(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 695 33 058 T2** 2005.05.12

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 726 520 B1**
(21) Deutsches Aktenzeichen: **695 33 058.6**
(96) Europäisches Aktenzeichen: **95 112 326.4**
(96) Europäischer Anmeldetag: **04.08.1995**
(97) Erstveröffentlichung durch das EPA: **14.08.1996**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **19.05.2004**
(47) Veröffentlichungstag im Patentblatt: **12.05.2005**

(51) Int Cl.[7]: **G06F 3/06**
G06F 11/10, G06F 11/20, G11B 20/18

(30) Unionspriorität:
386582 **10.02.1995** **US**

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:
Hewlett-Packard Co. (n.d.Ges.d.Staates Delaware), Palo Alto, Calif., US

(72) Erfinder:
Voigt, Douglas L., Boise, Idaho 83702, US; Nelson, Marvin D., Boise, Idaho 83704, US

(74) Vertreter:
Schoppe, Zimmermann, Stöckeler & Zinkler, 82049 Pullach

(54) Bezeichnung: **Speicherplattenanordnung mit redundanter Speicherung und Verfahren zur inkrementalen Redundanzerzeugung während des Datenschreibens auf die Speicherplattenanordnung**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

Gebiet der Erfindung

[0001] Diese Erfindung bezieht sich auf Plattenarrays und insbesondere auf redundante hierarchische Plattenarraydatenspeichersysteme, die redundante Speicherfähigkeiten aufweisen. Diese Erfindung bezieht sich ferner auf Verfahren zum inkrementalen Erzeugen von Redundanz, während Daten auf das Plattenarray geschrieben werden.

Hintergrund der Erfindung

[0002] Plattenarraydatenspeichersysteme weisen Mehrfach-Speicherplattenlaufwerksvorrichtungen auf, die angeordnet und koordiniert sind, um ein einziges Massenspeichersystem zu bilden. Für derartige Speichersysteme gibt es drei Hauptentwurfskriterien: Kosten, Leistungsfähigkeit und Verfügbarkeit. Am wünschenswertesten ist es, Speichervorrichtungen zu erzeugen, die geringe Kosten pro Megabyte, eine hohe Eingangs-/Ausgangsleistungsfähigkeit und eine hohe Datenverfügbarkeit aufweisen. „Verfügbarkeit" ist die Fähigkeit, auf Daten, die in dem Speichersystem gespeichert sind, zuzugreifen, und die Fähigkeit, im Fall eines Ausfalls einen fortgesetzten Betrieb zu gewährleisten. In der Regel wird eine Datenverfügbarkeit durch die Verwendung einer Redundanz geliefert, wobei Daten, oder Beziehungen zwischen Daten, an mehreren Plätzen gespeichert sind. Für den Fall, dass eine Speicherplatte in dem Plattearray teilweise oder vollständig ausfällt, können die Benutzerdaten über die auf den verbleibenden Platten gespeicherten redundanten Daten rekonstruiert werden.

[0003] Es gibt zwei übliche Verfahren zum Speichern von redundanten Daten. Gemäß dem ersten bzw. „Spiegel"-Verfahren werden Daten dupliziert und in zwei getrennten Bereichen des Speichersystems gespeichert. Bei einem Plattenarray werden die identischen Daten beispielsweise auf zwei getrennten Platten in dem Plattenarray bereitgestellt. Das Spiegelverfahren weist die Vorteile einer hohen Leistungsfähigkeit und einer hohen Datenverfügbarkeit aufgrund der Duplexspeichertechnik auf. Das Spiegelverfahren ist jedoch auch relativ teuer, da es die Kosten des Speicherns von Daten effektiv verdoppelt.

[0004] Bei dem zweiten bzw. „Paritäts"-Verfahren wird ein Teil des Speicherbereichs verwendet, um redundante Daten zu speichern, die Größe des redundanten Speicherbereichs ist jedoch geringer als der zum Speichern der ursprünglichen Daten verwendete verbleibende Speicherbereich. Bei einem Plattenarray, das fünf Platten aufweist, könnten beispielsweise vier Platten verwendet werden, um Daten zu speichern, wobei die fünfte Platte für das Speichern von redundanten Daten reserviert ist. Das Paritätsverfahren ist vorteilhaft, da es kostengünstiger ist als das Spiegelverfahren, im Vergleich zu dem Spiegelverfahren weist es jedoch auch geringere Leistungsfähigkeits- und Verfügbarkeitscharakteristika auf.

[0005] Diese Erfindung bezieht sich insbesondere auf ein Speichern von Daten gemäß Paritätstechniken. Bei herkömmlichen Plattenarrays ist der Raum auf den Speicherplatten zu mehreren Streifen konfiguriert, wobei sich jeder Streifen über die Speicherplatten erstreckt. Jeder Streifen besteht aus mehreren Segmenten eines Speicherraums bzw. Speicherplatzes, wobei jedes Segment ein Abschnitt des Streifens ist, der sich auf einer einzelnen Speicherplatte in dem Plattenarray befindet.

[0006] Während der Initialisierung eines bekannten Plattenarrays werden die Speicherplatten formatiert, und die Parität für jeden Streifen wird eingestellt. Nach der Initialisierung sind vier I/O-Zugriffe erforderlich, um Daten in das Plattenarray zu schreiben: ein erster I/O, um die Daten, die aktualisiert werden sollen, aus einem ausgewählten Streifen zu lesen, ein zweiter I/O, um die entsprechende Parität für Daten in diesem Streifen zu lesen, ein dritter I/O, um neue Daten zurück in den Streifen zu schreiben, und ein vierter I/O, um eine neue Parität, die die neuen Daten berücksichtigt, zurück in den Streifen zu schreiben. Es wäre wünschenswert, die Anzahl von I/Os, die erforderlich sind, um Daten auf Streifen in Plattenarrays zu schreiben, zu verringern.

[0007] Eine Technik, die bei manchen bekannten Plattenarrays bisher verwendet wurde, besteht darin, die Paritätswerte Cache-zu-speichern. Dies verringert das Erfordernis, die Parität während jedes Schreibvorgangs aus dem Plattenarray zu lesen, wodurch die Anzahl von I/Os auf drei verringert wird. Es bleibt jedoch weiterhin ein Bedarf bestehen, die Anzahl von I/Os, die erforderlich sind, um Daten auf Streifen in dem Plattenarray zu schreiben, weiter zu verringern.

[0008] Das Dokument US-A-5,166,939 lehrt ein Plattenarrayspeichersystem, bei dem die Platten in eine Mehrzahl von Redundanzgruppen unterteilt sind, wobei jede Redundanzgruppe einzeln angepasst ist, um eine geeignete Streifenbildungstiefe zu verwenden, um entweder eine Zugriffsleistungsfähigkeit oder eine Bandbreite gemäß dem darin gespeicherten Datentyp zu optimieren.

ZUSAMMENFASSUNG DER ERFINDUNG

[0009] Diese Erfindung liefert ein Plattenarraydatenspeichersystem, das eine verbesserte Leistungsfähigkeit während Paritätsschreibvorgängen aufweist und weniger I/Os erfordert. Das Plattenarray schreibt Daten in einzelne Segmente in einem Streifen, vor-

zugsweise auf sequentielle Weise, zu einem Zeitpunkt jeweils ein Segment. Es ist ein Speicherverwalter vorgesehen, der die Größen der Streifen und den Paritätsstatus jedes Segments in den Streifen bezüglich dessen, ob das Segment Daten enthält, die Bestandteil des Paritätswerts für den zugeordneten Streifen sind, kennt.

[0010] Bei der bevorzugten Implementierung verfolgt der Speicherverwalter einen Paritätsstatus jedes Segments durch die Verwendung von zwei Arten von Zeigern von einem oder mehreren Teilstreifenzeigern, die Streifen, die gefüllt werden, nachverfolgen, und von einem oder mehreren Segmentfüllzeigern, die auf spezifische Segmente in den Streifen, die gerade gefüllt werden, Bezug nehmen. Bezüglich eines ausgewählten Streifens ist bekannt, dass Segmente vor dem Segmentfüllzeiger Daten enthalten, die in die Streifenparität integriert sind, und Segmente, die auf den Segmentfüllzeiger folgen, leer und noch nicht in die Streifenparität integriert sind. In Bezug auf Streifen, auf die nicht durch den Teilstreifenzeiger Bezug genommen wird, geht man davon aus, dass bei Streifen, die Daten enthalten, alle Datensegmente durch die Streifenparität dargestellt werden, und man geht davon aus, dass alle leeren Streifen keine Daten enthalten oder keinerlei Paritätswert, der diese Segmente darstellt, aufweisen.

[0011] Nachdem ein Streifen ausgewählt wird, werden Daten in das Segment geschrieben, auf das durch den Segmentfüllzeiger Bezug genommen wird. Während dieses Datenschreibvorgangs wird ein Paritätswert, der den Daten entspricht, die in allen gefüllten Segmenten in dem ausgewählten Streifen enthalten sind, inkremental erzeugt und in einem separaten, nicht-flüchtigen Speicher wie z. B. einem NV-RAM, Cache-gespeichert. Die leeren Segmente in dem Streifen werden während des Schreibvorgangs nicht verwaltet und sind somit nicht in der inkrementalen Paritätserzeugung enthalten. Nachdem alle Datensegmente in dem Streifen gefüllt wurden, wird der endgültige Paritätswert in das Paritätssegment in dem Streifen geschrieben. Der Vorgang kann dann für den nächsten nicht-gefüllten oder teilweise gefüllten Streifen wiederholt werden.

[0012] Ein Plattenarray, das mit einem sequentiellen Segmentschreiben, einer inkrementalen Paritätserzeugung und einer Paritäts-Cache-Speicherung gemäß dieser Erfindung, die in den beigefügten unabhängigen Patentansprüchen 1 und 3 ausführlich definiert ist, implementiert ist, eliminiert die bekannten I/Os des Lesens der alten Daten und des Lesens der alten Parität. Ferner kann es die Anzahl von Malen, die ein Paritätswert in das Datenarray geschrieben wird, auf einen einzigen Schreibvorgang verringern, nachdem der ganze Streifen gefüllt wurde. Dies führt zu einer beträchtlichen Verbesserung der Schreibleistungsfähigkeit.

Beschreibung der Zeichnungen

[0013] Unter Bezugnahme auf die folgenden beigefügten Zeichnungen, in denen Beispiele gezeigt sind, die den besten Modus zum Praktizieren der Erfindung verkörpern, werden nachfolgend bevorzugte Ausführungsbeispiele der Erfindung beschrieben.

[0014] Fig. 1 ist ein schematisches Blockdiagramm eines Plattenarraydatenspeichersystems gemäß dieser Erfindung.

[0015] Fig. 2 ist eine schematische Veranschaulichung eines Speicherplatzes auf mehreren Speicherplatten und zeigt eine Datenspeicherung gemäß einer RAID-Ebene **1**.

[0016] Fig. 3 ist eine schematische Veranschaulichung eines Speicherplatzes auf mehreren Speicherplatten und zeigt eine Datenspeicherung gemäß einer RAID-Ebene **5**.

[0017] Fig. 4 ist eine schematische Veranschaulichung einer Speicherabbildungsanordnung dieser Erfindung, bei der zwei virtuelle Speicherplätze auf einen physischen Speicherplatz abgebildet werden.

[0018] Fig. 5 zeigt ein schematisches Blockdiagramm, das die Wechselwirkung zwischen Zeigern, die in einem nicht-flüchtigen Speicher und einem Speicherraum auf dem Plattenarray gehalten werden;

[0019] Fig. 6 und Fig. 7 zeigen ein Flussdiagramm von bevorzugten Schritten zum Schreiben von Daten auf das Plattenarray gemäß Paritätsredundanztechniken; und

[0020] Fig. 8 zeigt eine schematische Veranschaulichung eines Speicherraums auf mehreren Speicherplatten, die eine Müllsammeltechnik demonstriert.

Ausführliche Beschreibung der Erfindung

[0021] Fig. 1 zeigt ein Datenspeichersystem **10**, das gemäß dieser Erfindung aufgebaut ist. Vorzugsweise ist das Datenspeichersystem **10** ein Plattenarraydatenspeichersystem, das folgende Merkmale umfasst: ein hierarchisches Plattenarray **11**, das eine Mehrzahl von Speicherplatten **12** aufweist, eine Plattenarraysteuerung **14**, die mit dem Plattenarray **11** gekoppelt ist, um einen Transfer von Benutzerdaten zu und von den Speicherplatten **12** zu koordinieren, und ein RAID-Verwaltungssystem **16**. Diese Erfindung wird im Kontext ihrer bevorzugten Implementierung als redundantes hierarchisches Plattenarraysystem beschrieben. Man beachte jedoch, dass Aspekte und Konzepte dieser Erfindung auch bei anderen Arten von Plattenarraydatenspeichersystemen,

beispielsweise nicht-hierarchischen redundanten Plattenarrays, verwendet werden können.

[0022] Für die Zwecke dieser Offenbarung ist eine „Platte" jegliche nicht-flüchtige, direkt zugreifbare, wiederbeschreibbare Massenspeichervorrichtung, die die Fähigkeit aufweist, ihre eigenen Speicherausfälle zu erfassen. Sie umfasst sowohl sich drehende magnetische und optische Platten als auch Festzustandsplatten, oder nicht-flüchtige elektronische Speicherelemente (z.B. PROMs, EPROMs und EEPROMs). Der Begriff „Plattenarray" umfasst eine Sammlung von Platten, die Hardware, die erforderlich ist, um sie mit einem oder mehreren Hostcomputern zu verbinden, und eine Verwaltungssoftware, die verwendet wird, um den Betrieb der physischen Platten zu steuern und sie der Host-Betriebsumgebung als eine oder mehrere virtuelle Platten zu präsentieren. Eine „virtuelle Platte" ist eine abstrakte Entität, die durch die Verwaltungssoftware in dem Plattenarray verwirklicht ist.

[0023] Der Begriff „RAID" (Redundant Array of Independent Plattes, redundantes Array aus unabhängigen Platten) bedeutet ein Plattenarray, bei dem ein Teil der physischen Speicherkapazität verwendet wird, um redundante Informationen über Benutzerdaten, die auf dem Rest der Speicherkapazität gespeichert sind, zu speichern. Die redundanten Informationen ermöglichen eine Regeneration von Benutzerdaten für den Fall, dass eine der zu dem Array gehörenden Platten oder der Zugriffspfad zu derselben ausfällt. Eine ausführlichere Erläuterung von RAID-Systemen findet sich in einem Buch mit dem Titel The RAIDBook: A Source Book for RAID Technology, das am 9. Juni 1993 von RAID Advisory Board, Lino Lakes, Minnesota, veröffentlicht wurde.

[0024] Die Plattenarraysteuerung **14** ist über einen oder mehrere Schnittstellenbusse **13**, beispielsweise eine Kleincomputer-Schnittstelle (SCSI – small computer system interface) mit dem Plattenarray **11** gekoppelt. Das RAID-Verwaltungssystem **16** ist über ein Schnittstellenprotokoll **15** wirksam mit der Plattenarraysteuerung **14** gekoppelt. Ferner ist das Datenspeichersystem **10** über einen I/O-Schnittstellenbus **17** mit einem Hostcomputer (nicht gezeigt) gekoppelt. Das RAID-Verwaltungssystem **16** kann als separate Komponente verkörpert sein oder kann in der Plattenarraysteuerung **14** oder in dem Hostcomputer konfiguriert sein, um eine Datenverwaltungseinrichtung zum Steuern von Plattenspeicherungs- und von Zuverlässigkeitsebenen sowie zum Transferieren von Daten zwischen verschiedenen Zuverlässigkeits-Speicherstufen bereitzustellen. Diese Zuverlässigkeits-Speicherebenen sind, wie nachfolgend beschrieben wird, vorzugsweise Spiegel- oder Paritäts-Redundanzebenen, können jedoch auch eine Zuverlässigkeits-Speicherebene ohne jegliche Redundanz umfassen.

[0025] Die Plattenarraysteuerung **14** ist vorzugsweise als duale Steuerung implementiert, die aus der Plattenarraysteuerung A, die durch Bezugszeichen **14a** bezeichnet ist, und aus der Plattenarraysteuerung B, die durch Bezugszeichen **14b** bezeichnet ist, besteht. Die dualen Steuerungen **14a** und **14b** verbessern die Zuverlässigkeit, indem sie für den Fall, dass eine Steuerung funktionsuntüchtig wird, eine fortlaufende Sicherung und Redundanz liefern. Diese Erfindung kann jedoch auch mit einer einzigen Steuerung oder mit anderen Architekturen praktiziert werden.

[0026] Das hierarchische Plattenarray **11** kann als unterschiedliche Speicherplätze, einschließlich seines physischen Speicherplatzes und eines oder mehrerer virtueller Speicherplätze, charakterisiert werden. Diese verschiedenen Speicheransichten sind durch Abbildungstechniken verbunden. Beispielsweise kann der physische Speicherplatz des Plattenarrays in einen virtuellen Speicherplatz abgebildet werden, der Speicherbereiche gemäß den verschiedenen Datenzuverlässigkeitsebenen darstellt. Manche Bereiche in dem virtuellen Speicherplatz können einer ersten Zuverlässigkeitsspeicherebene zugewiesen sein, beispielsweise Spiegel- oder RAID-Ebene **1**, und andere Bereiche können einer zweiten Zuverlässigkeitsspeicherebene zugewiesen sein, beispielsweise Paritäts- oder RAID-Ebene **5**. Die verschiedenen Abbildungstechniken und virtuellen Plätze, die RAID-Ebenen betreffen, werden nachfolgend ausführlicher beschrieben.

[0027] Das Datenspeichersystem **10** umfasst einen Speicherabbildungsspeicher **21**, der eine durchgängige Speicherung der virtuellen Abbildungsinformationen liefert, die verwendet werden, um verschiedene Speicherplätze ineinander abzubilden. Der Speicherabbildungsspeicher befindet sich außerhalb des Plattenarrays und vorzugsweise in der Plattenarraysteuerung **14**. Die Speicherabbildungsinformationen können durch die Steuerung oder das RAID-Verwaltungssystem kontinuierlich oder periodisch aktualisiert werden, während sich die verschiedenen Abbildungskonfigurationen unter den verschiedenen Ansichten ändern.

[0028] Vorzugsweise ist der Speicherabbildungsspeicher **21** als zwei nicht-flüchtige RAMs (Direktzugriffsspeicher) **21a** und **21b** verkörpert, die in jeweiligen Steuerungen **14a** und **14b** angeordnet sind. Ein beispielhafter nicht-flüchtiger RAM (NVRAM) ist ein batteriegepufferter RAM. Ein batteriegepufferter RAM verwendet Energie von einer unabhängigen Batteriequelle, um im Falle eines Verlustes von Leistung an das Datenspeichersystem **10** die Daten über einen bestimmten Zeitraum in dem Speicher beizubehalten. Ein bevorzugter Aufbau ist ein selbstauffrischender batteriegepufferter DRAM (dynamischer RAM).

[0029] Die dualen NVRAMs **21a** und **21b** sorgen für eine redundante Speicherung der Speicherabbildungsinformationen. Die virtuellen Abbildungsinformationen werden dupliziert und gemäß Spiegelredundanztechniken in beiden NVRAMs **21a** und **21b** gespeichert. Auf diese Weise kann der NVRAM **21a** dafür reserviert sein, die ursprünglichen Abbildungsinformationen zu speichern, und der NVRAM **21b** kann dafür reserviert sein, die redundanten Abbildungsinformationen zu speichern. Bei einem alternativen Aufbau kann ein gespiegelter Speicherabbildungsspeicher unter Verwendung eines einzelnen nichtflüchtigen RAMs mit einem ausreichenden Platz, um die Daten im Duplikat zu speichern, konfiguriert sein.

[0030] Wie in Fig. 1 gezeigt ist, weist das Plattenarray **11** mehrere Speicherplattenlaufwerksvorrichtungen **12** auf. Beispielhafte Größen dieser Speicherplatten betragen ein bis drei Gigabytes. Die Speicherplatten können unabhängig voneinander mit mechanischen Buchten verbunden oder von denselben getrennt sein, die eine Schnittstellenbildung mit dem SCSI-Bus **13** liefern. Bei einer Implementierung ist das Datenspeichersystem mit zwölf aktiven mechanischen Buchten entworfen. Vier SCSI-Busse werden verwendet, um eine Schnittstelle zwischen diesen Buchten und der Plattenarraysteuerung **14** zu bilden (d.h. ein Bus pro drei mechanischen Buchten). Falls die aktiven Buchten vollständig beladen sind, weist das Datenspeichersystem eine beispielhafte kombinierte Kapazität von 12 bis 36 Gigabytes auf. Die Plattenarraysteuerung **14** erkennt die Speicherplatten **12** unabhängig davon, in welche Bucht sie eingesteckt sind. Das Datenspeichersystem **10** ist entworfen, um ein „heißes Einstecken" zusätzlicher Platten in verfügbare mechanische Buchten in dem Plattenarray, während das Plattenarray in Betrieb ist, zu ermöglichen.

[0031] Die Speicherplatten **12** in dem Plattenarray **11** können zu Erläuterungszwecken vom Konzept her so erklärt werden, dass sie in einer Spiegelgruppe **18** aus mehreren Platten **20** und einer Paritätsgruppe **22** aus mehreren Platten **24** angeordnet sind. Die Spiegelgruppe **18** stellt eine erste Speicherposition oder einen ersten RAID-Bereich des Plattenarrays dar, die bzw. der Daten gemäß einer ersten bzw. Spiegelredundanzebene speichert. Diese Spiegelredundanzebene wird auch als RAID-Ebene **1** betrachtet. Die RAID-Ebene **1** bzw. oder ein Spiegeln von Platten bietet die höchste Datenzuverlässigkeit, indem sie insofern einen Eins-Zu-Eins-Schutz bietet, als jedes Datenbit dupliziert und in dem Datenspeichersystem gespeichert wird. Die Spiegelredundanz ist schematisch durch die drei Paare von Platten **20** in Fig. 1 dargestellt. Ursprüngliche Daten können auf einem ersten Satz von Platten **26** gespeichert werden, während dupliziert, redundante Daten auf dem gepaarten zweiten Satz von Platten **28** gespeichert werden.

[0032] Fig. 2 veranschaulicht die Speicherung von Daten gemäß der RAID-Ebene **1** noch ausführlicher. Die vertikalen Spalten stellen einzelne Platten dar, von denen Platten 0, 1, 2 und 3 veranschaulicht sind. Der in diesem Plattenarray aus vier Platten enthaltene physische Speicherplatz kann zu mehreren Streifen konfiguriert sein, wie sie durch die horizontalen Zeilen veranschaulicht sind. Ein „Streifen" erstreckt sich über die Speicherplatten und ist aus zahlreichen, vorzugsweise gleich großen Segmenten an Speicherplatz gebildet, wobei ein Segment jeder Platte in dem Array zugeordnet ist. Das heißt, dass ein Segment der Abschnitt eines Streifens ist, der sich auf einer einzelnen Platte befindet. Jeder Streifen hält eine vorbestimmte Datenmenge, die über die Speicherplatten verteilt ist. Manche Segmente eines Streifens werden für ursprüngliche Daten verwendet, während andere Segmente für redundante Daten verwendet werden.

[0033] Bei diesem Beispiel einer Spiegelredundanz (RAID-Ebene **1**) werden Daten, die auf der Platte 0 in dem Segment 0 des Streifens 0 gespeichert sind, dupliziert und auf der Platte 1 in dem Segment 0' des Streifens 0 gespeichert. Desgleichen werden Daten, die auf der Platte 2 in dem Segment 5 des Streifens 2 gespeichert sind, zu dem Segment 5' des Streifens 2 auf der Platte 3 gespiegelt. Auf diese Weise wird jede Dateneinheit dupliziert und auf den Platten gespeichert. Der Redundanzentwurf der Fig. 2 ist zu Erläuterungszwecken vorgesehen. Die redundanten Daten müssen nicht sauber in demselben Streifen plaziert werden, wie gezeigt ist. Beispielsweise könnten Daten, die auf der Platte 0 in dem Segment 2 des Streifens 1 gespeichert sind, dupliziert werden und auf der Platte 3 in dem Segment T' des Streifens S plaziert werden.

[0034] Unter erneuter Bezugnahme auf Fig. 1 stellt die Paritätsgruppe **22** der Platten **24** eine zweite Speicherposition bzw. einen zweiten RAID-Bereich dar, in der bzw. dem Daten gemäß einer zweiten Redundanzebene, z.B. RAID-Ebene **5**, gespeichert werden. In dieser erläuternden Veranschaulichung von sechs Platten sind ursprüngliche Daten auf der fünften Platte **30** gespeichert, und redundante „Paritäts"-Daten sind auf der sechsten Platte **32** gespeichert.

[0035] Fig. 3 zeigt einen Paritäts-RAID-Bereich-Entwurf etwas ausführlicher. Ähnlich dem Entwurf des Spiegel-RAID-Bereichs der Fig. 2 kann der physische Speicherplatz der Platten 0, 1, 2, 3 zu mehreren gleich großen Streifen konfiguriert sein. Bei diesem veranschaulichten Beispiel werden Daten gemäß der RAID-Ebene **5** gespeichert, und die redundanten Daten werden in den Segmenten gespeichert, auf die durch den Buchstaben P Bezug genommen wird. Die redundanten P-Segmente speichern die Parität der anderen Segmente in dem Streifen. In

dem Streifen 0 speichert das redundante P-Segment auf der Platte 3 beispielsweise die Parität der Platten 0, 1 und 2. Die Parität für jeden Streifen wird durch eine bestimmte Funktion berechnet, beispielsweise eine Exklusiv-ODER-Funktion, die durch das Symbol „⊕" dargestellt wird. Die Paritäten für die ersten vier Streifen (wobei die tiefgestellte Zahl den entsprechenden Streifen darstellt) lauten wie folgt:

P_0 = Segment 0 ⊕ Segment 1 ⊕ Segment 2 = Platte 0 ⊕ Platte 1 ⊕ Platte 2

P_1 = Segment 3 ⊕ Segment 4 ⊕ Segment 5 = Platte 0 ⊕ Platte 1 ⊕ Platte 3

P_2 = Segment 6 ⊕ Segment 7 ⊕ Segment 8 = Platte 0 ⊕ Platte 2 ⊕ Platte 3

P_3 = Segment 9 ⊕ Segment 10 ⊕ Segment 11 = Platte 1 ⊕ Platte 2 ⊕ Platte 3

[0036] Eine Paritätsredundanz ermöglicht eine Regeneration von Daten, die auf einer der Platten nicht mehr zur Verfügung stehen. Wenn beispielsweise die Daten in dem Segment 5 nicht mehr zur Verfügung stehen, kann ihr Inhalt aus den Segmenten 3 und 4 und den Paritätsdaten in dem Segment P ermittelt werden. Eine Paritätsspeicherung ist kostengünstiger als eine Spiegelspeicherung, ist jedoch auch weniger zuverlässig und weist eine geringere Leistungsfähigkeit auf.

[0037] Die Plattenanordnung der Fig. 1 ist zu Konzeptionszwecken vorgesehen. In der Praxis weist das Plattenarray **11** einfach eine Mehrzahl von Platten **12** auf, die in der Lage sind, Daten gemäß einer Spiegel- und Paritätsredundanz zu speichern. Aus dem durch alle Platten **12** gelieferten, zur Verfügung stehenden Speicherplatz wird ein Abschnitt dieses Speicherplatzes einer Spiegelredundanz zugewiesen, und ein weiterer Abschnitt wird einer Paritätsredundanz zugewiesen. Vorzugsweise sind die Platten **12** konfiguriert, um mehrere gleich große Speicherregionen (die bei Fig. 4 mit dem Bezugszeichen **35** benannt sind) zu enthalten, wobei einzelne Regionen mehrere Segmente aufweisen. Die Regionen sind gruppiert, um bei einer virtuellen Ansicht des Speicherplatzes RAID-Bereiche zu bilden. Zusätzlich wird eine weitere (durch einen Host definierte) Ansicht eines Speicherplatzes dem Benutzer oder Host präsentiert, so dass die RAID-Bereiche und die Datenredundanz-Speichertechniken für den Benutzer oder Host transparent sind. Diese Merkmale werden nachstehend unter Bezugnahme auf Fig. 4 ausführlicher erläutert.

[0038] Das Datenspeichersystem **10** verwaltet die „Migration bzw. Bewegung" von Daten zwischen Spiegel- und Paritätsspeicherschemata. Die Verwaltung beider Redundanztypen wird durch das RAID-Verwaltungssystem **16** (Fig. 1) koordiniert. Das RAID-Verwaltungssystem **16** verwaltet die zwei unterschiedlichen Typen von RAID-Bereichen in dem Plattenarray als Speicherhierarchie, wobei die Spiegel-RAID-Bereiche ähnlich einem Cache für die Paritäts-RAID-Bereiche agieren. Das RAID-Verwaltungssystem **16** verschiebt und organisiert die Daten zwischen dem Spiegel- und dem Paritäts-RAID-Bereich gemäß einem definierten Leistungsfähigkeitsprotokoll, und verwaltet die Daten ansonsten auf ebensolche Weise. Der Vorgang des Bewegens von Daten zwischen dem Spiegel- und dem Paritäts-RAID-Bereich wird als „Migration" bezeichnet.

[0039] Das Datenspeichersystem **10** versucht, die für die Leistungsfähigkeit kritischeren Daten in die Spiegel-RAID-Bereiche zu plazieren, da dies die höchste Leistungsfähigkeit und Zuverlässigkeit liefert. Die Leistungsfähigkeitsprotokolle, die durch das RAID-Verwaltungssystem **16** implementiert sind, umfassen eine von zwei bevorzugten Migrationsrichtlinien. Gemäß der ersten Migrationsrichtlinie, die als „Zugriffshäufigkeit" bekannt ist, werden die Daten in dem hierarchischen Plattenarray, auf die am häufigsten zugegriffen wird, in dem Spiegel-RAID-Bereich **18** beibehalten. Daten, auf die weniger häufig zugegriffen wird, werden in dem Paritäts-RAID-Bereich **22** gehalten. Gemäß einer zweiten Migrationsrichtlinie, die als „Zugriffsaktualität" bekannt ist, werden die Daten, deren Zugriff am wenigsten lange zurückliegt, in dem Spiegel-RAID-Bereich **18** beibehalten, während die Daten, deren Zugriff länger zurückliegt, in dem Paritäts-RAID-Bereich **22** gespeichert werden. Es können auch andere Leistungsfähigkeitsprotokolle verwendet werden. Idealerweise sind solche Protokolle auf der Basis der spezifischen Computeranwendung und der Erfordernisse des Benutzers definiert.

[0040] Zusätzlich „stimmt" das RAID-Verwaltungssystem **16** die Speicherressourcen eines Datenspeichersystems gemäß einer Funktion von zwei Parametern „ab". Größe der physischen Speicherkapazität und Größe der vorhandenen Menge an Benutzerdaten, die in dem Datenspeichersystem gespeichert sind. Anfänglich werden alle Daten in Spiegel-RAID-Bereichen gespeichert, da dies die höchste Leistungsfähigkeit und Zuverlässigkeit liefert. Während mehr Daten zu dem Datenspeichersystem hinzugefügt werden, werden die Daten zwischen Spiegel-RAID-Bereichen und Paritäts-RAID-Bereichen bewegt, um die Leistungsfähigkeit und Zuverlässigkeit zu optimieren. Während sich das Datenspeichersystem an eine volle Kapazität annähert, werden im Rahmen des Bemühens, alle durch den Benutzer gestellten Anforderungen zu erfüllen und trotzdem durch Redundanz eine Zuverlässigkeit zu liefern, immer mehr Daten in Paritäts-RAID-Bereiche bewegt. Dementsprechend liefert das Datenspeichersystem dieser Erfindung eine maximale Flexibilität und Anpassung. Es erfordert nicht, dass der Benutzer einen spezifischen Speicherbetriebszustand auswählt, sondern kann sich statt dessen an jegliche durch den Benutzer an das System gestellte Forderung anpassen.

[0041] Fig. 4 veranschaulicht ein Speicherabbilden des verfügbaren Speicherplatzes des Datenspeichersystems **10** als mehrere Lagen eines abgebildeten virtuellen Speicherplatzes. Jedes vertikal verlängerte Rechteck in dem Diagramm stellt eine Ansicht des physischen Speicherplatzes dar. Bei diesem Diagramm wird auf den physischen Speicherplatz **34** durch zwei virtuelle Speicheransichten **40** und **50** Bezug genommen. Der physische Speicherplatz **34** wird durch vier Platten (z.B. Platten 12 in Fig. 1) dargestellt, die durch die Bezugszeichen 0, 1, 2 und 3 benannt sind. Die den Platten zugeordneten vier Rechtecke stellen eine Ansicht des physischen Speicherplatzes dar, bei der die Platten 1, 2 und 3 eine ungefähr gleiche Speicherkapazität aufweisen und die Platte 0 eine etwas geringere Speicherkapazität aufweist. Beispielhafte Speicherkapazitäten für derartige Platten betragen 1 bis 3 Gigabytes. Der Speicherplatz **34** ist in Bereiche A0, A1, A2 usw. unterteilt. Einzelne Bereiche enthalten zahlreiche Streifen, im Bereich A0 z.B. Streifen O–Q. Einzelne Bereiche enthalten auch zahlreiche Regionen **35**. Die Regionen **35** bestehen vorzugsweise aus einer ausgewählten Anzahl von Segmenten einer gleichmäßigen Größe auf jeder Speicherplatte, so dass die Regionen über das gesamte Plattenarray eine gleiche Größe aufweisen. Eine beispielhafte Größe einer Region **35** ist ein Megabyte.

[0042] Der Speicherplatz der Platten wird in eine erste virtuelle Zwischenansicht **40** der RAID-Ebene des physischen Speicherplatzes **34** abgebildet. Diese erste virtuelle Ansicht ist vom Konzept her ein Satz von RAID-Bereichen, die auf eine zweite Anwendungsansicht abgebildet werden können, die einen angrenzend adressierbaren Speicherplatz darstellt. Die physische Konfiguration und RAID-Ansichten des Speicherplatzes entziehen sich der Anwendungsansicht.

[0043] Der RAID-Bereich-Speicherplatz **40** ist die Speicheransicht, die den Spiegel- und Paritätsspeicherplatz identifiziert. Beispielsweise kann ein RAID-Bereich **42** einen Spiegel-RAID-Bereich von M Zuweisungsblöcken **43** darstellen, während der RAID-Bereich **44** einen Paritäts-RAID-Bereich von N Zuweisungsblöcken **45** darstellen kann. Die Zuweisungsblöcke **43** und **45** weisen vorzugsweise eine gleiche Größe auf, wobei eine beispielhafte Größe **64** Kilobytes beträgt. Diese RAID-Bereiche beziehen sich auf entsprechende physische Bereiche A0, A1, A2 usw. auf dem physischen Speicherplatz **34**. Beispielsweise können sechzehn 64K-Zuweisungsblöcke **43** oder **45** an der virtuellen RAID-Ansicht auf eine einzige 1M-Region **35** abgebildet werden.

[0044] Die Spiegel- und Paritäts-RAID-Bereiche können den gesamten Speicherplatz **34** des Plattenarrays einnehmen, müssen aber nicht. Dementsprechend kann während bestimmter Anwendungen ein unbenutzter und unbezeichneter Speicherplatz vorliegen, der keinem bestimmten RAID-Bereich entspricht. Ein solcher Speicherplatz kann jedoch in einen Spiegel- oder Paritäts-RAID-Bereich umgewandelt werden. Ferner ist zu beachten, dass die RAID-Bereiche in der Abbildung in benachbarte Bereiche auf dem Plattenarray abgebildet werden, wobei jede einem RAID-Bereich zugeordnete Region an derselben physischen Adresse auf jeder Speicherplatte angeordnet ist. Die RAID-Bereiche können alternativ auch zu nicht-benachbarten Bereichen auf dem Plattenarray abgebildet werden.

[0045] Der in den RAID-Bereichen verfügbare Speicherplatz wird in eine zweite virtuelle Vorderes-Ende-Ansicht **50** der Anwendungsebene abgebildet, die eine Speicheransicht ist, wie sie durch den Benutzer oder das Hostanwendungsprogramm definiert und demselben präsentiert wird. Bei einer Betrachtung durch den Benutzer oder das Hostanwendungsprogramm kann die virtuelle Ansicht **50** der Anwendungsebene eine einzige große Speicherkapazität darstellen, die den verfügbaren Speicherplatz auf den Speicherplatten **12** angibt. Der virtuelle Speicherplatz **50** präsentiert eine Ansicht eines linearen Satzes von gleich großen virtuellen Speicherblöcken **52** und **53**, die einzeln als 0, 1, 2, ... J–1, J, J+1, ..., L–1, L, L+1, ..., usw. bezeichnet werden. Die virtuellen Blöcke **52** und **53** weisen vorzugsweise dieselbe Größe auf wie die Zuweisungsblöcke in dem RAID-Bereich **40**, wobei eine beispielhafte Größe **64** Kilobytes beträgt. Der virtuelle Blockspeicherplatz **50** wird in der durch die RAID-Bereiche **40** dargestellten Ansicht durch eine Tabelle von Bezugnahmen oder Zeigern (wie sie durch Pfeile **54** dargestellt sind) auf Zuweisungsblöcke, dargestellt. Die virtuellen Blöcke **52** und **53** bei der virtuellen Anwendungsansicht **50** sind somit über die in der virtuellen Blocktabelle beibehaltenen Zeiger den Zuweisungsblöcken **43** und **45** an der virtuellen RAID-Ansicht **50** zugeordnet. Es gibt mindestens zwei Arten von RAID-Bereichen, auf die von der virtuellen Blocktabelle verwiesen werden kann: Spiegel und Parität.

[0046] Das RAID-Verwaltungssystem **16** kann die Konfiguration der RAID-Bereiche über den physischen Speicherplatz dynamisch verändern. Die Anzahl von RAID-Bereichen für jeden Typ kann je nach der Menge von Benutzerdaten, die in dem System gespeichert sind, und je nach der Größe des physischen Plattenspeicherplatzes erhöht oder verringert werden. Folglich befinden sich das Abbilden der RAID-Bereiche in der virtuellen Ansicht **40** der RAID-Ebene auf die Platten und das Abbilden der virtuellen Ansicht **50** der Anwendungsebene auf die RAID-Ansicht **40** allgemein in einem Veränderungszustand. Der Speicherabbildungsspeicher in den NVRAMs **21a** und **21b** (Fig. 1) behält die aktuellen Abbildungsinformationen bei, die durch das RAID-Verwaltungssystem **16** verwendet werden, um die

RAID-Bereiche auf die Platten abzubilden, und behält die Informationen bei, die verwendet werden, um zwischen die zwei virtuellen Ansichten abzubilden. Während das RAID-Verwaltungssystem die Abbildungen der RAID-Ebene dynamisch verändert, aktualisiert es auch die Abbildungsinformationen in dem Speicherabbildungsspeicher, um die Veränderungen zu reflektieren.

[0047] Die Migrationsoperation des Speichersystems **10** wird nun unter Bezugnahme auf Fig. 1 und Fig. 4 beschrieben.

[0048] Für die Zwecke einer fortführenden Erläuterung nehmen virtuelle Blöcke **53** des virtuellen Speicherplatzes **50** der Anwendungsebene Bezug auf zugeordnete Zuweisungsblöcke **45** in dem Paritäts-RAID-Bereich **44**, die in dem Bereich A1 des physischen Speicherplatzes **34** gespeichert sind. Derartige virtuelle Blöcke **53** können als „virtuelle Paritätsblöcke" bezeichnet werden, während die zugeordneten Zuweisungsblöcke **45** als „Paritätszuweisungsblöcke" bezeichnet werden. Desgleichen nehmen virtuelle Blöcke **52** Bezug auf zugeordnete Zuweisungsblöcke **43** in dem Spiegel-RAID-Bereich **42**, die in dem Bereich A0 des physischen Speicherplatzes **34** gespeichert sind. Derartige virtuelle Blöcke **52** können hierin als „virtuelle Spiegelblöcke" bezeichnet werden, während die zugeordneten Zuweisungsblöcke **43** als „Spiegelzuweisungsblöcke" bezeichnet werden.

[0049] Um Daten von einem RAID-Bereich zu einem anderen zu bewegen, wird allgemein ein virtueller Block, der einem Zuweisungsblock eines Typs der ersten RAID-Ebene (z.B. Spiegel oder Ebene **1**) zugeordnet ist, ausgewählt. Anschließend wird ein nicht benutzter Zuweisungsblock, der einen Typ der zweiten RAID-Ebene (beispielsweise Parität oder Ebene **5**) darstellt, lokalisiert. Falls ein nicht benutzter Zuweisungsblock nicht lokalisiert werden kann, wird einer erzeugt. Als nächstes werden Daten von dem Zuweisungsblock, der zuvor dem ausgewählten virtuellen Block zugeordnet war, an den nicht benutzten Zuweisungsblock transferiert, der bewirkt, dass die Daten einer Redundanzebenenänderung unterzogen werden. Beispielsweise werden Daten, die einst gemäß der Spiegelredundanz gespeichert wurden, nun gemäß einer Paritätsredundanz gespeichert, oder umgekehrt. Als abschließenden Schritt wird die Abb. **54** des virtuellen Speicherplatzes **50** der Anwendungsebene auf den virtuellen Speicherplatz **40** der RAID-Ebene modifiziert und aktualisiert, um die Datenverschiebung widerzuspiegeln. Der ausgewählte virtuelle Block, der zuvor einem Zuweisungsblock des Typs der ersten RAID-Ebene zugeordnet war, nimmt nun über einen aktualisierten Zeiger Bezug auf einen Zuweisungsblock des Typs der zweiten RAID-Ebene, der die migrierten Daten enthält. Jegliche Abbildungsveränderung, die während dieser Transaktion auftritt, würde in dem Speicherabbildungsspeicher **21** aktualisiert werden.

[0050] Die fortdauernde Erörterung liefert eine ausführlichere Erklärung des Bewegens von Daten zwischen Spiegel- und Paritätsspeicherbereichen gemäß bevorzugten Verfahren und Sequenzen dieser Erfindung. Zum Bewegen von Daten von einem Paritäts- zu einem Spiegelspeicher wird die folgende Sequenz verwendet:

1. Das RAID-Verwaltungssystem lokalisiert einen unbenutzten Spiegelzuweisungsblock **43** in einem Spiegel-RAID-Bereich **42**.
2. Falls keiner gefunden wird, erzeugt das RAID-Verwaltungssystem einen Spiegelzuweisungsblock (nachfolgend erläutert).
3. Das RAID-Verwaltungssystem hebt neue Speicheranforderungen an den zu bewegenden virtuellen Block zeitweilig auf.
4. Das RAID-Verwaltungssystem wartet, bis alle aktiven Datenspeicherungsanforderungen an den virtuellen Block abgeschlossen sind.
5. Die Daten von dem Paritätszuweisungsblock **45**, der dem virtuellen Block **53** zugewiesen ist, werden in einen vorübergehenden Speicherpuffer eingelesen.
6. Die Daten werden anschließend in den bei Schritt **2** gewählten Spiegelzuweisungsblock **43** geschrieben.
7. Die virtuelle Blocktabelle wird modifiziert, um auf die neue Position der Daten in dem Spiegelzuweisungsblock **43** zu verweisen.
8. Die zeitweilig aufgehobenen Speicheranforderungen werden wiederaufgenommen.

[0051] Gemäß der obigen Prozedur bewegte sich ein virtueller Block **53** in der virtuellen Ansicht **50** der Anwendungsebene von einer Paritäts- zu einer Spiegelspeicherung. Relativ zu der virtuellen Zwischenansicht **40** bewegten sich Daten von einem Paritätszuweisungsblock **45** in dem Paritäts-RAID-Bereich **44** zu einem Spiegelzuweisungsblock **43** in dem Spiegel-RAID-Bereich **42**. In dem physischen Speicherplatz bewegten sich Daten von dem Bereich A1 zum Bereich A0.

[0052] Falls kein unbenutzter Spiegelzuweisungsblock lokalisiert werden kann (Schritt 1 oben), versucht das RAID-Verwaltungssystem die folgende bevorzugte Sequenz von drei Techniken. Erstens versucht das RAID-Verwaltungssystem, einen unbenutzten (und somit unbenannten) RAID-Bereich zu lokalisieren, der in einen Spiegel-RAID-Bereich umgewandelt werden kann, ohne die Systemschwelle eines unbenutzten Speichers der RAID-Ebene zu verletzen, die benötigt wird, um zu garantieren, dass eine Migration immer stattfinden kann. Falls dies fehlschlägt und das System mehr als die reservierte Menge an unbenutztem Speicher der RAID-Ebene aufweist, bewegt das System Daten innerhalb des

Paritätsspeichers, um unbenutzte Paritätszuweisungsblöcke zu unbenutzten RAID-Bereichen zu sammeln. Falls diese Migration einen unbenutzten RAID-Bereich ergibt, der, wie oben erwähnt, in einen Spiegel-RAID-Bereich umgewandelt werden kann, wandelt das System ihn in einen Spiegel-RAID-Bereich um. Andernfalls bewegt das System alternativ dazu Daten von einem Spiegel- zu einem Paritätsspeicher, verdichtet den Spiegelspeicher und wandelt einen unbenutzten Speicher der RAID-Ebene in einen Paritätsspeicher um, bis das System den unbenutzten Speicher der RAID-Ebene ausreichend für die Lokalisierung eines unbenutzten Spiegelzuweisungsblocks oder eine Umwandlung eines unbenutzten RAID-Bereichs in einen Spiegel-RAID-Bereich erhöht. Da Spiegelzuweisungsblöcke mehr physischen Speicherplatz belegen als Paritätszuweisungsblöcke, führt diese letzte Technik zu einer Nettoerhöhung des Umfangs an unbenutztem Speicher der RAID-Ebene.

[0053] Das Erzeugungs-/Umwandlungsprotokoll, das verwendet wird, um unbenutzte Spiegelzuweisungsblöcke zu lokalisieren und einzurichten, ist vorteilhaft, da es dem RAID-Verwaltungssystem ermöglicht, die Speicherzuweisung gemäß der Menge an Benutzerdaten und der Größe des physischen Speicherplatzes zwischen Paritäts- und Spiegelbereichen selektiv einzustellen. Da die Datennutzung und die Speicherkapazität variieren, verwendet das RAID-Verwaltungssystem eine oder mehrere der obigen drei Techniken, um die Datenmenge, die in einem Spiegelspeicher gehalten wird, zu maximieren.

[0054] Das RAID-Verwaltungssystem versucht, die Situation zu vermeiden, bei der eine Speicheranforderung warten muss, bis die Platzschaffungssequenz einen unbenutzten Spiegelzuweisungsblock ergibt, indem sie während einer Leerlaufzeit unbenutzte RAID-Bereiche erzeugt. In manchen Situationen können Speicheranforderungen während der Platzschaffungssequenz jedoch zeitweilig aufgehoben werden. Das RAID-Verwaltungssystem konfiguriert den virtuellen Blockspeicherplatz derart, dass der virtuelle Platz geringer ist als die RAID-Ansicht. Dies gewährleistet, dass für eine Migration oder andere Zwecke ein freier Platz reserviert wird, der gleich zumindest einem RAID-Bereich ist. Auf diese Weise ergibt die Sequenz von Techniken immer einen unbenutzten Spiegelzuweisungsblock.

[0055] Um Daten von einem Spiegel- zu einem Paritätsspeicher zu bewegen, wird die folgende Sequenz verwendet:

1. Das RAID-Verwaltungssystem wählt einen virtuellen Block aus **52** aus, um gemäß einer Migrationsrichtlinie, beispielsweise Zugriffsaktualität oder Zugriffshäufigkeit, von einem Spiegel- zu einem Paritätsspeicher zu bewegen.
2. Das RAID-Verwaltungssystem lokalisiert einen unbenutzten Paritätszuweisungsblock **45** in einem Paritäts-RAID-Bereich **44**.
3. Falls kein solcher Block gefunden werden kann, wird ein für eine Migration reservierter Platz gemäß den oben beschriebenen Erzeugungstechniken in einen Paritäts-RAID-Bereich umgewandelt.
4. Neue Speicheranforderungen an den zu bewegenden virtuellen Block werden zeitweilig aufgehoben.
5. Das RAID-Verwaltungssystem wartet, bis alle aktiven Speicheranforderungen an den virtuellen Block abgeschlossen sind.
6. Daten werden von dem Spiegelzuweisungsblock **43**, der dem virtuellen Block **52** zugeordnet ist, in einen vorübergehenden Speicherpuffer gelesen.
7. Die Daten werden in den gewählten Paritätszuweisungsblock **45** geschrieben.
8. Die virtuelle Blocktabelle wird modifiziert, um auf die neue Position der Daten in dem Paritätszuweisungsblock **45** zu verweisen.
9. Datenanforderungen an den virtuellen Block werden wiederaufgenommen.

[0056] Die obigen zwei aufgezählten Sequenzen liefern Beispiele dessen, wie das Speichersystem dieser Erfindung arbeiten kann, um Daten zwischen zwei verschiedenen Redundanzebenen zu bewegen.

[0057] Obwohl sich die obige Erörterung auf eine Datenmigration zwischen einer Spiegel- und einer Paritätsspeicherung konzentrierte, ist diese Erfindung insbesondere auf den Prozess des Schreibens von Daten auf die Speicherplatten gemäß Paritätstechniken gerichtet. Die oben beschriebene Speicherverwaltungstechnik ermöglicht dieser Erfindung, eine Aktivität der Paritäts-RAID-Bereiche in segmentgroßen Stücken zu verwalten, wodurch eine inkrementale Paritätserzeugung ermöglicht wird.

[0058] Wie in dem Abschnitt „Hintergrund der Erfindung" erwähnt wurde, schreiben herkömmliche Plattenarrays während des Initialisierungsprozesses ein Voreinstellungsmuster, das sich in der Regel aus einem sich wiederholenden Bit oder Bytemuster zusammensetzt. Ein beispielhaftes Muster besteht darin, alle „1"en auf das Plattenarray zu schreiben. Herkömmliche Plattenarrays stellen während Initialisierungsprozeduren außerdem die Parität für jeden Streifen in denselben ein.

[0059] Nach diesem Initialisierungsprozess ist das herkömmliche Plattenarray bereit, Benutzerdaten zu schreiben. Gemäß seiner Verwendung in dieser Offenbarung bedeutet der Begriff „Benutzerdaten" Datenbytes, die Informationen bezüglich der Anwendung des Rechensystems enthalten. Beispielsweise können Benutzerdaten aus den Daten bestehen, die infolge einer Anwendung von dem Hostcomputer erhalten werden. Benutzerdaten könnten auch aus Da-

ten bestehen, die von der Speicherplatte wiederhergestellt wurden und die zuvor das Ergebnis einer Rechenanwendung gewesen waren. Der Begriff „Benutzerdaten" soll nicht einfache Formatmuster, die während der Initialisierung verwendet werden, beispielsweise ein Muster aller „1"en, umfassen.

[0060] Bei herkömmlichen Plattenarrays sind vier I/O-Zugriffe erforderlich, um Benutzerdaten in das Plattenarray zu schreiben: ein erster I/O, um die Daten, die aktualisiert werden sollen, aus einem ausgewählten Streifen zu lesen, ein zweiter I/O, um die entsprechende Parität für Daten in diesem Streifen zu lesen, ein dritter I/O, um neue Daten zurück in den Streifen zu schreiben, und ein vierter I/O, um eine neue Parität, die die neuen Daten berücksichtigt, zurück in den Streifen zu schreiben.

[0061] Gemäß dem Plattenarray dieser Erfindung schreibt ein Speicherverwalter Benutzerdaten sequentiell auf den Streifen, zu jeglichem Zeitpunkt ein Segment, und erzeugt während jedes Datenschreibvorgangs inkremental einen Paritätswert. Der Speicherverwalter Cache-speichert den Paritätswert, bis eine Parität für den gesamten Streifen berechnet werden kann. Der abschließende Paritätswert wird danach auf den Streifen geschrieben. Das Plattenarray verringert die Anzahl von I/Os während einer Schreibprozedur beträchtlich.

[0062] Der Speicherverwalter umfasst eine Einrichtung zum Bestimmen, welche Datensegmente in eine Streifenparität integriert sind und welche nicht. Das heißt, dass der Datenverwalter die Größe der Streifen sowie den Paritätsstatus jedes Segments in den Streifen bezüglich dessen, ob das einzelne Segment Daten enthält, die Bestandteil des Paritätswerts für den zugeordneten Streifen sind, kennt. Bei der bevorzugten Implementierung verfolgt der Speicherverwalter einen Paritätsstatus jedes Segments durch die Verwendung von zwei Arten von Zeigern: von einem oder mehreren Teilstreifenzeigern, die Streifen, die gefüllt werden, nachverfolgen, und von einem oder mehreren Segmentfüllzeigern, die auf spezifische Segmente in den Streifen, die gerade gefüllt werden, Bezug nehmen. Diese Zeiger werden zum Zweck einer raschen Nutzung durch den Speicherverwalter vorzugsweise in einem nicht-flüchtigen Speicher gehalten.

[0063] Fig. 5 zeigt einen NVRAM **21a**, der einen Teilstreifenzeiger **62** und einen Segmentfüllzeiger **64** enthält. Bei dem veranschaulichten Beispiel werden die Zeiger verwendet, um auf Segmente in dem Streifen **66** Bezug zu nehmen, der sich über Platten 0 – 4 des Plattenarrays **11** erstreckt. Der NVRAM **21a** enthält ferner eine Speicherabbildung **60**, die das Abbilden der virtuellen Speicherplätze auf RAID-Ebene und Anwendungsebene nachverfolgt. Der Spiegel-NVRAM **21b** ist in dieser Figur nicht gezeigt, würde jedoch identische Informationen, wie sie bei dem NVRAM **21a** veranschaulicht sind, enthalten.

[0064] Der Teilstreifenzeiger **62** nimmt auf einzelne Streifen und insbesondere auf nicht-gefüllte Streifen in dem Plattenarray Bezug. Anfänglich identifiziert der Teilstreifenzeiger einen leeren Streifen, der keine Benutzerdaten enthält. Der Segmentfüllzeiger **64** nimmt auf einzelne Segmente in dem Streifen, der durch den Teilstreifenzeiger **62** ausgewählt ist, Bezug. Der Segmentfüllzeiger **64** lokalisiert das leere Segment, das dazu bereit ist, mit Benutzerdaten gefüllt zu werden. Bei Fig. 5 nimmt der Teilstreifenzeiger **62** auf einen nicht-gefüllten Streifen **66**, wie er durch Pfeil **68** dargestellt ist, Bezug, und der Segmentfüllzeiger **64** nimmt auf das Segment 2 (wie es durch Pfeil **70** dargestellt ist), das nächste leere Segment in dem Streifen **66**, Bezug. Vorzugsweise durchläuft der Segmentfüllzeiger sequentiell den Streifen, um nacheinander auf jedes leere Segment Bezug zu nehmen, wobei er mit dem Segment 1 beginnt und mit dem Segment 5 endet.

[0065] Gemäß dieser bevorzugten Implementierung verfolgt der Speicherverwalter effektiv den Paritätsstatus aller Segmente in dem Plattenarray durch die Hilfe der zwei Zeiger nach. In Bezug auf den durch den Teilstreifenzeiger **62** ausgewählten Streifen ist bekannt, dass Segmente vor dem Segmentfüllzeiger **64** Daten enthalten, die in die Streifenparität integriert sind, und dass Segmente, die auf den Segmentfüllzeiger folgen, leer sind und noch nicht in die Streifenparität integriert sind. In Bezug auf diejenigen Streifen, auf die nicht durch den Teilstreifenzeiger **64** Bezug genommen wird, geht man davon aus, dass bei Streifen, die Daten enthalten, alle Datensegmente durch die Streifenparität dargestellt werden. Man geht davon aus, dass die leeren Streifen keine Daten enthalten oder keinerlei Paritätswert, der diese Segmente darstellt, aufweisen.

[0066] Nachdem ein leerer Streifen durch den Teilstreifenzeiger identifiziert wurde, werden Benutzerdaten in das Segment geschrieben, auf das durch den Segmentfüllzeiger Bezug genommen wurde. Bei Fig. 5 wurden Benutzerdaten bereits in das erste Segment 1 in Streifen **66** geschrieben. Während der Segmentfüllzeiger **64** nacheinander durch die verbleibenden Segmente läuft, werden Benutzerdaten in jedes Segment in dem Streifen geschrieben, mit Ausnahme des Segments, das für Parität reserviert ist. In diesem Fall wird die Parität in das letzte Segment 5 geschrieben. Falls das Paritätssegment nicht das letzte in dem Streifen ist, überspringt der Segmentfüllzeiger jedoch das Paritätssegment während seines sequentiellen Durchlaufens des Streifens und kehrt dann zurück, um die Parität in das Paritätssegment zu schreiben, nachdem die Benutzerdaten in alle Datensegmente des Streifens platziert wurden. Die Position des Paritätssegments in dem Streifen

wird durch ein Paritätsplatzierungsverfahren bestimmt, das auf dem technischen Gebiet üblich ist. Es sei erwähnt, dass Benutzerdaten nur in das Segment geschrieben werden, auf das durch den Zeiger **64** Bezug genommen wird, und dass die anderen Datensegmente in Ruhe gelassen werden, bis auf sie Bezug genommen wird.

[0067] Gemäß einem Aspekt dieser Erfindung wird anschließend an jeden Datenschreibvorgang in ein Segment in dem Streifen inkremental ein Paritätswert erzeugt. Wie oben unter Bezugnahme auf Fig. 3 beschrieben wurde, wird ein Paritätswert gemäß einer Logikfunktion, z. B. einer Exklusiv-ODER-Funktion (XOR-Funktion) berechnet. Wenn lediglich ein Segment gefüllt ist, wie in Fig. 5 gezeigt ist, ist der Paritätswert identisch mit den Benutzerdaten. Während sich jedoch mehr Segmente füllen, verändert sich der Paritätswert inkremental, um die Parität für die Benutzerdaten, die in allen gefüllten Segmenten enthalten sind, widerzuspiegeln. Die leeren oder ungenutzten Segmente in dem Streifen werden während des sequentiellen Schreibvorgangs nicht verwaltet und sind somit in der inkrementalen Paritätserzeugung nicht enthalten.

[0068] Der NVRAM **21a** umfasst einen Paritätscache **72**, der den inkrementalen Paritätswert vorübergehend speichert. Es wird ein neues Segment beschrieben, der Speicherverwalter berechnet einen neuen Paritätswert auf der Basis der Logikfunktion (d. h. XOR) des zuvor Cache-gespeicherten Paritätswerts und der soeben in das nächste Segment geschriebenen Benutzerdaten. Das sequentielle Segmentbeschreiben und inkrementale Cache-Speichern wird Segment um Segment wiederholt, bis alle Daten tragenden Segmente gefüllt sind.

[0069] Infolge des Paritäts-Cache-Speicherns muss der Paritätswert nicht nach jedem Schreibvorgang in das Plattenarray geschrieben werden. Statt dessen wird der Paritätswert erst in das Plattenarray geschrieben, nachdem alle Datensegmente gefüllt wurden. Das heißt, wenn der Segmentfüllzeiger **64** das letzte Segment 5 erreicht, schreibt das RAID-Verwaltungssystem den Paritätswert in das Paritätssegment für den Streifen **66**, was in diesem Fall das Segment 5 ist, obwohl jegliches Segment in dem Streifen zum Speichern der Parität verwendet werden kann.

[0070] Das Plattenarray dieser Erfindung eliminiert das Erfordernis, Benutzerdaten und Parität während eines Schreibvorgangs aus dem Plattenarray zu lesen, wodurch die ersten zwei I/Os herkömmlicher Plattenarrays eliminiert werden. Durch ein Cache-Speichern des Paritätswerts verringert das Plattenarray außerdem auf effektive Weise die Anzahl von Malen, die die Parität für jeden Streifen auf die Platte geschrieben wird. Obwohl bekannte Plattenarrays vier I/Os erforderten, um Benutzerdaten mit Parität zu schreiben, erfordert das Plattenarray dieser Erfindung im einzelnen lediglich 1 + 1/(N – 1) I/Os, um Benutzerdaten mit Parität zu schreiben (wobei N gleich der Anzahl von Platten in dem Streifen ist).

[0071] Wie oben erwähnt wurde, werden die Teilstreifenzeiger **62**, die Segmentfüllzeiger **64** und der Paritätscachespeicher **72** durch den Speicherverwalter verwaltet und aufrechterhalten. Bei der bevorzugten Implementierung ist der Speicherverwalter als das RAID-Verwaltungssystem **16** (Fig. 1) verkörpert, das ferner die Benutzerdaten, redundanten Daten und ungenutzten Platz auf dem Plattenarray nachverfolgt. Jedoch kann in dem Kontext dieser Erfindung auch ein Speicherverwalter verwendet werden, der konfiguriert ist, um einzig und allein die Speicherzuweisung gemäß einer Paritätsspeicherung zu verwalten.

[0072] Fig. 6 und Fig. 7 veranschaulichen ein bevorzugtes Verfahren zum Schreiben von Benutzerdaten in ein Plattenarray gemäß Paritätsredundanztechniken. Der Prozess beginnt bei Schritt **100** (Fig. 6), wo bestimmt wird, ob der Teilstreifenzeiger **62** null ist, was bedeutet, dass er derzeit nicht auf einen nicht-gefüllten Streifen Bezug nimmt. Falls der Teilstreifenzeiger auf einen Streifen zeigt (d. h. der „Nein"-Zweig aus Schritt **100**), werden Daten in das nächste ungenutzte Segment geschrieben, auf das durch den Segmentfüllzeiger **64** Bezug genommen wird (Schritt **102**). Ein inkrementaler Paritätswert für alle Benutzerdaten, die in diejenigen Segmenten enthalten sind, die gefüllt wurden, wird durch ein XOR-Verknüpfen der neuen Benutzerdaten in die Cache-gespeicherte Parität (Schritt **104**) berechnet und erneut Cache-gespeichert (Schritt **106**).

[0073] Bei Schritt **108** wird bestimmt, ob der Segmentfüllzeiger **64** auf das letzte Segment in dem Streifen Bezug nimmt. Das RAID-Verwaltungssystem verfolgt die Zuweisung jedes Platzes in einem Streifen. Es weiß, wie viele Segmente in einem Streifen sind und kennt alle ungenutzten Segmente, die in dem Streifen noch nicht zugewiesen wurden. Wie oben erörtert wurde, geht das RAID-Verwaltungssystem ferner davon aus, dass alle Segmente vor dem Segmentfüllzeiger in der Streifenparität dargestellt sind, und dass alle Segmente hinter dem Zeiger leer sind und in der Streifenparität nicht berücksichtigt sind. Ferner geht das RAID-Verwaltungssystem davon aus, dass bei Streifen, auf die nicht Bezug genommen wird und die Daten enthalten, alle Segmente in ihre Streifenparität integriert sind, und dass alle leeren Streifen, auf die nicht Bezug genommen wurde, Segmente aufweisen, die nicht Bestandteil irgendeiner Parität sind.

[0074] Infolge dieser Betriebsparameter kennt das RAID-Verwaltungssystem den Paritätszustand jedes Segments in dem Plattenarray sowie die Position der

Zeiger in dem Array. Das RAID-Verwaltungssystem weiß somit, ob der Segmentfüllzeiger das Ende des Streifens erreicht hat oder nicht. Falls der Zeiger das letzte Segment noch nicht erreicht hat (d. h. der „Nein"-Zweig aus Schritt **108**), wird der Segmentfüllzeiger zu dem nächsten Datensegment bewegt (Schritt **110**, Fig. 7). Der bei Schritt **104** berechnete Paritätswert kann dann optional in das Paritätssegment in dem Streifen geschrieben werden (Schritt **112**).

[0075] Unter erneuter Bezugnahme auf Fig. 6 wird, falls der Segmentfüllzeiger das letzte Segment erreicht hat, der Cache-gespeicherte Paritätswert, der eine Parität für den gesamten Streifen darstellt, in das Paritätssegment in dem Streifen geschrieben (Schritt **114**). An diesem Punkt wird der Streifen vollständig mit Benutzerdaten und redundanten Informationen gefüllt. Demgemäss kann der Teilstreifenzeiger aufgehoben werden, bis er wieder verwendet wird, um den nächsten leeren Streifen zu identifizieren (Schritt **116**), wie nachstehend ausführlicher beschrieben wird.

[0076] Bei Schritt **118** wertet das RAID-Verwaltungssystem aus, ob die Benutzerdaten, die soeben in die Segmente in dem Streifen geschrieben wurden, zuvor einem anderen Ort auf dem Plattenarray zugewiesen wurden. Eine derartige Situation entsteht, wenn das RAID-Verwaltungssystem beispielsweise Benutzerdaten auf dem Plattenarray umorganisiert und Benutzerdaten von einem Ort zu einem anderen bewegt. Falls es neugeschriebene Benutzerdaten sind (d. h. der „Ja"-Zweig aus Schritt **118**), hebt das RAID-Verwaltungssystem die Zuordnung der Benutzerdaten zu ihrer alten Position auf, wodurch ein „Loch" (d. h. ein leeres Segment) in dem vorherigen Streifen erzeugt wird (Schritt **120**). Bei Schritt **122** wird die Speicherabbildung **60** aktualisiert, um die neue Datenposition widerzuspiegeln.

[0077] Unter erneuter Bezugnahme auf den anfänglichen Schritt **100** ist der Teilstreifenzeiger während der Zeit zwischen dem Abschluss eines Streifens und der Identifizierung eines neuen leeren Streifens null. Falls der Teilstreifenzeiger null ist (d. h. der „Ja"-Zweig aus Schritt **100**), sucht das RAID-Verwaltungssystem einen neuen leeren Streifen (Schritt **124**, Fig. 7). In dem Fall, dass kein leerer Streifen identifiziert werden kann (d. h. der „Nein"-Zweig aus Schritt **124**), findet das RAID-Verwaltungssystem einen teilweise gefüllten Streifen, der „Löcher" in demselben aufweist, und leert ihn (Schritt **126**). Dieser Prozess ist auch als „Müllsammlung" bekannt, und ein Beispiel ist in Fig. 8 gezeigt.

[0078] Fig. 8 zeigt Streifen **150** – **156**, die die Platten 0 – 4 des Plattenarrays überqueren. Die Segmente, die Benutzerdaten enthalten, sind durch den Buchstaben „D" bezeichnet, auf den eine Zahl folgt. Leere oder ungenutzte Segmente sind durch den Buchstaben „U" bezeichnet. Man beachte, dass ungenutzte Segmente U „Löcher" in den Streifen **150** – **156** erzeugen. Zu Beispielszwecken sei angenommen, dass das RAID-Verwaltungssystem entscheidet, den Streifen **150** zu leeren. Benutzerdaten D0 in dem Segment auf der Platte 0 werden zu einem ungenutzten Segment in dem Streifen **154** bewegt, und Benutzerdaten D1 in dem Segment auf der Platte 2 werden zu einem ungenutzten Segment in dem Streifen **152** bewegt. Diese Datenbewegung leert den Streifen **150**, der nun aus ungenutzten Datensegmenten und einem Paritätssegment besteht. Gleichzeitig füllt die Datenbewegung die „Löcher" in anderen Streifen, wodurch diese Streifen voller werden.

[0079] Dieser beispielhafte Typ des Müllsammelprozesses bewegt Benutzerdaten willkürlich von dem zu leerenden Streifen zu Löchern in anderen Streifen. Bei einer anderen Implementierung schreibt der Müllsammelprozess Benutzerdaten sequentiell von dem bezeichneten Streifen auf dieselbe Weise, wie sie oben zum sequentiellen Schreiben von Benutzerdaten nach Segment beschrieben wurde, in einen neuen leeren Streifen. Gleichermaßen können auch andere Müllsammeltechniken verwendet werden, um Streifen durch den Schritt **126** der Fig. 7 zu leeren.

[0080] Unter erneuter Bezugnahme auf Fig. 7 wählt das RAID-Verwaltungssystem den leeren Streifen aus (Schritt **128**) und stellt den Teilstreifenzeiger ein, um auf den Streifen Bezug zu nehmen (Schritt **130**). Der Segmentfüllzeiger wird initialisiert, um auf den Anfang des Streifens Bezug zu nehmen (Schritt **132**). Anschließend werden Benutzerdaten in das erste ungenutzte Segment geschrieben (Schritt **134**). Bei Schritt **136** wird der anfängliche Paritätswert den Benutzerdaten gleichgesetzt und Cache-gespeichert. Danach wird der Segmentfüllzeiger zu dem nächsten Datensegment bewegt (Schritt **110**), und der Prozess wird für die verbleibenden Segmente des Streifens fortgesetzt.

[0081] Die oben beschriebene Implementierung verwendet zwei Arten von Zeigern, die leere Streifen und Segmente in diesen Streifen identifizieren. Bei einem weiteren Ausführungsbeispiel können mehrere Paare von Zeigern in NVRAM zurückgehalten werden, z. B. ein Paar pro RAID-Bereich. Die Verwaltung einer Cache-gespeicherten Parität wird mit dem inkrementalen Füllen mehrerer Streifen koordiniert. Falls mehrere Paare von Zeigern verwendet werden, kann eine Koordinierung ein Cache-Speichern mehrerer Paritätssegmente aus verschiedenen Streifen und/oder ein Platzieren der Cache-Parität in einem Muster beinhalten, das eine zeitlich günstige Wiederverwendung von Cache-Ressourcen ermöglicht, z. B. ein Platzieren einer Parität, nachdem jedes Datensegment geschrieben wurde.

[0082] Bei einer weiteren optionalen Implementierung kann das RAID-Verwaltungssystem konfiguriert sein, um teilweise gefüllte Streifen auszuwählen, die Benutzerdaten in einem oder mehreren zusammenhängenden Segmenten am Anfang des Streifens enthalten, während die Segmente am Ende des Streifens leer sind. Das RAID-Verwaltungssystem initialisiert den Segmentfüllzeiger auf das erste leere Segment in dem teilweise gefüllten Streifen und setzt anschließend den sequentiellen Schreibvorgang gemäß derselben Techniken, wie sie oben beschrieben wurden, fort. Ferner sei erwähnt, dass der Prozess Streifen um Streifen fortgesetzt werden kann, um ganze RAID-Bereiche zu schreiben.

[0083] Bei einer weiteren bevorzugten Implementierung wird jedes Paar von Zeigern zu einem einzigen Zeiger kombiniert, der sowohl einen Streifen als auch ein Segment in demselben identifiziert.

[0084] Als Ausführungsbeispiel, das alternativ zu einer Verwendung von zwei Arten von Zeigern ist, kann das RAID-Verwaltungssystem eine Bitabbildung als sein Mittel zum Bestimmen verwenden, welche Datensegmente in eine Streifenparität integriert sind und welche nicht. Die Bitabbildung würde Informationen über jedes einzelne Segment in dem Plattenarray enthalten. Dieses Ausführungsbeispiel erfordert im Vergleich zu den Zeigern mehr Speicherraum in dem NVRAM, ist jedoch wirksam bezüglich des Nachverfolgens der Parität der Datensegmente.

[0085] Vorschriftsgemäß wurde die Erfindung in einer Sprache beschrieben, die mehr oder weniger spezifisch für strukturelle und methodische Merkmale ist. Man sollte jedoch verstehen, dass die Erfindung nicht auf die gezeigten und beschriebenen spezifischen Merkmale beschränkt ist, da die hierin offenbarten Einrichtungen bevorzugte Formen der Umsetzung der Erfindung umfassen.

Patentansprüche

1. Ein Verfahren zum Schreiben von Benutzerdaten in ein Plattenarray gemäß redundanten Paritätsspeichertechniken, wobei das Plattenarray (**11**) mehrere Speicherplatten (**12**) aufweist und zu mehreren Streifen konfiguriert ist, bei denen sich jeder Streifen über mehrere der Speicherplatten (**12**) erstreckt und wobei jeder Streifen mehrere Segmente eines Speicherraums umfasst, in dem jedes Segment ein Abschnitt des Streifens ist, der sich auf einer einzigen Speicherplatte in dem Plattenarray befindet, wobei das Verfahren folgende Schritte umfasst:
(a) Differenzieren zwischen Segmenten, die Benutzerdaten oder Paritätsinformationen enthalten, und Segmenten, die leer sind;
(b) Identifizieren eines nicht gefüllten Streifens (**66**) in dem Plattenarray (**12**) unter Verwendung eines Teilstreifenzeigers (**62**), um auf den nicht gefüllten Streifen (**66**) Bezug zu nehmen, wobei der nicht gefüllte Streifen leere Segmente aufweist;
(c) Bereitstellen eines von dem Plattenarray (**11**) separaten nicht-flüchtigen Speichers (**21**);
(d) Lokalisieren eines leeren Segments in dem identifizierten Streifen (**66**) unter Verwendung eines Segmentfüllzeigers (**64**), um auf ein Segment, das in den Streifen geschrieben werden soll, Bezug zu nehmen, und sequentielles und nacheinander erfolgendes Bezugnehmen auf die leeren Segmente gemäß ihrer Reihenfolge in dem nicht gefüllten Streifen;
(e) Schreiben von Benutzerdaten in das leere Segment in dem Streifen, ohne Benutzerdaten in andere Segmente in dem Streifen zu schreiben;
(f) Bestimmen, nachdem die Benutzerdaten durch Schritt (e) in das Segment geschrieben wurden, eines Paritätswerts für die Benutzerdaten, die lediglich in denjenigen Segmenten in dem Streifen enthalten sind, in die geschrieben wurde, indem ein Paritätswert auf der Basis einer Logikfunktion aller Benutzerdaten in den Segmenten berechnet wird;
(g) vorübergehendes Cache-Speichern des Paritätswerts in dem nicht-flüchtigen Speicher (**21**) nach jedem der Bestimmungsschritte (f);
(h) Berechnen eines neuen Paritätswerts auf der Basis einer Logikfunktion des zuvor Cache-gespeicherten Paritätswerts und von Benutzerdaten, die zuletzt in ein leeres Segment in dem Streifen geschrieben wurden;
(i) Wiederholen der Schritte (d) bis (g), um eines oder mehrere der anderen Segmente in dem Streifen zu füllen; und
(j) Schreiben des Paritätswerts in ein Segment in dem Streifen des Plattenarrays (**11**), nachdem alle Segmente in dem Streifen, die zum Speichern von Benutzerdaten verwendet wurden, gefüllt wurden.

2. Ein Verfahren gemäß Anspruch 1, das ferner folgenden Schritt umfasst:
Speichern des Teilstreifenzeigers (**62**) und des Segmentfüllzeigers (**64**) in dem nicht-flüchtigen Speicher (**21**).

3. Ein Datenspeichersystem, das folgende Merkmale aufweist:
ein Plattenarray (**11**), das mehrere Speicherplatten (**12**) aufweist, wobei das Plattenarray (**11**) zu mehreren Streifen konfiguriert ist, bei denen sich jeder Streifen über mehrere der Speicherplatten erstreckt und wobei jeder Streifen mehrere Segmente eines Speicherraums umfasst, in dem jedes Segment ein Abschnitt des Streifens ist, der sich auf einer einzelnen Speicherplatte in dem Plattenarray (**11**) befindet;
eine Plattenarraysteuerung (**14**), die mit dem Plattenarray (**11**) gekoppelt ist, um einen Datentransfer zu und von den Speicherplatten (**12**) zu koordinieren;
einen Speicherverwalter (**16**), der wirksam mit der Plattenarraysteuerung (**14**) gekoppelt ist, um eine Speicherzuweisung zum redundanten Speichern von Benutzerdaten gemäß Paritätsredundanztechniken

zu verwalten, wobei der Speicherverwalter (**16**) Benutzerdaten in Datensegmente und Paritätsinformationen in Paritätssegmente plaziert, wodurch zumindest ein Paritätssegment und ein oder mehrere Datensegmente in jedem Streifen enthalten sind, wobei der Speicherverwalter (**16**) eine Einrichtung zum Bestimmen aufweist, welche Datensegmente in die Paritätsinformationen des Streifens integriert sind, wobei die Bestimmungseinrichtung folgende Merkmale aufweist:

einen Teilstreifenzeiger (**62**), um auf einzelne Streifen in dem Plattenarray (**11**) Bezug zu nehmen;

einen Segmentfüllzeiger (**64**), um auf einzelne Segmente in dem Streifen, auf den Bezug genommen wurde, Bezug zu nehmen; und

wobei der Speicherverwalter (**16**) die Zeiger verwendet, um einen nicht gefüllten Streifen und ein Segment in diesem Streifen auszuwählen;

wobei der Speicherverwalter (**16**) einen nicht gefüllten Streifen (**66**) und ein leeres Segment in diesem Streifen auswählt;

wobei der Speicherverwalter (**16**) Benutzerdaten in das ausgewählte Segment schreibt und inkremental einen Paritätswert für den Streifen bestimmt, nachdem in jedes Segment geschrieben wurde, wobei der Paritätswert auf den Benutzerdaten beruht, die in denjenigen Segmenten des Streifens enthalten sind, in die geschrieben wurde;

einen nicht-flüchtigen Speicher (**21**), um den Teilstreifenzeiger und den Segmentfüllzeiger zu speichern, wobei der nicht-flüchtige Speicher auch den Paritätswert Cache-speichert;

wobei der Speicherverwalter (**16**) einen neuen Paritätswert auf der Basis einer Logikfunktion des zuvor Cache-gespeicherten Paritätswerts und von Benutzerdaten, die zuletzt in ein leeres Segment geschrieben wurden, berechnet; und

wobei der Speicherverwalter (**16**) den Paritätswert in ein Segment in dem Streifen des Plattenarrays (**11**) schreibt, nachdem alle Segmente in dem Streifen, die zum Speichern von Benutzerdaten verwendet wurden, gefüllt wurden.

Es folgen 7 Blatt Zeichnungen

Anhängende Zeichnungen

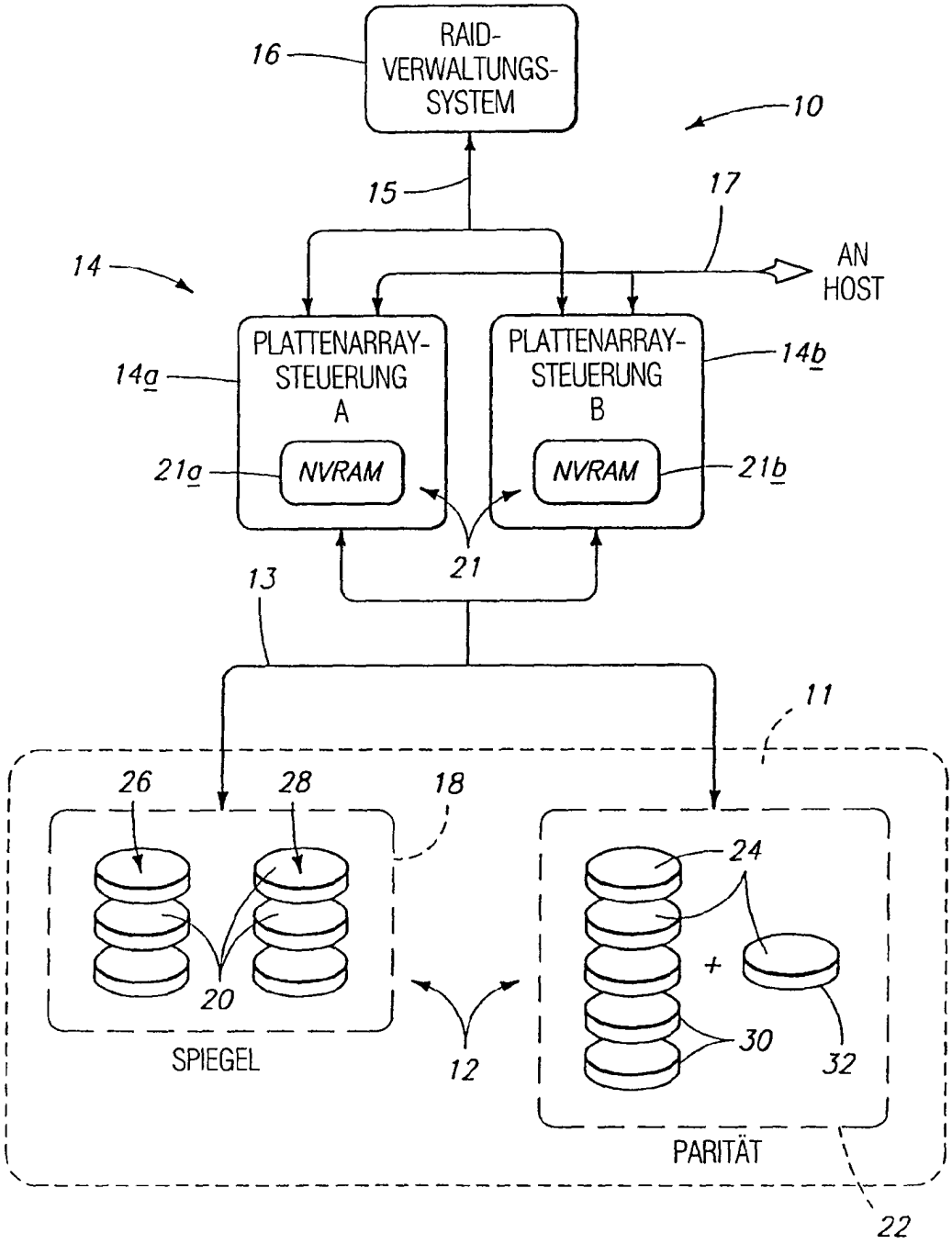

Fig 1

PLATTEN				STREIFEN-NUMMER
0	1	2	3	
0	0'	1	1'	0
2	2'	3	3'	1
4	4'	5	5'	2
6	6'	7	7'	3
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
T−1	T−1'	T	T'	S

Fig 2

PLATTEN				STREIFEN-NUMMER
0	1	2	3	
0	1	2	P	0
3	4	P	5	1
6	P	7	8	2
P	9	10	11	3
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
P	R−2	R−1	R	Q

Fig 3

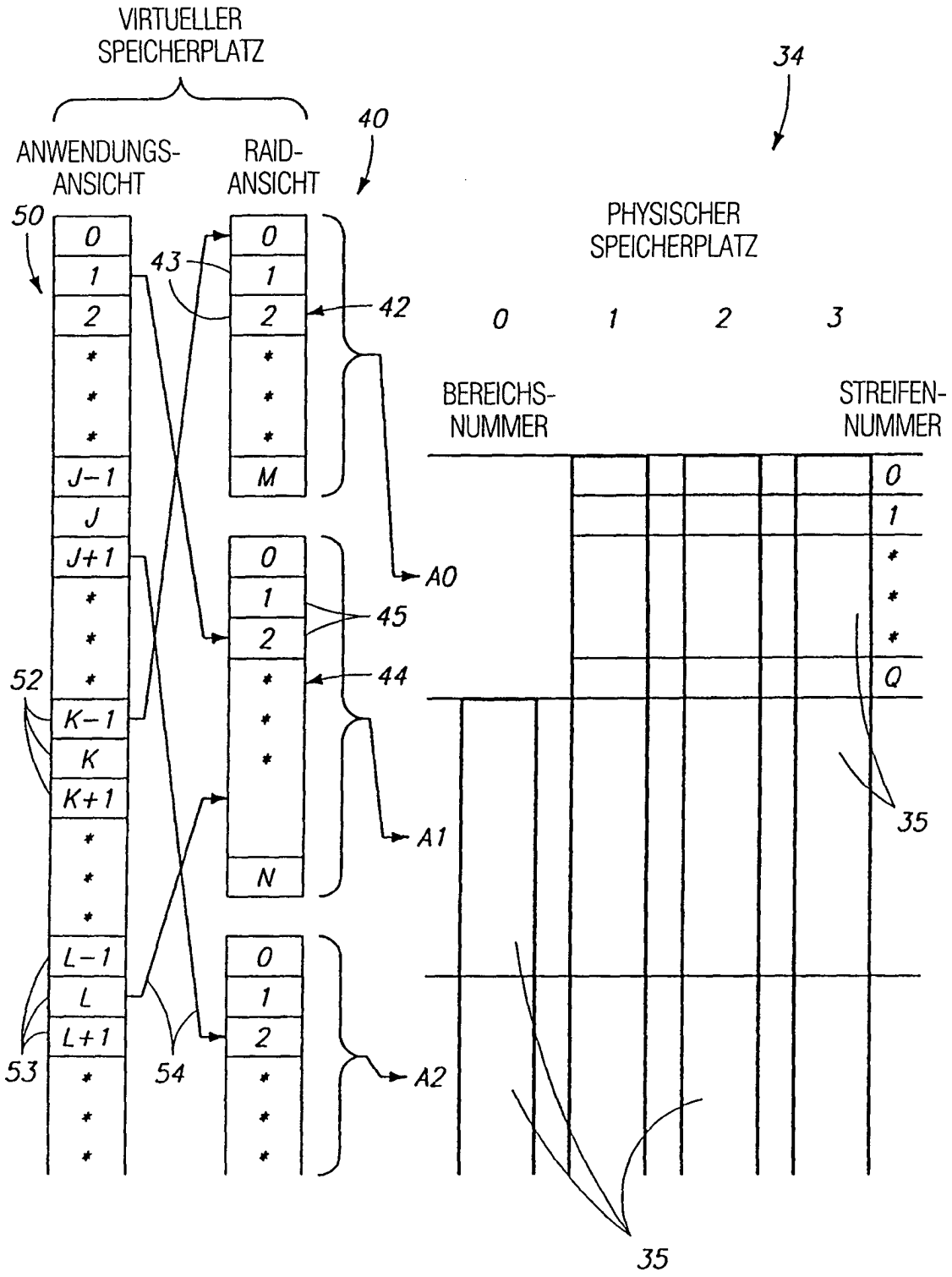
VIRTUELLER SPEICHERPLATZ
ANWENDUNGS-ANSICHT
RAID-ANSICHT
40
34
50
PHYSISCHER SPEICHERPLATZ
0
1
2
3
BEREICHS-NUMMER
STREIFEN-NUMMER
43
42
45
44
52
53
54
A0
A1
A2
35
35

Fig 4

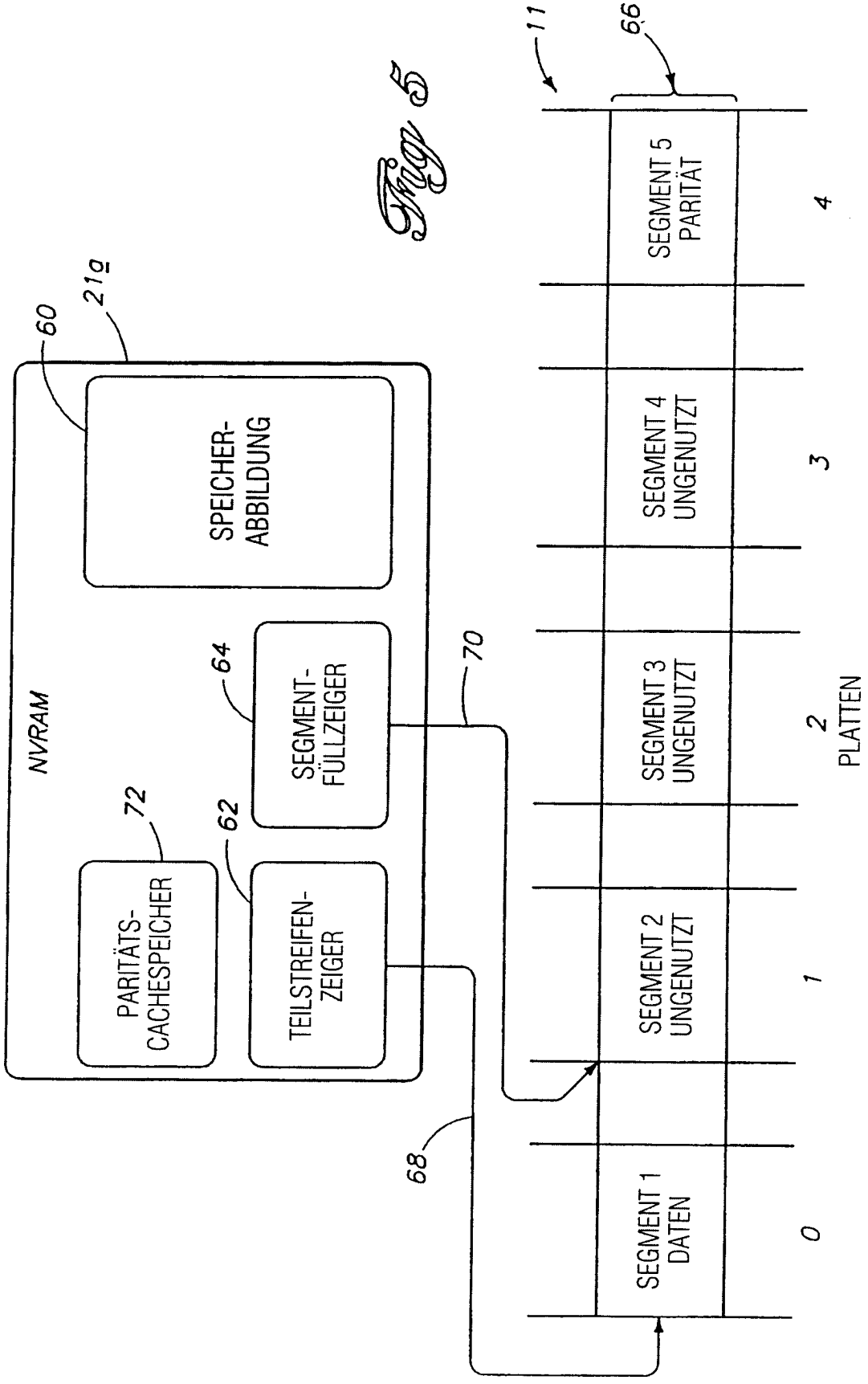
11
66
60
21g
NVRAM
SPEICHER-ABBILDUNG
64
SEGMENT-FÜLLZEIGER
70
72
PARITÄTS-CACHESPEICHER
62
TEILSTREIFEN-ZEIGER
68
SEGMENT 1 DATEN
SEGMENT 2 UNGENUTZT
SEGMENT 3 UNGENUTZT
SEGMENT 4 UNGENUTZT
SEGMENT 5 PARITÄT
0
1
2
3
4
PLATTEN

Fig 5

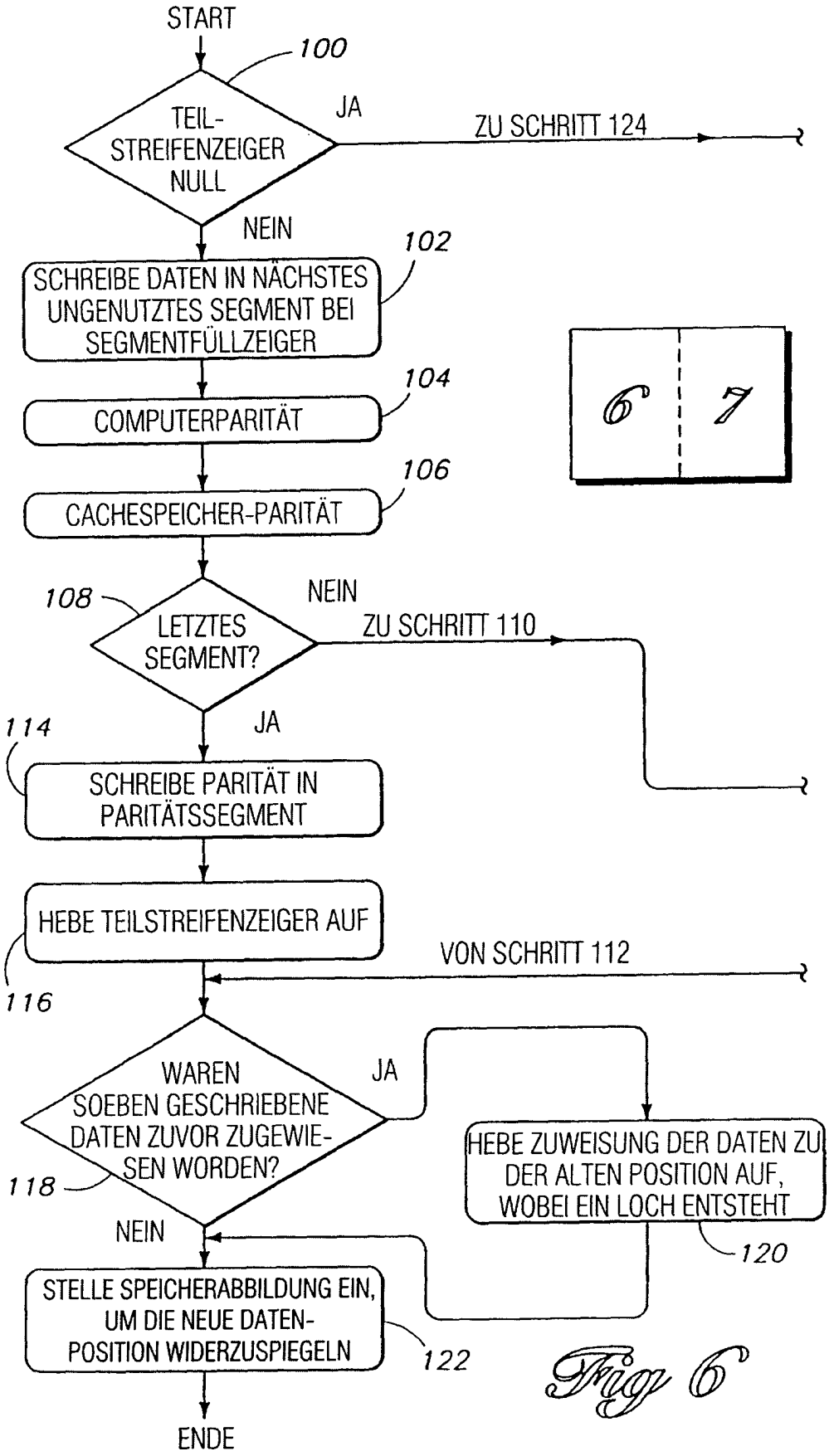
START
100
TEIL-STREIFENZEIGER NULL
JA
ZU SCHRITT 124
NEIN
102
SCHREIBE DATEN IN NÄCHSTES UNGENUTZTES SEGMENT BEI SEGMENTFÜLLZEIGER
104
COMPUTERPARITÄT
106
CACHESPEICHER-PARITÄT
108
LETZTES SEGMENT?
NEIN
ZU SCHRITT 110
JA
114
SCHREIBE PARITÄT IN PARITÄTSSEGMENT
HEBE TEILSTREIFENZEIGER AUF
116
VON SCHRITT 112
WAREN SOEBEN GESCHRIEBENE DATEN ZUVOR ZUGEWIESEN WORDEN?
JA
118
HEBE ZUWEISUNG DER DATEN ZU DER ALTEN POSITION AUF, WOBEI EIN LOCH ENTSTEHT
120
NEIN
STELLE SPEICHERABBILDUNG EIN, UM DIE NEUE DATEN-POSITION WIDERZUSPIEGELN
122
ENDE

Fig 6

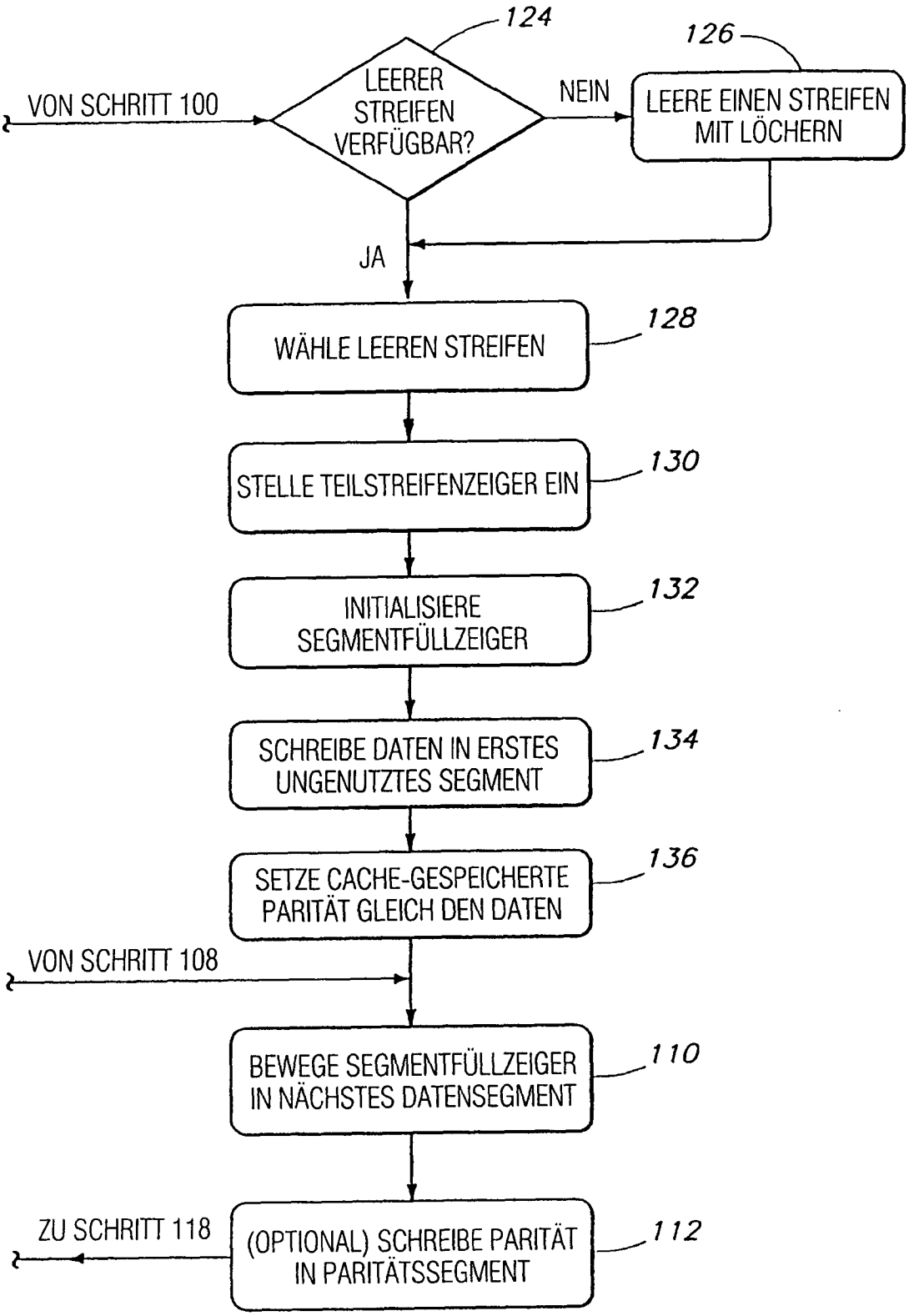
124
126
VON SCHRITT 100
LEERER STREIFEN VERFÜGBAR?
NEIN
LEERE EINEN STREIFEN MIT LÖCHERN
JA
WÄHLE LEEREN STREIFEN
128
STELLE TEILSTREIFENZEIGER EIN
130
INITIALISIERE SEGMENTFÜLLZEIGER
132
SCHREIBE DATEN IN ERSTES UNGENUTZTES SEGMENT
134
SETZE CACHE-GESPEICHERTE PARITÄT GLEICH DEN DATEN
136
VON SCHRITT 108
BEWEGE SEGMENTFÜLLZEIGER IN NÄCHSTES DATENSEGMENT
110
ZU SCHRITT 118
(OPTIONAL) SCHREIBE PARITÄT IN PARITÄTSSEGMENT
112

Fig 7

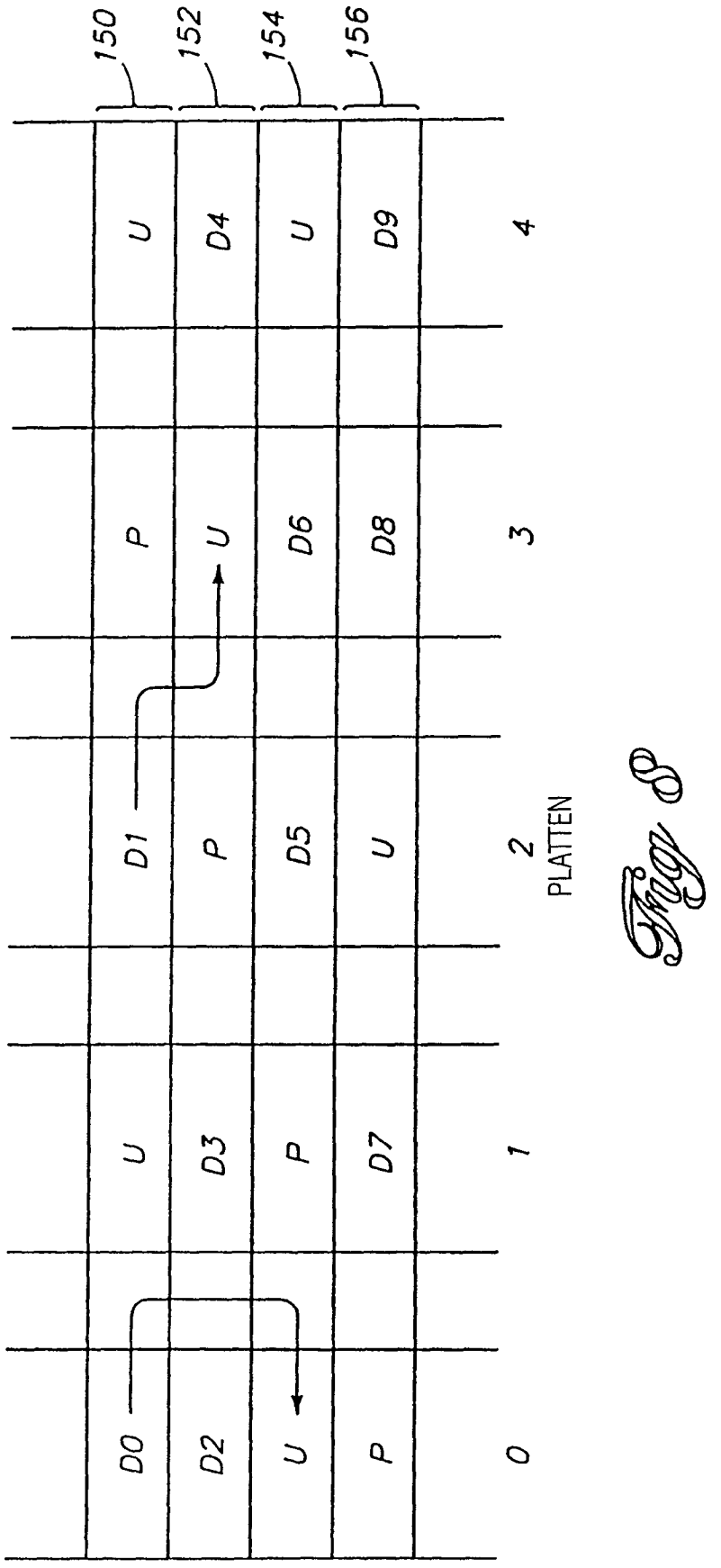
150
152
154
156
U
D4
U
D9
4
P
U
D6
D8
3
D1
P
D5
U
2
PLATTEN
U
D3
P
D7
1
D0
D2
U
P
0

Fig 8