(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5102776号
(P5102776)

(45)発行日　平成24年12月19日(2012.12.19)　(24)登録日　平成24年10月5日(2012.10.5)

(51)Int.Cl.　　F I
G06F 3/06 (2006.01)　　G06F 3/06 305C
G06F 12/16 (2006.01)　　G06F 3/06 540
　　G06F 12/16 320L

請求項の数 34　(全 38 頁)

(21)出願番号　特願2008-545792 (P2008-545792)
(86)(22)出願日　平成18年12月14日 (2006.12.14)
(65)公表番号　特表2009-524124 (P2009-524124A)
(43)公表日　平成21年6月25日 (2009.6.25)
(86)国際出願番号　PCT/US2006/047647
(87)国際公開番号　WO2007/078803
(87)国際公開日　平成19年7月12日 (2007.7.12)
審査請求日　平成21年9月24日 (2009.9.24)
(31)優先権主張番号　11/304,369
(32)優先日　平成17年12月15日 (2005.12.15)
(33)優先権主張国　米国 (US)

(73)特許権者　303039534
ネットアップ，インコーポレイテッド
アメリカ合衆国　カリフォルニア　９４０８９，　サニーヴェール，　イースト　ジャバ　ドライブ　４９５
(74)代理人　100087642
弁理士　古谷　聡
(74)代理人　100076680
弁理士　溝部　孝彦
(74)代理人　100121061
弁理士　西山　清春
(72)発明者　コルベット，ピーター，エフ.
アメリカ合衆国マサチューセッツ州０２４２０，レキシントン，サマー・ストリート・３３

最終頁に続く

(54)【発明の名称】ストレージアレイにおける三重故障からの効率的な復旧を可能にする三重パリティ技術

(57)【特許請求の範囲】
【請求項１】
ストレージアレイにおける記憶装置の３以下の同時故障からの復旧を可能にする方法であって、
データ、及び行パリティを格納するように構成された複数の第１の装置、対角パリティを格納するように構成された１つの対角パリティ装置、並びに反対角パリティを格納するように構成された反対角パリティ装置を含む所定数の記憶装置を備え、前記記憶装置の所定数ｎがｐ＋２であり、ｐが素数であるアレイを用意するステップと、
各装置を複数のブロックに分割するステップと、
前記ブロックを各装置上の同数のブロックを含む複数のストライプに編成するステップであって、各ストライプがｎ－３行のブロックを含むように前記ブロックを編成するステップと、
前記複数の第１の装置にわたって規定される対角パリティ集合に沿って対角パリティを定義するステップであって、前記対角パリティ集合が、ｎ－３行のグループの中で循環し、あるストライプの対角パリティ集合に属するブロックが全て、そのストライプに格納されるように、対角パリティを定義するステップと、
１つを除く全ての対角パリティ集合について、対角パリティを計算し、前記対角パリティ装置に格納するステップと、
前記複数の第１の装置にわたって規定される反対角パリティ集合に沿って反対角パリティを定義するステップであって、前記反対角パリティ集合が、ｎ－３行のグループの中

で循環し、あるストライプの反対角パリティ集合に属するブロックが全て、そのストライプに格納されるように、反対角パリティを定義するステップと、
　１つを除く全ての反対角パリティ集合について、反対角パリティを計算し、前記反対角パリティ装置に格納するステップと
からなる方法。
【請求項２】
あるストライプ中の行パリティブロックが全て、単一の装置に格納される、請求項１に記載の方法。
【請求項３】
前記第１の装置は複数のデータ装置を含み、前記データ装置は全て存在するのではなく、不在のデータ装置はゼロ値のデータを格納するものとして処理される、請求項１に記載の方法。
【請求項４】
異なる装置内において、パリティブロックの位置が、装置ごとにシフトされる、請求項１に記載の方法。
【請求項５】
前記ストライプのサイズは、２の乗数個のビットである、請求項１に記載の方法。
【請求項６】
ストレージアレイにおける記憶装置の３以下の同時故障からの復旧を可能にするように構成されたシステムであって、
　データ、及びパリティを格納するように構成された複数の第１の装置、対角パリティを格納するように構成された１つの対角パリティ装置、並びに反対角パリティを格納するように構成された反対角パリティ装置を含む所定数の記憶装置を備え、前記記憶装置の所定数ｎがｐ＋２であり、ｐが素数である前記ストレージアレイと、
(i)前記複数の第１の装置にわたって規定される対角パリティ集合に沿って対角パリティを計算し、(ii)１つを除く全ての対角パリティ集合について対角パリティを前記対角パリティ装置に格納し、(iii)前記複数の第１の装置にわたって規定される反対角パリティ集合に沿って反対角パリティを計算し、(iv)１つを除く全ての反対角パリティ集合について反対角パリティを前記反対角パリティ装置に格納する、三重パリティ(ＴＰ）技術を実施するように構成されたデバイスストレージ層を含むストレージオペレーティングシステムと、
　前記ストレージオペレーティングシステムを実行することにより、前記ＴＰパリティ技術にしたがって前記ストレージアレイに対する双方向のストレージアクセス操作を実施するように構成された処理要素と
からなるシステム。
【請求項７】
あるストライプ中の行パリティブロックが全て、単一の装置に格納される、請求項６に記載のシステム。
【請求項８】
前記デバイスストレージ層はＲＡＩＤシステムであり、前記記憶装置はディスクである、請求項６に記載のシステム。
【請求項９】
前記ＲＡＩＤシステムは、各ディスクをさらに複数のブロックに分割し、該ブロックを複数のストライプに編成する、請求項８に記載のシステム。
【請求項１０】
ｎ＝ｐ＋２として、各ストライプが、ｎ－３行のブロックを含み、各行が、各ディスクから１つのブロックを有する、請求項９に記載のシステム。
【請求項１１】
前記装置は、ビデオテープ、磁気テープ、光学媒体、ＤＶＤ、バブルメモリ、磁気ディスク、電気的ランダムアクセスメモリ、及びＭＥＭＳデバイスのうちのいずれかである、

請求項６に記載のシステム。
【請求項１２】
前記第１の装置は複数のデータ装置を含み、前記データ装置は全て存在するのではなく、不在のデータ装置はゼロ値のデータを格納するものとして処理される、請求項６に記載のシステム。
【請求項１３】
ストレージアレイにおける記憶装置の３以下の同時故障からの復旧を可能にする装置であって、
データ、及び行パリティを格納するように構成された複数の第１の装置、対角パリティを格納するように構成された１つの対角パリティ装置、並びに反対角パリティを格納するように構成された反対角パリティ装置を含む所定数の記憶装置を備え、前記記憶装置の所定数ｎがｐ＋２であり、ｐが素数であるアレイを設ける手段と、
各装置を複数のブロックに分割する手段と、
前記複数の第１の装置にわたって規定される対角パリティ集合に沿って対角パリティを定義する手段であって、前記対角パリティ集合が、ｎ－３行のグループの中で循環し、あるストライプの対角パリティ集合に属するブロックが全て、そのストライプに格納されるように、対角パリティを定義する手段と、
１つを除く全ての対角パリティ集合について、対角パリティを計算し、前記対角パリティ装置に格納する手段と、
前記複数の第１の装置にわたって規定される反対角パリティ集合に沿って反対角パリティを定義する手段であって、前記反対角パリティ集合が、ｎ－３行のグループの中で循環し、あるストライプの反対角パリティ集合に属するブロックが全て、そのストライプに格納されるように、反対角パリティを定義する手段と、
１つを除く全ての反対角パリティ集合について、反対角パリティを計算し、前記反対角パリティ装置に格納する手段と
からなる装置。
【請求項１４】
あるストライプ中の行パリティブロックが全て、単一の装置に格納される、請求項１３に記載の装置。
【請求項１５】
各装置を複数のブロックに分割する手段と、
前記ブロックを複数のストライプに編成する手段と
をさらに含む、請求項１３に記載の装置。
【請求項１６】
前記格納する手段は、１つを除き、ストライプの対角パリティ集合のそれぞれについて、対角パリティブロックを前記対角パリティ装置に格納する手段を含む、請求項１５に記載の装置。
【請求項１７】
前記格納する手段は、１つを除き、ストライプの反対角パリティ集合のそれぞれについて、反対角パリティブロックを前記反対角パリティ装置に格納する手段を含む、請求項１５に記載の装置。
【請求項１８】
前記第1の装置は複数のデータ装置を含み、前記データ装置は全て存在するのではなく、不在のデータ装置はゼロ値のデータを格納するものとして処理される、請求項１３に記載の装置。
【請求項１９】
前記ストライプのサイズは、２の乗数個のビットである、請求項１５に記載の装置。
【請求項２０】
各ストライプ中のブロックの数に第1のブロックサイズを乗じたものは、ファイルシステムがストレージアレイにアクセスするために使用する第２のブロックサイズに等しい、

請求項１５に記載の装置。
【請求項２１】
ストレージアレイにおける２つの記憶装置の２以下の同時故障からの復旧を可能にする実行可能プログラム命令が格納されたコンピュータ読み取り可能な記憶媒体であって、前記実行可能プログラム命令が、
データ、及び行パリティを格納するように構成された複数の第1の装置、対角パリティを格納するように構成された１つの対角パリティ装置、並びに反対角パリティを格納するように構成された反対角パリティ装置を含む所定数の記憶装置を備え、前期記憶装置の所定数ｎがｐ＋２であり、ｐが素数であるアレイを形成し、
各装置を複数のブロックに分割し、
各ストライプがｎ－３行のブロックを含むように、前記ブロックを各装置上の同数のブロックを含む複数のストライプに編成し、
前記複数の第１の装置にわたって規定される対角パリティ集合に沿って対角パリティを定義し、前記対角パリティ集合が、ｎ－３行のグループの中で循環し、あるストライプの対角パリティ集合に属するブロックが全て、そのストライプに格納されるようにし、
１つを除く全ての対角パリティ集合について、対角パリティを計算し、前記対角パリティ装置に格納し、
前記複数の第１の装置にわたって規定される反対角パリティ集合に沿って反対角パリティを定義し、前記反対角パリティ集合が、ｎ－３行のグループの中で循環し、あるストライプの反対角パリティ集合に属するブロックが全て、そのストライプに格納されるようにし、
１つを除く全ての反対角パリティ集合について、反対角パリティを計算し、前記反対角パリティ装置に格納するためのプログラム命令からなる、コンピュータ読み取り可能な記憶媒体。
【請求項２２】
あるストライプ中の行パリティブロックが全て、単一の装置に格納される、請求項２１に記載のコンピュータ読み取り可能な記憶媒体。
【請求項２３】
ストレージアレイにおける３つの記憶装置の３以下の同時故障からの復旧を可能にする方法であって、
データ、及び行パリティを格納するように構成された複数の第１の装置、対角パリティを格納するように構成された対角パリティ装置、並びに反対角パリティを格納するように構成された反対角パリティ装置を含む所定数の記憶装置を備えたアレイを用意するステップと、
前記複数の第１の装置にわたって規定される対角パリティ集合に沿って対角パリティを計算するステップと、
１つを除く全て前記対角パリティ集合について、対角パリティを前記対角パリティ装置に格納するステップと、
前記複数の第１の装置にわたって規定される反対角パリティ集合に沿って反対角パリティを計算するステップと、
１つを除く全ての前記反対角パリティ集合について、反対角パリティを前記反対角パリティ装置に格納するステップと
からなる方法。
【請求項２４】
各装置を複数のブロックに分割するステップと、
前記ブロックを複数のストライプに編成するステップと
をさらに含む、請求項２３に記載の方法。
【請求項２５】
前記第１の装置は複数のデータ装置を含み、前記データ装置は全て存在するのではなく、不在のデータ装置はゼロ値のデータを格納するものとして処理される、請求項２３に記

載の方法。
【請求項２６】
前記記憶装置の所定数はｎであり、ｎ＝ｐ＋２であり、ｐは素数である、請求項２３に記載の方法。
【請求項２７】
ストレージアレイにおける記憶装置の３つの同時故障からの復旧を可能にする方法であって、
失われた対角パリティ、及び反対角パリティを計算するステップと、
行、対角、及び反対角のそれぞれに沿って、一組の故障した記憶装置のそれぞれに対し、代数計算を実施するステップと、
故障した中間記憶装置に対して４タプル一組の和を計算するステップと
からなる方法。
【請求項２８】
前記失われた対角パリティを計算するステップは、対角パリティ記憶装置上の複数のブロックを計算することからなる、請求項２７に記載の方法。
【請求項２９】
前記失われた反対角パリティを計算するステップは、反対角パリティ記憶装置上の複数のブロックを計算することからなる、請求項２７に記載の方法。
【請求項３０】
前記中間記憶装置に対して４タプル一組の和を計算するステップは、
故障した記憶装置に順位を付けるステップと、
前記記憶装置の各行について
(a) 一行の記憶装置を選択し、
(b) 前記選択された行に対応する失われた記憶装置上のブロックの行パリティ和を読み出し、
(c) 第１の故障した記憶装置について、前記選択された行にあるブロックの対角パリティを読み出し、
(d) 第２の故障した記憶装置について、前記選択された行にあるブロックの反対角パリティを読み出し、
(e) 前記反対角、及び対角の最後の行に対応する失われた記憶装置のブロックの行パリティ和を読み出すステップと、
前記ステップ(b)、(c)、(d)、及び(e)の結果に対して排他的論理和（ＸＯＲ）演算を実施することにより、４タプル和を形成するステップと
をさらに含む、請求項２７に記載の方法。
【請求項３１】
前記４タプル和を前記中間記憶装置上での二つ一組の和に低減するステップと、
前記中間記憶装置を復旧するステップと
をさらに含む、請求項２７に記載の方法。
【請求項３２】
行－対角パリティ復元技術を実施し、残りの故障した記憶装置を復元するステップをさらに含む、請求項３１に記載の方法。
【請求項３３】
前記記憶装置はディスクである、請求項２７に記載の方法。
【請求項３４】
前記代数演算は、排他的論理和（ＸＯＲ）からなる、請求項２７に記載の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
［発明の分野］
本発明は、ストレージシステムのアレイに関し、より具体的には、ストレージアレイの

任意の３つの記憶装置の故障を効率的に復元する技術に関する。
【背景技術】
【０００２】
[発明の背景]
ストレージシステムは通常、要求に応じてデータを出し入れすることが可能な１以上の記憶装置を含む。ストレージシステムは、種々のストレージアーキテクチャにしたがって実施され、限定はしないが、例えば、ネットワーク・アタッチド・ストレージ環境、ストレージ・エリア・ネットワーク、あるいは、クライアント、若しくはホストコンピュータに直接取り付けられたディスクアセンブリのような種々のストレージアーキテクチャにしたがって実施される。記憶装置は通常、ディスクドライブであり、ここで言う「ディスク」という用語は一般に、内蔵型の回転式磁気媒体記憶装置を意味する。この文脈において「ディスク」という用語とは、ハードディスクドライブ（ＨＤＤ）やダイレクト・アクセス・ストレージ・デバイス（ＤＡＳＤ）と同義である。
【０００３】
ストレージシステム内のディスクは一般に１以上のグループに編成され、各グループが、ＲＡＩＤ（Redundant Array of Independent (Inexpensive) Disks）として運用される。大半のＲＡＩＤ実施形態は、ＲＡＩＤグループを成す所与の数の物理ディスクにわたってデータ「ストライプ」を冗長書き込みし、そのストライピングされたデータに対する冗長情報を適切に記憶することによって、データ記憶の信頼性／完全性を向上させる。冗長情報により、記憶装置が故障したときに失われたデータの復元が可能になる。
【０００４】
ディスクアレイの動作中に、ディスクは故障することがある。高い処理能力を持つストレージシステムの目標は、平均データ紛失時間（ＭＴＴＤＬ）を出来る限り長くすることであり、ＭＴＴＤＬは、システムの期待サービス寿命よりも遥かに長いことが好ましい。１以上のディスクが故障すると、データは失われることがあり、装置からデータを復旧できなくなることがある。データ紛失を回避する典型的な手法としては、ミラーリング、バックアップ、及びパリティ保護などがある。ディスクのようなストレージリソースの消費の観点からすれば、ミラーリングは高価な解決策である。バックアップは、バックアップ作成後に変更が加えられたデータを保護することができない。パリティ技術が一般的である理由は、システムに一台のディスクドライブを追加するだけで、単一消去（一台のディスクの紛失）を許容するデータの冗長符号化が可能となるからである。
【０００５】
パリティ保護は、コンピュータシステムにおいて、ディスクのような記憶装置上のデータの紛失に対する保護を提供するために使用される。パリティ値は、異なるデータを有する複数の同様のディスクにわたって特定ワードサイズ（通常は１ビット）のデータを加算（通常はモジューロ２）することによって計算され、その結果が、さらに別の同様のディスクに格納される。すなわち、パリティは、各ディスク上の対応する位置にあるビットから構成される１ビット幅の種々のベクトルに対して計算される。パリティが１ビット幅のベクトルに対して計算される場合、パリティは、和として計算される場合もあれば、その補数として計算される場合もある。それらのパリティはそれぞれ、偶数（ＥＶＥＮ）パリティ、及び奇数（ＯＤＤ）パリティと呼ばれる。１ビットベクトルに対する加算、及び減算はいずれも、排他的論理和（ＸＯＲ）演算と等価である。次に、いずれかの１つのディスクの紛失、またはいずれか１つのディスクにおける任意部分のデータの紛失から、データは保護される。パリティを格納するディスクが失われた場合、そのパリティは、データから復元することができる。データディスクの１つが失われた場合、そのデータは、生き残っているデータディスクの中身を加算し、その結果を格納されたパリティから減算することによって再現することができる。
【０００６】
通常、ディスクは幾つかのパリティグループに分割され、各パリティグループは、１以上のデータディスクと、１つのパリティディスクとを含む。パリティ集合は、幾つかのデ

ータブロックと、１つのパリティブロックとを含むブロックの集合であり、パリティブロックは、全てのデータブロックをＸＯＲ演算したものである。パリティグループは、１以上のパリティ集合を選出する元になる一組のディスクである。ディスク空間は幾つかのストライプに分割され、各ストライプは、各ディスクから１つのブロックを格納する。ストライプを形成する幾つかのブロックは通常、パリティグループ内の各ディスク上の同じ位置にある。ストライプ内で、１つを除く全てのブロックは、データを格納するブロック（「データブロック」）であり、１つのブロックだけは、全てのデータのＸＯＲを取ることによって計算されたパリティを格納するブロック（「パリティブロック」）である。パリティブックが全て１つのディスクに格納され、それによってパリティ情報を全て含む（且つ、パリティ情報しか持たない）単一のディスクが形成される場合、ＲＡＩＤ－４実施形態が提供される。各ストライプにおいて異なるディスク上に（通常は、巡回パターンを成すようにして）パリティブロックが格納される場合、実施形態はＲＡＩＤ－５である。ＲＡＩＤという用語、及びその種々の実施形態については広く知られており、１９９８年６月に、D.A. Gibson 、及びR.H. Katzにより、「A Case for Redundant Arrays of Inexpensive Disks (RAID)」と題するデータ管理に関する国際会議論文（Proceedings of the International Conference on Management of Data）に開示されている。

【０００７】

本明細書において、「符号化（encoding）」という用語は、データブロックの所定のサブセットに対する冗長値の計算を指し、「復号（decoding）」という用語は、データブロック、及びパリティブロックのサブセットを利用した、データブロック、又はパリティブロックの復元を意味する。パリティグループ内の１つのディスクが故障した場合、そのディスクの中身は、残りのデータブロックの中身を全て加算し、その結果をパリティブロックから差し引くことにより、予備ディスク（複数の場合もあり）上に復号（復元）することができる。１ビット幅での２の補数による加算と減算はいずれも、ＸＯＲ演算と等価であるから、復元は、生き残ったデータブロックとパリティブロックを全てＸＯＲ演算することからなる。同様に、パリティディスクが失われた場合も、同様の仕方で、生き残ったデータからそれを再計算することが出来る。

【０００８】

パリティ技術は通常、パリティグループ内の単一のディスク故障に対する保護を提供する。各故障が異なるパリティグループで発生する限り、パリティ技術は、複数のディスク故障に対する保護を提供することも可能である。ただし、１つのパリティグループ内で２つのディスクが同時に故障した場合、復元不能なデータ紛失を被る。１つのパリティグループ内で２つのディスクが同時に故障することは、極めて一般的に起こりうる。特に、ディスクの「磨耗」やディスクの動作に関する環境要因が原因で発生する。この文脈において、１つのパリティグループ内における２つのディスクの同時故障は、「二重故障」と呼ばれる。

【０００９】

二重故障は一般に、一台のディスクが故障した後、その最初の故障からの復元を試みている間に、別のディスクが続けて故障する結果として発生する。復元時間、すなわち復旧時間は、ストレージシステムの活動レベルに応じて変わる。すなわち、故障したディスクを復元している間も、ストレージシステムは「オンライン」状態にあり、（クライアント、又はユーザからの）データアクセス（すなわち、読み出し、及び／又は書き込み）の要求に対してサービスを提供し続けることができる。ストレージシステムが要求に対する応答に忙しい場合、復元のための経過時間は増大する。また、失われたデータを復元するためには、生き残ったディスクを全て読み出さなければならないため、ストレージシステム内のディスクのサイズや数が増えるほど、復元処理時間も増大する。さらに、二重故障率は、パリティグループ内のディスク数の二乗に比例する。しかしながら、パリティグループを小さくすると、各パリティグループにつき、一台のディスク全体を冗長データの記憶のために専用に使用しなければならないため、費用がかかる。

【００１０】

ディスクのさらに別の故障形態は、ディスクの単一のブロック、又はセクタが読み出せなくなるメディア読み出しエラーである。ストレージアレイにパリティが保持されていれば、読み出せないデータを復元できることがある。ただし、アレイ内の一台のディスクが既に故障しているときに、さらに別のディスク上にメディア読み出しエラーが発生した場合、データは失われる。これが、二重故障の第２の形態である。

【００１１】

二重故障の訂正に必要となる冗長情報の最小量が２単位であることは、簡単に示すことができる。したがって、データディスクに追加することが可能なパリティディスクの最小数は、２である。これは、複数のディスクにわたってパリティが分散配置されるか、追加された２台のディスク上にパリティが集中配置されるかに関わらず、常に当てはまる。

【００１２】

二重故障を訂正する既知のパリティ技術として、失われた（故障した）ディスクの逐次復元が可能なＥＶＥＮＯＤＤ　ＸＯＲを利用する技術がある。ＥＶＥＮＯＤＤパリティは、ちょうどディスク２台分の冗長データを必要とし、この量が最適である。このパリティ技術によれば、ディスクブロックは全て２つのパリティ集合に属する。一方は、全てのデータディスクにわたる通常のＲＡＩＤ－４スタイルでのＸＯＲ演算によって計算され、他方は、斜めに隣り合う幾つかのディスクブロックの集合から計算される。対角パリティ集合は、１つを除く全てのデータディスクからブロックを含む。ｎ個のデータディスクに対し、１つのストライプ中には、ｎ－１行のブロックが存在する。各ブロックは、一本の対角上にあり、ｎ－１ブロック分の長さをそれぞれ有するｎ本の対角が存在する。なお、ＥＶＥＮＯＤＤ技術は、ｎが素数でないと動作しない点に注意して欲しい。ＥＶＥＮＯＤＤ技術は、１９９５年２月に「EVENODD: An Efficient Scheme for Tolerating Double Disk Failures in RAID Architectures」と題するIEEE Transactions on Computers, Vol. 44, No.2の記事にBlaum他により開示されている。ＥＶＥＮＯＤＤの変形は、１９９６年１１月２６日に、「METHOD AND MEANS FOR ENCODING AND REBUILDING THE DATA CONTENTS OF UP TO TWO UNAVAILABLE DASDS IN A DASD ARRAY USING SIMPLE NON-RECURSIVE DIAGONAL AND ROW PARITY」と題する米国特許第５，５７９，４７５号に開示されている。上記の記事、及び特許は、参照により、本明細書の中で完全に説明されたものとして本明細書に援用される。

【００１３】

ＥＶＥＮＯＤＤ技術は、ｐを素数として、総数ｐ＋２個のディスクを使用する。うち、ｐ個のディスクはデータを格納し、残り２つのディスクはパリティ情報を格納する。一方のパリティディスクは、行パリティブロックを格納する。行パリティは、各データディスク上の同じ位置にあるデータブロック全てのＸＯＲとして計算される。他方のパリティディスクは、対角パリティブロックを格納する。対角パリティは、複数のデータディスクにわたって対角パターンを成して配置されるｐ－１個のデータブロックから構成される。ブロックは、ｐ－１行のストライプにグループ化される。これは、行パリティ集合へのデータブロックの割り当てには影響を及ぼさない。ただし、対角は、ある対角内のブロックが全て、同じストライプに属するようなパターンを成すように構成される。これは、対角が複数のディスクにわたって規定されるときに、大抵の対角は、ストライプ中で「循環する」ことを意味する。

【００１４】

具体的には、ｎ×（ｎ－１）個のデータブロックからなるアレイの場合、アレイの端部で対角が「循環」されるとすれば、ｎ－１の長さをそれぞれ有する対角が、ちょうどｎ個存在する。ＥＶＥＮＯＤＤパリティ構成の復元において重要な点は、各対角パリティ集合が、データディスクの１つからは、何も情報を有しないことにある。ただし、パリティを格納するブロックを有する対角の他に、さらにもう１つ対角が存在する。すなわち、ＥＶＥＮＯＤＤパリティ構成では、ある１つの対角パリティ集合については、独立したパリティブロックを持たない。この余分な「抜けている」パリティブロックを許容するために、ＥＶＥＮＯＤＤ構成では、ある特定の対角に対するパリティ計算の結果は、残りの対角の

それぞれに関するパリティブロックにＸＯＲ演算される。

【００１５】

図１は、従来のＥＶＥＮＯＤＤパリティ構成に従って構成された従来のディスクアレイ１００を示す略ブロック図である。各データブロックＤ$_{ab}$は、パリティ集合ａ及びｂに属する。ただし、各パリティ集合のパリティブロックは、Ｐ$_{a}$と表記する。また、１つの特別な対角（Ｘ）については、対応するパリティブロックが格納されない。ここに、ＥＶＥＮＯＤＤの特徴が現れる。２つの故障からの復元を可能にするために、各データディスクは、少なくとも１つの対角パリティ集合に属してはならない。ｎ×（ｎ－１）個のデータブロックからなる矩形アレイを採用する場合、対角パリティ集合は、ｎ－１個のデータブロック要素を有する。また、上記のように、そのような構成は、全ての対角についてパリティブロックを格納する位置を持つのではない。したがって、余分な（抜けている）対角パリティブロック（Ｘ）のパリティは、その対角パリティを残りの対角パリティブロックにそれぞれＸＯＲ演算することによって記録される。具体的には、抜けているパリティ集合のパリティは、対角パリティブロックＰ４～Ｐ７のそれぞれにＸＯＲ演算され、それらのブロックがＰ４Ｘ～Ｐ７Ｘと表記される。

【００１６】

２つのデータディスクの故障を復旧させる場合、まず、全てのパリティブロックのＸＯＲを求めることによって、対角パリティを持たない対角のパリティを計算する。例えば、全ての行パリティの和は、全てのデータブロックの和に等しい。全ての対角パリティの和は、全てのデータブロックの和から、抜けている対角パリティブロックの和を差し引いたものに等しい。したがって、全てのパリティブロックのＸＯＲは、全てのブロックの和（行パリティ和）から、抜けている対角を除く全てのブロックの和を差し引いたものに等しい。ここで、抜けている対角とは、要するに、抜けている対角のパリティである。実際には、各対角パリティブロックについて１つ、抜けている対角パリティのｎ－１個のコピーが、その結果に加算される。ｎは２よりも大きな素数であるから、ｎ－１は偶数であり、したがって、あるブロックをそれ自体と偶数回だけＸＯＲ演算した結果は、ゼロブロックになる。したがって、抜けているパリティを各対角パリティブロックに加えた後の対角パリティブロックの和は、その追加の対角パリティ以外の対角パリティブロックの和に等しい。

【００１７】

次に、対角パリティブロックのそれぞれから、抜けている対角パリティを差し引く。２つのデータディスクが故障した後、１ブロックだけ抜けているブロックを含む対角パリティ集合が少なくとも２つ存在する。それらのパリティ集合のそれぞれにおいて抜けているブロックは、それらの対角パリティ集合のうちの一方が、たとえパリティブロックを持たない対角であったとしても、復元することができる。それらのブロックが復元されると、２つの行パリティ集合の１つを除く全ての要素が、利用可能になる。その結果、それらの行の抜けている要素の復元が可能になる。この復元は、他の対角に対しても行われ、それらの対角上の最後の抜けている幾つかのブロックを復元するための十分な情報を提供する。行パリティと対角パリティを交互に使用したこの復元このパターンは、抜けているブロックの復元が完了するまで、繰り返される。

【００１８】

ｎは素数であるから、復元中、全ての対角に遭遇する前に、すなわち、抜けているデータブロックの復元が全て完了する前に、循環が形成されることはない。もしｎが素数でなければ、これは成り立たないこともある。パリティディスクが両方とも失われた場合、データからのパリティの単純な復元を実施することができる。データディスクと対角パリティディスクが失われた場合、行パリティを使用して単純なＲＡＩＤ－４スタイルでデータディスクの復元が実施された後、続いて、対角パリティディスクの復元が実施される。データディスクと行パリティディスクが失われた場合、１つの対角パリティを計算できる場合がある。対角は全て同じパリティを有しているので、続いて、各対角上の抜けているブロックを計算することが出来る。

【００１９】
各データブロックは、いずれかの対角パリティ集合の要素であるから、２つのデータディスクが失われた場合（二重故障）、１要素しか失われてない対角パリティ集合が２つ存在する。各ディスクは、そのディスク上に表現されない対角パリティ集合を１つ有する。したがって、二重故障の場合、復元可能な対角パリティ集合は２つ存在する。ＥＶＥＮＯＤＤも、両方のパリティディスクの故障からの復旧、及び１つのデータディスクと１つのパリティディスクの任意の組み合わせからの復旧が可能である。この技術も、任意の単一の故障からの復旧が可能である。

【００２０】
ＥＶＥＮＯＤＤ技術は、パリティ情報の量の点では最適であるが、エンコードとデコードの両方に要する計算量の点では、最適とは言えない。その理由は、抜けている対角パリティを対角パリティブロックのそれぞれに加算するために、余分な計算が必要になるからである。つまり、１ストライプ中のｐ－１個のブロックは、ｐ個の対角から生成されるｐ個のパリティブロックを保持するのに十分ではない。これを克服するために、ＥＶＥＮＯＤＤ技術では、いずれか１つの対角のパリティを残りの全ての対角のパリティブロックにＸＯＲ演算しなければならず、その結果、計算オーバヘッドが増大する。

【００２１】
一般に、直接的なパリティブロックを持たない対角上のデータブロックに対する小さな書き込み処理については常に、全ての対角パリティブロックを更新しなければならない。大きな書き込み処理の場合は、余分な計算が更に必要となる。本明細書では、「大きな書き込み」処理とは、１ストライプ中の全てのブロックの書き換えが必要な処理を言い、「小さな書き込み」処理とは、少なくとも１つのデータブロック、及びその関連パリティの変更が必要な処理を言う。

【００２２】
ストレージ環境によっては、データをテープその他の長期保管システムにバックアップするまでの短期記憶装置として、相当な数の、例えばニア・ライン記憶システムのような低品質ディスクドライブを使用することがよくある。しかしながら、アレイ内のディスク数が増えるにつれて、多重故障が発生する確率も増大する。この確率は、安価な記憶装置のＭＴＴＦ（故障までの平均時間）が短くなるほど悪化する。つまり、ストレージシステムは三重故障、すなわち、ストレージアレイ内の３つの装置の同時故障を被る可能性がある。また、ＳＡＳ（Serial Attached SCSI）、ファイバ・チャネル（ＦＣ）などのような膨大な数のストレージプロトコルが存在する結果として、ディスクシェルフのアーキテクチャは益々複雑になっており、それに伴い、ディスクシェルフが受ける故障の数も増加しており、その結果、故障したディスクシェルフに接続された各ディスクへのアクセスは失われることがある。

【００２３】
三重故障を訂正する一つの技術として、ＳＴＡＲ技術と呼ばれるＥＶＥＮＯＤＤ技術の拡張がある。この技術は、２００５年８月、Cheng Wang著の「Efficient and Effective Schemes for Streaming Media Deliveryに記載されており、この文献は参照により本明細書に援用される。

【００２４】
ＳＴＡＲ技術は、ｐを素数としてｐ＋３個のディスクからなるアレイを使用し、ＥＶＥＮＯＤＤ符号化方式を使用して、対角パリティ集合と反対角パリティ集合の両方をエンコードし、ｐ個のデータディスク、１つの行パリティディスク、１つの対角パリティディスク、及び１つの反対角パリティディスクを作成する。反対角パリティ集合は、対角パリティ集合と同様に計算されるが、対角パリティ集合では傾き１を使用するのに対し、反対角パリティ集合は、傾き－１を使用して計算される点が異なる。

【００２５】
ＳＴＡＲ技術の顕著な欠点は、ＥＶＥＮＯＤＤ符号化技術を使用する点にある。実際、ＳＴＡＲ技術は、ＥＶＥＮＯＤＤを使用してアレイを符号化し、行パリティ、及び対角パ

リティを生成した後、ＥＶＥＮＯＤＤ符号化を使用して第２のパリティ計算を実施し、反対角パリティを生成する。ＥＶＥＮＯＤＤ技術の欠点は、データブロックに書き込みを行うときに、対角パリティ集合と反対角パリティ集合の両方を更新しなければならないことによって悪化する。

【００２６】

ＳＴＡＲ技術が有する更に別の顕著な欠点は、大きな素数に関わるアレイ、及び／又は最悪故障条件に関わるアレイの場合、復元処理の複雑度が非常に高くなることである。この非常に高い複雑度は、復元処理を実施するために必要となる計算回数を増加させる。

【発明の開示】

【課題を解決するための手段】

【００２７】

[発明の概要]

本発明は、アレイ内の最大３つまでの記憶装置の同時故障からの効率的復旧が可能であるように構成されたストレージアレイに対し、パリティ計算のオーバヘッドを低減する三重パリティ（ＴＰ）技術を含む。このＴＰ技術は、好ましくは、複数のデータディスク、１つの行パリティディスク、１つの対角パリティディスク、及び１つの反対角パリティディスクを含むｎ個のディスクのような記憶装置を含むアレイにおいて実施される。ただし、ｐを素数として、数ｎ＝ｐ＋２である。ディスクは、複数のブロックに分割され、ブロックは複数のストライプに編成される。ただし、各ストライプは、ｎ－３（又は、ｐ－１）行を含む。

【００２８】

１つのストライプを形成するように選択された複数行のブロックは通常、各ディスク上で連続しているが、それは本発明の必須条件ではない。対角パリティディスクは、アレイの複数の対角パリティ集合（「対角」）に沿って計算されたパリティ情報を格納している。１つのストライプを形成するブロックは、ｎ－２本の対角に編成され、各対角は、データディスク、及び行パリティディスクからｎ－３個のブロックを含み、１つを除く全ての対角は、自分のパリティを対角パリティディスク上の１つのブロックに格納する。同様に、反対角パリティディスクは、アレイの複数の反対角パリティ集合（「反対角」）に沿って計算されたパリティ情報を格納している。特に、反対角は、対角に対して直交する傾きを有する。その結果、新規なＴＰ技術は、一様なストライプ深さ、及び３台のディスク分に相当するパリティ情報の量を提供する。この量は、任意の３台のディスク故障からの復旧に必要となる最小量である。

【００２９】

本発明によれば、ＴＰ技術は、アレイ内のデータディスクの各行に沿った行パリティの計算を含み、以後、（反）対角パリティディスクに格納される対角パリティ、及び反対角パリティを計算するときには、行パリティブロックとデータブロックを区別しない。すなわち、（反）対角パリティは、全データディスク、及び行パリティディスクにわたって規定される幾つかの（反）対角に沿って計算される。また、１つを除く全ての（反）対角について、（反）対角パリティディスクにパリティが格納される。換言すれば、（反）対角パリティディスクは、１つのストライプの１つを除く各（反）対角のパリティを格納する。ただし、（反）対角パリティのうちの１つについては、パリティが計算も格納もされないが、本発明の技術によれば、アレイ内の任意の３つの同時ディスク故障から復旧するために十分なだけのパリティ情報が提供される。

【００３０】

アレイ内の１以上の記憶装置故障に応答し、本発明は例えば、適切な復元技術を決定するために、ストレージ・オペレーティング・システムのディスクストレージ層（ＲＡＩＤシステム）においてマルチステップ・ステートマシンを実施する。発明の目的のために、データディスクと行パリティディスクは、まとめて「ＲＡＩＤ４ディスク」と呼ばれる。一台のディスクが故障した場合、故障したディスクから失われたブロックは、従来の行パリティ技術を使用して復元される。（反）対角パリティディスクがｍ故障した場合、デー

タディスクと行パリティディスクを使用して、適当な（反）対角パリティが再計算される。二重ＲＡＩＤ４ディスク故障が発生した場合、対角パリティと反対角パリティのいずれかを使用し、行対角（Ｒ－Ｄ）パリティ復元技術にしたがって、データは復元される。

【００３１】

三重ディスク故障が発生した場合、どのディスクが故障したかに関する判定がなされる。一台のＲＡＩＤ４ディスクに加え、対角パリティディスク、及び反対角パリティディスクが故障した場合、失われた対角パリティ、及び反対角パリティを再計算する前に、従来の行パリティディスクを使用して、故障したＲＡＩＤ４ディスクが復元される。２台のＲＡＩＤ４ディスク、及び１つの（反）対角パリティディスクが故障した場合、失われた対角パリティ、又は反対角パリティを計算する前に、Ｒ－Ｄパリティ技術を使用して、失われたＲＡＩＤ４ディスクが復元される。

【００３２】

ただし、３台のＲＡＩＤ４ディスクが故障した場合、「抜けている」対角パリティ、及び反対角パリティ、すなわち、以前に格納されなかった対角パリティ、及び反対角パリティをまず計算することによって、三重パリティ復元技術が実施される。次にＲＡＩＤシステムは、多数の十字を生成することにより、故障したディスクの中間（すなわち、真ん中）にあるディスクに沿って、合計ｐ個の四要素和を計算する。その後、この四要素和は、中間ディスク上の合計ｐ－１個の二つ一組の要素和にまで減らされる。生成された二つ一組の要素和は、解放可能な一組の一次方程式を形成する。例えば、方程式の系を解くことによって、中間ディスク上のデータがまず復元される。中間ディスク上の第１のブロックが求まると、中間ディスク上のブロックが全て復元されるまで、その解は、他の式にも代入される。中間ディスクが復元された後、システムは、Ｒ－Ｄパリティ技術を使用して、残りの２つのディスクも復元する。

【００３３】

有利なことに、本発明の技術によれば、故障なし条件の下でアレイに記憶されるパリティを計算する際の計算負荷を最小限に抑えることができる。また、本発明の技術は、パリティ計算のオーバヘッドを低減し、ＳＴＡＲのような従来の手法に比べて、所与の数のデータディスクに対する計算量も少なくて済む。さらに、本発明は、行パリティブロックが全て同じディスク上に格納される集中パリティ方式を使用して実施することができ、既存のパリティ情報を再フォーマットしたり、再計算したりすることなく、データディスクを少しづつアレイに追加することができる。アレイへのデータディスクの追加に関する唯一の制限は、アレイ内で使用可能なディスクの最大数を、前述のようにして（前もって）決定しなければならないことだけである。この制限は、（反）対角の使用に起因するものであり、（反）対角の長さは、ストライプ深さによって決まる。

【００３４】

存在する実際のディスクの数と、アレイ内のディスクの最大数との差は、例えば、全てゼロ値のデータを有する「仮想」ディスクを使用して埋められる。

【００３５】

本発明の上記の利点、及びその他の利点は、添付の図面と併せて下記の説明を読めれば、より深く理解できるであろう。図中、同じ参照符号は、同一の要素、又は機能的に同じ要素を意味している。

【発明を実施するための最良の形態】

【００３６】

[例示的実施形態の詳細な説明]

Ａ．ストレージシステム環境

図２は、本発明とともに有利に使用されるストレージシステム２２０を含む環境２００を示す略ブロック図である。本明細書に記載する本発明の技術は、ストレージシステム２００として実施され、又はストレージシステム２００を含む形で実施されるスタンドアロンのコンピュータ、又はその一部を含む、いかなるタイプのコンピュータにも適用することができ、特殊な用途のコンピュータ（例えばファイルサーバ、又はファイラ）にも、汎

用コンピュータにも適用することができる。また、本明細書の教示は、限定はしないが、ネットワーク・アタッチド・ストレージ環境、ストレージ・エリア・ネットワーク、あるいは、クライアント若しくはホストコンピュータに直接取り付けられたディスクアセンブリを含む、種々のストレージシステムアーキテクチャに適合する。したがって、「ストレージシステム」という用語は、記憶機能を実施するように構成され、他の装置、又はシステムに関連する何らかのサブシステムだけでなく、そうした構成も含むものとして広い意味で解釈しなければならない。

【００３７】

図示の実施形態において、ストレージシステム２２０は、システムバス２３２によって相互接続されたプロセッサ２２２、メモリ２２４、ネットワークアダプタ２２５、及びストレージアダプタ２２８を含む。メモリ２２４は、本発明に関連するソフトウェアプログラムコード、及びデータ構造を格納するために、プロセッサ、及びアダプタによってアドレス指定可能な複数の記憶場所を有する。そして、プロセッサ、及びアダプタは、そのソフトウェアコードを実行し、データ構造を操作するように構成された処理要素、及び／又はロジック回路を含む。ストレージオペレーティングシステム３００は、その一部が通常、メモリに常駐し、処理要素によって実行され、とりわけ、ストレージシステムによって実行される記憶機能を実施することによって、ストレージシステム２２０を機能的に編成する。当業者には明らかなように、本明細書に記載する本発明の技術に関連するプログラム命令の格納、及び実行には、種々のコンピュータ読取可能媒体を含む他のプロセッサや記憶手段を使用してもよい。

【００３８】

ネットワークアダプタ２２５は、ポイント・ツー・ポイントリンク、ワイド・エリア・ネットワーク、公共ネットワーク（インターネット）上で実施される仮想私設ネットワーク、あるいは、共有ローカルエリアネットワークを介して、ストレージシステム２２０を１以上のクライアント２１０に接続するように構成された複数のポートを有する。したがって、ネットワークアダプタ２２５は、ノードをネットワークに接続するために必要とされる機械的、電気的、及び信号回路を含む。たとえば、ネットワーク２０５は、イーサネット(R)ネットワーク、またはファイバチャネル（ＦＣ）ネットワークとして実施される場合がある。各クライアント２１０は、ＴＣＰ／ＩＰのような所定のプロトコルにしたがって、ネットワーク２０５を介して個々のデータフレームやデータパケットを交換することにより、ストレージシステム２２０と通信する。

【００３９】

ストレージアダプタ２２８は、ストレージシステム２２０上で実行されているストレージオペレーティングシステム３００と協働し、ユーザ（又は、クライアント）から要求された情報をアクセスする。情報は、ビデオテープ、光学、ＤＶＤ、磁気テープ、バブルメモリ、電気的ランダムアクセスメモリ、ＭＥＭＳのような書換型記憶装置媒体の任意のタイプのアタッチド・アレイ、及びデータやパリティ情報のような情報を記憶するように構成された任意の他の同様の媒体に格納される。ただし、本明細書に例示的に記載されるように、情報は、好ましくは、アレイ２４０のＨＤＤ、及び／又はＤＡＳＤのようなディスク２５０に格納される。ストレージアダプタは、従来の高性能ＦＣシリアルリンクトポロジのようなＩ／Ｏ相互接続構成を介してディスクを接続するための入出力（Ｉ／Ｏ）インタフェース回路を含む。

【００４０】

アレイ２４０への情報の格納は、ディスク空間の全体的論理構成を規定する、一群の物理記憶ディスク２５０を含む１以上の「ボリューム」として実施されることが好ましい。もちろん必須ではないが、各ボリュームは一般に、独自のファイルシステムに関連する。ボリューム／ファイルシステム内のディスクは通常、１以上のグループに編成され、各グループが、ＲＡＩＤ（Redundant Array of Independent Disks）として運用される。大半のＲＡＩＤ実施形態は、所与の数の物理ディスクにわたってデータ「ストライプ」を冗長書き込みし、そのストライピングされたデータに関する適当なパリティ情報を格納するこ

とによって、データ記憶の信頼性／完全性を向上させる。

【００４１】

本発明は、ディスクアレイ上の行パリティ、対角パリティ、及び反対角パリティを使用した三重故障パリティ訂正復旧を提供する「三重」パリティ（ＴＰ）技術を含む。本発明の技術は、好ましくは、ストレージオペレーティングシステム３０のディスクドライバ層（図３に符号３４０で示す）によって実施され、ストレージシステム内のディスクのような複数の記憶装置にわたって規定される複数のストライプにパリティを形成する方法、及びシステムを提供する。アレイ内の３台のディスクはパリティ専用に使用され、残りのディスクがデータを保持する。データディスク上のデータは、「クリアな状態の」で格納される。「クリアな状態」とは、格納の際にそれ以上エンコードされないことを意味する。任意の１台、２台、又は３台の同時故障の後でも、データを失うことなく、アレイの内容を完全に復元することができる。本発明は、従来の方法に比べて、必要とされるパリティ情報の計算量を減らすことができるだけでなく、３台のディスク故障からの復旧に必要とされる計算量も低減する。さらに、本発明は、均一なストライプ深さ（各ディスクが、１ストライプあたり同数のブロックを有する）を提供し、また、任意の３台のディスク故障からの復旧を可能にするために必要となる最少量である、ディスク３台分の量のパリティ情報を提供する。

【００４２】

概して言えば、本発明は、ｎ台の記憶装置を含む。ただし、ｐは素数であり、ｎ＝ｐ＋２である。記憶装置は、複数の同じサイズのブロックに分割される。全ての記憶装置にわたって、各記憶装置の中からｎ－３個のブロックが自由に選択され、グループ化され、ストライプが形成される。ストライプ中で、１台の記憶装置は、他の記憶装置からブロックを入力として選択することによって形成されたパリティを保持するように指定される。後で詳しく説明する単純化された形成技術の結果として、この記憶装置は、「対角パリティ装置」と呼ばれ、対角パリティ装置に保持されるパリティは「対角パリティ」と呼ばれる。同様に、各ストライプにおいて、１台の記憶装置は、「反対角パリティ装置」としてパリティを保持するように指定され、反対角パリティ装置に保持されるパリティは「反対角パリティ」と呼ばれる。

例えば、対角パリティと反対角パリティはアレイ上で直交する傾きを有し、例えば、対角パリティが傾き１を有するのに対し、反対角パリティは傾き－１を有する。各ストライプにおいて、そのストライプ中の（反）対角パリティ装置以外の記憶装置からそれぞれ、１つのブロックが選択される。このブロックの集合は「行」と呼ばれる。行内の１つのブロックは、その行のパリティ（行パリティ）を保持するために選択され、残りのブロックはデータを保持する。行の形成は、ストライプ中の対角パリティ装置、又は反対角パリティ装置上にないブロックが全て、ちょうど１行に割り当てられるまで継続される。全部でｎ－３行になる。

【００４３】

各ストライプ中で、そのストライプ中の対角パリティ装置でも反対角パリティ装置でもない記憶装置のうち、一つを除く全ての記憶装置から、１つのブロックが選択される。ただし、その際、選択されたブロックのうちの２つが同じ行に属することがないようにするという条件が課される。これは、「対角パリティ集合」、又は「対角」と呼ばれる。対角の形成は、例えば、データディスク、行パリティディスク、及び対角パリティディスクに０からｎ－２の番号を付け、行に０からｎ－３まで番号を付けた後、装置ｉの行ｊにあるブロックを対角（ｉ＋ｊ）ｍｏｄ（ｎ－２）に割り当てることによってなされる。対角の形成は、ストライプ内の対角装置、及び反対角装置上にないブロックが全て、対角に割り当てられるまで継続される。ただし、その際、同じ装置から選択されたブロックを全く有しない対角が２つ存在しないようにするという条件が更に課される。ｎ－２本の対角が存在し、そのｎ－２本の対角から、ｎ－３本の対角が選択される。これらの対角上のブロックは、データを保持しているかパリティを保持しているかに関わらず、対角パリティブロ

ックを形成するために結合されるｎ－３個の対角パリティブロックは、ストライプ中の対角パリティを保持する装置上にある、ストライプ中のｎ－３個のブロックに任意の順序で格納される。同様の技術は、パリティやブロックを「反対角パリティ集合」、すなわち「反対角」に割り当てるときにも使用される。反対角の形成は、対角パリティ装置上にも反対角パリティ装置上にもないブロックが全て、反対角に割り当てられるまで継続される。ただし、その際、同じ装置から選択されたブロックを全く有しない反対角が２つ存在しないようにするという条件が更に課される。反対角の形成は、例えば、データデバイス、行パリティデバイス、反対角パリティデバイスに０からｎ－２まで番号を付け、行に０からｎ－３の番号を付けた後、デバイスｉの行ｊにあるブロックを対角（ｎ－３－ｉ＋ｊ）ｍｏｄ（ｎ－２）に割り当てることによってなされる。

【００４４】

本発明は、各ディスク上の同じ位置にあるブロックを含む行を選択し、ｎ－３行の連続したグループを選択して複数のストライプを形成し、さらにストライプ内のブロックを選択し、各（反）対角上のブロックにより循環（反）対角パターンが形成されるようにすることで、単純に実施されることがある。さらに、本発明は、ストライプ中の全ての行パリティブロックを同じ装置に記憶することによって実施されることがある。好ましい実施形態として、本発明は、ストライプごとに、行パリティ装置、（反）対角パリティ装置、及びデータ装置のような装置の使用を同様に維持することによって実施されることがある。あるいは、本発明の他の好ましい実施形態では、行パリティ装置、（反）対角パリティ装置、及びデータ装置のような装置がストライプごとに異なるように、装置の使用を循環、その他の方法で異ならせる場合がある。

【００４５】

パリティブロックを形成する際、パリティは一般に、データブロックの排他的論理和（ＸＯＲ）として計算される。ＸＯＲ演算は一般に、各入力ブロック中の同じ１ビットフィールドに対して実施され、対応する１ビットの出力を生成する。上記のように、ＸＯＲ演算は、２つの１ビットフィールドにおける２の補数による加算、又は減算に等しい。また、冗長パリティ情報は、全ての入力における同じサイズの多数ビットフィールド（例えば８、１６、３２、６４、１２８ビットなど）の和として計算される場合もある。例えば、パリティに相当するものは、３２ビットフィールドに対して２の補数加算を使用してデータを加算することによって計算され、それぞれ３２ビットの冗長情報を生成する場合がある。あるブロックをそれ自体とＸＯＲ演算したものはゼロになることから、これは、同じ入力を２回、あるブロックに対してＸＯＲ演算すれば、そのブロックの元の内容が得られるという事が信頼できないと想定される場合だけである。

【００４６】

当業者には明らかなように、ブロック（パリティ計算のための）は、ファイルブロック、データブロック、ディスクセクタ、又は何らかの他の便利なサイズの単位に対応する場合もあれば、対応しない場合もある。パリティ計算に使用されるブロックサイズが、システム内で使用される他の何らかブロックサイズと何らかの関係を有している必要はない。しかしながら、１以上の整数個のパリティブロックは、１以上の整数個のディスクセクタとして規定される単位にぴったりと収まるものであることが期待される。多くの場合、幾つかのブロックが、ファイルシステム、又は幾つかのデータベースブロックに対応し、通常は、４ｋ（４０９６）バイト、又は２バイトのそれより大きな次数の乗数（例えば、８ｋ、１６ｋ、３２ｋ、６４ｋ、１２８ｋ、２５６ｋ）のサイズを有する。

【００４７】

本明細書に記載するシステムは、好ましくは、フルストライプ書き込み処理を実施する。具体的には、一般に４ｋバイト、又は８ｋバイトである個々のファイルブロックは、パリティ計算のときにしか使用されないより小さな複数のブロックに分割され、全ストライプの例えば、４ｋバイトサイズのブロックが、ディスクアレイに書き込まれる。全ストライプをディスクに書き込むとき、パリティ計算は全てメモリ上で実施され、その後、その結果がディスクに書き込まれる。したがって、ディスク上でのパリティの計算、及び更新

に関する負担が軽減される。

【００４８】

Ｂ．ストレージオペレーティングシステム

ディスクに対するアクセスを容易にするために、ストレージオペレーティングシステム３００は、仮想化モジュールと協働するｗｒｉｔｅ－ａｎｙｓｈｅｒｅファイルシステムを実施し、ディスクによって提供される記憶空間を「仮想化」する。ファイルシステムは、情報を名前付きディレクトリ、及びファイルオブジェクト（以後、「ディレクトリ」、及び「ファイル」）の階層構造としてディスク上に論理編成する。「ディスク上」の各ファイルは、データのような情報を格納するように構成されたディスクブロックの集合として実施される一方、ディレクトリは、特殊フォーマットのファイルとして実施され、その中に、名前や、他のファイル、及びディレクトリへのリンクが格納される。仮想化システムによれば、ファイルシステムは、情報を名前付きｖｄｉｓｋの階層構造としてディスク上にさらに論理編成することが可能となり、それによって、ＮＡＳシステムとＳＡＮシステムの統合アプローチ提供し、ファイルやディレクトリに対するアクセスにはファイルベースのアクセス（ＮＡＳ）を可能にする一方、ファイルベースのストレージプラットフォーム上のｖｄｉｓｋに対するアクセスには、ブロックベースのアクセス（ＳＡＮ）を可能にする。

【００４９】

例示的実施形態として、ストレージオペレーティングシステムは、カリフォルニア州サニーベイルにあるネットワーク・アプライアンス・インコーポレイテッドから販売されているＮｅｔＡｐｐ　Ｄａｔａ　ＯＮＴＡＰオペレーティングシステムであることが好ましい。このオペレーティングシステムは、Ｗｒｉｔｅ　Ａｎｙｗｈｅｒｅ　Ｆｉｌｅ　Ｌａｙｏｕｔ（ＷＡＦＬ）ファイルシステムを実施する。ただし、当然ながら、ｗｒｉｔｅ－ｉｎ－ｐｌａｃｅファイルシステムのような任意の他のストレージオペレーティングシステムを、本明細書に記載する本発明の原理にしたがって使用されるように拡張してもよい。したがって、「ＯＮＴＡＰ」という用語を使用した場合であっても、この用語は、本発明の教示に適合させることが可能な任意のストレージオペレーティングシステムを指すものとして広い意味で捉えなければならない。

【００５０】

本明細書では、「ストレージオペレーティングシステム」とは、コンピュータ上で実行可能な、データアクセスを管理するためのコンピュータ実行可能コードを言い、ストレージシステムの場合、マイクロカーネルとして実施されるＤａｔａ　ＯＮＴＡＰストレージオペレーティングシステムのように、データアクセスセマンティックを実施する場合がある。また、ストレージオペレーティングシステムは、ＵＮＩＸやＷｉｎｄｏｗｓ　ＮＴのような汎用コンピュータ上で動作するアプリケーションプログラムとして実施してもよいし、あるいは、本明細書に記載するようなストレージアプリケーションのために構成された構成変更機能を備えた汎用オペレーティングシステムとして実施してもよい。

【００５１】

また、当業者には明らかなように、本明細書に記載する本発明の技術は、いかなるタイプの特殊目的のコンピュータ（例えば、ストレージを提供するアプライアンス）にも、汎用コンピュータにも適用することができ、ストレージシステムとして実施され、又はストレージシステムを含む形で実施されるスタンドアロンのコンピュータ、又はその一部にも適用することができる。さらに、本発明の教示は、種々のストレージシステムアーキテクチャに適合させることができ、限定はしないが例えば、ネットワーク・アタッチド・ストレージ環境、ストレージ・エリア・ネットワーク、及びクライアントやホストコンピュータに直接取り付けられるディスクアセンブリにも適合させることができる。したがって、「ストレージシステム」という用語は、ストレージ機能を実施するように構成され、他の装置、又はシステムに関連する任意のサブシステムだけでなく、それらの構成も含むものとして広い意味で解釈しなければならない。

【００５２】

図３は、本発明とともに有利に使用されるストレージオペレーティングシステム３００を示す略ブロック図である。ストレージオペレーティングシステムは、統合ネットワークプロトコルスタック、すなわち、より一般的には、マルチプロトコルストレージシステム上に格納された情報をクライアントがブロックアクセスプロトコルやファイルアクセスプロトコルを使用してアクセスするためのデータパスを提供するマルチプロトコルエンジンを形成するように編成された一連のソフトウェア層を含む。プロトコルスタックは、ＩＰ層３１２、並びに、その支持搬送機構であるＴＣＰ層３１４、及びユーザデータグラムプロトコル（ＵＤＰ）層３１６といったネットワークプロトコル層へのインタフェースを提供するネットワークドライバ（例えば、ギガビットイーサネットドライバ）のメディアアクセス層３１０を含む。ファイルシステムプロトコル層は、マルチプロトコルファイルアクセスを提供し、その目的のために、ＤＡＦＳプロトコル３１８、ＮＦＳプロトコル３２０、ＣＩＦＳプロトコル３２２、及びハイパーテキストトランスファプロトコル（ＨＴＴＰ）プロトコル３２４をサポートする。ＶＩ層３２６は、ＶＩアーキテクチャを実施し、ＤＡＦＳプロトコル３１８に必要とされるＲＤＭＡのようなダイレクトアクセストランスポート（ＤＡＴ）機能を提供する。

【００５３】

ｉＳＣＳＩドライバ層３２８は、ＴＣＰ／ＩＰネットワークプロトコル層を介したブロックプロトコルアクセスを可能にする一方、ＦＣドライバ層３３０は、ネットワークアダプタと協働し、ストレージシステムに対するブロックアクセス要求、及び応答の送受信を行う。ＦＣドライバ、及びｉＳＣＳＩドライバは、ＬＵＮ（ｖｄｉｓｋ）に対するＦＣ固有の、及びｉＳＣＳＩ固有のアクセス制御を提供し、したがって、マルチプロトコルストレージシステム上の単一のｖｄｉｓｋをアクセスするときに、ｉＳＣＳＩとＦＣＰのどちらか一方、あるいは両方へのｖｄｉｓｋのエキスポートを管理する。さらに、ストレージオペレーティングシステムは、ＲＡＩＤプロトコルやディスクドライバ層３５０のようなディスクストレージプロトコルを実施するＲＡＩＤシステムのようなディスクストレージ層３４０を含み、ディスクドライバ層３５０は、例えばＳＣＳＩプロトコルのようなディスクアクセスプロトコルを実施する。

【００５４】

本発明の例示的実施形態として、ディスクストレージ層（例えばＲＡＩＤシステム３４０）は、新規なＴＰ技術を実施する。例えば、書き込み処理の際に、ＲＡＩＤシステム３４０は、データを以下に説明する符号化技術に従ってデータをエンコードし、記憶装置の１以上の故障の検出に応答して、後で詳しく説明される新規な復元技術を実施する。なお、代替実施形態では、この新規なＴＰ技術は、ＲＡＩＤシステム３４０以外のストレージオペレーティングシステムのモジュールによって実施される場合もある。したがって、新規なＴＰ技術を実施するＲＡＩＤシステム３４０の説明は、単なる例として捉えなければならない。

【００５５】

ディスクソフトウェア層を統合ネットワークプロトコルスタック層に橋渡しするのは、仮想化システム３５５である。仮想化システム３５５はファイルシステム３６５によって実施され、ファイルシステム３６５は、例えばｖｄｉｓｋモジュール３７０、及びＳＣＳＩターゲットモジュール３６０として実施される仮想化モジュールと対話する。なお、ｖｄｉｓｋモジュール３７０、ファイルシステム３６５、及びＳＣＳＩターゲットモジュール３６０は、ソフトウェアで実施しても、ハードウェアで実施しても、ファームウェアで実施しても、それらの組み合わせにより実施してもよい。ｖｄｉｓｋモジュール３７０は、ファイルシステム３６５と対話し、システム管理者がマルチプロトコルストレージシステム２２０に対して発行したコマンドに応答して、管理者インタフェースを使用したアクセスを可能にする。実際、ｖｄｉｓｋモジュール３７０は、とりわけ、システム管理者がユーザインタフェースを介して発行したｖｄｉｓｋ（ＬＵＮ）コマンドの複雑な組み合わせを実施することにより、ＳＡＮデプロイメントを管理する。こうしたｖｄｉｓｋコマンドは、ｖｄｉｓｋを実施するためのファイルシステム３６５やＳＣＳＩターゲットモジュ

ール３６０と対話する原始的なファイルシステムオペレーション（「プリミティブ」）に変換される。

【００５６】

次に、ＳＣＳＩターゲットモジュール３６０は、ＬＵＮを特殊なｖｄｉｓｋファイルタイプに変換するマッピング手順を提供することにより、ディスク、又はＬＵＮのエミュレーションを開始する。ＳＣＳＩターゲットモジュールは、例えば、ＦＣドライバ、ｉＳＣＳＩドライバ３３０、３２８と、ファイルシステム３６５との間に配置され、ＳＡＮブロック（ＬＵＮ）空間と、ファイルシステム空間（ＬＵＮがｖｄｉｓｋとして表現される）との間に、仮想化システム３５５の変換層を提供する。ファイルシステム３６５の上にＳＡＮ仮想化を「配置」することにより、マルチプロトコルストレージシステムは、従来のシステムによって行われるアプローチの逆を行うことができ、それによって、実質的に全てのアクセスプロトコルに対して単一の統一されたストレージプラットフォームを提供することができる。

【００５７】

ファイルシステム３６５は、例えば、メッセージベースのシステムである。したがって、ＳＣＳＩターゲットモジュール３６０は、ＳＣＳＩ要求を、ファイルシステムに対する操作を表わすメッセージに変換する。例えば、ＳＣＳＩターゲットモジュールによって生成されるメッセージは、操作のタイプ（たとえば、読み出し、書き込み）だけでなく、ファイルシステム上に表現されるｖｄｉｓｋオブジェクトのパス名（例えば、パス記述子）、及びファイル名（例えば、特殊なファイル名）を含む場合がある。ＳＣＳＩターゲットモジュール３６０は、そのメッセージを例えばファンクションコールとして、操作が実施される場所であるファイルシステム３６５に渡す。

【００５８】

ファイルシステム３６５は例えば、例えば４キロバイト（ＫＢ）ブロックを使用し、ｉｎｏｄｅを使用してファイルを表現するブロックベースのオンディスクフォーマット表現を備えたＷＡＦＬファイルシステムを実施する。ＷＡＦＬファイルシステムは、ファイルを使用して、自己のファイルシステムのレイアウトを表わすメタデータを格納する。そうしたメタデータには、とりわけ、ｉｎｏｄｅファイルがある。ディスクからｉｎｏｄｅを読み出すために、ファイルハンドル、すなわちｉｎｏｄｅ番号を含む識別子が使用される。オンディスクｉｎｏｄｅ、及びｉｎｏｄｅファイルを含む、ファイルシステムの構造の説明については、「METHOD FOR MAINTAINING CONSISTENT STATES OF A FILE SYSTEM AND FOR CREATING USER-ACCESSIBLE READ-ONLY COPIES OF A FILE SYSTEM」と題する米国特許第５，８１９，２９２号に、David Hitz他が記載している。

【００５９】

動作的に、クライアント２１０からの要求は、コンピュータネットワーク２０５を介してパケットとしてストレージシステム２２０へと転送され、そこで要求はネットワークアダプタ２２５によって受信される。ネットワークドライバは、そのパケットを処理し、必要であれば、それをネットワークプロトコル層やファイルシステム層に渡して更なる処理を施した後、それをｗｒｉｔｅ－ａｎｙｗｈｅｒｅファイルシステム３６５に転送する。ここで、要求されたデータが「コア内」になければ、すなわち、メモリ２２４上になければ、ファイルシステムは、要求されたデータをディスク２５０からロードする（読み出す）ための処理を生成する。情報がメモリ上になければ、ファイルシステム３６５は、ｉｎｏｄｅ番号を使用してｉｎｏｄｅファイル内を検索し、適当なエントリにアクセスし、論理ボリュームブロック番号（ｖｂｎ）を読み出す。次にファイルシステムは、その論理ｖｂｎを含むメッセージをＲＡＩＤシステム３４０に渡す。論理ｖｂｎは、ディスク識別子、及びディスクブロック番号（ｄｉｓｋ、ｄｂｎ）にマッピングされ、ディスクドライバシステム３５０の適当なドライバ（例えば、ＳＣＳＩ）に送られる。ディスクドライバは、指定されたディスク２５０からそのｄｂｎにアクセスし、要求されたデータブロック（複数の場合もあり）をメモリ上にロードし、ストレージシステムによって処理する。要求の処理が完了すると、ストレージシステム（及び、オペレーティングシステム）は、ネッ

トワーク２０５を介してクライアント２１０に返答を返す。

【００６０】

なお、ストレージシステムで受信されたクライアント要求に対し、データストレージアクセスを実施するために必要とされる、ストレージオペレーティングシステム層を通る上記のソフトウェア「パス」は、代替として、ハードウェアといて実施してもよい。すなわち、本発明の代替実施形態において、ストレージアクセス要求データパスは、フィールド・プログラマブル・ゲート・アレイ（ＦＰＧＡ）や特定用途向け集積回路（ＡＳＩＣ）の中に論理回路として実施される場合がある。この手のハードウェア実施形態によれば、クライアント２１０によって発行される要求に応答してストレージシステム２２０が提供するストレージサービスの性能を向上させることができる。また、本発明の更に他の実施形態において、アダプタ２２５、２２８の処理要素は、プロセッサ２２２から、パケット処理やストレージアクセス処理の負荷の一部、又は全部を取り除くように構成され、それによって、システムによって提供されるストレージサービスの性能を向上させる場合もある。本明細書に記載する種々の処理、アーキテクチャ、及び手順は、ハードウェアで実施しても、ファームウェアで実施しても、ソフトウェアで実施してもよいものと考えられる。

【００６１】

本明細書では、「ストレージオペレーティングシステム」は通常、ストレージシステムにおけるストレージ機能、例えば、データアクセスを管理するコンピュータ実行可能コードを意味し、場合によっては、ファイルシステムセマンティックを実施する場合もある。その意味で、ＯＮＴＡＰソフトウェアは、マイクロカーネルとして実施され、ＷＡＦＬファイルシステムセマンティックを実施し、データアクセスを管理するためのＷＡＦＬ層を含むストレージオペレーティングシステムの一例である。ストレージオペレーティングシステムは、ＵＮＩＸやＷｉｎｄｏｗｓ　ＮＴのような汎用オペレーティングシステム上で動作するアプリケーションとして実施してもよいし、あるいは、本明細書に記載するストレージアプリケーションに合わせて構成された構成変更機能を備えた汎用オペレーティングシステムとして実施してもよい。

【００６２】

さらに、当業者には明らかなように、本明細書に記載する本発明の教示は、いかなるタイプの特殊目的のコンピュータ（例えば、ファイルサーバ、ファイラ、又はストレージシステム）にも、汎用コンピュータにも適用することができ、ストレージシステム２２０として実施され、又はストレージシステム２２０を含む形で実施されるスタンドアロンのコンピュータ、又はその一部にも適用することができる。本発明とともに有利に使用されるストレージシステムの一例は、２００２年８月８日に出願されたBrian Pawlowski他による「MULTI-PROTOCOL STORAGE APPLIANCE THAT PROVIDES INTEGRATED SUPPORT FOR FILE AND BLOCK ACCESS PROTOCOLS」と題する米国特許出願第１０／２１５，９１７号に記載されている。また、本発明の教示は、種々のストレージシステムアーキテクチャに適合させることができ、限定はしないが例えば、ネットワーク・アタッチド・ストレージ環境、ストレージエリアネットワーク、及びクライアントやホストコンピュータに直接取り付けられたディスクアセンブリに適合させることができる。したがって、「ストレージシステム」という用語は、ストレージ機能を実施するように構成され、他の装置、又はシステムに関連する任意のサブシステムだけでなく、そうした構成も含むものとして広い意味で捉えなければならない。

【００６３】

Ｃ．三重パリティ符号化

本発明は、アレイ内の最大３つまでの記憶装置の同時故障からの効率的な復旧が可能となるように構成された、ストレージアレイに関するパリティ計算のオーバヘッドを低減する三重パリティ（ＴＰ）技術を含む。ＴＰ技術は、好ましくは、ｐを素数として、ｐ＝ｎ＋２個のディスクのような記憶装置を含むアレイにおいて実施され、例えば、複数のデータディスク、１つの行パリティディスク、１つの対角パリティディスク、及び１つの反対角パリティディスクからなるアレイにおいて実施される。ディスクは複数のブロックに分

割され、ブロックは複数のストライプに編成され、各ストライプが、ｎ－３（すなわちｐ－１）行を含む。１つのストライプを形成するように選択された複数行のブロックは通常、各ディスク上で連続しているが、これは、本発明にとって必須ではない。対角パリティディスクは、アレイの対角パリティ集合（「対角」）に沿って計算されたパリティ情報を記憶する。１ストライプ中のブロックは、ｎ－２本の対角に編成され、各対角は、データディスクと行パリティディスクからｎ－３個のブロックを含み、１つを除く全ての対角が、自己のパリティをブロックとして対角パリティディスク上に格納する。同様に、反対角パリティディスクは、アレイの反対角パリティ集合（「反対角」）に沿って計算されたパリティ情報を記憶する。特に、反対角は、対角に対して直交する傾きを有する。そのため、新規なＴＰ技術によれば、一様なストライプ深さ、及びディスク３台分に相当するパリティ情報の量が得られる。ディスク３台分は、任意の３つのディスク故障からの復旧に必要となる最小量である。

【００６４】

本発明によれば、ＴＰ技術は、アレイ内のデータディスクの各行における行パリティの計算を必要とし、その後、（反）対角パリティディスク上に格納される（反）対角パリティを計算するときに、行パリティブロックとデータブロックを区別しない。すなわち、（反）対角パリティは、全データディスク、及び行パリティディスクにわたって規定される幾つかの（反）対角に沿って計算される。また、１つを除く全ての（反）対角について、パリティが（反）対角パリティディスク上に格納される。換言すれば、（反）対角パリティディスクは、ストライプ中の１つを除く全ての（反）対角について、パリティブロックを有する。さらに、（反）対角のうちの１つについては、パリティが計算も格納もされないが、本発明の技術によれば、アレイ内の３台のディスクの同時故障からの復旧に十分なだけのパリティ情報が提供される。

【００６５】

図４は、本発明の一実施形態によるＴＰ技術を実施する手順４００のステップを示すフロー図である。手順４００はステップ４０６から始まり、ステップ４１０へ進み、そこでまず、素数ｐに等しい数のディスクのような記憶装置でアレイを構成する。ｐ個のディスクは、幾つかのデータディスク、及び１つの行パリティディスクに相当する。ステップ４１５において、さらにもう１つ別の対角パリティディスクをアレイに含め、アレイ全体がｐ＋１個のディスクから構成されるようにする。本明細書に記載されるように、対角パリティディスクは、アレイ内の全データディスク、及び行パリティディスクにわたって規定される幾つかの対角に沿って計算された対角パリティを記憶する。したがって、この時点で、アレイは、ｐ－１個のデータディスク、１つの行パリティディスク、及び１つの対角パリティディスクを含む。ステップ４２０において、反対角パリティをアレイに追加する。対角パリティディスクと同様に、反対角パリティディスクも、アレイ内の全データディスク、及び行パリティディスクにわたって規定される幾つかの反対角に沿って計算された反対角パリティを記憶する。特に、対角と反対角は互いに直交し、例えば、傾き±１を有する。したがってアレイは、ｐ－１個のデータディスク、１つの行パリティディスク、１つの対角パリティディスク、及び１つの反対角パリティディスクを含み、総数ｎ＝ｐ＋２個のディスクを含む。ステップ４２５において、これらのディスクは複数のブロックに分割され、ステップ４３０において、ブロックはストライプに編成され、各ストライプは、ｎ－３行のブロックを含む（ただし、ｎ＝ｐ＋２）。ステップ４３５では、ある行の各データブロックは、各データディスク上の同じ位置にあるデータブロックを全てＸＯＲ演算したものを保持する、その行の行パリティブロックにＸＯＲ演算される。

【００６６】

次に、ステップ４４０において、全データブロック、及び行パリティブロックが、対角に割り当てられる。ｐ個のディスクを含むアレイの場合、対角は、ｐ－１行のブロックのグループに収容される。ちょうどｐ本の対角が存在し、各対角は、ちょうどｐ－１個のデータブロック、及び／又は行パリティブロックをＸＯＲ演算したものを含む１つの対角パリティブロックを含む。ｐ個の対角集合がそれぞれ、ちょうど１つのディスクを含まない

ようにして、対角はアレイの端部で循環される。各対角は異なる１つのディスクを含んではならない。ｐ－１行の集合内で、あらゆるディスクブロックは、ｐ個の対角のうちのちょうど１つ上にある。表１は、０から４まで番号の付いた対角を有する、ｐ＝５の場合のアレイの一実施形態を示している。表中の番号は、各ブロックが属する対角パリティ集合を示している。

【００６７】

【表１】

01234
12340
23401
34012

【００６８】

なお、１行中の２つのブロックが同じ対角パリティ集合に属することがなく、任意の２つのディスク故障からアレイを復元できるという性質が変わらない限り、列の順序は変更してもよく、各列における要素の位置も変更してよい。一般性を失うことなく、ブロックを対角パリティ集合に割り当てる方法は、表１に実質的に従うものと仮定してよい。さらに、行内のブロックの要素は、順序変更してもよい。

【００６９】

上記のように、対角パリティ集合のパリティは、対角パリティディスク上に格納される。本発明によれば、ＴＰパリティ技術は、対角パリティディスク上に格納される対角パリティを計算するときに、行パリティブロックとデータブロックを区別しない。換言すれば、元のアレイの全てのディスクは等しく扱われ、ディスクの１つに格納された情報が、行パリティ集合中の他の全てのディスクのＸＯＲから復元できるような形で扱われる。したがって、対角パリティディスクは、アレイ内の全データディスク、及び行パリティディスクにわたって規定される幾つかの対角パリティ集合に沿って計算された対角パリティを格納する（ステップ４４５）。なお、ＲＡＩＤ５スタイルの分散パリティ実施形態が可能となるように、データディスク、行パリティディスク、又は対角パリティディスクのようなディスクの役割は、ストライプごとに違っていてもよい。

【００７０】

しかしながら、ｐ－１本の行に対して規定されるｐ本の対角に関するパリティ情報を全て保持するだけの十分な空間が、対角パリティディスク上には無い。具体的には、対角パリティディスク上には、ｐ－１ブロック分の対角パリティを入れる空間しか存在しない。データディスク、及び行パリティディスクはそれぞれ、多くとも１ブロックしか対角に貢献せず、また、データブロックであるか行パリティブロックであるかに関わらず、１つの行が、同じ対角の要素である２つのブロックを有することはない。１ストライプ中にはちょうどｐ本の対角が存在するが、対角パリティディスク上にはｐ－１個の対角パリティブロックしか存在しない。

【００７１】

これを克服するために、対角パリティ集合のうちの１つについては、対角パリティを、対角パリティディスクに格納しない（ステップ４５０）。すなわち、対角パリティディスクは、ストライプ中の幾つかの対角のうちの１つを除くそれぞれについて、パリティブロックを保持する。どの対角パリティブロックを格納しないかは、自由である。そのパリティは格納されないので、計算もされない。対角のうちの１つについてはパリティが格納されないが、本発明の技術によれば、反対角パリティをさらに使用することにより、アレイ内の任意の３つの同時ディスク故障から復旧するのに十分なだけのパリティ情報が提供される。つまり、本発明の一復旧態様によれば、アレイ内の任意の３台のディスクが失われたときでも、ストライプの中身を完全に復元することができる。

【００７２】
対角パリティを計算し、格納した後、ＲＡＩＤシステムは、ステップ４５５～４６５の処理を実施し（ステップ４４０～４５０の処理と同様に）、反対角パリティを計算し、格納する。したがって、ステップ４５５において、データブロック、及び行パリティブロックは全て、反対角に割り当てられる。上記のように、反対角の傾きは－１であり、すなわち、対角の傾きに対して直交している。表２は、０から４まで番号を付けた反対角を有する、ｐ＝５の場合のアレイの一実施形態を示している。表中の数字は、各ブロックが属する反対角パリティ集合を示している。
【００７３】
【表２】

43210
04321
10432
21043

【００７４】
次に、ステップ４６０において、全データディスク、及び行パリティディスクにわたって規定される幾つかの反対角に沿って反対角パリティを計算し、ステップ４６５において、１つを除く全ての反対角の反対角パリティを反対角パリティディスクに格納する。そして、手順４００はステップ４７０で終了する。
【００７５】
図５は、本発明の新規なＴＰ技術にしたがって編成されたディスクアレイ５００を示すブロック図である。ｎをアレイ内のディスク数とし、ｎ＝ｐ＋２であるものと仮定する。最初のｎ－３台のディスク（Ｄ０～Ｄ３）はデータを保持し、ディスクｎ－２（ＲＰ）は、データディスクＤ０～Ｄ３に対する行パリティを保持し、ディスクｎ－１（ＤＰ）は対角パリティを保持し、ディスクｎ（ＡＤＰ）は反対角パリティを保持している。この実施形態の場合、アレイ内のディスク数ｎは７（ｐ＝５）である。ディスクはブロックに分割され、ブロックはストライプにグループ化され、各ストライプは、ｎ－３（例えば４）行に相当する。また、１つの対角あたり、ｎ－２（例えば５）個の対角が存在する。
【００７６】
各行において、各ブロックが１つの対角パリティ集合に属し、且つ各ブロックが異なる対角パリティ集合に属するように、データブロックと行パリティブロックに番号が付けられている。$D_{a,b,c}$、及び$P_{a,b,c}$という記述は、特定の行（ａ）、対角パリティ（ｂ）、及び反対角パリティ（ｃ）の計算に対するデータブロック（Ｄ）、及びパリティブロック（Ｐ）それぞれの貢献を意味する。すなわち、$D_{a,b,c}$という記述は、そのデータブロックが、行パリティａ、対角パリティｂ、及び反対角パリティｃの計算に使用される行、又は対角に属することを意味し、$P_{a,b,c}$は、そのデータブロックが、行パリティ集合ａのパリティを格納し、且つ対角パリティ集合ｂ、及び反対角パリティｃに貢献することを意味する。例えば、下記のようなものである。
【００７７】
【数１】

$$P_{0,8,10} = D_{0,4,9} \oplus D_{0,5,13} \oplus D_{0,6,12} \oplus D_{0,7,11}$$

【００７８】
また、特定の対角の対角パリティの計算に使用される行パリティブロックを含む記述もある。例えば、下記のようなものである。
【００７９】

【数２】

$$P_4 = D_{0,4,9} \oplus D_{3,4,10} \oplus D_{2,4,13} \oplus P_{1,4,11}$$

【００８０】

なお、対角パリティディスクに格納される対角パリティブロックはそれぞれ、アレイ内の他のディスク（行パリティディスクは含むが、反対角ディスクは含まない）のうちの１つを除く全てのディスクからの貢献を含む。例えば、対角パリティブロックＰ$_4$は、Ｄ０（Ｄ$_{0,4,9}$）、Ｄ２（Ｄ$_{3,4,10}$）、Ｄ３（Ｄ$_{2,4,13}$）、及びＲＰ（Ｐ$_{1,4,11}$）からの貢献はあるが、Ｄ１からの貢献はない。また、対角８（Ｐ$_8$）は、計算されず、対角パリティディスクＤＰにも格納されない。

【００８１】

図６、及び図７は、対角、及び反対角へのブロックの割り当てをそれぞれ示すアレイの概略図である。図６は、対角へのブロックの割り当てを示すアレイ６００を示し、各ブロックには、そのブロックが属する対角に対応する番号が付けられている。同図にさらに（破線で）示されているのは、アレイを符号化するときに格納されなかった抜けている対角である。同様に、図７は、反対角へのブロックの割り当てを示すアレイ７００を示し、アレイに格納されなかった抜けている反対角を（破線で）示している。表１、及び表２に関して上で述べたように、単一のディスクが、同じ（反）対角のブロックを２つ有することがないようにさえすれば、（反）対角へのブロックの割り当ては、自由に変更してよい。

【００８２】

Ｄ．ディスク故障と復元

図８は、本発明の新規なＴＰ技術を使用するときに実施される適当な復元手順を判定する手順８００のステップの詳細を示すフロー図である。上記のように、説明の都合上、「ＲＡＩＤ４ディスク」は、データディスク、及び行パリティを意味するものとする。なお、データディスクと行パリティディスクは、ＲＡＩＤ４構成以外の構成、例えばＲＡＩＤ５で構成される場合もある。手順８００はステップ８０５から始まり、ステップ８１０へ進み、そこで、１以上のディスク故障が発生する。故障の原因には例えば、ディスクの完全な故障や、ディスクの一部に対するメディアエラーがある。ステップ８１５において、ＲＡＩＤシステム３４０は、故障したディスクが１つであるか、２つであるか、それとも３つであるかを判定し、適当な復旧技術を使用して、故障したディスクの復旧を行う。故障したディスクが１つである場合、手順はステップ８２０へ分岐する。そこでシステムは、故障したディスクがＲＡＩＤ４ディスクであれば、従来の行パリティを使用してその失われたブロックを計算することにより、あるいは故障したディスクが（反）対角パリティディスクであれば、（反）対角パリティを計算することにより、その単一のディスクを復旧する。

【００８３】

２台のディスクが故障した場合、手順はステップ９００へ分岐し、そこで、Ｒ－Ｄパリティ技術を実施し、二重ディスク故障からの復旧を行う。Ｒ－Ｄパリティ復元技術は、行パリティと対角パリティ、あるいは行パリティと反対角パリティを復旧に使用する場合がある。４以上のディスクが故障した場合、手順８００は、ステップ８４０で完了する前に、エラー条件により、ステップ８３５で終了する。

【００８４】

一方、３台のディスクが故障した場合、ステップ８４５において、１つのＲＡＩＤ４ディスク、対角パリティディスク、及び反対角パリティディスクが故障したか否かが判定される。そうであれば、まずステップ８５０において、従来の行パリティ技術を使用して失われたＲＡＩＤ４ディスクを復元し、その後ステップ８５５において、対角パリティと反対角パリティを再計算することによって、アレイは復元される。そうでなければ、手順はステップ８６０へ進み、そこで、ＲＡＩＤシステムは、２つのＲＡＩＤ４ディスクと、１つの（反）対角パリティディスクが故障したか否かを判定する。そうであれば、ステップ９００において、システムは、Ｒ－Ｄパリティ復元技術を使用して、故障したＲＡＩＤ４

ディスクを復元する。この復元は、良好な（反）対角パリティを使用して実施される。すなわち、対角パリティディスクが故障した場合、Ｒ－Ｄパリティ復元技術は反対角パリティを使用するが、反対角パリティディスクが故障した場合、Ｒ－Ｄパリティ復元技術は対角パリティを使用する。ステップ９００におけるＲ－Ｄパリティ復元が完了した後、ステップ８７０において、システムは次に、失われた（反）対角パリティを再計算する。一方、ステップ８６０において、３台のＲＡＩＤ４ディスクが故障したものと判定された場合、手順はステップ１０００へと分岐し、そこでＲＡＩＤシステムは、新規な三重ＲＡＩＤ４故障手順を実施する。

【００８５】

Ｅ．行－（反）対角復元

ディスクを対角に割り当てるときに、行パリティディスクとデータディスクの間に区別はないので、（反）対角パリティ集合からの復元を処理するときに、行パリティディスクとデータディスクの違いは無視することができる。例えば、任意の２台のデータディスク、あるいは、任意の１つのデータディスクと行パリティディスクが故障したものと仮定する。あるパリティ集合中の失われたブロックを復元できるのは、そのパリティ集合を構成する残りのブロックが全て、利用可能である場合だけである。ＸＯＲパリティのアーチファクトは、最初にデータを保持していたかパリティを保持していたかに関わらず、全てのブロックが数学的に等価である点にある。例えば、

【００８６】

【数３】

$$a \oplus b \oplus c = d$$

【００８７】

というパリティ構成を考える。ただし、

【００８８】

【数４】

「 ⊕ 」

【００８９】

は、ＸＯＲ演算を表わす。式の両辺にｄをＸＯＲ演算すると、

【数５】

$$a \oplus b \oplus c \oplus d = 0$$

【００９０】

となる。したがって、復元の際に、データディスクと行パリティディスクは全て、同様に処理することができる。

【００９１】

これらのディスクのそれぞれにおいて、ちょうど１つの（反）対角だけは現れない。したがって、復元は、その（反）対角の要素を含まない他のディスクから開始することができる。２つのディスクが故障しているので、大抵の場合は、（反）対角パリティデータから直ぐに復元可能なブロックが２つ存在する。これは、１ブロックだけを失った（反）対角の一方が、パリティを持たない（反）対角でない限り成り立つ。しかしながら、そのパリティ集合について多くとも１つのディスクしかデータを失っていないので、直ぐに復元可能なブロックが少なくとも１つ存在する。１つ、又は２つのブロックを（反）対角パリティから復元した後、次に、その行、又はそれらの行にある残りの失われたブロックを行パリティから復元することができる。なぜなら、この時点で、（反）対角パリティ（（反）対角パリティブロックを含まない）を使用して復元されたブロックを有する行パリティ集合から失われているのは、１ブロックだけだからである。それらのブロックを復元した後、行ブロックと同じ（反）対角（複数の場合もあり）上にある１、又は２以上のブロックを復元することができる。

【００９２】
このように、復元は、一連の（反）対角「移動」、及び水平「移動」によって進められる。ｐが素数であるから、一連の水平移動、及び（反）対角移動は全て、同じ行に２回遭遇するより前に、ストライプのあらゆる行に「遭遇」する。ただし、（反）対角上で（反）対角移動が全くできない（反）対角が１つ存在する。なぜなら、その（反）対角については、パリティが格納されていないからである。一般性を損なわずに言えば、（反）対角に０からｐ－１の番号を付した場合、パリティは、（反）対角０を除く全ての（反）対角について計算される。アレイ内でディスクが互いに所定距離だけ離れている場合、（反）対角０で復元を完了することが可能な（反）対角の固定シーケンスが常に存在する。ディスクに０からｐ－１まで番号を付し、ディスクｐ－１（行パリティディスク）がディスク０の隣りにくるようなディスクの循環を想定すると、（ｐ－１）個のシーケンスが考えられる。各シーケンスは、その距離だけ隔てられた任意の対を成すディスクの復元に対応する。表３は、例えば、ｐ＝３の場合のシーケンスを示している。

【００９３】
【表３】

Disks 1 apart: 1 2 3 4 5 6 7 8 9 10 11 12 0
Disks 2 apart: 2 4 6 8 10 12 1 3 5 7 9 11 0
Disks 3 apart: 3 6 9 12 2 5 8 11 1 4 7 10 0
Disks 4 apart: 4 8 12 3 7 11 2 6 10 1 5 9 0
Disks 5 apart: 5 10 2 7 12 4 9 1 6 11 3 8 0
Disks 6 apart: 6 12 5 11 4 10 3 9 2 8 1 7 0
Disks 7 apart: 7 1 8 2 9 3 10 4 11 5 12 6 0
Disks 8 apart: 8 3 11 6 1 9 4 12 7 2 10 5 0
Disks 9 apart: 9 5 1 10 6 2 11 7 3 12 8 4 0
Disks 10 apart: 10 7 4 1 11 8 5 2 12 9 6 3 0
Disks 11 apart: 11 9 7 5 3 1 12 10 8 6 4 2 0
Disks 12 apart: 12 11 10 9 8 7 6 5 4 3 2 1 0

【００９４】
なお、ディスクｋ離れている場合のシーケンスは、常に対角ｋから開始され、その対角を毎回モジューロｐでｋだけインクリメントすることによって継続され、ｐ　ＭＯＤ　ｐ＝０で終了する。また、ディスクｋ離れている場合のシーケンスの最初のｐ－１項は、ディスクｐ－ｋ離れている場合のシーケンスの最初のｐ－１項を逆にしたものである。

【００９５】
シーケンス上の開始位置は、どのディスク対が故障したかによって変わる。前述のようにディスクと対角に番号を付した場合、すなわち、ディスクに０からｎ－２まで順番に番号を付し、行に０からｎ－３まで順番に番号を付し、故障した各ディスクｊにおいて、ディスクｊのブロックｉが、対角パリティ集合（ｉ＋ｊ＋１）ＭＯＤ（ｎ－２）に属している場合、失われる対角は常に、対角ｊである。したがって、ｋだけ離れた一対のディスクの場合、修復を開始することが可能な２つの対角は、ｊと、（ｊ＋ｋ）ＭＯＤ（ｎ－２）である。なお、ディスクｋ離れている場合、復旧シーケンスにおいて、それら２つの対角は常に隣り合うものとなる。反対角についても、同様の計算を行うことが出来る。復旧は、２つのシーケンス上の開始点から右へ移動することによって決定される対角のシーケンスに従って進められ、ｋ＜ｐ／２とした場合、ディスクｋ離れている場合は記号（ｊ＋ｋ）ＭＯＤ（ｎ－２）から開始され、ディスクｐ－ｋ離れている場合は記号ｊから開始される。したがって、２つのデータディスクのどのような組み合わせが故障しても、また、１つのデータディスクと行パリティパリティディスクのどのような組み合わせが故障した場

合でも、常に完全な復旧が可能である。対角パリティディスクと、もう１つ他のディスクが故障した場合は、それがデータであるか行パリティであるかに関わらず、格納されている行パリティから他のディスクを復元した後、対角パリティディスクを復元することは簡単なことである。

【００９６】

なお、全てのデータブロックが、パリティの計算される（反）対角に属するとは限らない。実際、（反）対角パリティは、データブロック、及び行パリティブロックのうちの（ｐ－１）／ｐに対してしか計算されない。単一ブロックの書き換えは、そのブロックの行パリティを更新しなければならないだけでなく、そのブロックの（反）対角パリティも再計算しなければならないため、高くつく。また、そのブロックの行パリティを更新するときに、さらに、変化分をその行パリティブロックの（反）対角パリティブロックに加算しなければならない。ただし、１つのストライプが「１ブロック」幅であり、パリティ計算にしか使用されない幾つかのサブブロックから構成されるシステムの場合は、この計算を簡略化出来ることもある。ここで、計算されたパリティ更新は、行パリティに加算される。同じパリティ更新ブロックの幾つかの部分は、そのストライプの（反）対角パリティブロックの幾つかの部分にも直接加算される。

【００９７】

ディスク（ＡＤＰ）ＤＰ上の（反）対角パリティブロックは、自己のＸＯＲ計算に行パリティブロックを含める。換言すれば、ディスク（ＡＤＰ）ＤＰに格納された（反）対角パリティは、データディスクの内容に従って計算されるだけでなく、行パリティディスクの内容にも従って計算される。アレイ５００に示されているように（反）対角パリティを符号化することによって、システムは、抜けている（反）対角パリティ（例えば、対角パリティの場合、対角番号８）を除く任意の２つの同時ディスク故障からの復旧が可能となる。なぜなら、行パリティブロックが、（反）対角パリティディスクＤＰ／ＡＤＰに格納される（反）対角パリティブロックの計算に要素として含まれるからである。これに対し、従来のＥＶＥＮＯＤＤ技術は、行パリティブロックを対角パリティ集合の計算に要素として含めない。むしろ、従来のＥＶＥＮＯＤＤアプローチは、抜けている対角パリティブロックを自己の対角パリティディスクに格納される他の対角パリティブロックのそれぞれに要素として含める。

【００９８】

動作として、対角パリティディスクと、何らかのデータディスクが故障した場合、まず、そのデータディスクを行パリティディスクに基づいて復元し（例えば、従来のＲＡＩＤ－４復元技術にしたがって）、次に（反）対角パリティディスクを復元することによって、復旧は達成される。同様に、２つのパリティディスクが故障した場合、まず、データディスクから行パリティディスクを復元し、次に（反）対角パリティディスクを復元することによって、復旧は達成される。一方、任意の一対のデータディスクが故障した場合は、少なくとも１つの、大抵は２つの（反）対角パリティ集合から、一方のブロックを直ちに復元することができる。そしてシステムは、失われたデータブロックのうちの残りの一方を復元することができる。なお、行パリティディスクとデータディスクを失うことは、２つのデータディスクが失われた場合と全く同じであり、その復旧も、同様の仕方で達成される。

【００９９】

図９は、行－（反）対角パリティのための復旧手順（復元プロセス）９００に必要とされる一連のステップを示すフロー図である。図示のように、ＲＡＩＤシステム３４０が、二重故障が発生したものと判断すると、手順９００が開始される。あるいは、手順９００は、三重故障においてディスクの１つが復元された後、二重故障が残っているときに実施される場合がある。なお、手順９００は、対角を使用して実施してもよいし、反対角を使用して実施してもよい。一つを除く全ての（反）対角について、ＤＰ／ＡＤＰディスク上に（反）対角パリティブロックが格納される。したがって、手順９００はステップ９０５から開始され、ステップ９１０へ進み、そこで、（反）対角パリティを使用して、失われ

たブロックのうちの少なくとも一方、大抵は２つの復元を開始する。

【０１００】

一方の失われたブロックを復元した後、行パリティを使用して、その行にある他方の失われたブロックを復元することにより、行の復元は完了する（ステップ９１５）。他方のブロックを復元するときに、ステップ９２０において、そのブロックが、パリティを有する（反）対角に属しているか否かの判定がなされる。そのブロックが、パリティを有する（反）対角に属している場合、（反）対角パリティを使用して、その（反）対角パリティ上にある他のディスクから、（反）対角上の他方の失われたブロックを復元することができる（ステップ９２５）。つまり、抜けている（反）対角を除く全ての（反）対角について、その（反）対角上にある一方のブロックが復元されれば、他方のブロックも復元することができる。次に、手順はステップ９１５へ戻り、そこで、行パリティ集合中の他方の失われたブロックが復元される。一方、そのブロックが、パリティを有しない（反）対角（すなわち、抜けている（反）対角））に属している場合、ステップ９３０において、全てのブロックが復元されたか否かに関する判定がなされる。否であれば、手順はステップ９１０へ戻り、そこで、まず、（反）対角パリティを利用した復元を行い、次に行パリティを利用した復元を行うというパターンが、抜けている（反）対角パリティ集合の計算に使用される最後のデータブロックに達するまで続けられる。全てのブロックの復元が完了すれば、ステップ９３５において手順は終了する。一連の復元手順は常に、抜けている（反）対角パリティ集合で終了する。

【０１０１】

要するに、復元手順は、復元される最初の（反）対角から始まって、抜けている（反）対角パリティで終わる、復元可能な（反）対角を順番に列記することによって表される。互いにｋだけ離れた２つのディスクｊ、及びディスクｊ＋ｋが故障した場合、復元可能な（反）対角の一方のシーケンスは毎回ｋだけインクリメントされ、他方のシーケンスは毎回ｋだけデクリメントされる。なぜなら、行復元はディスクｋ個分右（又は左）へ移動し、且つ更にモジューロｐでそこからｋだけ高い（又は低い）（反）対角へ移動するからである。大抵の場合、複数（例えば、少なくとも２つ）の並列復元スレッドが存在する。例外は、「抜けている」（反）対角のブロックを有しないディスクであるディスクＤ０と、（反）対角パリティディスクＤＰ／ＡＤＰ以外のいずれかの他のディスクとが失われた場合である。その場合、復元されるブロックの流れは、失われた他のディスク上にある抜けている（反）対角パリティ集合に属するブロックで終了する単一の流れしか存在しない。

【０１０２】

Ｆ．三重パリティ復元

アレイ内の１以上の記憶装置故障に応答し、適当な復元技術を判定するために、本発明は例えば、ストレージオペレーティングシステムのディスクストレージ層（ＲＡＩＤシステム）において、マルチステップ・ステートマシンを実施する。具体的には、３つのＲＡＩＤ４ディスクが故障した場合、まず、「抜けている」対角パリティ、及び反対角パリティ、すなわち、先に格納されなかった対角パリティブロック、及び反対角パリティブロックを計算することにより、三重復元技術を実施する。次にＲＡＩＤシステムは、多数のクロスを生成することにより、故障したディスクのうちの中間の（すなわち、中間）ディスクに沿って、多数の４タプル和を計算する。その後、それらの４タプル和は、中間ディスク上の多数の二つ一組の和にまで低減される。生成された二つ一組の和は、解法可能な一組の一次方程式を形成する。中間ディスク上のデータは、例えば、この一組の方程式を解くことによって復元される。中間ディスク上の最初のブロックが復元された後、その解は他の方程式に代入され、これは、中間ディスク上の全てのブロックが復元されるまで継続される。中間ディスクが復元された後、次にシステムは、Ｒ－Ｄパリティ技術を実施し、残り２つのディスクを復旧する。

【０１０３】

図１０は、本発明の一実施形態による、３つの故障したＲＡＩＤ４ディスクを復元するための手順１０００のステップの詳細を示すフロー図である。手順１０００は、ステップ

１００５から開始され、ステップ１０１０へ進み、そこで、ＲＡＩＤシステム３４０は、抜けている対角パリティ、及び反対角パリティを計算する。上記のように、抜けている対角、及び反対角パリティは、ディスクアレイに格納されていない（反）対角に関係する。例えば、図６のアレイ６００では、第４の対角については、パリティが格納されていない。同様に、図７のアレイ７００では、第０の反対角については、パリティが格納されていない。この抜けている（反）対角の計算は、比較的簡単である。抜けている（反）対角のパリティは、例えば、（反）対角パリティディスク上のブロックの和として計算される場合がある。すなわち、下記のように計算される。

【０１０４】

【数６】

$$\sum\oplus Diagonal\_Parity\_Blocks = \sum\oplus Data\_Blocks \oplus \sum\oplus Data\_Blocks\_on\_Dropped\_Diagonal \oplus \sum\oplus Row\_Parity\_Blocks \oplus \sum\oplus Row\_Parity\_Blocks\_on\_Dropped\_Diagonal$$

【０１０５】

ただし、

【０１０６】

【数７】

$$\sum\oplus$$

【０１０７】

は、指定されたブロックのＸＯＲ演算による和を表わす。

【０１０８】

【数８】

$$\sum\oplus Row\_Parity\_Blocks \equiv \sum\oplus Data\_Blocks$$

【０１０９】

であり、２つの同一の対象をＸＯＲ演算した結果が０になることに留意すると、この式は、

【０１１０】

【数９】

$$\sum\oplus Data\_Blocks\_on\_Dropped\_Diagonal \oplus \sum\oplus Row\_Parity\_Blocks\_on\_Dropped\_Diagonal = \sum\oplus Blocks\_on\_Dropped\_Diagonal$$

【０１１１】

のように低減することができる。

【０１１２】

次にＲＡＩＤシステムは、故障したディスクをアレイ内の３つの故障したディスクの索引値に等しい値を有するＸ、Ｙ、及びＺとして識別し、ディスクに０－ｐから始まるラベルを付ける。つまり、例えば、アレイ５００のうちディスクＤ０、Ｄ１、及びＤ３が故障した場合は、Ｘ＝０、Ｙ＝１、Ｚ＝３となる。次に、ステップ１０１２においてシステムは、代数演算を行い、例えば、行、対角、及び反対角の３つの次元のそれぞれについて、

３つの故障したディスク上の失われたブロックのＸＯＲを計算する。例えば、この時点で和を計算をしておくことは、中間ディスクの復元が終わった後、残りの２つのディスクを復元するために必要となるＸＯＲ演算の回数を低減するのに役立つ。ステップ１０１０における抜けている／失われた（反）対角パリティの復元の結果、行パリティ集合、対角パリティ集合、及び反対角パリティ集合のそれぞれに沿って、３つのＸＯＲを計算することが可能となる。

【０１１３】

例えば、失われたブロックD_{00}、D_{01}、及びD_{03}（最初の行にあるブロック）の行パリティ和は、下記のように計算することができる。

【０１１４】

【数１０】

$$D_{00} \oplus D_{01} \oplus D_{03} = RP_0 \oplus D_{02}$$

【０１１５】

同様に、失われたブロックD_{30}、D_{31}、及びD_{33}（第４の行にあるブロック）の行パリティ和は、下記のように計算することができる。

【０１１６】

【数１１】

$$D_{30} \oplus D_{31} \oplus D_{33} = RP_3 \oplus D_{32}$$

【０１１７】

失われたブロックD_{00}、D_{11}、及びD_{33}（図７を参照すると、これらのブロックは、反対角４上のブロックである）の反対角パリティ和は、下記のように計算することができる。

【０１１８】

【数１２】

$$D_{00} \oplus D_{11} \oplus D_{33} = ADP_0 \oplus D_{22}$$

【０１１９】

失われたブロックD_{30}、D_{21}、及びD_{03}（図６を参照すると、これらは、対角３上にあるブロックである）の対角パリティ和は、下記のように計算することができる。

【０１２０】

【数１３】

$$D_{30} \oplus D_{21} \oplus D_{03} = DP_3 \oplus D_{12}$$

【０１２１】

次にＲＡＩＤシステムは、故障した中間ディスク上のｐ個の４タプル和の総計を計算する（ステップ１０１３～１０１８）。ステップ１０１３では、故障したディスクの行間距離を下記のように定義することにより、ディスクに順番を付ける。

ｇ＝Ｙ－Ｘ

ｈ＝Ｚ－Ｙ

【０１２２】

したがって、Ｘ＝０、Ｙ＝１、及びＺ＝３である上記の例を使用すると、ｇ＝１－０＝１となり、ｈ＝３－１＝２となる。この定義により、ディスクＹが中間ディスクとなる。

【０１２３】

ステップ１０１４において、システムは次に、行ｋを選択する。例えば、ｋ＝０であるものと仮定する。そして、システムは、その選択された行に対応する、失われたディスク上のブロックの行パリティの和を読み出す（ステップ１０１５）。この例では、行ｋ＝０の場合の和は、下記のように既に計算されている。

【０１２４】

【数１４】

$$D_{00} \oplus D_{01} \oplus D_{03} = RP_0 \oplus D_{02}$$

【０１２５】

次に、ステップ１０１６において、システムは、ディスクＺ上の行ｋにあるブロックの対角を読み出す。例えば、ｋ＝０であるものと仮定すると、この対角上にある失われたブロックの和は、

【０１２６】

【数１５】

$$D_{30} \oplus D_{21} \oplus D_{03} = DP_3 \oplus D_{12}$$

【０１２７】

となる。次に、ステップ１０１７において、ＲＡＩＤシステムは、ディスクＸ上の行ｋにあるブロックの反対角パリティを読み出す。例えばこれは、下記のようになる。

【０１２８】

【数１６】

$$D_{00} \oplus D_{11} \oplus D_{33} = ADP_0 \oplus D_{22}$$

【０１２９】

（反）対角の最後の行を行ｑと呼ぶものとする。次に、ステップ１０１８において、ＲＡＩＤシステムは、行ｑに対応する失われたディスク上のブロックの行パリティ和を読み出す。これは例えば、下記のようになる。

【０１３０】

【数１７】

$$D_{30} \oplus D_{31} \oplus D_{33} = RP_3 \oplus D_{32}$$

【０１３１】

図示の実施形態では、各ステップ１０１５、１０１６、１０１７において、読み出した和が、前回の和とＸＯＲ演算される。例えば、行ｋ＝０である場合、総計は、下記のようになる。

【０１３２】

【数１８】

$$D_{00} \oplus D_{01} \oplus D_{03} \oplus D_{30} \oplus D_{31} \oplus D_{33} \oplus D_{00} \oplus D_{11} \oplus D_{33} \oplus D_{30} \oplus D_{21} \oplus D_{03} =$$

$$RP_0 \oplus D_{02} \oplus RP_3 \oplus D_{32} \oplus ADP_0 \oplus D_{22} \oplus DP_3 \oplus D_{12}$$

【０１３３】

この式は、下記のように低減される。

【０１３４】

【数１９】

$$D_{01} \oplus D_{11} \oplus D_{21} \oplus D_{31} = RP_0 \oplus D_{02} \oplus RP_3 \oplus D_{32} \oplus ADP_0 \oplus D_{22} \oplus DP_3 \oplus D_{12}$$

【０１３５】

この式の右辺は分かっているから、この式には、中間ディスク上の４つの不明なものが残っている。より一般的には、各クロスにおける一番上の行と一番下の行にある重複項を削除すれば、中間ディスク上の多くとも４つのブロックの和が得られる。失われたデータを求めるために、異なるストライプからクロスを開始することにより、ｐ個のそのような和が計算される。アレイは、ｐ－１行しか有していないので、第ｐ番目の４タプル和は、ディスクＺとディスクＸのそれぞれに対応する抜けている対角ディスク、及び反対角ディスクを使用し、クロスを形成することによって形成される。例えば、図１１において、第

ｐ番目の４タプル和は、下記４つのＸＯＲ和を使用して形成される。
【０１３６】
【数２０】

（対角）　$D_{40} \oplus D_{01} \oplus D_{23}$

（反対角）　$D_{43} \oplus D_{11} \oplus D_{20}$

（行）　$D_{40} \oplus D_{41} \oplus D_{43}$

$D_{20} \oplus D_{21} \oplus D_{23}$

【０１３７】
したがって、下記のような４タプル和が得られる。
【０１３８】
【数２１】

$$D_{41} \oplus D_{01} \oplus D_{11} \oplus D_{21}$$

【０１３９】
この例において、第１のクロス（行０に基づく）の結果、［０，１，２，３］からなるタプルが得られる。このタプルは、和の計算対象となる中間ディスク上のブロックを表わす。ｐ個のクロスを生成することによって、行ｋに対応するタプルは、［ｋ，ｋ＋ｇ，ｋ＋ｈ，ｋ＋ｈ＋ｇ］で表わされる。ただし、加算は全てモジューロｐで実施される。
【０１４０】
なお、４タプル和を計算するステップは、故障したディスクの任意の順序の和として実施される。上記の例では、この順序が、Ｘ＝０、Ｙ＝１、Ｚ＝２のように選択された。あるいは、別の順序Ｘ＝０、Ｙ＝２、及びＺ＝１を選択してもよい。この場合、中間ディスクはＹ＝２になる。その場合の値は、ｇ＝Ｙ－Ｘ＝２、ｈ＝Ｚ－Ｙ＝－１となる。３つのディスク故障の場合、全部で６つの順序が可能であることが、簡単に見て取れる。各順序付けの結果、ｇとｈに対し、異なる値の集合が生成されるため、４タプル和を二つ一組の和に減らすために必要となるタプル数は異なる。したがって、必要となるＸＯＲ演算の回数を最小限に抑えるために、４タプル和を二つ一組の和に減らすために必要になるタプル集合の数が結果として最小になる順序が選択される。
【０１４１】
また、このステップでは、削除列間の距離が重要となる。当然ながら、ｇ＝ｈであれば、中間ディスクに更に２つのブロックを追加することで、４タプルから２タプルへの変換は不要となる。したがって、等距離／対称故障の場合、次のステップは不要となる。また、行「０」に対応するタプルを［０，ｇ，ｈ，ｈ＋ｇ］と表した場合、等距離故障は、条件ｇ＝＝ｈ　ＭＯＤ　（ｐ）、すなわち［（ｇ－ｈ）ＭＯＤｐ　＝０］のように一般化される。この条件は、４タプル中の第２のブロックと第３のブロックが同一のものであるため、削除されることを意味する。
【０１４２】
４タプル和を計算した後、ステップ１０２０において、ＲＡＩＤシステムは、４タプル和を中間ディスク上の二つ一組の和に低減する。二つ一組の和を形成するために、システムは、２ブロックだけ残して共通のブロックを削除するように、一部の式を選択する。一部について和を計算すると、中間ディスク上に二つ一組のブロックの和が得られる。
【０１４３】
タプルを二つ一組の和に減らすために一部のタプルを選択する方法は、多数存在する。一実施形態では、行ｋに対応するタプルから開始して、最後からオフセットｇ（又はｈ）

の位置までにある後続のタプルを選択することによって、一部のタプルを選択する。各ステップにおいて共通のブロックは削除され、手順は、２つの不明なブロックだけが残るまで続けられる。その結果、二つ一組の和が得られる。

【０１４４】

例えば、行０に対応する４タプル和、すなわち［０，ｇ，ｈ，ｇ＋ｈ］から開始する場合、オフセットｇにある次のタプルを選択すると、２つの新たなブロックを追加するとともに２つのブロックを削除することが可能となり、その結果、不明なブロックの総数をそのまま維持できることが、簡単に分かる。なぜなら、行ｇに対応するタプルは［ｇ，２ｇ，ｈ＋ｇ，２ｇ＋ｈ］であり、ブロックｇとブロックｈ＋ｇは両方のタプルに存在するため、共通ブロックｇとｈ＋ｇは削除できるからである（ただし、加算、及び乗算は全て、モジューロｐで行われるものと仮定する）。したがって、行０に対応する４タプルから開始する場合（これを第１ステップと呼ぶことにする）、ステップｍの結果、オフセットｇにある連続したタプルが選択され、ブロック［０，（ｍ＊ｇ）ＭＯＤ　ｐ，ｈ，（ｍ＊ｇ＋ｈ）ＭＯＤ　ｐ］の和が得られる。

【０１４５】

ｐが素数であり、条件｛ｇ，ｈ＜ｐ｝が変わらないものと仮定すれば、［（ｍ＊ｇ＋ｈ）ＭＯＤ　ｐ］＝０が成り立つようなｍ（０＜ｍ＜ｐ）が、常に見つかる。同様に、［（ｍ＊ｇ－ｈ）ＭＯＤ　ｐ］＝０が成り立つようなｍが、常に見つかる。したがって、［（ｍ＊ｇ＋ｈ）ＭＯＤ　ｐ］＝０が成り立つようなｍを選択すれば、第ｍのステップの後、得られる結果［０，(ｍ＊ｇ)ＭＯＤ　ｐ，ｈ，（ｍ＊ｇ＋ｈ）ＭＯＤ　ｐ］において、第１のブロックと第４のブロックは削除することが出来る。あるいは、［（ｍ＊ｇ－ｈ）ＭＯＤ　ｐ］＝０が成り立つようなｍを選択すれば、第ｍのステップの後、第２のブロックと第３のブロックは削除することが出来る。不明なブロックは２つしか残っていないので、タプルを選択するプロセスは、このステップで終了する。ｐ－１行のそれぞれにおいて４タプル和から開始し、上記のステップを繰り返すことにより、ｐ－１個の二つ一組の和が得られる。

【０１４６】

ステップ１０２５では、代数演算を実施することにより、式の一つから、既知の大きさに等しい単一の未知の値が得られる。そして、その値は、前述の式に代入され、中間ディスク上の全ての未知の値が解明され、それによって中間ディスクが復旧される。

【０１４７】

具体的には、アレイの形成には、ｐ－１行しか使用されないので、ディスクＹ上の第ｐのブロックはゼロであるものと仮定してよい。したがって、第ｐのブロックと対を成してＸＯＲ演算されるブロックの値は、二つ一組の和の生成の完了時に分かっている。すなわち、式は未知の値を１つしか有しない。解を復元されたブロックに代入し、他の二つ一組の和を使用することにより、中間ディスク上の残りのブロックを復元することができる。この時点で、ＲＡＩＤシステムは、故障したディスクを２つだけ残して、中間ディスクの復元を完了する。この問題は、行－対角パリティを使用して既に解決されているので、システムは、そのようなＲ－Ｄパリティを実施し、失われた２つのディスクを復元する。したがって、中間ディスクの復旧が完了した後、ステップ９００において、ＲＡＩＤシステムは、Ｒ－Ｄパリティを使用し、残りの２つのディスクを復元し、その後ステップ１０３５において終了する。

【０１４８】

アレイの構築に特に良好な幾つかの素数が存在する。それは、２の乗数に１を加算したものであって、且つディスクアクセスに使用されるブロックサイズよりも小さく、さらに、アレイ上に想定されるディスク数と同じ、又はそれよりも大きな素数である。２の乗数に１を加えたものである最初の幾つかの素数は、５、１７、及び２５７である。そのうち５は、ディスクアレイに、多くても４つのデータディスクまで含めることができないので、多くの場合、小さすぎる。一方、１７と２５７はいずれも、良い選択肢である。なぜなら、大半のストレージシステムは、ディスクストレージを通常、４ｋ（４０９６）バイト

、８ｋ（８１９２）バイト、又は他の同様の２の乗数のサイズの複数のブロックに分割するからである。最大で１５個、又は２５５個のデータディスクを有するアレイにおいてそれぞれ、ｐ＝１７、又はｐ＝２５７にすることにより、１６行、又は２５６行のグループの中で、対角パリティが計算される。これらはいずれも妥当な選択肢である。なぜなら、例えば４ｋバイトサイズのデータブロックを４ｋ／１６＝２５６バイト、又は４ｋ／２５６＝１６バイトのサブブロックに均等に分割することができるからである。（反）対角パリティは、サブブロックに沿って（反）対角を定義することによって計算される。行パリティは、例えば４ｋブロック全体に対してパリティを計算することによって計算され、対角パリティを持たないＲＡＩＤ４、若しくはＲＡＩＤ５のアレイにおける計算と全く同じ方法で計算することができる。

【０１４９】

（反）対角パリティ集合の計算のために、各４ｋディスクブロックを１６個、又は２５６個のサブブロックに均等に分割することも可能であるが、例示的実施形態として、このアプローチのソフトウェア、又はハードウェアによる実施形態は、各４ｋブロックのうちの、４ｋ対角パリティブロックと重ならない１つ、又は２つの連続した非重複領域を、１以上のサブブロックから構成される各領域にＸＯＲ演算しなければならない。データブロックの中身は、シフトパターンを成すようにして（反）対角パリティブロックにＸＯＲ演算され、抜けている（反）対角パリティ集合に属する各データブロックのサブブロックが、（反)対角パリティブロックに貢献しないようにする。（反）対角パリティデータを（反）対角パリティブロックに加算するための総計算時間は、データブロックを行パリティディスクに加算するための計算時間に匹敵することがある。

【０１５０】

有利なことに、本発明によれば、例えば、パリティ情報が全てディスクのような３つの装置に格納される、ＲＡＩＤ４スタイルの集中パリティフォーマットの三重ディスク故障保護が提供される。したがって、本発明のパリティ技術によれば、既存のパリティ情報を再形成、すなわち再計算することなく、データディスクをディスクアレイに徐々に追加してゆくことが可能となる。本発明は、最少量の冗長ディスク空間、すなわち、アレイ一つあたりちょうど３つのディスクしか使用しない。また、本発明によれば、所与の数のデータディスクに対し、ＥＶＥＮＯＤＤやＳＴＡＲといった従来技術のアプローチに比べて、パリティ計算のオーバヘッドも低減される。パリティ計算オーバヘッドは、本明細書に記載するＴＰ技術の場合に最適なものとなる。

【０１５１】

なお、本発明のＴＰ技術は、冗長データストリームに依存する他の用途における三重故障からの復旧にも使用される場合がある。例えば、ＴＰ技術は、データ通信アプリケーションにおいて使用されることがある。データ通信アプリケーションでは、最大で３つまでの失われた、及び／又は壊れたパケットを復元するために、追加のデータを伝送し、再送への依存性を低減する。また、さらに別の実施形態として、ＸＯＲ演算以外の他の代数演算が使用される場合もある。

【０１５２】

上記の説明は、本発明の特定の幾つかの実施形態に関するものである。しかしながら、当業者には明らかなように、それらの利点の一部、又はは全部を獲得しつつも、記載した実施形態に対して他の変更、または修正を施すことも可能である。したがって、添付の特許請求の範囲の目的は、そうした変形や修正も、本発明の真の思想、及び範囲に入るものとしてカバーすることにある。

【図面の簡単な説明】

【０１５３】

【図１】従来のＥＶＥＮＯＤＤパリティ構成に従って構成された従来のディスクアレイを示す略ブロック図である。

【図２】本発明の一実施形態によるストレージシステムを含む環境を示す略ブロック図である。

【図３】本発明の一実施形態による、図２のストレージシステム上で使用される例示的ストレージオペレーティングシステムを示す略ブロック図である。
【図４】本発明の一実施形態による三重パリティを符号化する手順のステップの詳細を示すフロー図である。
【図５】本発明の一実施形態により編成されたディスクアレイを示すブロック図である。
【図６】本発明の一実施形態による対角パリティストライプを示すディスクアレイの略ブロック図である。
【図７】本発明の一実施形態による反対角パリティストライプを示すディスクアレイの略ブロック図である。
【図８】本発明の一実施形態による、復元を実施する手順のステップの詳細を示すフロー図である。
【図９】本発明の一実施形態による、行－対角（Ｒ－Ｄ）パリティ復元を実施する手順のステップの詳細を示すフロー図である。
【図１０】本発明の一実施形態による、三重パリティ復元を実施する手順のステップの詳細を示すフロー図である。
【図１１】本発明の一実施形態による、ディスク識別子を示すディスクアレイの略ブロック図である。

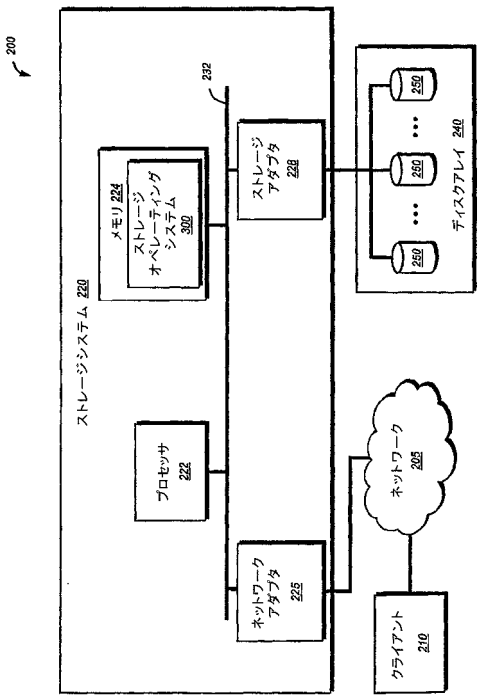
【図１】

100

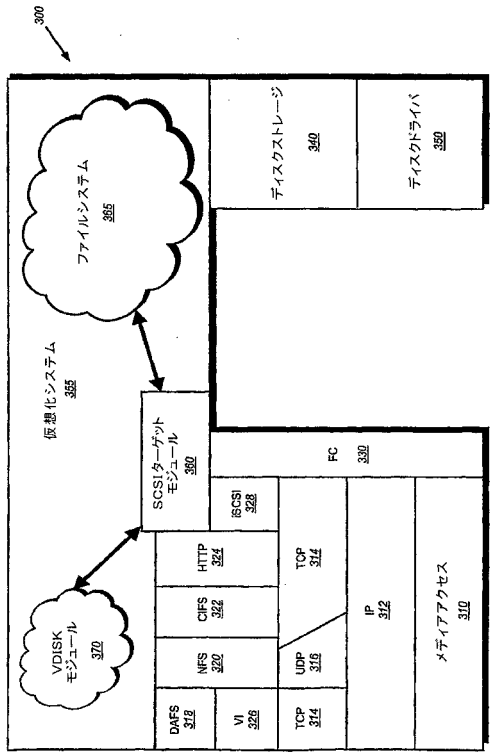
データディスク0	データディスク1	データディスク2	データディスク3	データディスク4	行パリティディスク	対角パリティディスク
D04	D05	D06	D07	D0X	P0	P4X
D15	D16	D17	D1X	D14	P1	P5X
D26	D27	D2X	D24	D25	P2	P6X
D37	D3X	D34	D35	D36	P3	P7X

（従来技術）

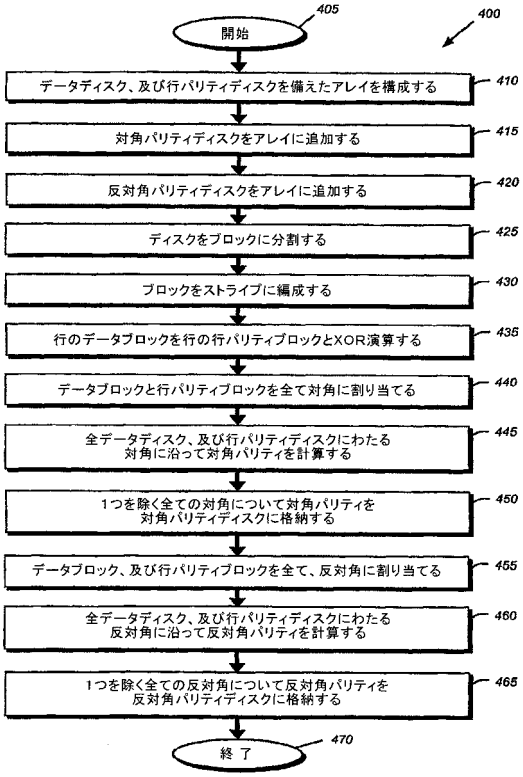
【図２】



【図３】



【図４】



【図５】

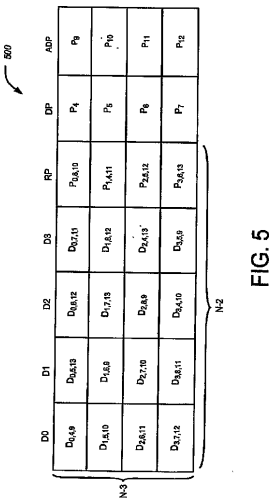


FIG. 5

【図６】

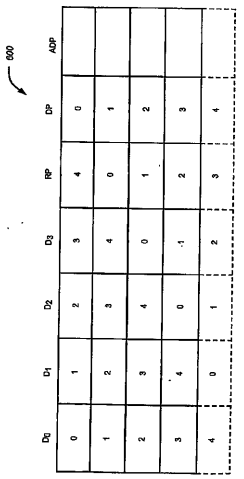


FIG. 6

【図７】

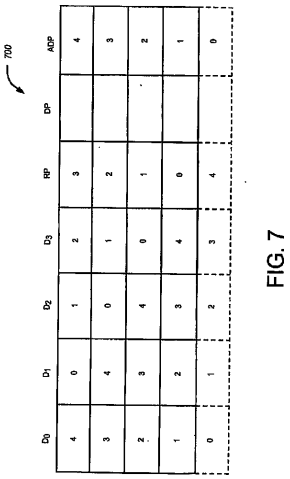
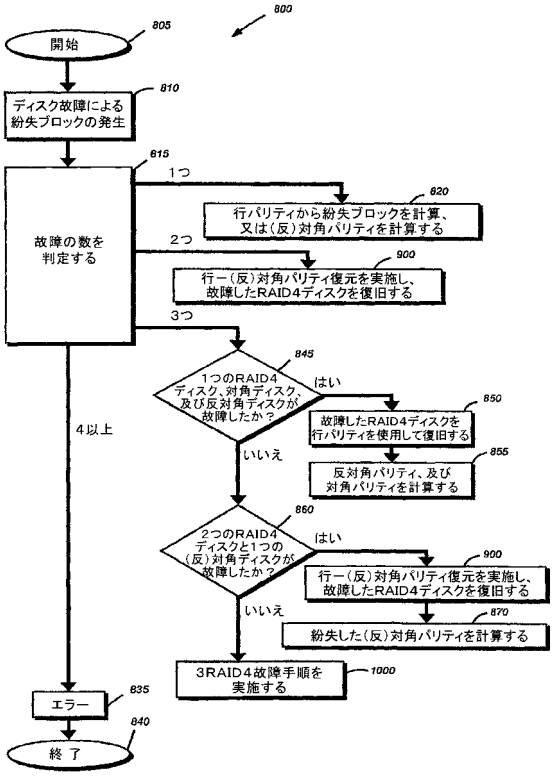
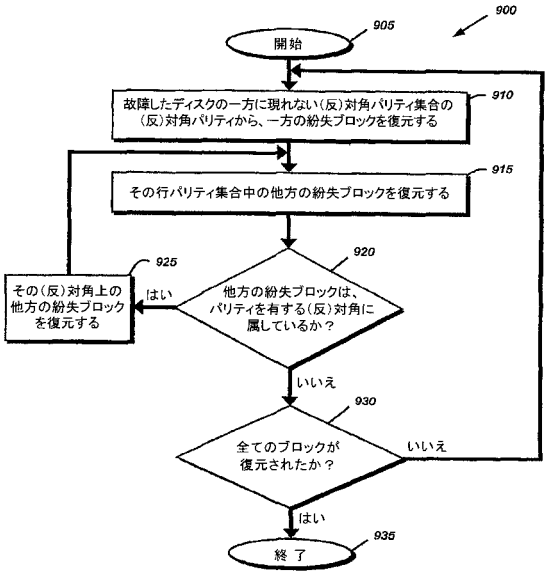


FIG. 7

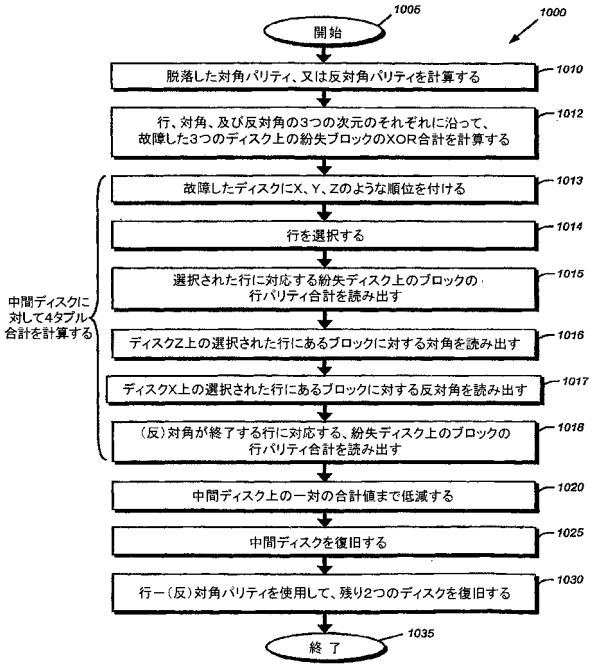
【図８】



【図９】



【図１０】



【図１１】

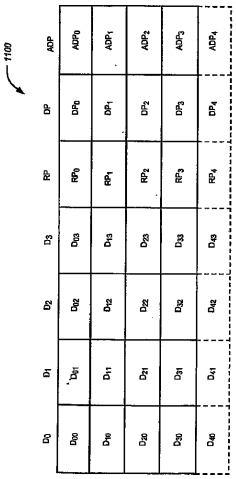


FIG. 11

フロントページの続き

(72)発明者　ゴエル，アトゥル
アメリカ合衆国カリフォルニア州９４４０４，フォスター・シティ，フォスター・シティ・ブルバード・１１１５，アパートメント・ナンバー４

審査官　菅原　浩二

(56)参考文献　特開２００３－２３３４６８（ＪＰ，Ａ）
特開平１１－０３９１０４（ＪＰ，Ａ）
特開２００５－１６６０１６（ＪＰ，Ａ）
特表２００６－５０５０３５（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F　3/06
G06F　12/16