(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5298393号
(P5298393)
(45)発行日　平成25年9月25日(2013.9.25)　(24)登録日　平成25年6月28日(2013.6.28)

(51)Int.Cl.　F I
G06F 3/06 (2006.01)　G06F 3/06 305C

請求項の数 26 (全 19 頁)

(21)出願番号	特願2010-534958 (P2010-534958)
(86)(22)出願日	平成20年11月20日 (2008.11.20)
(65)公表番号	特表2011-504269 (P2011-504269A)
(43)公表日	平成23年2月3日 (2011.2.3)
(86)国際出願番号	PCT/US2008/012944
(87)国際公開番号	WO2009/070235
(87)国際公開日	平成21年6月4日 (2009.6.4)
審査請求日	平成23年11月17日 (2011.11.17)
(31)優先権主張番号	60/989,670
(32)優先日	平成19年11月21日 (2007.11.21)
(33)優先権主張国	米国(US)

(73)特許権者　502188642
マーベル　ワールド　トレード　リミテッド
バルバドス国　ビービー１４０２７，セントマイケル、ブリトンズ　ヒル、ガンサイトロード、エル　ホライズン
(74)代理人　110000877
龍華国際特許業務法人
(72)発明者　ブルーチ、アービンド
アメリカ合衆国、９５０５４　カリフォルニア州、サンタ　クララ、マーベル　レーン　５４８８　マーベル　セミコンダクター　インコーポレイテッド内

審査官　坂東　博司

最終頁に続く

(54)【発明の名称】並列リードソロモンＲＡＩＤ（ＲＳ－ＲＡＩＤ）アーキテクチャ、デバイス、および方法

(57)【特許請求の範囲】
【請求項１】
分散データ記憶デバイスであって、
複数の並列データ記憶クラスタであって、
複数のデータシンボルを記憶する複数のデータ記憶デバイス、および、
前記複数のデータ記憶デバイスのそれぞれに結合し、前記複数のデータシンボルからローカルな中間チェックサムを計算し、前記ローカルな中間チェックサムを含む複数の中間チェックサムに基づいて前記複数のデータシンボルのチェックサムを計算するよう構成されたＲＡＩＤコントローラを含む、複数の並列データ記憶クラスタと、
前記複数の並列データ記憶クラスタのそれぞれに結合し、前記複数の中間チェックサムを前記複数の並列データ記憶クラスタに配信する通信ファブリックとを備える分散データ記憶デバイス。
【請求項２】
前記ＲＡＩＤコントローラは、前記複数のデータシンボルの重み付き和に基づいて前記ローカルな中間チェックサムを計算する、請求項１に記載の分散データ記憶デバイス。
【請求項３】
前記複数の並列データ記憶クラスタの複数のＲＡＩＤコントローラの少なくとも１つは、他の並列データ記憶クラスタから前記中間チェックサムを受信し、前記複数の並列なデータ記憶クラスタの複数のデータ記憶デバイスのうちの１つに記憶するためのチェックサムを、前記受信した中間チェックサムと、前記ローカルな中間チェックサムとに基づいて

生成する、請求項１又は２に記載の分散データ記憶デバイス。
【請求項４】
前記ＲＡＩＤコントローラは、前記チェックサムをデータ記憶デバイスに記憶するようさらに構成される請求項１から３のいずれか１項に記載の分散データ記憶デバイス。
【請求項５】
前記通信ファブリックは、前記複数のデータシンボルを含む複数の情報シンボルを前記複数の並列データ記憶クラスタに配信するようさらに構成される請求項１から４のいずれか１項に記載の分散データ記憶デバイス。
【請求項６】
前記ＲＡＩＤコントローラは、前記複数の情報シンボルのサブセットから前記複数のデータシンボルを選択するＲＡＩＤコントロールユニットをさらに備える請求項５に記載の分散データ記憶デバイス。
【請求項７】
前記ＲＡＩＤコントロールユニットは、前記複数のデータ記憶デバイスから作動可能なデータ記憶デバイスのリストを確定する記憶デバイス故障センスユニットをさらに備える請求項６に記載の分散データ記憶デバイス。
【請求項８】
前記ＲＡＩＤコントロールユニットは、作動可能なデータ記憶デバイスの前記リストに基づいてデータ回復行列を計算するようさらに構成される請求項７に記載の分散データ記憶デバイス。
【請求項９】
前記通信ファブリックは、作動可能なデータ記憶デバイスの前記リストが変化すると、前記データ回復行列を前記複数の並列データ記憶クラスタに配信するようさらに構成される請求項８に記載の分散データ記憶デバイス。
【請求項１０】
前記ＲＡＩＤコントローラは、作動可能なデータ記憶デバイスのリストに基づいて前記中間チェックサムを計算するよう構成された中間和デバイスをさらに備える請求項６に記載の分散データ記憶デバイス。
【請求項１１】
前記中間和デバイスは、データシンボルが変化すると、前記ローカルな中間チェクサムを更新するよう構成された再計算器をさらに備える請求項１０に記載の分散データ記憶デバイス。
【請求項１２】
前記ＲＡＩＤコントロールユニットは、前記データ回復行列を計算するために、添加された単位行列とファンデルモンデ行列の逆行列を求めるようさらに構成される請求項８に記載の分散データ記憶デバイス。
【請求項１３】
中間和デバイスは、前記データ回復行列に基づく中間データシンボル、ならびに、前記記憶デバイスから読出されるリードデータシンボルおよびリードチェックサムシンボルの少なくとも一方のベクトルを計算するようさらに構成される請求項１２に記載の分散データ記憶デバイス。
【請求項１４】
前記ＲＡＩＤコントロールユニットは、前記添加された単位行列とファンデルモンデ行列の行の集合ならびに作動可能なデータ記憶デバイスの前記リストに基づくベクトルを選択し、前記データ回復行列を形成するために前記行の集合の逆行列を求めるようさらに構成される請求項１２に記載の分散データ記憶デバイス。
【請求項１５】
前記ＲＡＩＤコントローラは、中間和デバイスに中間データシンボルを計算させるメッセージを前記複数の並列データ記憶クラスタに送出するようさらに構成される請求項１４に記載の分散データ記憶デバイス。

【請求項１６】
前記通信ファブリックは、前記中間データシンボルを前記複数の並列データ記憶クラスタのそれぞれに配信するようさらに構成される請求項１５に記載の分散データ記憶デバイス。
【請求項１７】
前記ＲＡＩＤコントローラは、複数の中間データシンボルから、回復されるデータシンボルを計算するようさらに構成される請求項１６に記載の分散データ記憶デバイス。
【請求項１８】
前記ローカルな中間チェックサムは、前記ＲＡＩＤコントローラに割当てられた前記複数のデータ記憶デバイスに対応する前記複数のデータシンボルの部分集合の部分和(partial sum)である請求項１から１７のいずれか１項に記載の分散データ記憶デバイス。
【請求項１９】
前記中間データシンボルは、前記ＲＡＩＤコントローラに割当てられた前記データ記憶デバイスに対応する前記データシンボルの部分集合および前記リードチェックサムシンボルの部分和である請求項１３に記載の分散データ記憶デバイス。
【請求項２０】
データ記憶方法であって、
複数のデータ記憶デバイスを複数の並列データ記憶クラスタに割当てること、
前記複数の並列データ記憶クラスタのそれぞれに複数のデータシンボルを記憶すること、
前記複数の並列データ記憶クラスタのそれぞれに含まれる複数のＲＡＩＤコントローラのそれぞれが、前記複数のデータシンボルの重み付き和から中間チェックサムを計算すること、
前記複数のＲＡＩＤコントローラの少なくとも１つが、前記複数のＲＡＩＤコントローラの他のＲＡＩＤコントローラのそれぞれから前記中間チェックサムを受信すること、
前記複数のＲＡＩＤコントローラの少なくとも１つが、計算した前記中間チェックサムおよび前記他のＲＡＩＤコントローラのそれぞれから受信した前記中間チェックサムに基づいて前記複数のデータシンボルのチェックサムを計算すること、および、
前記複数のデータ記憶デバイスの少なくとも１つに前記チェックサムを記憶することを含むデータ記憶方法。
【請求項２１】
情報シンボルの部分集合から前記複数のデータシンボルを選択すること、および、
通信ファブリックを使用して、前記複数の並列データ記憶クラスタに前記選択されたデータシンボルを配信することをさらに含む請求項２０に記載のデータ記憶方法。
【請求項２２】
前記複数の並列データ記憶クラスタのそれぞれの中の作動可能なデータ記憶デバイスの集合を検知することをさらに含む請求項２１に記載のデータ記憶方法。
【請求項２３】
作動可能なデータ記憶デバイスの前記集合に基づくデータ回復行列を、前記複数の並列データ記憶クラスタのそれぞれに配信することをさらに含む請求項２２に記載のデータ記憶方法。
【請求項２４】
前記データ回復行列に基づく中間データシンボルならびに前記記憶デバイスから読出されるリードデータシンボルおよびリードチェックサムシンボルの少なくとも一方のベクトルを計算することをさらに含む請求項２３に記載のデータ記憶方法。
【請求項２５】
前記中間データシンボルを前記複数の並列データ記憶クラスタのそれぞれに配信すること、および、
前記中間データシンボルの和から、回復されるデータシンボルを計算することをさらに含む請求項２４に記載のデータ記憶方法。

【請求項２６】
分散データアーキテクチャにおける誤り訂正のための方法であって、
複数のデータ記憶デバイスを複数の並列データ記憶クラスタに割当てる構成行列を読出すこと、
複数のデータシンボルを前記複数の並列データ記憶クラスタに記憶すること、
前記複数の並列データ記憶クラスタのそれぞれに含まれる複数のＲＡＩＤコントローラのそれぞれが、前記割当てられたデータ記憶デバイスに記憶された前記複数のデータシンボルから中間チェックサムを計算することであって、それにより、少なくとも１つの中間チェックサムがそれぞれのデータ記憶クラスタについて計算される、計算すること、
前記複数のＲＡＩＤコントローラの少なくとも１つが、前記複数のＲＡＩＤコントローラの他のＲＡＩＤコントローラのそれぞれから前記中間チェックサムを受信すること、
チェックサムを形成するために、前記複数のＲＡＩＤコントローラの少なくとも１つが、計算した前記中間チェックサムおよび前記他のＲＡＩＤコントローラのそれぞれから受信した前記中間チェックサムを合計すること、および、
前記チェックサムを少なくとも１つのデータ記憶デバイスに記憶する方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本出願は、２００７年１１月２１日に出願された米国仮出願第６０／９８９，６７０号「Ｐａｒａｌｌｅｌ　ＲＡＩＤ　Ｉｍｐｌｅｍｅｎｔａｔｉｏｎ　ｆｏｒ　ＲＡＩＤ６　ａｎｄ　Ｒｅｅｄ－Ｓｏｌｏｍｏｎ　Ｃｏｄｅ」の利益を主張し、参照によりその全体が本明細書に組込まれる全ての引用された参考文献を含む。
【背景技術】
【０００２】
ＲＡＩＤ（ｒｅｄｕｎｄａｎｔ　ａｒｒａｙ　ｏｆ　ｉｎｅｘｐｅｎｓｉｖｅ　ｄｉｓｋｓ）アーキテクチャは、ハードディスクなどのデータ記憶ユニットのグループを使用して、耐故障性を有するデータ記憶装置を提供する。ＲＡＩＤアーキテクチャは、誤りおよびディスク故障から情報を保護するために、前方誤り訂正（ｆｏｒｗａｒｄ　ｅｒｒｏｒ　ｃｏｒｒｅｃｔｉｏｎ）（ＦＥＣ）コードおよび予備のデータ記憶ユニットを使用する。情報シンボルは、ビット、バイト、またはワードであってよい。情報シンボルは、符合化されて、データおよびチェックサムまたはパリティシンボルを含むコードシンボルを形成しうる。組織的な前方誤り訂正コードの場合、情報シンボルは、コードシンボルのデータシンボル部分において明示的に表されうる。
【０００３】
リードソロモンコードは、チェックサムシンボルの数に等しい記憶ユニットの故障の数を許容するために、ＲＡＩＤアーキテクチャ（ＲＳ－ＲＡＩＤ）で使用されうる。たとえば、データ用に２０の記憶ユニットを、チェックサム用に４つの記憶ユニットを割当てる４重誤り訂正ＲＳ－ＲＡＩＤアーキテクチャは、４つを含む４つまでの記憶デバイスにおける故障を許容しうる。
【０００４】
ＲＳ－ＲＡＩＤアーキテクチャは、通常、データ記憶ユニットに書込まれるデータシンボルを保護するために単一のＲＡＩＤコントローラを使用する。単一のＲＡＩＤコントローラが使用されて、チェックサム、符号化、および復号化計算を実施するとき、ＲＡＩＤアーキテクチャのスループットまたはデータ記憶量および取出し速度は、ＲＡＩＤでなくかつ耐故障性を有さないデータ記憶アーキテクチャと比べて低減される可能性がある。
【発明の概要】
【発明が解決しようとする課題】
【０００５】
したがって、高スループットで耐故障性を有する分散型のデータ記憶アーキテクチャが望ましい場合がある。

【課題を解決するための手段】
【０００６】
高性能記憶アーキテクチャでは、複数のＲＡＩＤコントローラは、通信ファブリックと呼ぶ通信経路の共通集合を通じて互いに通信してもよい。通信ファブリックは、ＲＡＩＤコントローラと所与のＲＡＩＤコントローラに割当てられた記憶デバイスとの間の通信経路と比較して高いレーテンシを有する可能性がある。レーテンシの高い通信ファブリックは、ＲＡＩＤコントローラ間のデータ、メッセージ、構成などのトラフィックが、耐故障性を有する分散データ記憶装置のタスクに整合しなければ、ＲＡＩＤデータ記憶アーキテクチャのスループットを減少させる可能性がある。通信ファブリックと割当てられたデータ記憶デバイスの集合との間に介在してもよい、それぞれのＲＡＩＤコントローラは、データ記憶アーキテクチャのノードと呼ばれてもよい。ＲＡＩＤコントローラおよび割当てられたデータ記憶デバイスは、データ記憶クラスタと呼ばれてもよい。
【０００７】
リードソロモン（Ｒｅｅｄ－Ｓｏｌｏｍｏｎ）ＲＡＩＤ（ＲＳ－ＲＡＩＤ）アーキテクチャは、冗長なデータ記憶デバイスを含むことによって、ハードディスクなどの記憶デバイスに書込まれ、また、記憶デバイスから読出される情報シンボルを保護しうる。ｍのチェックサムデバイスを使用するＲＳ－ＲＡＩＤアーキテクチャは、データ記憶デバイスのｍ程度の同時故障を許容しうる。ｍのチェックサムシンボルはc_1，c_2、…、c_mで示されてもよい。ＲＳ－ＲＡＩＤアーキテクチャはまた、d_1，d_2、…、d_nで示す情報保持またはデータシンボル用の数ｎのデータ記憶デバイスを含みうる。
【０００８】
チェックサムおよびデータ記憶デバイスは、データおよびチェックサムシンボルを、ビット、バイト、ワードなどとして記憶してもよい。リードソロモン（ＲＳ）コードなどのあるタイプの前方誤り訂正コード（ｆｏｒｗａｒｄ　ｅｒｒｏｒ　ｃｏｒｒｅｃｔｉｏｎ　ｃｏｄｅｓ）（ＦＥＣ）は、通常、バイトを使用することが留意されてもよい。たとえば、ＲＳコードは、２５５のバイトブロック内の２３３のデータバイトと３２のチェックサムバイトに２３３の情報バイトを符合化するブロックなどのバイトブロックに作用しうる。
【０００９】
ＲＳ－ＲＡＩＤアーキテクチャは、対応するデータ記憶デバイスD_1、D_2、…、D_nによって保持されるデータシンボルd_1，d_2、…、d_nを使用して、ｉ番目のチェックサムデバイスC_iに記憶されるチェックサムシンボルc_iを計算しうる。ＲＳ－ＲＡＩＤアーキテクチャは、D_1、D_2、…、D_n、C_1、C_2、…、C_mの記憶デバイスのうちの任意のｍ以下の記憶デバイスが故障する場合、故障したデバイスのうちの任意のデバイスのコンテンツが、損なわれていないまたは故障していないデバイスから再構築されうるように各c_i（$1 \leqq i \leqq m$）を確定しうる。ＲＳ－ＲＡＩＤアーキテクチャは、ファンデルモンデ行列の特性により耐故障性を有する演算を提供することができ、ファンデルモンデ行列は、チェックサムシンボルを計算し維持し、記憶デバイスから読出したデータおよびチェックサムシンボルから情報を回復するのに使用される。ＲＳ－ＲＡＩＤコントローラは、記憶デバイスが故障しても、添加されるかまたは拡大されたファンデルモンデ行列と単位行列の（ｎ×ｎ）部分の逆行列を計算することによって、記憶デバイスにおけるデータおよび／またはチェックサムシンボルを回復しうる。
【００１０】
チェックサムシンボルを生成するために、ＲＳ－ＲＡＩＤアーキテクチャは、データシンボルを、ファンデルモンデ行列の要素で重み付けし、式１によって線形関数F_iを使用して重み付けされたデータシンボルを合計しうる。関数F_iは、ファンデルモンデ行列の要素のｉ番目の行から得られうるため、$F_i = [f_{i,1}; f_{i,2}; \cdots; f_{i,n}]^T$である。

【数１】

$$c_i = \sum_{j=1}^{n} d_j f_{i,j} \quad 式１$$

換言すれば、データおよびチェックサムシンボルが、それぞれ、（ｎ×１）次元および（ｍ×１）次元ベクトルＤ＝［ｄ$_1$，ｄ$_2$、…、ｄ$_n$］^{T}およびＣ＝［ｃ$_1$、ｃ$_2$、…、ｃ$_m$］^{T}として表され、また、関数Ｆ$_i$が行列Ｆの行として表される場合、ＲＳ－ＲＡＩＤアーキテクチャは、チェックサムシンボルを式２ａによって符合化しうる。

Ｃ＝ＦＤ　（式２ａ）

式２ａは、

【数２】

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,n} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,n} \\ \vdots & \vdots & & \vdots \\ f_{m,1} & f_{m,2} & \cdots & f_{m,n} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}. \quad 式２ｂ$$

に等しい。

【００１１】

有利に設計されたＲＳ－ＲＡＩＤ　ＦＥＣコードの場合、Ｆ行列は、要素：ｆ$_{i,j}$＝ｊ$^{i-1}$を有する（ｍ×ｎ）ファンデルモンデ行列でありうる。式中、インデックスｉ＝１、２、…、ｍおよびｊ＝１、２、…、ｎは、それぞれ、ファンデルモンデ行列の行および列に対応し、代数演算は、ガロア体の特性を使用して実施される。たとえば、（３×４）ファンデルモンデ行列は、

【数３】

$$F = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \end{bmatrix}. \quad 式３$$

として書かれうる。

【００１２】

誤りを含む可能性があるコードワードまたはコードシンボルから（ｎ×１）情報ベクトル

【数４】

$$\tilde{D} = \left[\tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_n\right]^T$$

を回復するために、並列ＲＳ－ＲＡＩＤアーキテクチャは、ファンデルモンデ行列およびＩで示す添加された（（ｎ－ｍ）×（ｎ－ｍ））単位行列を含む拡大されるかまたは分割された行列Ａの逆行列を求め、Ａの逆行列に、作動可能な記憶ユニットから読出されるデータおよびチェックサムシンボルの集合ＤおよびＣをそれぞれ右から乗算し(ｐｏｓｔ－ｍｕｌｔｉｐｌｙ)うる。記号的には、回復される情報ベクトル

【数５】

$$\tilde{D}$$

は、

【数６】

$$\tilde{D}=Inv(A)\bullet\lfloor E\rfloor$$

から得られうる。式中、拡大行列は

【数７】

$$A\triangleq\left[\frac{I}{F}\right]$$

であり、

【数８】

$$E\triangleq\left[\frac{D}{C}\right]$$

は、拡大されたデータおよびチェックサムシンボルベクトルである。表記Ｉｎｖ（Ａ）は、正則な（ｎ×ｎ）正方行列を形成するＡの行の部分集合の逆行列などのＡに基づく逆行列をもたらし、また、以下で述べるように、

【数９】

$$\lfloor \mathrm{E}\rfloor$$

で示す列行列Ｅのｎの行の、対応する選択されるかまたは選抜された集合に共形的である関数であると理解されてもよい。Ａ行列の逆行列を求めるプロセスは、Ａの行の選択された集合の反転とみなされてもよく、選択は、作動可能なデータ記憶デバイスのリストおよび行列にベクトルを掛ける計算における共形性についての要件によって確定される。（（ｎ＋ｍ）×ｎ）拡大行列Ａのｎの行の全ての部分集合は、Ｆがファンデルモンデ行列であるため反転可能であることが留意されてもよい。

【００１３】

拡張形態では、式

【数１０】

$$\tilde{\mathrm{D}}=\mathrm{Inv(A)}\bullet\lfloor \mathrm{E}\rfloor$$

は、

【数１１】

$$\begin{bmatrix}\tilde{d}_1\\ \tilde{d}_2\\ \vdots\\ \tilde{d}_n\end{bmatrix}=Inv\left(\begin{bmatrix}1&0&0&\cdots&0\\0&1&0&\cdots&0\\\vdots&\vdots&\vdots&&\vdots\\0&0&0&\cdots&1\\\hline 1&1&1&\cdots&1\\1&2&3&\cdots&n\\\vdots&\vdots&\vdots&&\vdots\\1&2^{m-1}&3^{m-1}&\cdots&n^{m-1}\end{bmatrix}\right)\bullet\left\lfloor\begin{bmatrix}d_1\\d_2\\\vdots\\d_n\\\hline c_1\\c_2\\\vdots\\c_m\end{bmatrix}\right\rfloor \quad \text{式４}$$

として表されうる。式中、共形性は、行列Ａの選択された部分を反転する前に、Ｅおよびの対応する行を選択することによって実施される。

【００１４】

換言すれば、ＲＳ－ＲＡＩＤアーキテクチャの各記憶デバイスは、拡大行列Ａの行および列ベクトルＥ＝［ｄ$_1$、ｄ$_2$、…、ｄ$_n$、ｃ$_1$、ｃ$_2$、…、ｃ$_m$］^{T}の対応する要素によって表されうる。ｍの冗長記憶デバイス中のいずれの冗長記憶デバイスも故障しない場合、回復される情報シンボルは、Ａのｎの行の任意の部分集合およびＥのｎの対応する要素を選択することによって確定されて、データ回復行列として記述されてもよい正方行列

A'および対応するデータ記憶ユニットから読出されるデータのベクトル

【数１２】

$$E'=\lfloor E \rfloor.$$

が形成されうる。換言すれば、Ｉｎｖ（Ａ）＝（Ａ'）$^{-1}$であり、また、

【数１３】

$$\tilde{D}=(A')^{-1}\bullet E'$$

である。たとえば、４＋２ＲＳ－ＲＡＩＤアーキテクチャの場合、回復されるかまたは復号化されるデータ

【数１４】

$$\tilde{D}$$

は、

【数１５】

$$\begin{bmatrix}\tilde{d}_1\\ \tilde{d}_2\\ \tilde{d}_3\\ \tilde{d}_4\end{bmatrix}=Inv\left(\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\\ \cancel{1}&\cancel{1}&\cancel{1}&\cancel{1}\\ \cancel{1}&\cancel{2}&\cancel{3}&\cancel{4}\end{bmatrix}\right)\bullet\begin{bmatrix}d_1\\d_2\\d_3\\d_4\\ \cancel{c_1}\\ \cancel{c_2}\end{bmatrix}=\begin{bmatrix}d_1\\d_2\\d_3\\d_4\end{bmatrix}$$ 式５

によって、拡大されたファンデルモンデ行列の最初の４つの行および記憶デバイスアレイから読出されたデータおよびチェックサムの最初の４つのエンティティから抽出された回復されるデータシンボルのベクトルでありうる。

【００１５】

たとえば、第３の、第５の、または、第３と第５の両方のデータ記憶デバイスが故障する場合、

【数１６】

$$\tilde{D},$$

は、以下の通りに、作動可能なデバイスに対応する４つの行を選択することによって、E'から回復されうる。

【数１７】

$$\begin{bmatrix}\tilde{d}_1\\ \tilde{d}_2\\ \tilde{d}_3\\ \tilde{d}_4\end{bmatrix}=Inv\left(\begin{bmatrix}1&0&0&0\\0&1&0&0\\ \cancel{0}&\cancel{0}&\cancel{1}&\cancel{0}\\0&0&0&1\\1&1&1&1\\ \cancel{1}&\cancel{2}&\cancel{3}&\cancel{4}\end{bmatrix}\right)\bullet\begin{bmatrix}d_1\\d_2\\ \cancel{d_3}\\d_4\\c_1\\ \cancel{c_2}\end{bmatrix}$$ 式６ａ

$$=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\1&1&1&1\end{bmatrix}^{-1}\begin{bmatrix}d_1\\d_2\\d_4\\c_1\end{bmatrix}$$

【数１８】

$$\begin{bmatrix}\tilde{d}_1\\ \tilde{d}_2\\ \tilde{d}_3\\ \tilde{d}_4\end{bmatrix} = Inv\left(\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\\\cancel{1}&\cancel{1}&\cancel{1}&\cancel{1}\\\cancel{1}&\cancel{2}&\cancel{3}&\cancel{4}\end{bmatrix}\bullet\begin{bmatrix}d_1\\d_2\\d_3\\d_4\\\cancel{c_1}\\c_2\end{bmatrix}\right) \quad \text{式６ｂ}$$

$$= \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}^{-1}\begin{bmatrix}d_1\\d_2\\d_3\\d_4\end{bmatrix}$$

【数１９】

$$\begin{bmatrix}\tilde{d}_1\\ \tilde{d}_2\\ \tilde{d}_3\\ \tilde{d}_4\end{bmatrix} = Inv\left(\begin{bmatrix}1&0&0&0\\0&1&0&0\\\cancel{0}&\cancel{0}&\cancel{1}&\cancel{0}\\0&0&0&1\\\cancel{1}&\cancel{1}&\cancel{1}&\cancel{1}\\1&2&3&4\end{bmatrix}\bullet\begin{bmatrix}d_1\\d_2\\\cancel{d_3}\\d_4\\\cancel{c_1}\\c_2\end{bmatrix}\right) \quad \text{式６ｃ}$$

$$= \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\1&2&3&4\end{bmatrix}^{-1}\begin{bmatrix}d_1\\d_2\\d_4\\c_2\end{bmatrix}$$

式中、２重末梢線は記憶デバイスの故障を示し、１重末梢線は、逆行列を形成し、その後の計算を実施するための記憶デバイスの選択解除を示しうる。逆行列は、ガウス消去法または別の方法によって計算されてもよい。

【数２０】

$$\tilde{D},$$

の値が得られると、回復されるかまたは推定される任意のチェックサムベクトル

【数２１】

$$\tilde{C}$$

が、

【数２２】

$$\tilde{C} = F\tilde{D}$$

を使用して、データベクトル

【数２３】

$$\tilde{D},$$

に基づいて計算されてもよい。

【００１６】

並列ＲＳ－ＲＡＩＤデータ記憶アーキテクチャは、各クラスタ内のデータおよびチェックサムを、全てのクラスタに転送されるかまたは配信される中間または部分和に集計しうる。中間データシンボル、中間チェックサムシンボル、データ記憶デバイスのクラスタへ

の割当てに関するクラスタ構成情報、およびデータ記憶デバイスの作動に関する状態などの使用は、並列ＲＳ－ＲＡＩＤ分散データ記憶アーキテクチャのスケーラビリティおよびスループットを向上させながら、誤り訂正計算についての計算負荷およびレーテンシを低減しうる。

【００１７】

本開示は、同じ数字が同じ要素を表す添付図面を参照することになる。

【図面の簡単な説明】

【００１８】

【図１】並列ＲＳ－ＲＡＩＤ（Ｒｅｅｄ－Ｓｏｌｏｍｏｎ　ｒｅｄｕｎｄａｎｔ　ａｒｒａｙ　ｏｆ　ｉｎｅｘｐｅｎｓｉｖｅ　ｄｉｓｋ）の例を示す図である。

【図２】構成行列の例を示す図である。

【図３】ＲＡＩＤコントローラの例を示す図である。

【図４Ａ】チェックサムプログラムの例のフローチャートである。

【図４Ｂ】チェックサム更新プログラムの例のフローチャートである。

【図５】データプログラムの例のフローチャートである。

【発明を実施するための形態】

【００１９】

図１は、データ記憶のための並列ＲＳ－ＲＡＩＤアーキテクチャ１００の例である。並列ＲＳ－ＲＡＩＤアーキテクチャ１００は、通信ファブリック１２００、ＲＡＩＤコントローラ１１１１～１１１３、および記憶デバイス１００１～１０１２を含みうる。記憶デバイス１００１～１００４、１００５～１００８、および１００９～１０１２は、それぞれ、ＲＡＩＤコントローラ１１１１～１１１３に結合されうる。換言すれば、記憶デバイス１００１～１０１２の部分集合またはクラスタは、それぞれの対応するＲＡＩＤコントローラ１１１１～１１１３に結合しうる。各ＲＡＩＤコントローラ１１１１～１１１３に結合しうる記憶デバイスの数は、等しくても、等しくなくてもよく、また、耐故障性を改善する、スループットを改善するなどのために、記憶デバイスのＲＡＩＤコントローラに対する構成またはマッピングは動的に変わってもよい。たとえば、記憶デバイス１００１～１０１２のＲＡＩＤコントローラ１１１１～１１１３に対する割当ては、構成行列または同様のデータ構造によって確定されてもよい。

【００２０】

図２は、変数「ｔ」を含みうる構成行列２００の例を示し、変数「ｔ」は、ＲＡＩＤコントローラの数のインデックスまたはカウンタである。たとえば、構成行列２００の行２０６は、ＲＡＩＤコントローラインデックス番号「ｔ」と、ＲＡＩＤコントローラ１１１１～１１１３などのそれぞれのＲＡＩＤコントローラとの間のマッピング関数Ｑ（ｔ）を示す。行２０２はＲＡＩＤ記憶デバイス開始インデックスＱＳ（ｔ）を示し、行２０４はＲＡＩＤ記憶デバイス終了インデックスＱＥ（ｔ）を示す。たとえば、ＱＳ（２）＝１００５であり、ＱＥ（２）＝１００８である。デバイス番号のオフセットが、関数Ｊ（）によって供給されてもよいため、Ｊ（ＱＳ（２））＝５であり、これは、たとえば、５番目の記憶デバイスが第２の記憶デバイスクラスタで開始することを示しうることが留意されてもよい。構成行列２００は、記憶デバイスを対応するＲＡＩＤコントローラにマッピングしうる。換言すれば、構成行列２００は、記憶デバイスのどの部分集合またはクラスタが、所与のＲＡＩＤコントローラに割当てられるかを確定するかまたは制御しうる。計算のために、構成行列２００は、以下で述べるように、コードワードを符合化するかまたは復号化しうる重み付き部分和の始まりと終わりを確定しうる、チェックサムおよびデータを更新するかまたは維持しうるなどを行いうる。

【００２１】

通信ファブリック１２００は、ＲＡＩＤコントローラ１１１１～１１１３間で、また、並列ＲＳ－ＲＡＩＤアーキテクチャ１００と外部デバイスとの間で入力および出力（Ｉ／Ｏ）デジタル信号を結合しうる。たとえば、通信ファブリック１２００は、ＲＡＩＤコントローラ１１１１～１１１３間で、データシンボル、チェックサムシンボル、中間データ

およびチェックサムシンボルなどのようなデジタル信号を結合しうる。通信ファブリック１２００は、並列バス構造、直列データリンク、光バックプレーンなどを使用してもよい。通信ファブリック１２００は、外部通信用に１つのタイプのバス、リンク、またはバックプレーン構造を、ＲＡＩＤコントローラ１１１１～１１１３間の通信用に別のタイプを使用してもよい。

【００２２】

ＲＡＩＤコントローラ１１１１～１１１３は、構成行列２００などの構成行列または他のデータ構造によって与えられた、割当てられた記憶デバイスの部分集合またはクラスタ内の各記憶デバイスについてデータチェックサムシンボルを計算しうる。ＲＡＩＤコントローラ１１１１～１１１３は、誤り訂正コード計算の部分和を集計するかまたは蓄積し、集計されたデータおよびパリティ計算結果を、通信ファブリック１２００を通じて、並列ＲＳ－ＲＡＩＤアーキテクチャ１００内の他のＲＡＩＤコントローラに報告しうる。データおよびチェックサムシンボルについての部分和計算の詳細が、特定のＲＡＩＤコントローラを参照して述べられてもよいが、対応する計算は、ＲＡＩＤコントローラ１１１１などの、並列ＲＳ－ＲＡＩＤアーキテクチャ１００内の任意のＲＡＩＤコントローラによって実施されてもよい。

【００２３】

図３は、通信ファブリックインタフェース１１１１ａ、ＲＡＩＤコントロールユニット１１１１ｂ、中間和デバイス１１１１ｃ、記憶デバイスインタフェース１１１１ｇ、および記憶デバイス故障センスユニット１１１１ｈを含みうるＲＡＩＤコントローラ１１１１の例を示す。通信ファブリックインタフェース１１１１ａは、通信ファブリック１２００などの通信ファブリックへのまた通信ファブリックからの信号を、中間和デバイス１１１１ｃおよびＲＡＩＤコントロールユニット１１１１ｂに結合しうる。ＲＡＩＤコントロールユニット１１１１ｂは、中間和デバイス１１１１ｃ、記憶デバイスインタフェース１１１１ｇ、および記憶デバイス故障センスユニット１１１１ｈに結合しうる。記憶デバイスインタフェース１１１１ｇは、ＲＡＩＤコントロールユニット１１１１ｂ、中間和デバイス１１１１ｃ、および記憶デバイス故障センスユニット１１１１ｈに結合しうる。ＲＡＩＤコントローラ１１１１は、先に述べたように、通信ファブリック１２００へまた通信ファブリック１２００から結合し、記憶デバイスインタフェース１１１１ｇを介して記憶デバイス１００１～１００４などの記憶デバイスへまた記憶デバイスから結合しうる。

【００２４】

中間和デバイス１１１１ｃは、中間和計算器１１１１ｄ、再計算器１１１１ｅ、および計算コントロール１１１１ｆを含みうる。中間和計算器１１１１ｄは、通信ファブリックインタフェース１１１１ａ、記憶デバイスインタフェース１１１１ｇ、再計算器１１１１ｅ、および計算コントロール１１１１ｆに結合しうる。再計算器１１１１ｅは、通信ファブリックインタフェース１１１１ａ、中間和計算器１１１１ｄ、計算コントロール１１１１ｆ、および記憶デバイスインタフェース１１１１ｇに結合しうる。計算コントロール１１１１ｆは、中間和計算器１１１１ｄ、再計算器１１１１ｅ、および記憶デバイスインタフェース１１１１ｇに結合しうる。

【００２５】

通信ファブリックインタフェース１１１１ａは、並列ＲＳ－ＲＡＩＤアーキテクチャ１００と外部デバイスとの間で情報シンボルを転送し、通信ファブリック１２００とＲＡＩＤコントローラ１１１１の要素との間で、情報シンボル、情報シンボルの所定部分、データシンボル、中間チェックサムシンボルなどのチェックサムシンボル、コントロール信号、クロック信号などを結合しうる。通信ファブリックインタフェース１１１１ａは、情報シンボルをビットからバイト、ワード、または他のシンボルにリフォーマットしうる、信号を多重化し逆多重化しうる、データ転送を同期化しうる、ラインドライバおよび受信機によって信号をバッファリングしうるなどを行いうる。換言すれば、通信ファブリックインタフェース１１１１ａは、デジタルバスなどの通信ファブリックを通じて送信するためにデジタル信号を調節しうる、データ転送をバッファリングしうるなどを行いうる。

【００２６】
ＲＡＩＤコントロールユニット１１１１ｂは、通信ファブリックインタフェース１１１１ａおよび記憶デバイスから信号を受信しうる、情報シンボルの部分集合からデータシンボルを選択しうる、記憶デバイスにわたってデータおよびチェックサムシンボルをストライピングしうる、前方誤り訂正コード（ｆｏｒｗａｒｄ－ｅｒｒｏｒ　ｃｏｒｒｅｃｔｉｏｎ　ｃｏｄｅ）（ＦＥＣｃｏｄｅ）によって中間和デバイス１１１１ｃの作動を制御しうるなどを行いうる。たとえば、情報シンボルの部分集合は、データシンボルによって表され、かつ、ＲＡＩＤコントローラ１１１１によって制御される作動可能なデータ記憶デバイスに記憶される情報シンボルでありうる。中間和デバイス１１１１ｃは、記憶デバイス故障センスユニット１１１１ｈから状態情報を得ることができるＲＡＩＤコントロールユニット１１１１ｂから作動可能な記憶デバイスの数に関する状態情報を受信してもよい。

【００２７】
記憶デバイス故障センスユニット１１１１ｈは、ＲＡＩＤコントローラ１１１１に結合する任意の記憶デバイスの作動に関する状態を確定し、作動可能な記憶デバイスのリストを確定しうる。換言すれば、記憶デバイス故障センスユニット１１１１ｈは、所与の記憶デバイスが、データおよびチェックサムの確実な記憶に適さなくなったかどうかを判定しうる。記憶デバイス故障センスユニット１１１１ｈは、信頼性のある作動について記憶デバイスを試験しうる、所与の記憶デバイスがオンラインであるかどうかを判定しうる、所与の記憶デバイスからの応答が、所定のタイムアウト間隔内に受信されない場合、ユニットオフラインを宣言しうる、信号品質メトリックが、記憶デバイスから読出されたデータについて閾品質より小さいかどうかを判定しうる、作動可能な記憶デバイスを挙げうるなどを行いうる。記憶デバイス故障センスユニット１１１１ｈは、こうした試験の結果を記録し、ＲＡＩＤコントロールユニット１１１１ｂなどのＲＡＩＤコントローラ１１１１の要素のために、作動可能な記憶デバイスのリストを配信しうる。

【００２８】
中間和計算器１１１１ｄは、中間的でローカルで部分的な和を計算することができ、この和内に、チェックサムおよびデータについての誤り訂正コード計算が、式８および式１３に関してそれぞれ述べたように分解されうる。中間的なまたは部分的な和は、ＲＡＩＤコントローラ１１１１に報告するクラスタ内の作動可能な記憶デバイスから読出されるシンボルの重み付き和であってよい。たとえば、記憶デバイスのクラスタおよびこうした部分和の合計の対応する制限は、構成行列２００などの構成行列または他のデータ構造から確定されてもよい。中間和計算器１１１１ｄは、ＲＡＩＤコントローラ１１１２またはＲＡＩＤコントローラ１１１３などの他のＲＡＩＤコントローラから対応する部分和を受信した後、データおよびチェックサムシンボルを計算しうる。

【００２９】
再計算器１１１１ｅは、ＲＡＩＤコントローラ１１１１に直接結合する記憶デバイスからのデータに基づいて中間的なローカルのチェックサムを、また、通信ファブリックインタフェース１１１１ａを通して転送される他のＲＡＩＤコントローラからのローカルでない中間チェックサムを再計算しうる。換言すれば、データまたはチェックサムシンボルの変化が、ＲＡＩＤコントローラ１１１１に直接結合するローカルの記憶デバイスにおいて起こるか、または、通信ファブリック１２００を通じてＲＡＩＤコントローラ１１１１に送信される中間チェックサムによって起こると、再計算器１１１１ｅは、中間和計算器１１１１ｄからの結果を相応して修正しうる。

【００３０】
計算コントロール１１１１ｆは、中間チェックサム計算結果または再計算されたチェックサムが、ＦＥＣのために使用されるべきかどうかを判定するために、中間和計算器１１１１ｄと再計算器１１１１ｅの両方を制御しうる。ＲＡＩＤコントロール１１１１ｂは、中間和計算器１１１１ｄの結果と再計算器１１１１ｅの結果のいずれが計算されるかを判定するために、直接にまたは通信ファブリックインタフェース１１１１ａを通して計算コ

ントロール１１１１ｆに合図し(signal)うる。ＲＡＩＤコントロール１１１１ｂは、記憶デバイス故障センスユニット１１１１ｈから、データ記憶デバイスに関する作動に関する状態などの状態情報を得ることができる。

【００３１】

並列ＲＡＩＤコントローラ１１１１～１１１３は、

【数２４】

$$c_i = \sum_{t=1}^{r} \sum_{j=J(QS(t))}^{J(QE(t))} d_j f_{i,j} = \sum_{t=1}^{r} c_{i,t} \quad \text{式７}$$

によって、チェックサムを計算し記憶しうる。式中、インデックスｔは、１からＲＡＩＤコントローラの数ｒまでの範囲にあることができ、$c_{i,t}$は、ｔ番目のインデックスについてのｉ番目の中間チェックサムである。たとえば、ｒは、並列ＲＡＩＤアーキテクチャ１００の場合、３に等しい。構成行列２００に関して述べたように、ＱＳ（ｔ）およびＱＥ（ｔ）は、開始および終了記憶デバイスをＲＡＩＤコントローラにマッピングし、それぞれの中間チェックサム$c_{i,j}$を生成する部分和の合計の制限を確定しうる。関数Ｊ（・）は、たとえば、Ｊ（１００２）＝２であるように、オフセットを減算しうる。

【００３２】

ＲＡＩＤコントローラ１１１１などのｔ番目のＲＳ－ＲＡＩＤコントローラは、

【数２５】

$$c_{i,t} = \sum_{j=J(QS(t))}^{J(QE(t))} d_j f_{i,j} \quad \text{式８}$$

によって、中間チェックサム$c_{i,t}$を計算しうる。

【００３３】

中間チェックサム$c_{i,t}$の使用は、通信ファブリック１２００上のデータトラフィックを低減することができ、並列ＲＳ－ＲＡＩＤアーキテクチャ１００のスループットを増加させることができる。たとえば、８＋４ＲＳ－ＲＡＩＤアーキテクチャでは、単一の主要なＲＡＩＤコントローラが、記憶デバイスの全てを制御し、チェックサムを計算する場合、８つのデータシンボルが、通信ファブリックを通じて転送されうる。対照的に、８＋４並列ＲＳ－ＲＡＩＤアーキテクチャからの中間チェックサム計算器結果を使用して、２つの中間チェックサムシンボルだけが、通信ファブリックを通じて転送される必要がある可能性がある。

【００３４】

中間チェックサムおよび全チェックサムを計算することに加えて、並列ＲＳ－ＲＡＩＤアーキテクチャ１００は、データシンボルが変化すると、チェックサムシンボルを修正するかまたは維持しうる。たとえば、データシンボルが、d_jからd'_jに変化すると、チェックサムは、

$c'_i = c_i + f_{i,j}(d'_j - d_j)$　式９

によって、再計算されうる。式９の計算を実施するとき、ＲＡＩＤコントローラ１１１１は、データ差（$d'_j - d_j$）を計算し、ファンデルモンデ要素$f_{i,j}$、すなわち、

【数２６】

$$c'_{i,t} = \sum_{j=J(QS(t))}^{J(QE(t))} f_{i,j}(d'_j - d_j) \quad \text{式１０}$$

によって、データ差に重み付けしうる。

【００３５】

個々の並列ＲＡＩＤコントローラ１１１１～１１１３は、一時的な成分$c'_{i,t}$をＲＡＩＤコントローラ１１１１～１１１３の他のコントローラに送出しうる。

【００３６】

ＲＳ－ＲＡＩＤコントローラ１１１１～１１１３は、

【数２７】

$$c'_i = c_i + \sum_{t=1}^{r} c'_{i,t} \quad \text{式１１}$$

によって、それぞれの割当てられた記憶デバイスを更新しうる。

【００３７】

記憶デバイスが故障する、たとえば、記憶デバイス故障センスユニット１１１１ｈがハードディスククラッシュを検出すると、拡大行列の逆行列Ｉｎｖ（Ａ）が、並列ＲＡＩＤコントローラ１１１１～１１１３によって修正されて、残りのまたは作動可能なデータ記憶デバイスに対応する逆行列Ｉｎｖ（Ａ'）が形成される。行列Ｉｎｖ（Ａ'）は、記憶デバイスのさらなる故障が起こらない限り、静的データ構造である可能性がある。別の記憶デバイスが故障すると、Ｉｎｖ（Ａ'）が一回計算され、その後、ＲＡＩＤコントローラ１１１１～１１１３などの全ての作動可能なＲＡＩＤコントローラにブロードキャストされうる。より多くの記憶デバイスが後で故障する場合、新しい逆行列Ｉｎｖ（Ａ''）が、再計算され、全てのＲＡＩＤコントローラにブロードキャストされてもよい。

【００３８】

並列ＲＳ－ＲＡＩＤアーキテクチャ１００は、記憶デバイスが故障しても、各ＲＡＩＤコントローラにおいてローカルで計算される中間和または部分和を使用して、データシンボルを回復しうる。回復されるデータ

【数２８】

$$\tilde{D} = \left[\in \tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_n \right]^T$$

は、

【数２９】

$$\begin{bmatrix} \tilde{d}_1 \\ \tilde{d}_2 \\ \vdots \\ \tilde{d}_n \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} \quad \text{式１２}$$

から回復される可能性がある。式中、Ｉｎｖ（Ａ'）の要素は、$a_{i,j}$（$1 \leqq i \leqq n$および$1 \leqq j \leqq n$）で示されてもよい。対応するデータおよびチェックサムシンボルＥ'＝［e_1、e_2、…、e_n］^{T}の要素は、作動可能でかつ選択されたデータ記憶デバイスから読出されうる。並列ＲＳ－ＲＡＩＤアーキテクチャ１００は、Ｅの行ならびに添加された単位行列とファンデルモンデ行列の対応する部分集合を選択するかまたは選抜して、それぞれ、Ｅ'およびＩｎｖ（Ａ'）が形成されうる。換言すれば、並列ＲＳ－ＲＡＩＤアーキテクチャ１００は、データ回復計算を、

【数３０】

$$\tilde{d}_{i,t} = \sum_{j=J(QS(t))}^{J(QE(t))} a_{i,j} \cdot e_j \quad \text{式１３}$$

によって、部分和または中間データシンボルの集合に分解しうる。式中、e_jは、ｔ番目のＲＳ－ＲＡＩＤコントローラの制御下にある全てのデータまたはチェックサムシンボルの集合であると理解される。

【００３９】

他の並列ＲＳ－ＲＡＩＤコントローラから、中間データシンボルなどのメッセージを受信すると、個々のＲＡＩＤコントローラは、最初に中間データシンボル

【数３１】

$$\tilde{d}_{i,t}$$

を計算し、次に、

【数３２】

$$\tilde{d}_i = \sum_{t=1}^{r} \tilde{d}_{i,t} . \quad 式１４$$

によって、回復されるデータ

【数３３】

$$\tilde{d}_i$$

を計算しうる。

【００４０】

図４Ａは、データ記憶用の並列ＲＳ－ＲＡＩＤアーキテクチャのためのチェックサムプログラムフローチャート４００Ａの例を示す。プログラムフローチャート４００Ａは、プログラムステップＳ４１０で開始し、並列ＲＳ－ＲＡＩＤアーキテクチャの構成行列が読出されうるプログラムステップＳ４２０に進みうる。たとえば、構成行列は、図２に関して述べた構成行列などの、所与のＲＡＩＤコントローラに関連する記憶デバイスについての開始デバイス番号と終了デバイス番号を指定しうる。各ＲＡＩＤコントローラは、構成行列のローカルなコピーを記憶しうる、構成行列を他のＲＡＩＤコントローラと調和させうる、構成行列を、高レベルのＲＡＩＤデバイスまたはネットワークコントローラから受信しうるなどを行いうることが理解されてもよい。

【００４１】

プログラムステップＳ４２０から、プログラムフローは、プログラムステップＳ４２５に進むことができ、プログラムステップＳ４２５にて、プログラムは、外部デバイスから、記憶される情報保持データシンボルを読出すことができる。たとえば、プログラムは、通信ファブリックを通じて受信される２Ｋビットのデータブロックの集合をフラッシュドライブから受信しうる。

【００４２】

プログラムステップＳ４２５から、プログラムは、中間チェックサムが計算されうるプログラムステップＳ４３０に進むことができる。たとえば、中間チェックサムまたはデータおよびパリティ計算は、式８およびガロア体の特性を使用した、データワードの線形結合から計算されうる。プログラムステップＳ４３０は、中間チェックサムを計算し、１）所与のＲＡＩＤコントローラにコードワードシンボルの所定部分を供給する個々の作動可能な記憶ユニットからの記憶されたコードワードシンボルを使用して中間チェックサムを更新するかまたは維持し、２）通信ファブリックを通じて所与のＲＡＩＤコントローラと通信する他のＲＡＩＤコントローラからの中間チェックサムを集計しうる。換言すれば、ローカルな部分集合データ記憶ユニットからの低レーテンシのデータおよびパリティビット、バイト、またはワードは、他のデータ記憶ユニットからの、$c_{i,t}$の形態の、高レーテンシの、蓄積されるかまたは部分的に合計されたデータおよびパリティと結合されうる。その後、プログラムは、プログラムステップＳ４６０に進む。

【００４３】

ステップＳ４６０にて、プログラムは、中間チェックサムを異なるＲＳ－ＲＡＩＤコントローラに配信しうる。たとえば、Ｑ（ｔ）＝ｔである場合、プログラムステップＳ４６０は、第１のＲＡＩＤコントローラからの第１の中間チェックサム$c_{1,1}$を第２および第３のＲＡＩＤコントローラに配信しうる。

【００４４】

プログラムステップＳ４６０から、プログラムフローは、プログラムステップＳ４７０に進むことができ、プログラムステップＳ４７０にて、プログラムは、他のＲＡＩＤコントローラから中間チェックサムを受信しうる。プログラムステップＳ４７０から、プログラムはプログラムステップＳ４８０に進みうる。中間チェックサムの集合によって、各ＲＡＩＤコントローラが、式８によって完全なチェックサムc_iを計算し、その後の誤り訂正および検出計算についてc_iを記憶することが可能になる。たとえば、プログラムは、第２および第３の中間チェックサム$c_{i,2}$および$c_{i,3}$を受信することができ、第２

および第３の中間チェックサム$c_{i,2}$および$c_{i,3}$は、ローカルに計算された第１のチェックサム$c_{i,1}$と共に、c_1を計算するためのチェックサムの十分な集合を形成しうる。

【００４５】

プログラムステップＳ４８０から、プログラムフローは、プログラムステップＳ４９０に進むことができ、プログラムステップＳ４９０にて、プログラムは、プログラムを実行するＲＡＩＤコントローラに割当てられるデータおよび完全なチェックサムシンボルを記憶しうる。たとえば、プログラムは、ディスクのアレイにわたってデータおよびチェックサムシンボルをストライピングしうる。プログラムステップＳ４９０から、プログラムフローは、プログラム実行が停止しうるプログラムステップＳ４９５に進みうる。

【００４６】

図４Ｂは、データ記憶用の並列ＲＳ－ＲＡＩＤアーキテクチャのためのチェックサム更新プログラムフローチャート４００Ｂの例を示す。プログラムフローチャート４００Ｂは、ステップＳ４４０で開始し、ステップＳ４４２に進む。

【００４７】

ステップＳ４４２にて、並列ＲＳ－ＲＡＩＤアーキテクチャは、データの変化を受信する可能性がある。たとえば、記憶デバイスは、古いデータシンボルを置換するために新しいデータシンボルを受信してもよい。その後、プログラムフローはステップＳ４４４に進みうる。

【００４８】

ステップＳ４４４にて、記憶デバイスに結合するＲＡＩＤコントローラは、式１０によって、一時的な成分を計算しうる。ＲＡＩＤコントローラは、新しいデータシンボルと古いデータシンボルとのデータ差を得、ファンデルモンデ行列要素によってデータ差に重み付けしてもよい。その後、プログラムフローはステップＳ４４６に進みうる。

【００４９】

ステップＳ４４６にて、一時的な成分は、他のＲＡＩＤコントローラに伝達されうる。ある実施形態では、通信ファブリックは、種々のＲＡＩＤコントローラを結合してもよい。通信ファブリックは、データ変化に対応する一時的な成分を、チェックサムを記憶する記憶デバイスを制御するＲＡＩＤコントローラに伝達しうる。その後、プログラムフローはステップＳ４４８に進みうる。

【００５０】

ステップＳ４４８にて、チェックサムを記憶する記憶デバイスを制御するＲＡＩＤコントローラは、たとえば、式１１によって、受信した一時的な成分に基づいてチェックサムを更新してもよい。その後、プログラムフローはステップＳ４５０に進み、停止しうる。

【００５１】

図５は、データ記憶用の並列ＲＳ－ＲＡＩＤアーキテクチャのためのデータプログラムフローチャート５００の例を示す。プログラムフローチャート５００は、ステップＳ５１０で開始し、ステップＳ５２０に進むことができ、ステップＳ５２０にて、並列ＲＳ－ＲＡＩＤアーキテクチャの構成行列が、図４Ａに関して説明したように読出されうる。プログラムステップＳ５２０から、プログラムフローは、プログラムステップＳ５２５に進むことができ、プログラムステップＳ５２５にて、データおよびチェックサムシンボルが、記憶デバイスから読出されうる。たとえば、８つのデータおよび４つのチェックサムシンボルは、１２の記憶デバイスから読出されうる。この例では、少なくとも８つのデータまたはチェックサムシンボルが、作動可能な記憶デバイスから読出されうる。

【００５２】

プログラムステップＳ４２５から、プログラムフローは、プログラムステップＳ５３０に進むことができ、プログラムステップＳ５３０にて、プログラムは、中間データシンボルを計算しうる。たとえば、プログラムは、式１３によって、中間データシンボルを計算しうる。式１３で使用される重み係数$a_{i,j}$は、予め計算され、ＲＡＩＤコントローラに配信されるか、または、プログラムステップＳ５２０において構成行列を読出した後な

どに、必要に応じて再計算されてもよいことが理解されてもよい。プログラムステップＳ５３０から、プログラムフローは、プログラムステップＳ５４０に進むことができ、プログラムステップＳ５４０にて、プログラムは、中間データシンボルを並列ＲＡＩＤコントローラに配信しうる。

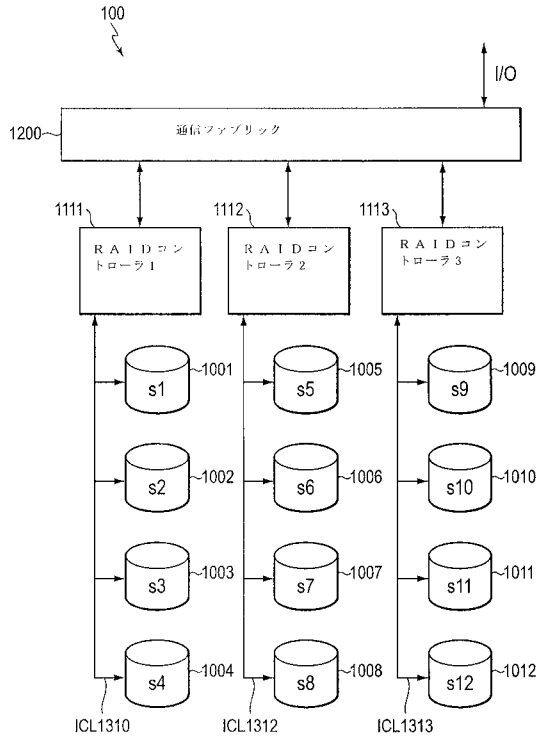
【００５３】

プログラムステップＳ５４０から、プログラムフローは、プログラムステップＳ５５０に進むことができ、プログラムステップＳ５５０にて、プログラムは、並列ＲＡＩＤコントローラから中間データシンボルを受信しうる。プログラムステップＳ５５０から、プログラムフローは、プログラムステップＳ５６０に進むことができ、プログラムステップＳ５６０にて、プログラムは、ローカルなＲＡＩＤコントローラと並列ＲＡＩＤコントローラの両方から来た中間データシンボルから、回復されるデータシンボルを計算しうる。換言すれば、プログラムは、式１４によって、中間データシンボルを合計しうる。プログラムステップＳ５６０から、プログラムフローは、プログラム実行が停止しうるプログラムステップＳ５７０に進むことができる。

【００５４】

本発明は、本発明の特定の例示的な実施形態に関連して述べられたが、多くの代替、修正、および変形が当業者に明らかになることが明らかである。したがって、本明細書で述べる本発明の実施形態は、制限的でなく、例証的であることを意図される。本発明の精神および範囲から逸脱することなく行われてもよい変更が存在する。

【図１】

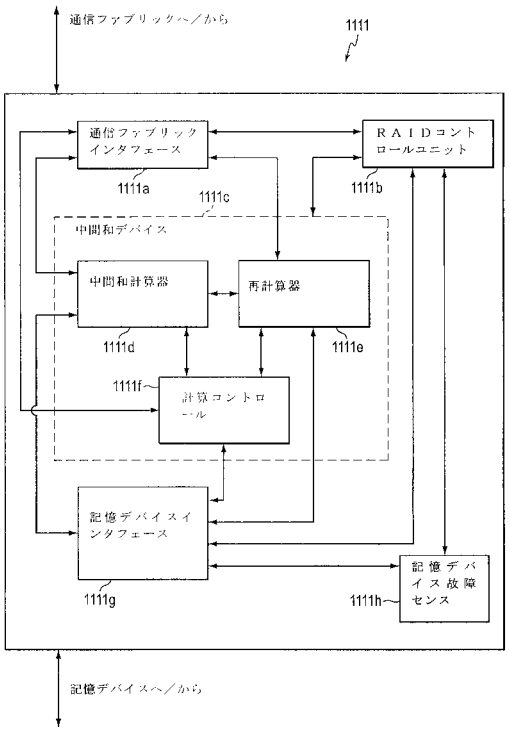


【図２】

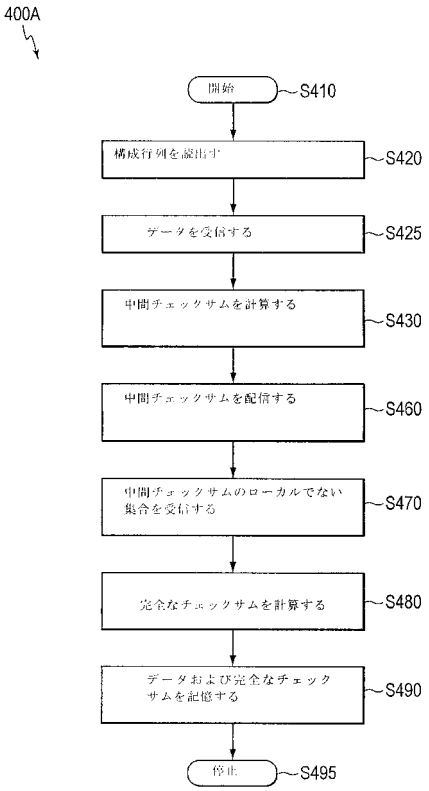
200

	t	1	2	3
202	QS (t)	1001	1005	1009
204	QE (t)	1004	1008	1012
206	Q (t)	1111	1112	1113

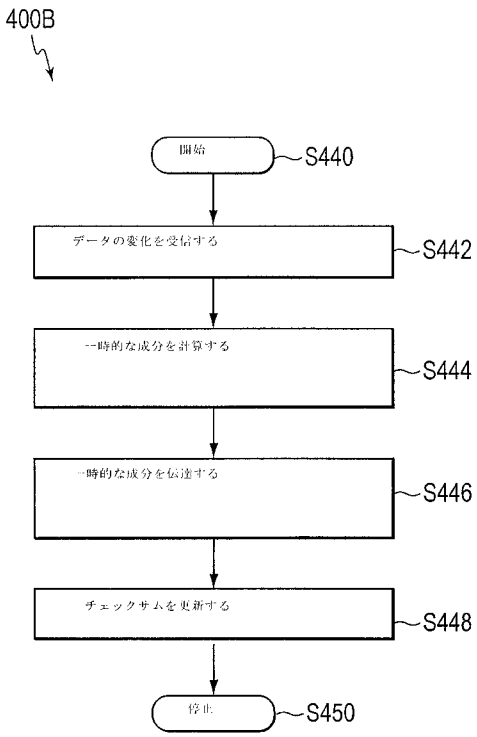
【図３】


通信ファブリックへ／から
1111
通信ファブリックインタフェース
RAIDコントロールユニット
1111a
1111c
1111b
中間和デバイス
中間和計算器
再計算器
1111d
1111e
1111f
計算コントロール
記憶デバイスインタフェース
記憶デバイス故障センス
1111g
1111h
記憶デバイスへ／から

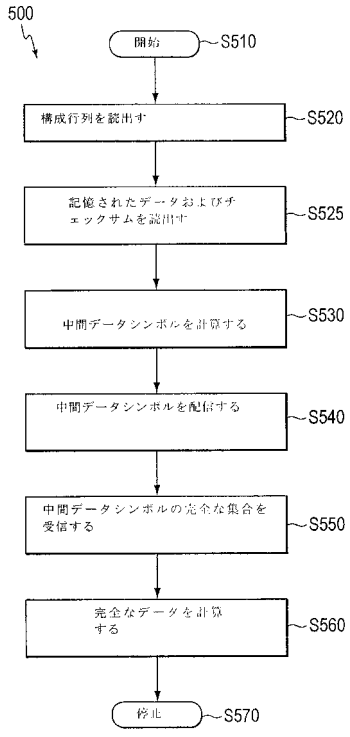
【図４Ａ】


400A
開始
S410
構成行列を読出す
S420
データを受信する
S425
中間チェックサムを計算する
S430
中間チェックサムを配信する
S460
中間チェックサムのローカルでない集合を受信する
S470
完全なチェックサムを計算する
S480
データおよび完全なチェックサムを記憶する
S490
停止
S495

【図４Ｂ】


400B
開始
S440
データの変化を受信する
S442
一時的な成分を計算する
S444
一時的な成分を伝達する
S446
チェックサムを更新する
S448
停止
S450

【図５】


500
開始
S510
構成行列を読出す
S520
記憶されたデータおよびチェックサムを読出す
S525
中間データシンボルを計算する
S530
中間データシンボルを配信する
S540
中間データシンボルの完全な集合を受信する
S550
完全なデータを計算する
S560
停止
S570

フロントページの続き

(56)参考文献 特開２００７－２５７６３０（ＪＰ，Ａ）
特開２００３－１３１８１８（ＪＰ，Ａ）
米国特許出願公開第２００７／０２４５１７３（ＵＳ，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　　３／０６