復号化を続ける。

40

【0155】

別の実施形態では、限られた回数の反復を実行した後で、 $Q_v$  値のいくつかまたは全て等、いくつかの変数に対して切捨て動作を追加し、次いで本アルゴリズムは従来の復号化を続ける。

【0156】

両方のアルゴリズムは極めて簡単であって、低い複雑度で実装でき、更に、従来の高い複雑度且つ特化された方法とは対照的に一般的な LDPC グラフに適用できる。

【0157】

50

弱い値  $Q_v$  の切り捨ては、以下のように、各種の非収束基準および遅い収束基準に対して有用である。

【 0 1 5 8 】

1. 所定回数の反復実行後、または所定時間経過後、または所定回数のメッセージ交換後に、所定のシンδροームの要素が非ゼロである場合。要素の所定個数の典型的な値は 1 である。

【 0 1 5 9 】

2. 連続的な 2 回の反復においてシンδροームの要素の高々所定の個数が非ゼロのままである場合。

【 0 1 6 0 】

3. 連続的な 2 回の反復においてシンδροームの非ゼロ要素の数の差違が所定の限度を下回り、遅い収束を示唆する場合。

【 0 1 6 1 】

4. 所定回数の反復（通常は 1 回の反復）の前後でビット推定値間のハミング距離が所定の限度を下回り、遅い収束を示唆する場合。

【 0 1 6 2 】

上述のように非収束および遅い収束を考慮すべく図 7 A、7 B のデコーダ 3 0、3 1 は容易に変更される。具体的には、上述の基準に従い非収束または遅い収束を検出すべくルーティング層 4 4 を変更し、ルーティング層 4 4 により判定された非収束または遅い収束に回答して、 $R_{cv}$  値のいくつかまたは全てをゼロにする、および / または  $Q_v$  値のいくつかまたは全てを切り捨てるべくプロセッサ 4 2 を修正する。

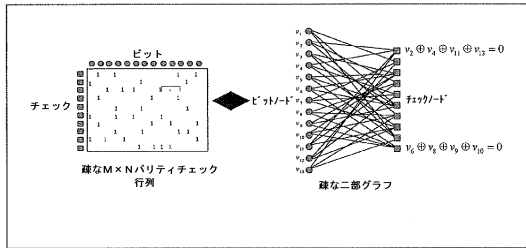
【 0 1 6 3 】

以上は、符号ワードの表現を復号化する方法、これらの方法を使用するデコーダ、自身のコントローラがこのようなデコーダを含むメモリ、および自身の受信器がこのようなデコーダを含む通信システムの限定された数の実施形態を記述したものである。本方法、デコーダ、メモリ、およびシステムの多くの変型、変更、およびその他の応用が行えることが理解されよう。

10

20

【図 1】



【図 2】

**Initialization :**  
 for all  $c \in C, v \in N(c, G)$   $R_{cv} \leftarrow 0$

**Iteration :**  
 for all  $v \in V$  (Compute bit to check messages)  
   for all  $c \in N(v, G)$   
      $Q_{vc} \leftarrow P_v + \sum_{c' \in N(v, G) \setminus c} R_{c'v}$   
   end of loop  
 end of loop  
 for all  $c \in C$  (Compute check to bit messages)  
   for all  $v \in N(c, G)$   
      $R_{cv} \leftarrow \varphi^{-1} \left( \sum_{v' \in N(c, G) \setminus v} \varphi(Q_{v'c}) \right)$   
   end of loop  
 end of loop  
 for all  $v \in V$  (update final bit estimates)  
    $Q_v \leftarrow P_v + \sum_{c \in N(v, G)} R_{cv}$   
 end of loop

**Stop Criteria :**  
 for all  $v \in V$   $\hat{v} = \text{sign}(Q_v)$   
 if  $H \cdot \hat{v} = 0$ , stop  
 else, continue to next iteration

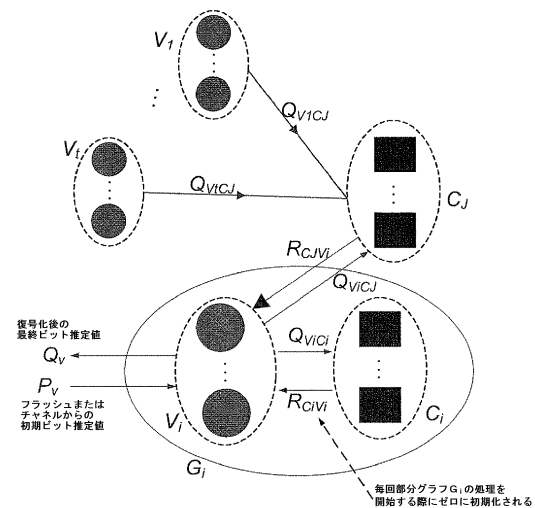
【図 3】

**Initialization :**  
 for all  $v \in V, c \in C$   $R_{cv} \leftarrow 0$   
 for all  $v \in V$   $Q_v \leftarrow P_v$

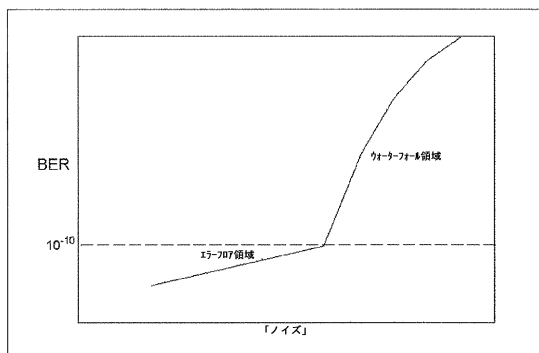
**Iteration :**  
 for all  $c \in C$  (Serially traversing the check nodes)  
    $S \leftarrow \sum_{v \in N(c, G)} \varphi(Q_v - R_{cv})$   
   for all  $v \in N(c, G)$   
      $Q_{vc} \leftarrow Q_v - R_{cv}$  (Sending  $Q_{vc}$  messages into the check nodes)  
      $R_{cv} \leftarrow \varphi^{-1}(S - \varphi(Q_{vc}))$  (Sending  $R_{cv}$  messages out from the check nodes)  
      $Q_v \leftarrow Q_v + R_{cv}$  (updating a-posteriori LLRs)  
 end of loop  
 end of loop

**Stop Criteria :**  
 for all  $v \in V$   $\hat{v} = \text{sign}(Q_v)$   
 if  $H \cdot \hat{v} = 0$ , stop  
 else, continue to next iteration

【図 5】



【図 4】



【図 6】

**Decoding phases :**  
 for all  $c \in C_j, v \in N(c, G) : Q_{cv} = P_c$

**for all  $i \in \{1, \dots, t\} :$  (go over sub-graphs)**

**Sub-graph initialization :**  
 for all  $v \in V_i, c \in C_j : R_{cv} \leftarrow 0$   
 for all  $v \in V_i : Q_v \leftarrow P_v = \text{function}(y_v)$   
 for all  $v \in V_i, c \in C_j : R_{cv} \leftarrow \varphi^{-1} \left( \sum_{u \in N(c, G) \cap V_i} \varphi(Q_{vu}) \right)$  (exchange inter sub-graph information)

**Sub-graph ( $i$ ) iterations :**  
 for all  $c \in C_j :$  (go over sub-graph check nodes)  
**Check node processing (send messages to and from check node):**  
 $S \leftarrow \sum_{v \in N(c, G)} \varphi(Q_v - R_{cv})$   
 for all  $v \in N(c, G) :$   
 $Q_{cv} \leftarrow Q_v - R_{cv}$   
 $R_{cv} \leftarrow \varphi^{-1} (S - \varphi(Q_{cv}))$   
 $Q_v \leftarrow Q_{cv} + R_{cv}$   
 end of loop  
 end of check nodes loop  
 end of iteration loop

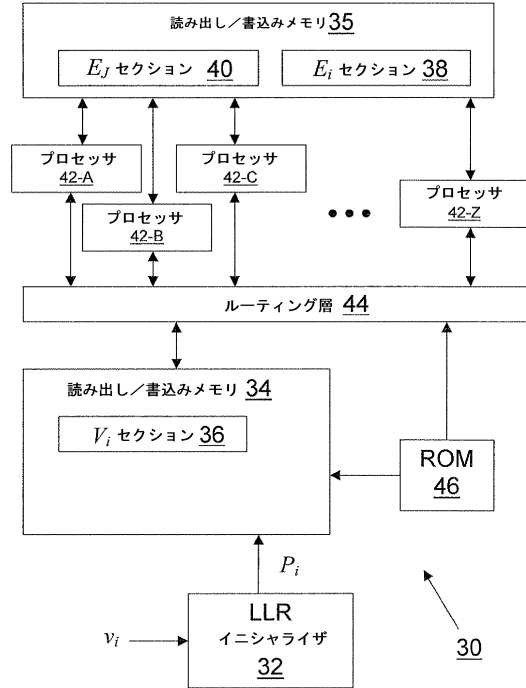
**Sub-Graph Stop Criteria :**  
 for all  $v \in V_i : \hat{v} = \text{sign}(Q_v)$   
 if  $H_i \cdot \hat{v} = 0$  or maximal number of iterations is reached, stop  
 $H_i \triangleq$  the parity-check matrix corresponding to sub-graph  $G_i$   
 else, continue to next iteration

**Sub-graph termination :**  
 for all  $v \in V_i, c \in C_j : Q_{cv} \leftarrow P_c + \sum_{u \in N(c, G) \cap V_i} R_{uv}$  (exchange inter sub-graph information)

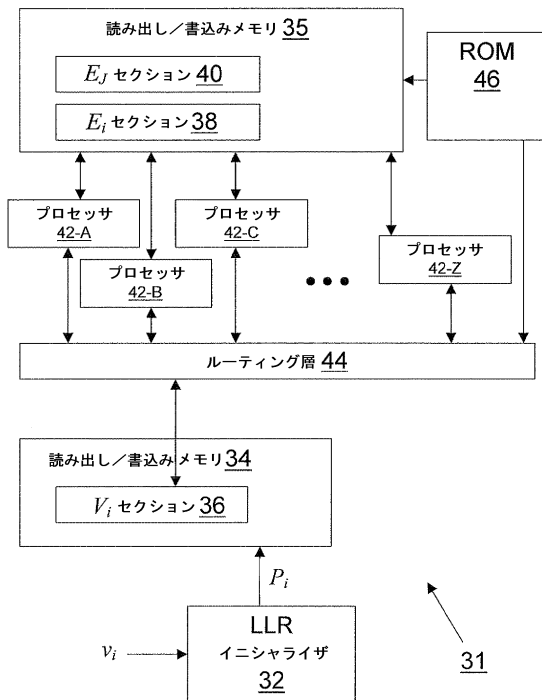
**end of sub-graphs loop**

**Stop Criteria :**  
 for all  $v \in V : \hat{v} = \text{sign}(Q_v)$   
 if  $H \cdot \hat{v} = 0$  or maximal number of phases, then stop  
 $H \triangleq$  the parity-check matrix corresponding to graph  $G$   
 else, continue to next phase  
**end of phases loop**

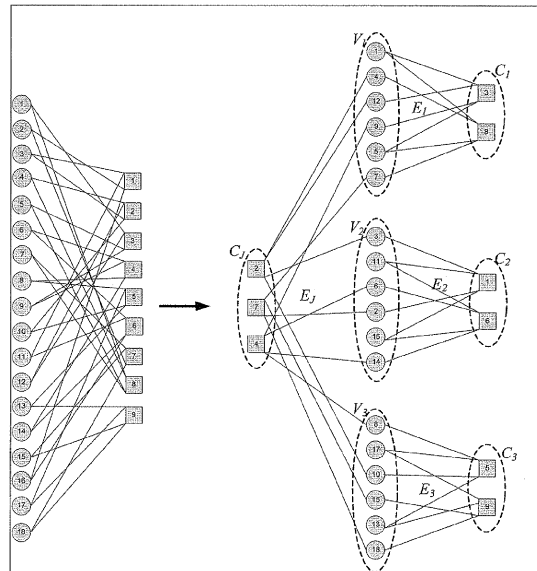
【図 7 A】



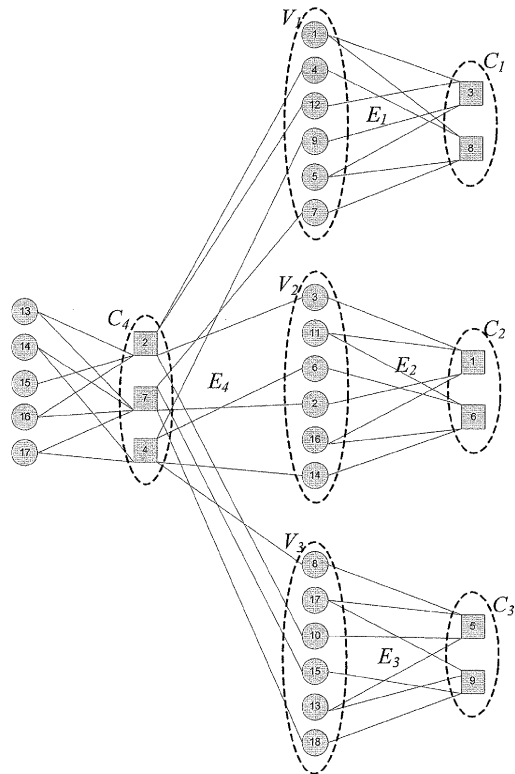
【図 7 B】



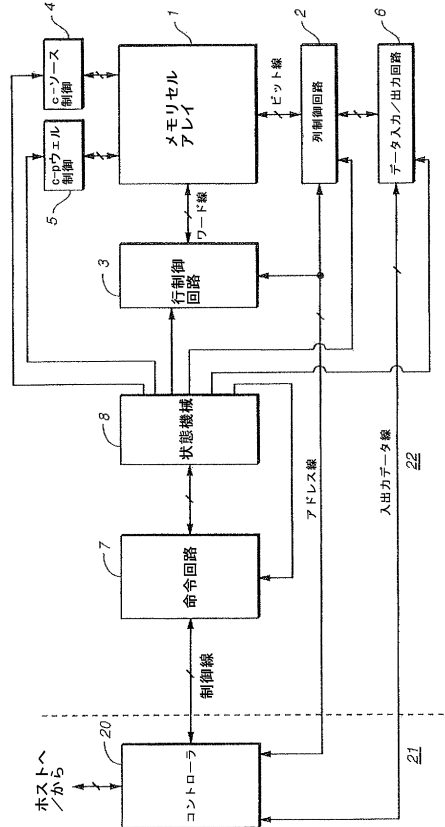
【図 8】



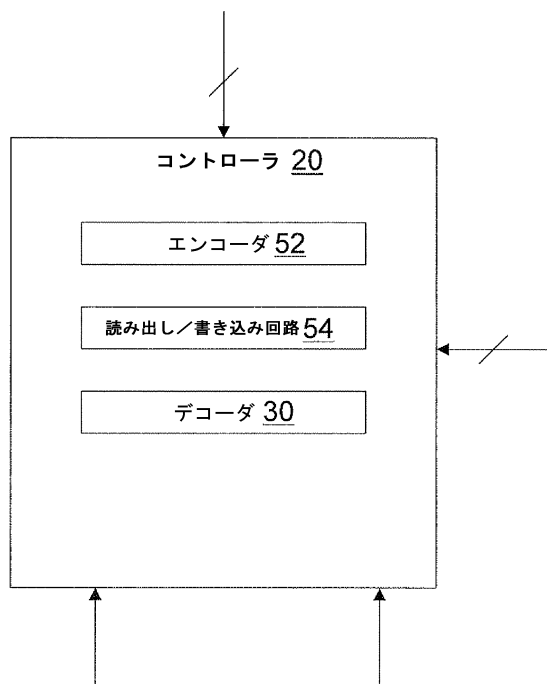
【図 9】



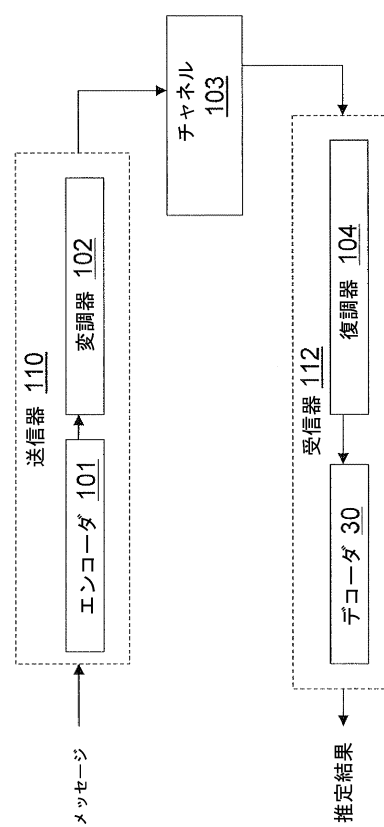
【図 10】



【図 11】



【図 12】





---

フロントページの続き

(72)発明者 リツィン、シモン

イスラエル国、54020 ギヴァット シュムエル、ズヴロン ハメール ストリート 5 / 3  
0

審査官 岡 裕之

(56)参考文献 米国特許出願公開第2005/0193320 (US, A1)

国際公開第2010/019168 (WO, A1)

Enver Cavus et al., A performance improvement and error floor avoidance technique for belief propagation decoding of LDPC, Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium, 2005年 9月14日, Vol.4, pp.2386-2390

Yang Han et al., LDPC decoder strategies for achieving low error floors, Information Theory and Applications Workshop, 2008, 2008年 2月 1日

Jingyu Kang et al., A Two-Stage Iterative Decoding of LDPC Codes for Lowering Error Floors, Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE, 2008年 12月 4日, pp.1-4

(58)調査した分野(Int.Cl., DB名)

H03M 13/19

IEEE Explore

Cinii