

(19)日本国特許庁(JP)

(12)公表特許公報(A)

(11)公表番号

特表2023-552994

(P2023-552994A)

(43)公表日 令和5年12月20日(2023.12.20)

(51)国際特許分類

H 0 3 M 13/47 (2006.01)

G 0 6 F 11/10 (2006.01)

F I

H 0 3 M 13/47

G 0 6 F 11/10

6 4 8

テーマコード (参考)

5 J 0 6 5

審査請求 未請求 予備審査請求 未請求 (全24頁)

(21)出願番号 特願2023-533832(P2023-533832)
(86)(22)出願日 令和3年12月7日(2021.12.7)
(85)翻訳文提出日 令和5年6月20日(2023.6.20)
(86)国際出願番号 PCT/US2021/062201
(87)国際公開番号 WO2022/125545
(87)国際公開日 令和4年6月16日(2022.6.16)
(31)優先権主張番号 17/116,952
(32)優先日 令和2年12月9日(2020.12.9)
(33)優先権主張国・地域又は機関
米国(US)
(81)指定国・地域 AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA
,RW,SD,SL,ST,SZ,TZ,UG,ZM,ZW),EA(
AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,A
T,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR
,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,
最終頁に続く

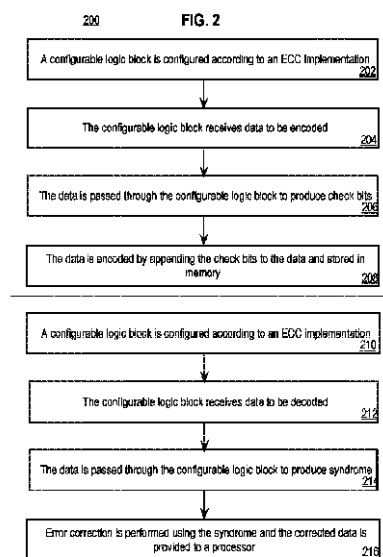
(71)出願人 591016172
アドバンスド・マイクロ・ディバイス
・インコーポレイテッド
ADVANCED MICRO DEVI
CES INCORPORATED
アメリカ合衆国 9 5 0 5 4 カリフォル
ニア州、 サンタ クララ、 オーガステ
ン ドライブ 2 4 8 5
(74)代理人 100108833
弁理士 早川 裕司
(74)代理人 100111615
弁理士 佐野 良太
(74)代理人 100162156
弁理士 村雨 圭介
(72)発明者 ロス ヴィー・ ラ フェトラ
最終頁に続く

(54)【発明の名称】 プログラム可能な誤り訂正コードの符号化及び復号論理

(57)【要約】

メモリモジュールは、特定のECC実装に対して構成可能な論理要素を含む。本明細書で使用される場合、「ECC実装」という用語は、例えば、誤り検出の結果を使用して誤り訂正を実行し、任意の誤りが後で識別及び訂正され得るようにデータを符号化する、誤り検出及びその後の処理を実行するためのECC機能を指す。このアプローチは、メモリモジュール又はコンピューティングデバイスが、要求がホスト間で往復して送信されることを必要とせずに、特定のECC実装に構成されることを可能にする。

【選択図】図2



【特許請求の範囲】

【請求項 1】

方法であって、

構成可能な論理要素のセットにおいて、誤り訂正コード（ECC）実装のために前記構成可能な論理要素のセットにおいてマスクするビットを定義する構成命令を受信することと、

前記構成可能な論理要素のセットにおいて、前記 ECC 実装に従って符号化又は復号されるデータを受信することと、

データ又は前記構成命令に従ってマスクされたコード語ビットを有する前記構成可能な論理要素のセットを使用して、前記 ECC 実装を実行することと、を含む、

方法。

【請求項 2】

前記構成命令は、前記 ECC 実装に対応する G 行列に従って、前記論理要素においてマスクするビットを定義することを含み、

前記 ECC 実装を実行することは、データを符号化することを含み、前記データを符号化することは、特定のデータ値について、

前記構成命令に従ってビットがマスクされた前記論理要素を使用して、前記データ値に対する複数のチェックビット値を決定することと、

前記特定のデータ値に対するコード語を生成及び記憶することであって、前記コード語は、前記データ値のデータ及び前記複数のチェックビットを含む、ことと、

を実行することを含む、

請求項 1 の方法。

【請求項 3】

前記複数のチェックビット値を決定することは、前記構成命令に従ってビットがマスクされた前記論理要素を使用して、前記コード語に前記 G 行列を乗算して前記複数のチェックビットを生成することを含む、

請求項 2 の方法。

【請求項 4】

前記構成命令は、前記 ECC 実装に対応する H 行列に従って、前記論理要素においてマスクするビットを定義することを含み、

前記 ECC 実装を実行することは、コード語を復号することを含み、前記コード語を復号することは、

前記構成命令に従ってビットがマスクされた前記論理要素を使用して、前記コード語からシンドローム列を計算することと、

前記シンドローム列に一致する前記 H 行列の特定の列を識別し、これに応じて、前記 H 行列の前記特定の列に対応する前記コード語の値に誤りが存在すると決定することと、

を含む、

請求項 1 の方法。

【請求項 5】

前記構成命令は、前記 ECC 実装に対応する H 行列に従って、前記論理要素においてマスクするビットを定義することを含み、

前記 ECC 実装を実行することは、コード語を復号することを含み、前記コード語を復号することは、

前記 H 行列のシンボルセットごとに、第 1 の行のセットに対応する第 1 の値のセットと第 2 の行のセットに対応する第 2 の値のセットとの比のセットを記憶することと、

前記構成命令に従ってビットがマスクされた前記論理要素を使用して、シンドロームを計算することと、

前記第 1 の行のセットに対応する前記シンドロームの第 1 の部分が、特定のシンボルセットについての比と、前記第 2 の行のセットに対応する前記シンドロームの第 2 の部分との積に等しいと決定し、これに応じて、前記特定のシンボルセットに対応する前記データ

10

20

30

40

50

の部分が誤りを含むと決定することと、
を含む、
請求項 1 の方法。

【請求項 6】

前記コード語を復号することは、
前記シンδροームの前記第 1 の部分と、前記特定のシンボルに対する前記第 1 の行のセットに対応する前記第 1 の値のセットとの商を計算することによって、前記誤りの大きさを決定することと、
前記誤りの大きさと、前記誤りを含むと決定された前記コード語の部分と、を使用して、前記データを訂正することと、
を含む、
請求項 5 の方法。

10

【請求項 7】

前記構成可能な論理要素は、コード語をメモリに記憶する前にプロセッサによって生成されたデータの符号化、又は、前記メモリに記憶された前記コード語の復号を実行することによって、メモリモジュールに対して前記 E C C 実装を提供するように構成されている、
請求項 1 の方法。

【請求項 8】

前記構成可能な論理要素は、ネットワークを介して送信又は受信されるデータに対して前記 E C C 実装を提供するように構成されている、
請求項 1 の方法。

20

【請求項 9】

装置であって、
構成可能な論理要素のセットを備え、
前記構成可能な論理要素のセットは、誤り訂正コード（E C C）実装のために前記構成可能な論理要素のセットにおいてマスクするビットを定義する構成命令に従って構成されており、
前記構成可能な論理要素のセットは、前記構成命令に従って前記構成可能な論理要素のセット内のデータ又はコード語ビットをマスクすることによって、前記 E C C 実装を実行するように構成されている、
装置。

30

【請求項 10】

前記構成命令は、前記 E C C 実装に対応する G 行列に従って、前記論理要素においてマスクするビットを定義することを含み、
前記 E C C 実装を実行することは、データを符号化することを含み、前記データを符号化することは、特定のデータ値について、
前記構成命令に従ってビットがマスクされた前記論理要素を使用して、前記データ値に対する複数のチェックビット値を決定することと、
前記特定のデータ値に対するコード語を生成及び記憶することであって、前記コード語は、前記データ値のデータ及び前記複数のチェックビットを含む、ことと、
を実行することを含む、
請求項 10 の装置。

40

【請求項 11】

前記複数のチェックビット値を決定することは、前記構成命令に従ってビットがマスクされた前記論理要素を使用して、前記コード語に前記 G 行列を乗算して前記複数のチェックビットを生成することを含む、
請求項 10 の装置。

【請求項 12】

前記構成命令は、前記 E C C 実装に対応する H 行列に従って、前記論理要素においてマ

50

スクするビットを定義することを含み、

前記 E C C 実装を実行することは、コード語を復号することを含み、前記コード語を復号することは、

前記構成命令に従ってビットがマスクされた前記論理要素を使用して、前記コード語からシンドローム列を計算することと、

前記シンドローム列に一致する前記 H 行列の特定の列を識別し、これに応じて、前記 H 行列の前記特定の列に対応する前記コード語の値に誤りが存在すると決定することと、

を含む、

請求項 9 の装置。

【請求項 13】

前記構成命令は、前記 E C C 実装に対応する H 行列に従って、前記論理要素においてマスクするビットを定義することを含み、

前記 E C C 実装を実行することは、コード語を復号することを含み、前記データを復号することは、

前記 H 行列のシンボルセットごとに、第 1 の行のセットに対応する第 1 の値のセットと第 2 の行のセットに対応する第 2 の値のセットとの比のセットを記憶することと、

前記構成命令に従ってビットがマスクされた前記論理要素を使用して、シンドロームを計算することと、

前記第 1 の行のセットに対応する前記シンドロームの第 1 の部分が、特定のシンボルセットについての比と、前記第 2 の行のセットに対応する前記シンドロームの第 2 の部分との積に等しいと決定し、これに応じて、前記特定のシンボルセットに対応する前記コード語の部分が誤りを含むと決定することと、

を含む、

請求項 9 の装置。

【請求項 14】

前記コード語を復号することは、

前記シンドロームの前記第 1 の部分と、前記特定のシンボルに対する前記第 1 の行のセットに対応する前記第 1 の値のセットとの商を計算することによって、前記誤りの大きさを決定することと、

前記誤りの大きさと、前記誤りを含むと決定された前記コード語の部分と、を使用して、前記データを訂正することと、

を含む、

請求項 13 の装置。

【請求項 15】

前記構成可能な論理要素は、コード語をメモリに記憶する前にプロセッサによって生成されたデータの符号化、又は、前記メモリに記憶されたコード語の復号を実行することによって、メモリモジュールに対して前記 E C C 実装を提供するように構成されている、

請求項 9 の装置。

【請求項 16】

前記構成可能な論理要素は、ネットワークを介して送信又は受信されるデータに対して前記 E C C 実装を提供するように構成されている、

請求項 9 の装置。

【請求項 17】

メモリモジュールであって、

誤り訂正コード (E C C) 実装に従って符号化されたデータを記憶するメモリと、

前記メモリに記憶されたデータに対して演算を実行し、前記メモリに追加のデータを書き込むように構成されたプロセッサと、

誤り訂正コード (E C C) 実装のために構成可能な論理要素のセットにおいてマスクするビットを定義する構成命令に従って構成された構成可能な論理要素のセットと、を備え、

、

10

20

30

40

50

前記構成可能な論理要素のセットは、前記メモリへの書き込みの前又は前記プロセッサによる読み出しの前に、前記構成命令に従って前記構成可能な論理要素のセット内のビットをマスクすることによって前記ECC実装を実行するように構成されている、メモリモジュール。

【請求項18】

前記構成命令は、前記ECC実装に対応するG行列に従って、前記論理要素においてマスクするビットを定義することを含み、

前記ECC実装を実行することは、データを符号化することを含み、前記データを符号化することは、特定のデータ値について、

前記構成命令に従ってビットがマスクされた前記論理要素を使用して、前記データ値に対する複数のチェックビット値を決定することと、

前記特定のデータ値に対するコード語を生成及び記憶することであって、前記コード語は、前記データ値のデータ及び前記複数のチェックビットを含む、ことと、

を実行することを含む、

請求項17のメモリモジュール。

【請求項19】

前記構成命令は、前記ECC実装に対応するH行列に従って、前記論理要素においてマスクするビットを定義することを含み、

前記ECC実装を実行することは、コード語を復号することを含み、前記コード語を復号することは、

前記構成命令に従ってビットがマスクされた前記論理要素を使用して、前記コード語からシンδροーム列を計算することと、

前記シンδροーム列に一致する前記H行列の特定の列を識別し、これに応じて、前記H行列の前記特定の列に対応する前記コード語の値に誤りが存在すると決定することと、

を含む、

請求項17のメモリモジュール。

【請求項20】

前記構成命令は、前記ECC実装に対応するH行列に従って、前記論理要素においてマスクするビットを定義することを含み、

前記ECC実装を実行することは、コード語を復号することを含み、前記コード語を復号することは、

前記H行列のシンボルセットごとに、第1の行のセットに対応する第1の値のセットと第2の行のセットに対応する第2の値のセットとの比のセットを記憶することと、

前記構成命令に従ってビットがマスクされた前記論理要素を使用して、シンδροームを計算することと、

前記第1の行のセットに対応する前記シンδροームの第1の部分が、特定のシンボルセットについての比と、前記第2のセットの行に対応する前記シンδροームの第2の部分との積に等しいと決定し、これに応じて、前記特定のシンボルセットに対応する前記コード語の部分が誤りを含まないと決定することと、

前記シンδροームの前記第1の部分と、前記特定のシンボルに対する前記第1の行のセットに対応する前記第1の値のセットとの商を計算することによって、前記誤りの大きさを決定することと、

前記誤りの大きさと、前記誤りを含まないと決定された前記コード語の部分と、を使用して、前記コード語を訂正することと、

を含む、

請求項17のメモリモジュール。

【発明の詳細な説明】

【背景技術】

【0001】

このセクションに記載されているアプローチは、遂行され得るアプローチであるが、必

10

20

30

40

50

ずしも以前に着想又は遂行されたアプローチではない。したがって、別段の指示がない限り、このセクションに記載されたアプローチの何れも、単にこのセクションに含まれることによって、従来技術として適格であると仮定すべきではない。更に、このセクションに記載されたアプローチの何れも、単にこのセクションに含まれることによって、よく理解されている、日常的である又は従来的であると仮定すべきではない。

【0002】

誤り訂正コード（Error Correction Code、ECC）メモリモジュールは、内部データ破損を検出及び訂正するためにECCエンジンを実装する。従来のECCメモリモジュールの制限の1つは、特定のECC実装が製造業者によって構成され、製造業者のアプローチに対して独自仕様となる傾向があることである。したがって、製造業者がホストを特定のECC実装で構成する場合、データにアクセスする任意のメモリコントローラは、同じECC実装で構成されなければならない。本明細書で使用される場合、「ホスト」という用語は、中央処理ユニット（Central Processing Unit、CPU）、グラフィックス処理ユニット（Graphics Processing Unit、GPU）、メモリコントローラ等の任意のタイプの処理要素を指す。例えば、プロセッサインメモリ（Processor-In-Memory、PIM）ベースのメモリモジュールが、内部誤り検出及び訂正のために及び訂正できない破損データを識別するために結果ビットを使用する場合、PIMベースのメモリモジュール及びホストは、同じレベルの誤り保護を保証し、結果ビットの一貫性のない使用を回避するために、同一のECC実装を使用しなければならない。これは、PIMベースのメモリモジュールが複数のホストで使用するよう設計される場合があるが、例えば、システムオンチップ（System-on-a-Chip、SoC）のホスト設計者又は会社は、それらが使用している、独自仕様であり得るECC実装をPIMベースのメモリベンダが知ることを望まない場合があるので、問題である。同様に、PIMベースのメモリモジュールを使用する利点は、データがPIMベースのメモリモジュールから外部プロセッサに転送される必要がないので、データが処理され得る速度である。したがって、メモリモジュールに記憶されたデータを復号するためにECCを実装するように構成され得るプログラム可能なメモリモジュールが必要とされている。

【0003】

加えて、符号化されたデータを受信する任意のプロセスが、元のエンコーダに要求を送り返すことなくデータを復号することができることが望ましい場合がある。例えば、ネットワークを介して符号化されたデータを受信するコンピューティングデバイスは、ネットワークを介して送信デバイスに要求を送信することなく、データに対して誤りチェックを実行することができることによって改善され得る。したがって、従来のECCメモリモジュールの代わりに又はそれに加えて、ECC実装を実装するように構成することができるシステムが必要である。

【0004】

実施形態は、添付の図面の図において、限定としてではなく例として示され、同様の符号は同様の要素を指す。

【図面の簡単な説明】

【0005】

【図1A】メモリ構成を示すブロック図である。

【図1B】ECCエンコーダ及び／又はデコーダ160の構成可能な論理要素の一例を示す図である。

【図2】ECCエンコーダ又はデコーダをプログラミングするためのアプローチを示すフロー図である。

【図3A】ECC実装を構成するためにH行列を使用する例を示す図である。

【図3B】ECC実装を構成するためにG行列を使用する例を示す図である。

【図3C】特定のデータ値を符号化する例を示す図である。

【図3D】記憶されたコード語を復号する例を示す図である。

【図3E】マルチビット誤りを識別するために記憶されたコード語を復号する例を示す図

10

20

30

40

50

である。

【発明を実施するための形態】

【0006】

以下の説明では、説明の目的のために、実施形態の十分な理解を提供するために、多くの具体的な詳細が記載される。しかしながら、これらの具体的な詳細なしに実施形態が実施され得ることが当業者には明らかであろう。他の例では、実施形態を不必要に不明瞭にすることを避けるために、周知の構造及びデバイスがブロック図で示されている。

【0007】

I. 概要

II. アーキテクチャ

III. 一般化されたエンコーダ／デコーダ

IV. 動作概要

V. 符号化

VI. 復号

I. 概要

【0008】

メモリモジュールは、特定の ECC 実装に一致するようにプログラムすることができる1つ以上の汎用 ECC エンジンを含む。本明細書で使用される場合、「ECC 実装」という用語は、例えば、誤り検出の結果を使用して誤り訂正を実行し、訂正することができない破損データを符号化する等の誤り検出及びその後の処理を実行するための ECC 機能を指す。

【0009】

汎用 ECC エンジンを提供することによって、メモリモジュールは、任意の特定の ECC 実装に事前にプログラムされることなく、ECC 実装を実装するように構成することができる。これは、異なる ECC 実装をサポートする異なるホストデバイスとペアにされ得る PIM ベースのメモリモジュールにおいて特に有益であり得る。メモリモジュールは、ホストによって使用される H 行列等の ECC 実装を定義する入力を受信するように構成される。メモリモジュールが、メモリモジュールに記憶されるデータを符号化するか又はメモリモジュールに記憶されたデータを復号する要求を受信する場合、メモリモジュールは、メモリモジュールの論理要素の選択的使用を通じてよって ECC 実装を実行する。

【0010】

II. アーキテクチャ

図 1 A は、メモリ構成 100 を示すブロック図である。メモリ構成 100 は、ホスト 110 及びメモリモジュール 120 を含む。ホスト 110 及びメモリモジュール 120 は、メモリバス 130 を介して通信可能に結合されている。本明細書で使用される場合、「バス」という用語は、ワイヤ、導体及び／又は無線通信リンク等の任意のタイプの有線又は無線結合を指す。加えて、実施形態は、本明細書においてバスのコンテキストで説明されるが、実施形態は、バス自体に限定されず、シリアルリンク及び容量性／誘導性結合等の他の形態のメモリ接続に適用可能である。実施形態は、説明のために単一のメモリモジュール 120 のコンテキストで図に示され、本明細書で説明されるが、実施形態は、図 1 の M 個のメモリモジュール等の任意の数のメモリモジュールを有するメモリ構成に適用可能である。

【0011】

ホストは、メモリモジュール 120 にデータを書き込み、そこからデータを受信する。ホストは、誤り訂正が実行されることなくメモリから直接、又は、誤り訂正コード (ECC) エンコーダ及び／又はデコーダ 160 からデータを取得するように構成され得る。メモリ構成 100 は、説明のために図に示されず、本明細書で説明されない他の要素を含み得る。

【0012】

例えば DRAM メモリモジュールであり得るメモリモジュール 120 は、N 個の PIM

10

20

30

40

50

ユニット（PIMユニット1～PIMユニットN）を有するメモリアレイを含む。各PIMユニットは、メモリ150、ECCエンコーダ及び／又はデコーダ160、並びに、プロセッサ170を含む。メモリモジュールは、バッファ及びデコーダ等の他の要素を含むことができ、これらの要素は、説明のために図に示されず、本明細書で説明されない。PIMユニットは、ローカルレジスタ等のような、算術演算を実行するための処理論理及びストレージを含む。実施形態は、PIMベースのメモリモジュールのコンテキストで図に示され、本明細書で説明されるが、実施形態は、この文脈に限定されず、非PIMベースのメモリモジュールに適用可能である。

【0013】

メモリモジュール120は、複数の論理要素を含み得る。論理要素は、論理演算を実行するように構成されたハードウェア構成要素を含む。例示的な論理要素は、ANDゲート、ORゲート、NORゲート、NANDゲート、XORゲート、及び、XNORゲートを含む。論理要素は、電子スイッチとして働くダイオード又はトランジスタを使用することによって等のように、ハードウェアで実装され得る。

【0014】

図1Bは、ECCエンコーダ及び／又はデコーダ160の構成可能な論理要素の一例を示している。「構成可能な論理要素」又は「構成可能な論理ブロック」は、本明細書で用いられる場合、特定のゲートにおいて特定のビットを設定することによって異なるECC実装に構成され得るハードウェア論理の固定セットを指す。図1Bの例は、構成可能なECCエンコーダ及び／又はデコーダとして使用され得る構成可能な論理要素の一実施形態である。他のオプションは、様々なレベルでマルチプレクサを使用する方式を含んでもよい。図1Bの実施形態では、複数のANDゲートが、何れのデータビットがXORゲートを通るかを決定する。図1BのAND論理要素の各々は、構成命令を通じて設定されるビットを含む。本明細書で用いられる場合、データビットを「マスクする」ことは、ビットを「0」に設定することに対応し、これは、ANDゲートの場合、入力値がXORゲートに通過しないようにさせ、一方、マスクされていないデータビットは、「1」に設定されたビットに対応し、それにより、入力値は、ANDゲートを通じてXORゲートに渡される。ECCエンコーダ及び／又はデコーダ160は、本明細書で更に説明されるように、パリティチェック行列（parity check matrix、H行列）及び／又は生成行列（generator matrix、G行列）に従ってマスクされ得る。

【0015】

III. 一般化されたエンコーダ／デコーダ

一実施形態によれば、メモリモジュール120は、ECC実装のための一般化されたエンコーダ及び／又はデコーダを用いて構成される。本明細書で用いられる場合、「一般化されたエンコーダ及び／又はデコーダ」という用語は、ECC実装のためにデータを符号化する及び／又はECC実装のためにデータを復号するプログラム可能な処理論理を指す。一般化されたエンコーダ及び／又はデコーダは、ECC実装に特有の情報をを用いて符号化される限り、複数のECC実装の何れかをを用いて実装され得るようにプログラム可能である。エンコーダ及び／又はデコーダを実装するためのエンジンは、特定の实装に応じて変化し得るいくつかの異なる方法で実装され得る。例としては、CPU、GPU、マイクロコントローラ、フィールドプログラマブルゲートアレイ（Field Programmable Gate Array、FPGA）、特定用途向け集積回路（Application Specific Integrated Circuit、ASIC）及び他のタイプの処理論理を挙げることができるが、これらに限定されない。ECCエンジンは、高帯域幅メモリ3（High Bandwidth Memory-3、HBM3）メモリモジュール内のオンダイECCエンジン等のように、メモリモジュール120の内部にある任意のECC能力を増強し得る。

【0016】

ECCエンコーダ及び／又はデコーダ160は、メモリ150からアクセスされているデータがプロセッサ170によって使用される前に復号され、プロセッサ170によって生成されたデータがメモリ150に記憶される前に符号化され得るように、メモリ150

とプロセッサ 170 との間に実装され得る。図 1 A において、メモリモジュールの各々は、PIM ユニット内の ECC エンコーダ及び／又はデコーダを通して実装される論理要素のためのマスキング命令を含む。他の実施形態では、マスキング命令は個別に記憶され、論理要素をマスクするために使用され得る。追加的又は代替的に、複数の PIM ユニットは、マスキング命令の単一のセットへのアクセスを共有してもよい。別の例として、マルチレイヤメモリモジュールが、レイヤごとに 1 セットのマスキング命令を用いて構成されてもよい。

【0017】

本明細書では「ホスト」とも呼ばれるホスト処理要素は、特定の ECC 実装のためにマスクすべき複数の論理要素を識別するマスキング命令を提供することによって、一般化されたエンコーダ／デコーダをプログラムする。例えば、ホストは、特定の ECC 実装で使用される H 行列に対応するデータを提供し得る。メモリモジュールは、H 行列に基づいて、アクティブ化する論理要素、及び、アクティブ化しない又はマスクする論理要素を識別し得る。例えば、FPGA、EPROM 等は、特定の H 行列、及び、データを符号化又は復号するために H 行列をどのように使用するかに関する命令を用いてプログラムされ得る。別の例として、ホストは、PIM ユニット又はマイクロコントローラによる実行のために、メモリモジュールに、特定の实装のための H 行列を定義するデータを提供し得る。マスキング命令は、PIM コマンドを処理するために使用される PIM ユニットの処理論理によって、PIM ユニット内の個別の処理論理、又は、PIM ユニット内のマイクロコントローラによって処理される。

【0018】

本明細書で説明されるマスキング命令及び任意の追加の ECC 実装命令は、図 1 に示されるようなメモリモジュール 120 のメモリアレイ、すなわちバンクに記憶され得る。あるいは、命令は、PIM ユニットに、又は、個別のデバイスに記憶されてもよい。マスキング命令及び任意の追加の ECC 実装命令は、任意のタイプ又は形態の命令であり得る。例えば、命令は、ソースコード、例えばレジスタ転送言語 (Register Transfer Language、RTL) 命令、構文解析されたソースコード、コンパイルされたコード (例えばオブジェクトコード)、又は、実行可能コード (例えばバイナリコード) の形態であってもよい。命令の別の例示的な形態は、Verilog 及び VHDL 等のような、FPGA で使用されるハードウェア記述言語である。ホストは、ソースコードをコンパイルして、記憶され実行される PIM モジュールに提供される実行可能コードを生成し得る。代替的に、ホストは、ソースコードを PIM 又はマイクロコントローラに提供してもよく、PIM 又はマイクロコントローラは、ソースコードをコンパイルして、記憶及び実行される実行可能コードを生成する。したがって、ソースコードからの実行可能コードの生成は、ホスト又はメモリモジュールにおいて実行され得る。

【0019】

ECC エンコーダ及び／又はデコーダは、プロセッサからメモリに書き込まれた全てのデータを符号化し、プロセッサによってメモリから読み出された全てのデータを復号するように構成され得る。プロセッサによってメモリからアクセスされ、メモリに記憶されるデータを自動的に符号化及び復号することによって、メモリモジュールは、復号を引き起こすか又は符号化されたデータを検出するための特殊コマンドを必要とせずに、メモリに記憶されたデータが正常に復号され得ることを確実にすることによって、システムの性能を改善する。

【0020】

一実施形態によれば、一般化された ECC エンコーダ／デコーダは、PIM ユニット内のプロセッサとメモリとの間で ECC 処理を実行する。例えば、ECC エンコーダ／デコーダは、PIM ユニットのメモリからデータを受信し、データに対して誤りチェックを実行し、結果を PIM ユニットのプロセッサに提供するように構成された構成可能な論理ブロックであってもよい。逆に、ECC エンコーダ／デコーダは、PIM ユニットのプロセッサから新しいデータを受信し、新しいデータをチェックビットで符号化し、符号化され

たデータをメモリに記憶させるように構成されてもよい。別の実施形態では、ECCエンコーダ／デコーダは、ネットワークを介してデータを受信し、データを利用するか又はデータをストレージに送信する前に、データに対して誤り訂正及び／又は符号化を実行するように構成され得る。

【0021】

一実施形態によれば、一般化されたエンコーダ／デコーダは、異なるECC実装を用いて実装されるように構成される。異なるECC実装は、ホストマシンによって使用される独自仕様のエンコーダ／デコーダを含み得る。一般化されたエンコーダ／デコーダは、ECC実装によって使用される特定のH行列を用いてECCをプログラムすることによって、特定のECC実装に合わせて構成され得る。これは、ホストマシンによって使用されるECCの正確な実装が異なる場合であっても、一般化されたエンコーダ／デコーダが異なるシステムに適合されることを可能にする。

10

【0022】

IV．動作概要

本明細書で説明される一般化されたエンコーダ／デコーダは、ECC実装のためにアクティブ化する論理要素を識別するために構成情報を使用することによって、ハードウェアを通して実装され得る。他の実施形態は、デジタル的にプログラムされた計算を通じてソフトウェアで同様の論理プロセスを実装し得る。

【0023】

図2は、ECCエンコーダ及び／又はデコーダを実装するためのアプローチを示すフロー図である。ステップ202～208は、構成可能な論理要素を使用して、保存前にデータを符号化する方法を含む。ステップ202において、構成可能な論理ブロックが、ECC実装に従って構成される。例えば、構成可能な論理ブロックは、複数のXORゲートを備えるXORツリーを備え得る。構成可能な論理ブロックは、構成命令のセットに従って、XORツリー内のXORゲートにつながる複数のANDゲート内のデータビットをマスクするように構成され得る。命令は、マスクすべきANDゲートのビットを直接示し得る。追加的又は代替的に、命令は、ANDゲートの何れのビットをマスクすべきかを決定するためにメモリモジュールが使用するバイナリ又はシンボル行列を備え得る。特定のビットをマスクしてECC実装に一致させるための実施形態について、本明細書で更に説明する。

20

30

【0024】

図3Aは、ECC実装を構成するためにH行列を使用する例を示している。H行列302は、データからコード語を生成する線形関数を表す、ホストデバイスから受信されたH行列を含む。図3Aに示されるH行列は、GF(2³)表現、 α 、及び、バイナリ表現を含む。本明細書で説明される方法は、バイナリ行列に関して説明されるが、同じ方法が、任意のH行列に適用され得る。列0～17は、データを表し、列18～23は、チェックビットを表す。H行列を使用して、システムは、構成可能な論理要素のANDゲートにおいてマスクされていないデータビット及びマスクされたデータビットを含むH行列XORゲート入力304を生成する。H行列XORゲート入力304の形式は、明確な例を提供するために、H行列と等価な形式で示されている。実施形態において、H行列XORゲート入力304の形式は、XORゲートにつながるANDゲートにおけるデータビットが構成命令に従ってマスクされるXORツリーを含む。データビットのマスクングを通して構成されると、H行列XORゲート入力304は、次いで、符号化されたデータに対して誤り訂正を実行するために利用され得る。

40

【0025】

図3Bは、ECC実装を構成するためにG行列を使用する例を示している。G行列303は、既知の技術を使用してH行列から導出され得る及び／又はホストデバイスによって提供され得るG行列を含む。図3Bに示されるG行列は、GF(2³)表現、 α 、及び、バイナリ表現を含む。本明細書で説明される方法は、バイナリ行列に関して説明されるが、同じ方法が、任意のG行列に適用され得る。G行列を使用して、システムは、構成可能

50

な論理要素のANDゲートにおいてマスクされていないデータビット及びマスクされたデータビットを含むG行列XORゲート入力305を生成する。G行列XORゲート入力305の形式は、明確な例を提供するために、G行列と等価な形式で示されている。実施形態において、G行列XORゲート入力305の形式は、XORゲートにつながるANDゲートにおけるデータビットが構成命令に従ってマスクされるXORツリーを含む。構成されると、G行列XORゲート入力305は、次いで、符号化されるべきデータに対するチェックビットを生成するために利用され得る。

【0026】

ステップ204において、構成可能なブロックは、符号化されるべきデータを受信する。例えば、PIMモジュールは、ホストプロセッサへの接続を必要とせずにメモリへの読み出し及び書き込みを可能にするプロセッサを備える。プロセッサが記憶されたデータを用いて計算を完了する場合、プロセッサは、結果をメモリに記憶する。結果を記憶する前に、メモリモジュールは、データを、ECCエンコーダを通して渡し、ECCエンコーダに、本明細書で説明する方法を使用して結果を符号化させ得る。

【0027】

実施形態は、PIMストレージへの要求及びデータの取り出しについて説明されるが、本明細書で説明されるシステム及び方法は、個別のデバイスによって符号化されたデータを記憶又は受信する任意のシステムに適用され得る。例えば、コンピューティングデバイスがネットワークを介して符号化された情報を受信した場合、コンピューティングデバイスは、本明細書で説明する方法を使用して、追加の要求が符号化デバイスに送信されることを必要とせずに、取り出し中にデータに対して誤り訂正を実行することができる。

【0028】

一実施形態では、コンピューティングデバイスは、複数の誤り訂正コード実装を用いて構成される。コンピューティングデバイスがネットワークを介して符号化されたデータを受信する場合、コンピューティングデバイスは、データの送信者に基づいて又はデータとともに送信された追加情報に基づいて等のように、データに適用される誤り訂正コード実装を識別し得る。コンピューティングデバイスは、識別された誤り訂正コード実装を使用してデータを復号し得る。加えて、データが個別のデバイスに送り返される場合、コンピューティングデバイスは、個別のデバイスに対応する誤り訂正コード実装を識別し、誤り訂正コード実装を使用してデータを符号化し得る。

【0029】

ステップ206において、データは、構成可能な論理ブロックを通過してチェックビットを生成する。例えば、構成可能な論理要素は、G行列に従ってマスクされたビット及びマスクされていないビットを用いて構成され得る。データは、チェックビットを生成するために、マスクされていないANDゲートに対応するデータがXORゲートに進む状態で、構成可能な論理要素を通過させられ得る。

【0030】

ステップ208において、データは、チェックビットをデータに付加することによって符号化され、メモリに記憶される。構成可能な論理要素を使用する符号化方式は、本明細書で更に説明される。

【0031】

ステップ210～216は、構成可能な論理要素を使用して、使用前にデータに対して誤り訂正を実行する方法を含む。ステップ210において、構成可能な論理ブロックは、ECC実装に従って構成される。例えば、構成可能な論理ブロックは、複数のXORゲートを備えるXORツリーを備え得る。構成可能な論理ブロックは、構成命令のセットに従って、XORツリー内のXORゲートにつながる複数のANDゲート内のデータビットをマスクするように構成され得る。命令は、マスクすべきANDゲートのビットを直接示し得る。追加的又は代替的に、命令は、ANDゲートの何れのビットをマスクすべきかを決定するためにメモリモジュールが使用するバイナリ又はシンボル行列を備え得る。特定のビットをマスクしてECC実装に一致させるための実施形態について、本明細書で更に説

明する。

【0032】

ステップ212において、構成可能な論理ブロックは、復号されるべきデータを受信する。例えば、PIMモジュールは、ホストプロセッサへの接続を必要とせずにメモリへの読み出し及び書き込みを可能にするプロセッサを備える。PIMのメモリに記憶されたデータがホストプロセッサによって生成され、こうして特定のECC実装で符号化された場合、PIMのプロセッサは、データに対して1つ以上の動作を実行するために、メモリからデータを要求し得る。実際の例として、PIMは、ホストによってメモリに書き込まれたデータを使用してシミュレーションを実行し、次いで、シミュレーションの結果をメモリに記憶するように構成され得る。シミュレーションを実行するためにデータをプロセッサに送信する前に、メモリモジュールは、データを、ECCデコーダを通して渡し、本明細書で説明する方法を使用してECCデコーダにデータを復号させ得る。

10

【0033】

ステップ214において、データは、構成可能な論理ブロックを通過してシンドロームを生成する。例えば、構成可能な論理要素は、H行列に従ってマスクされたビット及びマスクされていないビットを用いて構成され得る。データは、シンドロームを生成するために、マスクされていないANDゲートに対応するデータがXORゲートに進む状態で、構成可能な論理要素を通過させられ得る。

【0034】

ステップ216において、シンドロームを使用して誤り訂正が実行され、訂正されたデータがプロセッサに提供される。例えば、シンドロームは、誤りが訂正され得るように、誤り位置及び大きさを識別するために使用され得る。構成可能な論理要素を使用する復号方式は、本明細書で更に説明される。

20

【0035】

V. 符号化

一実施形態では、ECC実装は、メモリに記憶されるべき新しいデータを符号化することを含む。符号化は、データをメモリに記憶するためのプロセッサからの要求に応じて、及び／又は、他の符号化されたデータとともに記憶されるべきデータを、ネットワークを介して受信することに応じて、実行され得る。

【0036】

一実施形態では、データを符号化することは、特定のECC実装に従って構成された構成可能な論理要素を使用して、特定のデータ値に対して複数のチェックビットを決定することを含む。チェックビットは、特定のデータ値のビットに付加されて、特定のデータ値のためのコード語を生成し得る。次いで、コード語がメモリに記憶され得る。一実施形態では、ECCエンコーダは、データ値に対応するビットを構成可能な論理要素を通して送信することによって複数のチェックビット値を決定し、それによって行列乗算を生成する。例えば、ECCエンコーダは、H行列に従ってビットが設定された複数のANDゲートを含み得る。データ値からデータビットを受信するANDゲートのみがデータビットをXORゲートに送信するので、データ値のビットに対応しない列は、XORゲートに対応する行の値を計算するためにデータビットをXORゲートに与えない。したがって、特定のデータ値が、0番目のビット及び2番目のビットに「1」を含む場合、ECCエンコーダは、0番目の列及び2番目の列のANDゲートからXORゲートにビットを送信するだけでよい。

30

40

【0037】

図3Cは、特定のデータ値を符号化する例を示している。図3Cの例では、システムは、G行列を使用してデータ値に対するチェックビットを計算する。他の実施形態では、システムは、H行列を使用し、データ値が論理要素を介して乗算される場合に計算されたシンドロームを0になるようにさせるチェックビット列の値を計算し得る。

【0038】

図3Cにおいて、データ値306は、符号化されるべきデータ値のバイナリ表現を含む

50

。データ値 306 は、論理要素において符号化された G 行列によって乗算される。G 行列 XOR ゲート入力 305 の各円は、マスクされていないビットを有する AND ゲートに対応する。したがって、円を有さないゲートを通過する任意のビットは、XOR ゲートに渡されない。データ値 306 が G 行列 XOR ゲート入力 305 を通過する場合、データ値 306 の各データビット (1) は、対応する列の AND ゲートを通過し、それによって符号化 XOR ゲート入力 308 を生成する。符号化 XOR ゲート入力 304 は、データビットが通過する各ゲートを含む。データビットは、第 1 及び第 3 の列においてのみ「1」に設定されているため、符号化 XOR ゲート入力 308 では、G 行列 XOR ゲート入力 305 の第 1 及び第 3 の列のみが示されている。

【0039】

符号化 XOR ゲート入力 308 の残りのビットが XOR ゲートを通過した後、それらはチェックビット 310 を生成する。例えば、第 1 の行は、AND ゲートを通過した 2 つのビットを含み、それによって、それらが XOR ゲートを通過する場合に 0 を生成する。同様に、行 2 及び行 5 は、2 つのビットを含み、それによって、それらがそれぞれの XOR ゲートを通過する場合に 0 を生成する。行 3 及び行 4 の各々は、XOR ゲートを通過する単一のビットを有し、したがって 1 を生成し、行 6 は、XOR ゲートを通過するビットを有さない。したがって、最終チェックビット 310 は、(0 0 0 1 1 0 0) を含む。次いで、チェックビット 310 がデータ値 306 に付加されて、符号化されたコード語 312 を生成する。

【0040】

上記の説明は、チェックビット値を生成するために使用される行列乗算のハードウェア実装を提供する。行列乗算は、以下のように計算され得る。

【0041】

【数 1】

$$\sum_{j=1}^K G_{ij} D_j = C_i \text{ for } i = 1 \dots R$$

ここで、 G_{ij} は、 i 番目の行及び j 番目の列に対する G 行列の値であり、 D は、 j 番目の列に対するデータ値であり、 C_i は、 i 番目の行に対するチェックビット値であり、 K は、列の総数であり、 R は、G 行列の行の数である。上記の行列乗算は、同じ結果に対してシンボル空間又はバイナリ空間において実行され得る。

【0042】

VI. 復号

一実施形態では、ECC 実装は、ECC デコーダで受信されたデータを復号することを含む。復号は、メモリモジュールに記憶された記憶データにアクセスするためのプロセッサからの要求に応じて、及び／又は、ネットワークを介して符号化されたデータを受信することに応じて、実行され得る。データを復号することは、記憶されたコード語を使用して、コード語の 1 つ以上のビットにおける誤りを識別し、コード語のビットを調整して誤りを修正することを含み得る。

【0043】

一実施形態では、ECC デコーダは、最初に、特定の ECC 実装に従って構成された構成可能な論理要素を使用してシンドローム列を計算する。次いで、システムは、シンドローム列を構成可能な論理要素に対応する各列と比較して、シンドロームに一致する列を識別する。システムは、シンドロームに一致する列が誤りを含むと決定する。バイナリ実装では、誤りの値は常に 1 であり、したがって、列が識別されると、ビットを修正することができる。

【0044】

図 3 D は、記憶されたコード語を復号する例を示している。図 3 D において、コード語 3 1 4 は、誤りを含む ECC 実装を使用して符号化されたデータを含む。符号化と同様に、コード語 3 1 4 は、論理要素において符号化された H 行列によって乗算される。H 行列 X OR ゲート入力 3 0 4 の各円は、マスクされていないビットを有する AND ゲートに対応する。コード語 3 1 4 は、符号化ステップで生成されたチェックビットを含むので、チェックビットは、チェックビットに対応する列を含む H 行列を使用して構成された構成可能な論理要素を介して加えて供給される。こうして、復号 X OR ゲート入力 3 1 6 は、結果として生じるシンδροームを計算するためにビットが通過する 5 つの列を含む。メモリモジュールは、シンδροーム 3 1 8 を H 行列 X OR ゲート入力 3 0 4 の列と比較する。シンδροーム 3 1 8 が第 2、第 4 及び第 5 の行に 1 を含むので、メモリモジュールは、第 2、第 4 及び第 5 の行の列を検索する。シンδροーム 3 1 8 は、H 行列 X OR ゲート入力 3 0 4 の第 5 列に一致する。したがって、コード語 3 1 4 の第 5 のビットが誤りを含む。メモリモジュールは、誤り値によってビットを変更することによってコード語を訂正することができる。したがって、第 5 の列の 1 が 0 に置き換えられる。

10

【0045】

上記の説明は、シンδροームを計算するために使用される行列乗算のハードウェア実装を提供する。行列乗算は、以下のように計算され得る。

【0046】

【数 2】

20

$$\sum_{j=1}^{KN} H_{ij} D_j = S_i \text{ for } i = 1 \dots$$

ここで、 H_{ij} は、 i 番目の行及び j 番目の列に対する H 行列の値であり、 D は、コード語の j 番目の列に対するデータ値であり、 S_i は、 i 番目の行に対するシンδροーム値であり、 N は、列の総数であり、 R は、H 行列の行の数である。上記の行列乗算は、同じ結果に対してシンボル空間又はバイナリ空間において実行され得る。

30

【0047】

図 3 D の実施形態は、単一ビット誤りを識別し、修正する方法を表している。こうして、シンδροームを H 行列の列に一致させることは、同じ位置にシンδροームと同じ値を有する H 行列の列を識別することによって実行することができる。計算されたシンδροームが 0 値のみの列である場合、メモリモジュールは、コード語内に誤りが存在しないと決定し得る。計算されたシンδροームが、H 行列内の何れの列にも一致しない列である場合、メモリモジュールは、コード語がマルチビット又はマルチシンボル誤りを含むと決定し得る。

【0048】

マルチビット誤りの場合、システムは、シンボル表記を利用して、同じシンボル内のシングルビット誤り又はマルチビット誤りの何れかを識別し得る。本明細書で使用されるシンボルは、3 ビット等の複数のビットのセットを指す。例えば、図 3 A における H 行列の第 1 の表現は、1 を含む第 1 の行及び α のべき乗を含む第 2 の行によって示されるシンボルの 2 つの行を含み、その各値は 3 ビットシンボルを含む。シンボルに対応する 3×3 行列は、2 つの 3 ビットシンボルの乗算の表現を含むことができ、そのうちの 1 つは、行列に符号化される定数を含む。例えば、第 1 の 3 行及び第 1 の 3 列は、 D_0 と 1 との第 1 の乗算に対応し、第 2 の 3 行及び第 1 の 3 列は、 D_0 と α^1 との第 1 の乗算に対応し、ここで、 D_0 は、第 1 のデータシンボル (3 ビット) である。

40

【0049】

一実施形態では、復号は、シンボル空間において、第 1 の行の各列が 1 のシンボル値を含む H 行列を使用して実行される。例えば、図 3 A の H 行列において、第 1 の 3 行に対す

50

る 3 列の各繰り返しセットは、対角線に 1 を、他の場所に 0 を有する対角線行列を含む。行の値が各列に対して 1 である場合、シンドロームは誤りを含む。したがって、図 3 D のシンドローム 3 1 8 における第 1 の 3 つの値 $S_0 = 0 1 0$ は、誤りを含む。誤りは、各シンボル列に適用されて、シンドロームがシンボルに一致するときに決定し得る。各シンボル列に誤りを適用することは、H 行列のシンボルに対応する AND ゲートのセットに誤りを通過させることと、XOR ゲートを通して結果を計算することと、を含み得る。

【0 0 5 0】

実際の例として図 3 D を使用すると、0 1 0 の誤りが H 行列 XOR ゲート入力 3 0 4 の 3 つの列の第 1 のセットによって乗算される場合、第 2 の列のみが AND ゲートを通過するデータビットを受信する。第 2 の列は、第 2 及び第 6 の行にセットビットを含むので、XOR ゲートは (0 1 0 0 0 1) の値を計算する。これは、(0 1 0 1 1 0) のシンドロームと一致しないので、システムは、次の列のセットをチェックし得る。0 1 0 の誤りが H 行列 XOR ゲート入力 3 0 4 の列の第 2 のセットによって乗算される場合、第 5 の列のみが AND ゲートを通過するデータビットを受信する。第 5 の列は、第 2、第 4 及び第 5 の行にセットビットを含むので、XOR ゲートは、シンドロームに一致する (0 1 0 1 1 0) の値を計算する。したがって、誤りは第 5 の列にある。

【0 0 5 1】

図 3 E は、マルチビット誤りを識別するために記憶されたデータを復号する例を示している。誤りが 1 1 0 を含む場合、メモリモジュールは、第 1 の 2 つの列を保持し、それを誤りと比較する。例えば、コード語を第 4 及び第 5 の列に誤りを含むように変更すると (したがって、コード語の第 1 の 6 つの値は 1 0 1 1 1 0 である)、計算されたシンドロームは (1 1 0 1 1 1) を含むことになる。1 1 0 の誤りが第 1 のシンボル、すなわち H 行列 XOR ゲート入力 3 0 4 の第 1 の 3 つの列に適用される場合、第 1 及び第 2 の列が残りの、(1 1 0 0 1 1) の値を生成する。この値はシンドロームと一致しないので、メモリモジュールは次のシンボルに移動する。第 2 のシンボル、すなわち H 行列 XOR ゲート入力 3 0 4 の第 2 の 3 つの列に 1 1 0 を適用すると、第 4 及び第 5 の列が残りの、これは、XOR ゲートを介して結合される場合、シンドロームに一致する (1 1 0 1 1 1) の値を生成する。したがって、システムは、誤りが第 2 のシンボルに存在すると決定し、誤り値 1 1 0 を第 2 のシンボルに適用することは、誤りが第 4 及び第 5 の列にあることを示す。

【0 0 5 2】

一般的なシナリオでは、システムは、H 行列の n 番目の行と H 行列の複数の他の行との間の比を使用して誤り位置を計算するように構成され得る。これらの比は、事前に計算され、以下の式を使用して ECC 実装のために記憶され得る。

【0 0 5 3】

【数 3】

$$I_{ij} = \frac{H_{ij}}{H_{nj}}$$

ここで、 I_{ij} は、特定の行及び特定の列についての比率値であり、 H_{ij} は、特定の行及び特定の列における H 行列の値であり、 H_{nj} は、n 番目の行における H 行列の値である。上記のように、 H_{nj} が全ての行に対して「1」である場合、比は、特定の列及び行に対する H 行列の値に等しい。H 行列の複数の行から H_{nj} であるように選択された行は、何れかの行であり得る。一実施形態では、 H_{nj} に対して選択された行が列に「0」を含む場合、システムは、比の計算が不定にならないように、その列の H_{nj} に対して異なる行を選択する。

【0 0 5 4】

誤りの位置を計算するために、システムは、各列 (j) をテストして、列 (j) の全ての行について以下の式が真であるかどうかを決定し得る。

$$S_i = I_{ij} * S_n$$

ここで、 S_i は、 i 番目の行に対するシンドローム値であり、 S_{1n} は、 H_{nj} に使用されるのと同じ行に対するシンドローム値である。したがって、第1の列について、 H_{nj} が第2の行である場合、 S_n は、 n 番目の行に対するシンドローム値になる。加えて、第2の列について、 H_{nj} が第6の行である場合、 S_n は、第6の行に対するシンドローム値になる。特定の列について、シンドローム値と各行の比との積がその行に対するシンドローム値に等しい場合、システムは、誤りが特定の列に存在すると決定し得る。

【0055】

さらに、システムは、シンドロームの値を使用して誤りの大きさを計算し得る。このステップは、誤りの位置の計算の前、後、又は、それと並行して実行されてもよい。誤り値は、以下のように計算され得る。

$$E_j = S_n * H_{nj}^{-1}$$

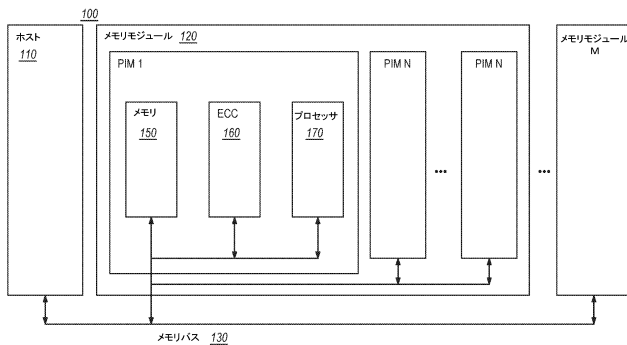
ここで、 E_j は、誤りである。 S_n 及び H_{nj}^{-1} に使用される行は、誤りの位置の計算に使用される同じ行であってもよく、及び／又は、誤り算出式において S_n 及び H_{nj}^{-1} に同じ行が使用される限り、異なる行であってもよい。 H_{nj}^{-1} の値は、実装前に各列に対して事前計算され得る。加えて、値は、各列に対して同じものを使用してもよく、又は、列間で行を変えてもよい。システムは、各列の H_{nj}^{-1} に何れの行が使用されているかを示すデータを記憶し得る。

【0056】

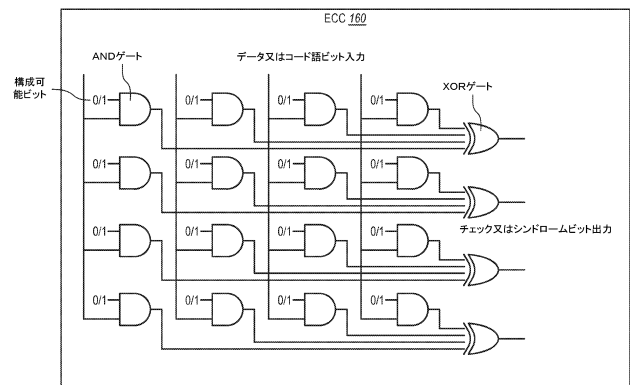
誤り値及び誤り位置が分かると、システムは、誤り値を誤り位置に適用することによって誤りを訂正し得る。システムが誤りの位置を識別するために上記の計算を実行し、全ての行について上記の式を満たす計算がない場合、システムは、コード語がマルチシンボル誤りを含むと決定し得る。概して、上述した方法は、初期ECCコードが検出することができ、任意の単一シンボル誤りを訂正することができる。本明細書で説明する方法は、任意の線形ブロックコードに関して実行することができる。

【図面】

【図1A】



【図1B】



10

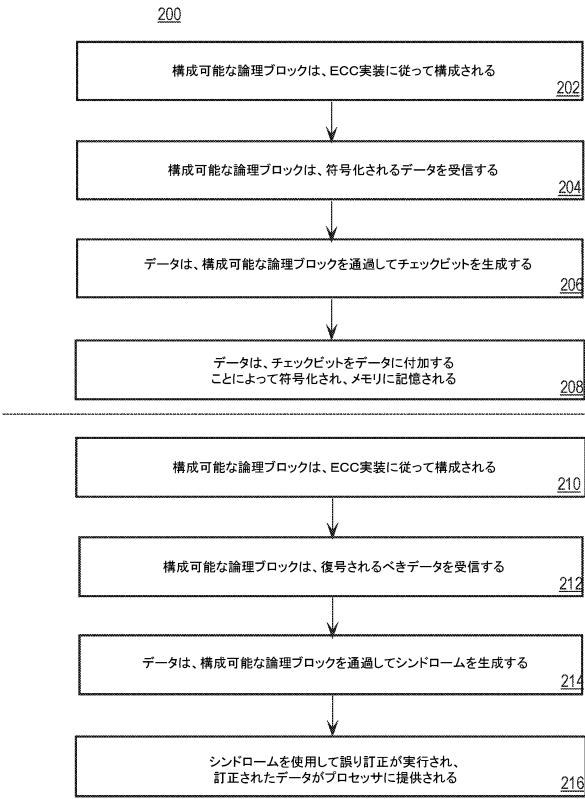
20

30

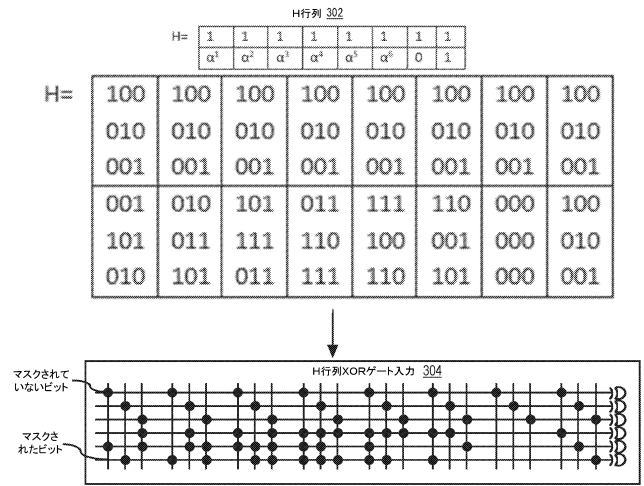
40

50

【図 2】



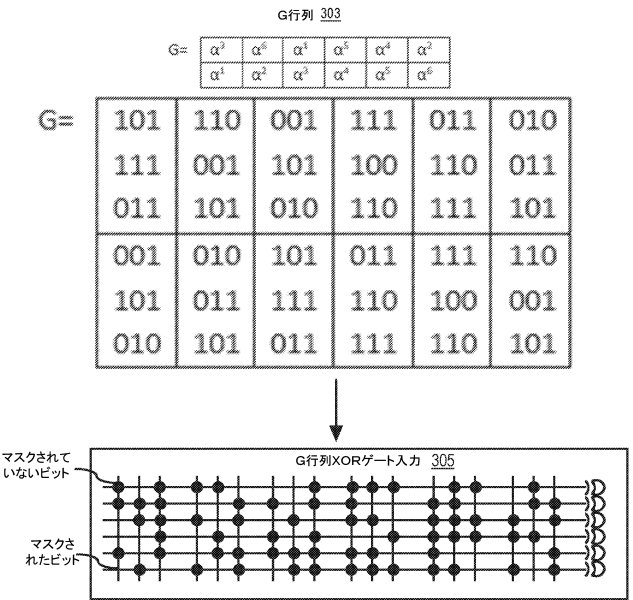
【図 3 A】



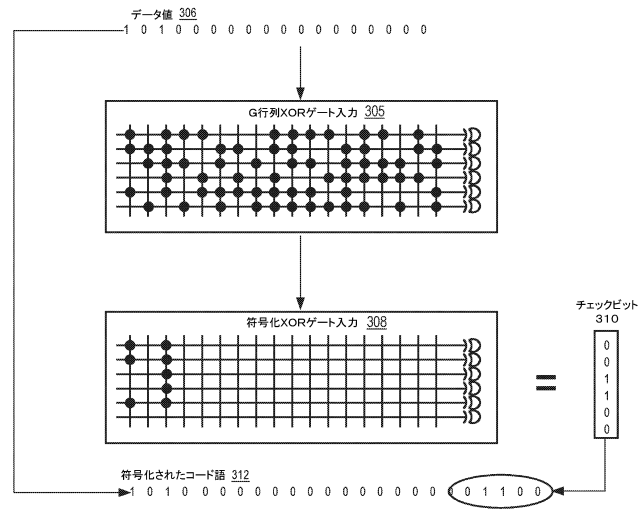
10

20

【図 3 B】



【図 3 C】

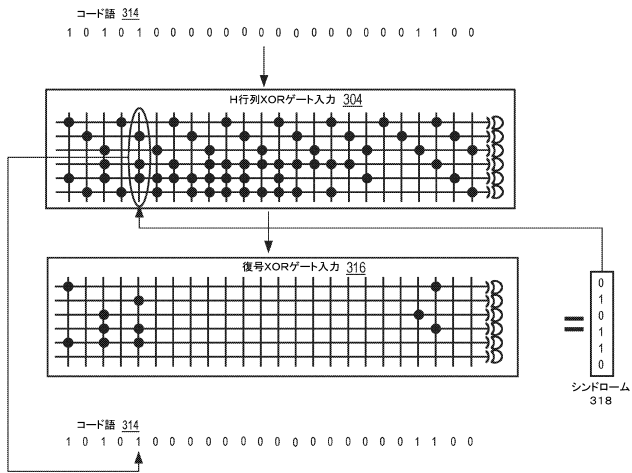


30

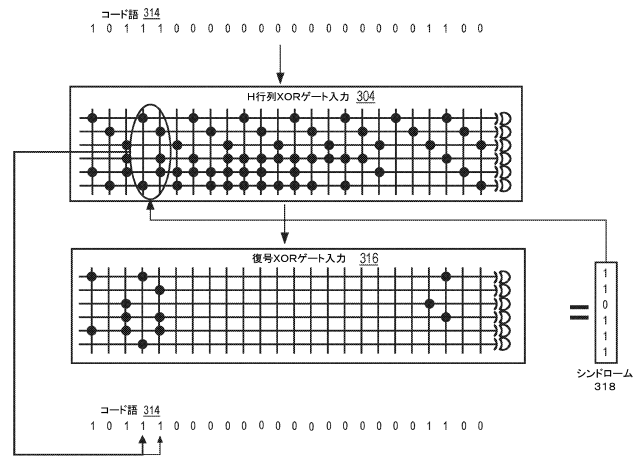
40

50

【図 3 D】



【図 3 E】



10

20

30

40

50

【手続補正書】

【提出日】令和5年8月4日(2023.8.4)

【手続補正1】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項1】

方法であって、

構成可能な論理要素のセットにおいて、誤り訂正コード（ECC）実装のために前記構成可能な論理要素のセットにおいてマスクするビットを定義する構成命令を受信することと、

前記構成可能な論理要素のセットにおいて、前記ECC実装に従って符号化又は復号されるデータを受信することと、

データ又は前記構成命令に従ってマスクされたコード語ビットを有する前記構成可能な論理要素のセットを使用して、前記ECC実装を実行することと、を含む、方法。

【請求項2】

前記構成命令は、前記ECC実装に対応するG行列に従って、前記論理要素においてマスクするビットを定義することを含み、

前記ECC実装を実行することは、データを符号化することを含み、前記データを符号化することは、特定のデータ値について、

前記構成命令に従ってビットがマスクされた前記論理要素を使用して、前記データ値に対する複数のチェックビット値を決定することと、

前記特定のデータ値に対するコード語を生成及び記憶することであって、前記コード語は、前記データ値のデータ及び前記複数のチェックビットを含む、ことと、

を実行することを含む、

請求項1の方法。

【請求項3】

前記複数のチェックビット値を決定することは、前記構成命令に従ってビットがマスクされた前記論理要素を使用して、前記コード語に前記G行列を乗算して前記複数のチェックビットを生成することを含む、

請求項2の方法。

【請求項4】

前記構成命令は、前記ECC実装に対応するH行列に従って、前記論理要素においてマスクするビットを定義することを含み、

前記ECC実装を実行することは、コード語を復号することを含み、前記コード語を復号することは、

前記構成命令に従ってビットがマスクされた前記論理要素を使用して、前記コード語からシンドローム列を計算することと、

前記シンドローム列に一致する前記H行列の特定の列を識別し、これに応じて、前記H行列の前記特定の列に対応する前記コード語の値に誤りが存在すると決定することと、

を含む、

請求項1の方法。

【請求項5】

前記構成命令は、前記ECC実装に対応するH行列に従って、前記論理要素においてマスクするビットを定義することを含み、

前記ECC実装を実行することは、コード語を復号することを含み、前記コード語を復号することは、

10

20

30

40

50

前記 H 行列のシンボルセットごとに、第 1 の行のセットに対応する第 1 の値のセットと第 2 の行のセットに対応する第 2 の値のセットとの比のセットを記憶することと、

前記構成命令に従ってビットがマスクされた前記論理要素を使用して、シンドロームを計算することと、

前記第 1 の行のセットに対応する前記シンドロームの第 1 の部分が、特定のシンボルセットについての比と、前記第 2 の行のセットに対応する前記シンドロームの第 2 の部分との積に等しいと決定し、これに応じて、前記特定のシンボルセットに対応する前記データの部分が誤りを含むと決定することと、

を含む、

請求項 1 の方法。

10

【請求項 6】

メモリモジュールであって、

誤り訂正コード（ECC）実装に従って符号化されたデータを記憶するメモリと、

前記メモリに記憶されたデータに対して演算を実行し、前記メモリに追加のデータを書き込むように構成されたプロセッサと、

前記 ECC 実装のために構成可能な論理要素のセットにおいてマスクするビットを定義する構成命令に従って構成された構成可能な論理要素のセットと、を備え、

前記構成可能な論理要素のセットは、前記メモリへの書き込みの前又は前記プロセッサによる読み出しの前に、前記構成命令に従って前記構成可能な論理要素のセット内のビットをマスクすることによって前記 ECC 実装を実行するように構成されている、

20

メモリモジュール。

【請求項 7】

前記構成命令は、前記 ECC 実装に対応する G 行列に従って、前記論理要素においてマスクするビットを定義することを含み、

前記 ECC 実装を実行することは、データを符号化することを含み、前記データを符号化することは、特定のデータ値について、

前記構成命令に従ってビットがマスクされた前記論理要素を使用して、前記データ値に対する複数のチェックビット値を決定することと、

前記特定のデータ値に対するコード語を生成及び記憶することであって、前記コード語は、前記データ値のデータ及び前記複数のチェックビットを含む、ことと、

30

を実行することを含む、

請求項 6 のメモリモジュール。

【請求項 8】

前記複数のチェックビット値を決定することは、前記構成命令に従ってビットがマスクされた前記論理要素を使用して、前記コード語に前記 G 行列を乗算して前記複数のチェックビットを生成することを含む、

請求項 7 のメモリモジュール。

【請求項 9】

前記構成命令は、前記 ECC 実装に対応する H 行列に従って、前記論理要素においてマスクするビットを定義することを含み、

40

前記 ECC 実装を実行することは、コード語を復号することを含み、前記コード語を復号することは、

前記構成命令に従ってビットがマスクされた前記論理要素を使用して、前記コード語からシンドローム列を計算することと、

前記シンドローム列に一致する前記 H 行列の特定の列を識別し、これに応じて、前記 H 行列の前記特定の列に対応する前記コード語の値に誤りが存在すると決定することと、

を含む、

請求項 6 のメモリモジュール。

【請求項 10】

前記構成命令は、前記 ECC 実装に対応する H 行列に従って、前記論理要素においてマ

50

スクするビットを定義することを含み、

前記 E C C 実装を実行することは、コード語を復号することを含み、前記コード語を復号することは、

前記 H 行列のシンボルセットごとに、第 1 の行のセットに対応する第 1 の値のセットと第 2 の行のセットに対応する第 2 の値のセットとの比のセットを記憶することと、

前記構成命令に従ってビットがマスクされた前記論理要素を使用して、シンδροームを計算することと、

前記第 1 の行のセットに対応する前記シンδροームの第 1 の部分が、特定のシンボルセットについての比と、前記第 2 のセットの行に対応する前記シンδροームの第 2 の部分との積に等しいと決定し、これに応じて、前記特定のシンボルセットに対応する前記コード語の部分が誤りを含むと決定することと、

前記シンδροームの前記第 1 の部分と、前記特定のシンボルに対する前記第 1 の行のセットに対応する前記第 1 の値のセットとの商を計算することによって、前記誤りの大きさを決定することと、

前記誤りの大きさと、前記誤りを含むと決定された前記コード語の部分と、を使用して、前記コード語を訂正することと、

を含む、

請求項 6 のメモリモジュール。

【請求項 1 1】

前記コード語を復号することは、

前記シンδροームの前記第 1 の部分と、前記特定のシンボルに対する前記第 1 の行のセットに対応する前記第 1 の値のセットとの商を計算することによって、前記誤りの大きさを決定することと、

前記誤りの大きさと、前記誤りを含むと決定された前記コード語の部分と、を使用して、前記データを訂正することと、

を含む、

請求項 1 0 のメモリモジュール。

【請求項 1 2】

前記構成可能な論理要素は、コード語をメモリに記憶する前にプロセッサによって生成されたデータの符号化、又は、前記メモリに記憶された前記コード語の復号を実行することによって、メモリモジュールに対して前記 E C C 実装を提供するように構成されている、請求項 6 のメモリモジュール。

【請求項 1 3】

前記構成可能な論理要素は、ネットワークを介して送信又は受信されるデータに対して前記 E C C 実装を提供するように構成されている、

請求項 6 のメモリモジュール。

10

20

30

40

50

【国際調査報告】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US2021/062201
A. CLASSIFICATION OF SUBJECT MATTER H03M 13/05(2006.01)i; G06F 11/10(2006.01)i According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) H03M 13/05(2006.01); G06F 11/10(2006.01); G06F 3/06(2006.01); G11C 29/42(2006.01); G11C 7/10(2006.01); H03M 13/29(2006.01) Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Korean utility models and applications for utility models Japanese utility models and applications for utility models Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) eKOMPASS(KIPO internal) & Keywords: configurable, logic, element, error correction code(ECC), implementation, encode, decode, cordward, G-matrix, H-matrix, syndrome, multiply, magnitude		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2017-0004037 A1 (SK HYNIX INC.) 05 January 2017 (2017-01-05) paragraphs [0001]-[0137]; and figures 1-10	1,7-9,15-17
Y		2-6,10-14,18-20
Y	US 2011-0138249 A1 (THOMAS KERN et al.) 09 June 2011 (2011-06-09) paragraphs [0001]-[0190]; and figures 1-10	2-6,10-14,18-20
A	US 2014-0075272 A1 (THOMAS KERN et al.) 13 March 2014 (2014-03-13) paragraphs [0001]-[0155]; and figures 1-12	1-20
A	US 2012-0079342 A1 (SHIH-LIEN LU et al.) 29 March 2012 (2012-03-29) paragraphs [0001]-[0083]; and figures 1-6	1-20
A	KR 10-2018-0019818 A (SAMSUNG ELECTRONICS CO., LTD.) 27 February 2018 (2018-02-27) paragraphs [0001]-[0133]; and figures 1-19	1-20
☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "D" document cited by the applicant in the international application "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 24 March 2022		Date of mailing of the international search report 24 March 2022
Name and mailing address of the ISA/KR Korean Intellectual Property Office 189 Cheongsu-ro, Seo-gu, Daejeon 35208, Republic of Korea Facsimile No. +82-42-481-8578		Authorized officer YANG, Jeong Rok Telephone No. +82-42-481-5709

Form PCT/ISA/210 (second sheet) (July 2019)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/US2021/062201

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2017-0004037	A1	05 January 2017	CN	106326028	A	11 January 2017
				CN	106326028	B	02 February 2021
				KR	10-2017-0004693	A	11 January 2017
				TW	201702878	A	16 January 2017
				TW	I679533	B	11 December 2019
				US	9977712	B2	22 May 2018
US	2011-0138249	A1	09 June 2011	CN	102442634	A	09 May 2012
				CN	102442634	B	20 April 2016
				CN	102442636	A	09 May 2012
				CN	102442636	B	27 May 2015
				CN	102487312	A	06 June 2012
				DE	102009039106	A1	06 May 2010
				DE	102011083962	A1	31 May 2012
				DE	102011083962	B4	19 November 2015
				DE	102011084024	A1	05 April 2012
				DE	102011084024	B4	21 July 2016
				DE	102011087457	A1	06 June 2012
				US	2010-0058876	A1	11 March 2010
				US	2011-0068420	A1	24 March 2011
				US	2012-0161254	A1	28 June 2012
				US	2013-0334624	A1	19 December 2013
				US	2014-0227818	A1	14 August 2014
				US	7832279	B2	16 November 2010
				US	8518732	B2	27 August 2013
				US	8723276	B2	13 May 2014
				US	8921954	B2	30 December 2014
				US	9527725	B2	27 December 2016
US	2014-0075272	A1	13 March 2014	US	8856629	B2	07 October 2014
US	2012-0079342	A1	29 March 2012	US	8533572	B2	10 September 2013
KR	10-2018-0019818	A	27 February 2018	CN	107767919	A	06 March 2018
				CN	107767919	B	19 February 2021
				US	2018-0053545	A1	22 February 2018
				US	9953702	B2	24 April 2018

Form PCT/ISA/210 (patent family annex) (July 2019)

フロントページの続き

MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,ML,MR,N
E,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,
CZ,DE,DJ,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IR,IS,IT,JO,JP,K
E,KG,KH,KN,KP,KR,KW,KZ,LA,LC,LK,LR,LS,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,N
G,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,ST,SV,SY,TH,TJ,TM,
TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,WS,ZA,ZM,ZW

(特許庁注：以下のものは登録商標)

1. VERILOG

アメリカ合衆国 9 5 0 5 4 カリフォルニア州、サンタ クララ、オーガスティン ドライブ 2 4 8 5
F ターム (参考) 5J065 AB01 AC03 AD03 AE06 AG02 AH01