(19) Deutsches Patent- und Markenamt

(10) **DE 11 2020 002 792 B4** 2023.05.04

(12) # Patentschrift

(21) Deutsches Aktenzeichen: **11 2020 002 792.9**
(86) PCT-Aktenzeichen: **PCT/IB2020/055061**
(87) PCT-Veröffentlichungs-Nr.: **WO 2020/250071**
(86) PCT-Anmeldetag: **28.05.2020**
(87) PCT-Veröffentlichungstag: **17.12.2020**
(43) Veröffentlichungstag der PCT Anmeldung in deutscher Übersetzung: **31.03.2022**
(45) Veröffentlichungstag der Patenterteilung: **04.05.2023**

(51) Int Cl.: ***G06F 12/16*** (2006.01)
G06F 3/06 (2006.01)

Innerhalb von neun Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

(30) Unionspriorität:
16/439,122 **12.06.2019** **US**

(73) Patentinhaber:
International Business Machines Corporation, Armonk, NY, US

(74) Vertreter:
LifeTech IP Spies & Behrndt Patentanwälte PartG mbB, 80687 München, DE

(72) Erfinder:
Tomic, Sasa, San Jose, CA, US; Pletka, Roman Alexander, Rueschlikon, CH; Ioannou, Nikolas, Rueschlikon, CH; Papandreou, Nikolaos, Rueschlikon, CH; Fry, Aaron Daniel, Houston, TX, US; Fisher, Timothy, Houston, TX, US; Stoica, Radu Ioan, Rueschlikon, CH

(56) Ermittelter Stand der Technik:

DE	**10 2017 128 941**	**A1**
US	**2012 / 0 240 012**	**A1**
US	**2013 / 0 124 787**	**A1**

(54) Bezeichnung: **VERSCHLEISSORIENTIERTE BLOCKMODUSUMWANDLUNG IN NICHTFLÜCHTIGEN SPEICHERN**

(57) Hauptanspruch: Verfahren zum Betreiben eines nichtflüchtigen Speichers, der eine Mehrzahl von physischen Speicherblöcken (204) aufweist, die jeweils eine entsprechende Mehrzahl von Zellen aufweisen, wobei jede der Mehrzahl von Zellen einzeln in der Lage ist, mehrere Daten-Bits zu speichern, wobei das Verfahren aufweist:
Zuweisen durch eine Steuereinheit (120) von physischen Blöcken aus der Mehrzahl von physischen Blöcken (204) zu einem ersten Pool (400), der physische Blöcke (204) aufweist, die in einem ersten Modus arbeiten, um eine größere Anzahl von Bits pro Zelle zu speichern, und Zuweisen von sonstigen physischen Blöcken (204) aus der Mehrzahl von physischen Blöcken zu einem zweiten Pool (402), der physische Blöcke aufweist, die in einem zweiten Modus arbeiten, um eine geringere Anzahl von Bits pro Zelle zu speichern; und
Übertragen von physischen Blöcken (204) zwischen dem ersten Pool (400) und dem zweiten Pool (402) durch die Steuereinheit (120) zumindest auf Grundlage von Bit-Fehlerraten, die für die übertragenen physischen Blöcke (204) gemessen werden.



Beschreibung

TECHNISCHES GEBIET

[0001] Die vorliegende Erfindung bezieht sich auf eine Datenverarbeitung und Datenspeicherung und im Besonderen auf ein nichtflüchtiges Speichersystem, in dem Blöcke in der Lage sind, in mehreren Blockmodi zu arbeiten, die verschiedene Bit-Speicherkapazitäten pro Zelle bereitstellen. Noch genauer gesagt, bezieht sich die Erfindung auf eine verschleißorientierte Umwandlung von Blöcken eines physischen Speichers in einem nichtflüchtigen Speichersystem zwischen verschiedenen Blockmodi.

HINTERGRUND

[0002] Bei einem NAND-Flash-Speicher handelt es sich um eine elektrisch programmierbare und löschbare nichtflüchtige Speichertechnologie, die ein oder mehrere Daten-Bits pro Speicherzelle als Ladung auf dem Floating-Gate eines Transistors oder einer ähnlichen Ladungsfallenstruktur speichert. Die Menge der Ladung auf dem Floating-Gate moduliert die Schwellenspannung des Transistors. Durch Anlegen einer geeigneten Lesespannung und Messen der Strommenge kann die programmierte Schwellenspannung der Speicherzelle ermittelt werden, und auf diese Weise kann die gespeicherte Information erkannt werden. Bei einer typischen Implementierung ist ein NAND-Flash-Speicher-Array in Blöcken (auch als „Löschblöcke“ bezeichnet) eines physischen Speichers organisiert, von denen jeder mehrere physische Seiten enthält, die wiederum eine Mehrzahl von Speicherzellen enthalten. Aufgrund der Anordnung der Wort- und Bit-Leitungen, die zum Zugreifen auf die Speicherzellen genutzt werden, sind Flash-Speicher-Arrays im Allgemeinen auf Grundlage von Seiten programmiert, jedoch auf Grundlage von Blöcken gelöscht worden.

[0003] Die Flash-Speichertechnologie hat im letzten Jahrzehnt ein beeindruckendes Wachstum der Bit-Speicherdichte erlebt, was auf das Ziel zurückzuführen ist, den Preis pro Gigabyte Flash-Speicher zu senken. Einer der Hauptgründe für diese Entwicklung ist die stetige Zunahme der Anzahl von pro Flash-Zelle gespeicherten Bits. Beispielsweise ist die Anzahl der Bits, die pro Flash-Zelle gespeichert werden kann, stetig gestiegen, von einem (Single Level Cell (SLC), Ein-Bit-Zelle) über zwei (Multi-Level Cell (MLC), Mehr-Bit-Zelle) und drei (Three Level Cell (TLC), Drei-Bit-Zelle) bis in der jüngsten Zeit hin zu vier (Quad Level Cell (QLC), Vier-Bit-Zelle). Die gestiegene Anzahl von Bits, die pro Flash-Zelle gespeichert werden kann, geht mit zunehmender Anzahl der Bits pro Zelle mit einer geringeren Lebensdauer und Leistungsfähigkeit einher.

[0004] Auf Unternehmen abgestimmte Flash-Steuereinheiten setzen eine Vielfalt von Techniken ein, um die Beschränkungen von Lebensdauer und Leistungsfähigkeit neuer Generationen von NAND-Flash-Speichern zu verringern. Beispielsweise können Flash-Steuereinheiten eine oder mehrere Flash-Verwaltungstechniken einsetzen, darunter ein Implementieren fortschrittlicher Fehlerkorrekturcodes, ein Anpassen von Spannungspegeln zum Verringern einer Bit-Fehlerrate, ein Verringern eines internen Garbage-Collection-Aufwands durch Trennen von Daten auf Grundlage der Schreibwärme, ein Verwalten von Blöcken auf Grundlage von Blöcken (d.h., Zustandsklasseneinteilung) und ein Durchführen eines/einer zustandsorientierten Verschleißausgleichs und Datenplatzierung. Trotz der Steigerungen der Langlebigkeit und der Leistungsfähigkeit, die diese fortschrittlichen Flash-Verwaltungstechniken mit sich bringen, können die bestehenden Flash-Verwaltungstechniken allein bei weiter zunehmender Bit-Dichte keine akzeptable Lebensdauer der Einheiten garantieren (z.B. eine 7-Jahres-Garantie oder mehr für auf Unternehmen abgestimmte SSDs bei hoher Schreiblast). Die QLC-NAND-Flash-Technologie stellt zum Beispiel einen Anstieg von 33 % bei der Speicherdichte im Vergleich mit der TLC-NAND-Flash-Technologie bereit, weist jedoch eine um ein Mehrfaches geringere Nennnutzungsdauer als TLC-NAND-Flash (z.B. nur wenige Tausend Programmier-/Löschzyklen (program/erase, P/E) pro Block) auf.

[0005] Ein vielversprechender Weg zum Erhöhen der Lebensdauer von Einheiten ist die Nutzung der Fähigkeit von NAND-Flash-Speichern, in mehreren Modi zu arbeiten. Einige TLC-NAND-Flash-Speicher unterstützen zum Beispiel einen SLC-Modus, in dem jeweilige ausgewählte Blöcke nur ein einzelnes Bit pro Zelle speichern, wodurch die Leistungsfähigkeit und die Lebensdauer für die ausgewählten Blöcke auf Kosten der Speicherdichte erhöht werden. Im SLC-Modus konfigurierte Blöcke können ein bis zwei Größenordnungen mehr P/E-Zyklen standhalten, stellen jedoch nur ein Drittel der Speicherkapazität von im TLC-Modus konfigurierten Blöcken bereit. In solchen Systemen können häufig geschriebene Daten in SLC-Blöcken gespeichert werden, wohingegen selten geschriebene Daten in Blöcken gespeichert werden können, die im nativen TLC-Modus arbeiten.

[0006] Bestehende Flash-Steuereinheiten, die mehrere Betriebsmodi unterstützen, setzen häufig eine mehrschichtige Architektur ein, bei der eingehende Schreibvorgänge eines Benutzers zunächst in den Blöcken einer Schicht mit niedrigerer Speicherdichte (z.B. einer SLC-Schicht) gespeichert werden. Wenn der Speicherplatz in der Schicht mit niedrigerer Speicherdichte erschöpft ist (oder eine sonstige Schwellenwertbelegung erreicht), initiiert die Flash-Steuer-

einheit einen Auslagerungsprozess, der Platz auf der Schicht mit niedrigerer Speicherdichte freimacht, indem sie Daten von der Schicht mit niedrigerer Speicherdichte auf eine Schicht mit höherer Speicherdichte (z.B. eine TLC-Schicht) verschiebt, bevorzugt beginnend mit den Daten, die in Zukunft am wenigsten wahrscheinlich überschrieben werden. Die Flash-Steuereinheit verwaltet dann die Daten, die sich in der Schicht mit höherer Speicherdichte befinden, auf herkömmliche Weise, indem sie zum Beispiel regelmäßig gültige Daten aus Blöcken verlagert, die für die Garbage Collection ausgewählt worden sind, die Garbage-Collection-Blöcke löscht und anschließend die gelöschten Blöcke mit Daten, die aus der Schicht mit niedrigerer Speicherdichte ausgelagert worden sind, neu beschreibt. In herkömmlichen mehrschichtigen Speicherarchitekturen dieses Typs werden Blöcke im Allgemeinen entweder der Schicht mit niedrigerer Speicherdichte oder der Schicht mit höherer Speicherdichte statisch zugewiesen.

[0007] In diesem Zusammenhang wurden bereits Dokumente veröffentlicht: Das Dokument US 2013 / 0 124 787 A1 beschreibt ein Festkörperlaufwerk mit mindestens einer NAND-Speicherkomponente, die in Blöcken, Seiten und Zellen organisiert ist. Jede Zelle der Speicherkomponente ist geeignet, mindestens zwei Bits zu speichern. Dabei kann jeder Block der Speicherkomponente dynamisch so konfiguriert werden, dass er mindestens ein Bit pro Zelle in einer ersten Betriebsart speichert oder mindestens zwei Bits pro Zelle in einer anderen Betriebsart speichert. Weiterhin beschreibt das Dokument US 2012 / 0 240 012 A1 eine Vorrichtung und ein Verfahren zum Betrieb einer Multi-Level-Cell-(MLC)-Flash-Speicherschaltung. Dabei werden Daten werden aus einem Speicherblock einer Vielzahl von Speicherblöcken in der MLC-Fash-Speicherschaltung gelesen, wobei jeder der Vielzahl von Speicherblöcke in einem von mindestens drei Betriebsmodi arbeiten kann: einem MLC-Modus, einen Single-Level-Cell-(SLC)-Modus und einem Defekt-Modus. Und das Dokument DE 10 2017 128 941 A1 beschreibt ein Speichersystem und ein Verfahren zur prädiktiven Blockzuweisung zur effizienteren Speicherbereinigung des Speichersystems. Dabei werden verschiedene Blockzuweisungsverfahren angewendet.

KURZDARSTELLUNG

[0008] Die vorliegende Erfindung erkennt, dass die rein statische Zuweisung von Blöcken zu den verschiedenen Schichten eines mehrschichtigen Speichersystems, das Blöcke verschiedener Bit-Speicherdichten einsetzt, dazu führen kann, dass eine jeweilige Schicht vorzeitig ihre Speicherkapazität verliert oder ihre Lebensdauergrenze erreicht. Um dieses Problem zu lösen, können Blöcke des nichtflüchtigen Speichers selektiv zwischen Betriebsmodi umgewandelt werden. Eine entsprechende detaillierte Lösung ist durch die unabhängigen Ansprüche beschrieben. Weitere Ausführungsbeispiele sind durch die abhängigen Ansprüche beschrieben.

[0009] Bei zumindest einer Ausführungsform überträgt die Steuereinheit physische Blöcke mit einer höheren Bit-Fehlerrate von dem ersten Pool an den zweiten Pool und überträgt physische Blöcke mit einer niedrigeren Bit-Fehlerrate von dem zweiten Pool an den ersten Pool.

[0010] Bei zumindest einer Ausführungsform verfolgt die Steuereinheit jeweils für den ersten und den zweiten Pool die Anzahl von physischen Blöcken, die zum Speichern von Daten für jeden einer Mehrzahl von Schreibdatenströmen mit verschiedenen Schreibwärmen zugewiesen worden sind, und die Anzahl von Programmier-/Lösch(P/E)-Zyklen in jedem der Mehrzahl von Schreibdatenströmen. Zusätzlich berechnet die Steuereinheit für jeden der Mehrzahl von Schreibdatenströmen eine entsprechende Datenstrom-Verschleißmessgröße auf Grundlage der Anzahl von dem Schreibdatenstrom zugewiesenen physischen Blöcken und einer Gesamtanzahl von P/E-Zyklen der dem Schreibdatenstrom zugewiesenen physischen Blöcke. Die Steuereinheit kann anschließend physische Blöcke auf Grundlage (1) eines ausgewählten Schreibdatenstroms aus der Mehrzahl von Schreibdatenströmen, (2) der Datenstrom-Verschleißmessgröße des ausgewählten Schreibdatenstroms und (3) des Blockzustands von physischen Blöcken, die zum Speichern von Daten für den ausgewählten Schreibdatenstrom zugewiesen worden sind, zum Übertragen auswählen. Der Blockzustand der physischen Blöcke in sämtlichen der Mehrzahl von Schreibdatenströmen kann zum Beispiel aus einer Bit-Fehlerrate ermittelt werden, die kurz nach dem Programmieren der physischen Blöcke gemessen wird.

[0011] Bei zumindest einer Ausführungsform wählt die Steuereinheit physische Blöcke zum Übertragen so aus, dass, wenn der ausgewählte Schreibdatenstrom eine Datenstrom-Verschleißmessgröße aufweist, die einen größten Verschleiß unter sämtlichen der Mehrzahl von Schreibdatenströmen widerspiegelt, die Steuereinheit zum Übertragen die im besten Zustand befindlichen physischen Blöcke aus den physischen Blöcken auswählt, die dem ausgewählten Schreibdatenstrom zugewiesen worden sind, und wenn der ausgewählte Schreibdatenstrom eine Datenstrom-Verschleißmessgröße aufweist, die einen geringsten Verschleiß unter sämtlichen der Mehrzahl von Schreibdatenströmen widerspiegelt, die Steuereinheit zum Übertragen die im schlechtesten Zustand befindlichen physischen Blöcke aus den physischen Blöcken auswählt, die dem ausgewählten Schreibdatenstrom zugewiesen worden sind.

[0012] Bei zumindest einer Ausführungsform überträgt die Steuereinheit die physischen Blöcke zwischen dem ersten und dem zweiten Pool auf Grundlage einer verfügbaren Anzahl von unprogrammierten physischen Blöcken aus einem jeweiligen des ersten und des zweiten Pools, die einen Schwellenwert nicht erfüllen.

Figurenliste

Fig. 1A ist ein Übersichts-Blockschaubild einer Datenverarbeitungsumgebung gemäß einer Ausführungsform;

Fig. 18 ist ein ausführlicheres Blockschaubild einer Flash-Karte des Datenspeichersystems von **Fig. 1A**;

Fig. 2 stellt ein beispielhaftes NAND-Flash-Speichermodul gemäß einer Ausführungsform dar;

Fig. 3 ist ein Übersichtsablaufplan der Flash-Verwaltungsfunktionen und Datenstrukturen, die bei einer Flash-Verwaltung gemäß einer Ausführungsform eingesetzt werden;

Fig. 4 stellt eine beispielhafte mehrschichtige Speicherarchitektur dar, in der Pools von physischen Blöcken dazu konfiguriert sind, in verschiedenen Modi zu arbeiten, in denen verschiedene Anzahlen von Bits pro Zelle gespeichert werden;

Fig. 5 ist ein Graph einer Schreibverstärkung gegenüber einer Einheitennutzung für eine beispielhafte mehrschichtige Speicherarchitektur;

Fig. 6 ist ein Übersichtsplan eines logischen Ablaufs eines beispielhaften Prozesses zum Konfigurieren eines nichtflüchtigen Speichers gemäß einer Ausführungsform; und

Fig. 7 ist ein Übersichtsplan eines logischen Ablaufs eines beispielhaften Prozesses zum dynamischen Übertragen von Blöcken zwischen Block-Pools gemäß einer Ausführungsform.

AUSFÜHRLICHE BESCHREIBUNG

[0013] Unter Bezugnahme auf die Figuren und unter besonderer Bezugnahme auf **Fig. 1A** wird ein Übersichts-Blockschaubild einer beispielhaften Datenverarbeitungsumgebung 100 veranschaulicht, die ein Datenspeichersystem 120 mit einem nichtflüchtigen Speicher-Array enthält, wie im Folgenden beschrieben. Wie dargestellt, enthält die Datenverarbeitungsumgebung 100 einen oder mehrere Hosts wie zum Beispiel ein Prozessorsystem 102 mit einem oder mehreren Prozessoren 104, die Anweisungen und Daten verarbeiten. Ein Prozessorsystem 102 kann zusätzlich einen lokalen Speicher 106 (z.B. einen dynamischen Direktzugriffsspeicher (dynamic random access memory, DRAM) oder Festplatten) enthalten, die Programmcode, Operanden und/oder Ausführungsergebnisse der durch den/die Prozessor(en) 104 durchgeführte Verarbeitung speichern kann. Bei verschiedenen Ausführungsformen kann es sich bei einem Prozessorsystem 102 zum Beispiel um eine mobile Datenverarbeitungseinheit (wie etwa ein Smartphone oder ein Tablet), einen Laptop oder ein Desktop-Personalcomputersystem, ein Server-Computersystem (wie etwa einen aus der POWER-Serie von Servern, die bei International Business Machines Corporation erhältlich sind) oder ein Großrechner-Computersystem handeln. Bei einem Prozessorsystem 102 kann es sich darüber hinaus um ein eingebettetes Prozessorsystem handeln, das verschiedene Prozessoren wie zum Beispiel ARM, POWER, Intel x86 oder einen beliebigen sonstigen Prozessor in Kombination mit beliebigen Speicher-Caches, Speichersteuereinheiten, lokalem Speicher, E/A-Bus-Hubs usw. verwendet.

[0014] Jedes Prozessorsystem 102 enthält des Weiteren einen Ein-/Ausgabe(E/A)-Adapter 108, der direkt (d.h., ohne jegliche dazwischenliegende Einheit) oder indirekt (d.h., über zumindest eine Zwischeneinheit) mit einem Datenspeichersystem 120 über einen E/A-Kanal 110 verbunden ist. Bei einigen Ausführungsformen kann das Datenspeichersystem 120 in ein Prozessorsystem 102 integriert sein. Bei verschiedenen Ausführungsformen kann der E/A-Kanal 110 ein beliebiges oder eine Kombination von bekannten oder künftig entwickelten Datenübertragungsprotokollen einsetzen, darunter zum Beispiel Fibre Channel (FC), FC over Ethernet (FCoE), Internet Small Computer System Interface (iSCSI), InfiniBand, Transport Control Protocol/Internet Protocol (TCP/IP), Peripheral Component Interconnect Express (PCIe) usw. Über den E/A-Kanal 110 übertragene E/A-Anforderungen enthalten Leseanforderungen, durch die ein Prozessorsystem 102 Daten aus dem Datenspeichersystem 120 anfordert, und Schreibanforderungen, durch die ein Prozessorsystem 102 eine Speicherung von Daten in dem Datenspeichersystem 120 anfordert.

[0015] Wenngleich dies nicht erforderlich ist, enthält das Datenspeichersystem 120 bei der veranschaulichten Ausführungsform mehrere Schnittstellenkarten 122, durch die das Datenspeichersystem 120 E/A-Anforderungen von Hosts über die E/A-Kanäle 110 empfängt und auf diese antwortet. Jede Schnittstellenkarte 122 ist mit jeder von mehreren Steuereinheiten 124 eines redundanten Arrays kostengünstiger Festplatten (Redundant Array of Inexpensive Disks, RAID) verbunden, um eine Fehlertoleranz und einen Lastenausgleich zu erleichtern. Jede der RAID-Steuereinheiten 124 ist wiederum (z.B. durch einen PCIe-Bus) mit nichtflüchtigen Speichermedien verbunden, die in dem veranschaulichten Beispiel mehrere Flash-Karten 126 mit NAND-Flash-Speicher enthalten. Bei sonstigen Ausführungsformen

können alternative und/oder zusätzliche nichtflüchtige Speichereinheiten eingesetzt werden.

[0016] Bei der dargestellten Ausführungsform wird der Betrieb des Datenspeichersystems 120 durch redundante Systemverwaltungs-Steuereinheiten (system management controllers, SMCs) 123 verwaltet, die mit den Schnittstellenkarten 122 und den RAID-Steuereinheiten 124 verbunden sind. Bei verschiedenen Ausführungsformen kann die Systemverwaltungs-Steuereinheit 123 unter Nutzung von Hardware oder von Hardware, die Firmware und/oder Software ausführt, implementiert sein.

[0017] **Fig. 18** stellt ein ausführlicheres Blockschaubild einer beispielhaften Ausführungsform einer Flash-Karte 126 des Datenspeichersystems 120 von **Fig. 1A** dar. Die Flash-Karte 126 enthält ein Gateway 130, das als Schnittstelle zwischen der Flash-Karte 126 und den RAID-Steuereinheiten 124 dient. Das Gateway 130 ist mit einem Universalprozessor (general-purpose processor, GPP) 132 verbunden, der (z.B. durch einen Programmcode) dazu konfiguriert sein kann, eine Vorverarbeitung an Anforderungen durchzuführen, die durch das Gateway 130 empfangen werden, und/oder ein Bearbeiten der Anforderungen durch die Flash-Karte 126 zu planen. Der GPP 132 ist mit einem GPP-Speicher 134 (z.B. einem dynamischen Direktzugriffsspeicher (Dynamic Random Access Memory, DRAM)) verbunden, der Daten, die durch den GPP 132 im Verlauf seiner Verarbeitung erstellt, referenziert und/oder modifiziert werden, oder Daten, die durch das Gateway 130 fließen, die für eine oder mehrere der Flash-Steuereinheiten 140 bestimmt sind, in geeigneter Weise zwischenspeichern kann.

[0018] Das Gateway 130 ist des Weiteren mit mehreren Flash-Steuereinheiten 140 verbunden, die jeweils ein entsprechendes NAND-Flash-Speichersystem 150 steuern. Die Flash-Steuereinheiten 140 können zum Beispiel durch eine anwendungsspezifische integrierte Schaltung (Application Specific Integrated Circuit, ASIC) und/oder ein feldprogrammierbares Gate-Array (FPGA) und/oder einen Mikroprozessor implementiert sein, und jede kann einen zugehörigen Flash-Steuereinheitenspeicher 142 (z.B. einen DRAM) aufweisen. Bei Ausführungsformen, bei denen die Flash-Steuereinheiten 140 mit einem FPGA implementiert sind, kann der GPP 132 die Flash-Steuereinheiten 140 während eines Starts des Datenspeichersystems 120 programmieren und konfigurieren. Nach dem Start empfangen die Flash-Steuereinheiten 140 in einem allgemeinen Betrieb Lese- und Schreibanforderungen von dem Gateway 130, die anfordern, dass Daten, die in dem NAND-Flash-Speichersystem 150 gespeichert sind, gelesen werden und/oder dass Daten in dem NAND-Flash-Speichersystem 150 gespeichert werden. Die Flash-Steuereinheiten 140 bearbeiten diese Anforderungen zum Beispiel durch Zugreifen auf das NAND-Flash-Speichersystem 150, um die angeforderten Daten aus dem NAND-Flash-Speichersystem 150 zu lesen oder in dieses zu schreiben, oder durch Zugreifen auf einen (nicht veranschaulichten) Speicher-Cache, der dem NAND-Flash-Speichersystem 150 zugehörig ist.

[0019] Die Flash-Steuereinheiten 140 implementieren eine Flash-Umsetzungsschicht (flash translation layer, FTL), die eine Umsetzung von logischen in physische Adressen bereitstellt, um einen Zugriff auf spezifische Speicherpositionen innerhalb der NAND-Flash-Speichersysteme 150 zu ermöglichen. Im Allgemeinen enthält eine Anforderung, die durch die Flash-Steuereinheit 140 von einer Host-Einheit wie zum Beispiel einem Prozessorsystem 102 empfangen wird, die logische Blockadresse (LBA), an der auf die Daten zuzugreifen ist (an der sie zu lesen oder zu schreiben sind) und, im Falle einer Schreibanforderung, die Schreibdaten in das Datenspeichersystem 120 zu speichern sind. Die Anforderung kann darüber hinaus die Menge (oder Größe) der Daten angeben, auf die zuzugreifen ist. Sonstige Informationen können abhängig von dem Protokoll und den Funktionen, die durch das Datenspeichersystem 120 unterstützt werden, ebenfalls ausgetauscht werden. Die Flash-Umsetzungsschicht setzt von einer RAID-Steuereinheit 124 empfangene LBAs in physische Adressen um, die einer entsprechenden physischen Position in den NAND-Flash-Speichersystemen 150 zugewiesen sind. Die Flash-Steuereinheiten 140 können eine Adressumsetzung durchführen und/oder Zuordnungen zwischen logischen und physischen Adressen in einer Datenstruktur für eine logische-physische Umsetzung wie zum Beispiel eine Tabelle für eine logische-physische Umsetzung (LPT) speichern, die in geeigneter Weise in dem Flash-Steuereinheitenspeicher 142 gespeichert sein kann.

[0020] Bei verschiedenen Ausführungsformen können die NAND-Flash-Speichersysteme 150 zahlreiche Formen annehmen. Bei der in **Fig. 18** dargestellten Ausführungsform enthält jedes NAND-Flash-Speichersystem 150 mehrere (z.B. 32) einzeln adressierbare NAND-Flash-Speichereinheiten 152. In dem veranschaulichten Beispiel nehmen die Flash-Speichereinheiten 152 die Form von auf einer Leiterplatte montierten Flash-Speichermodulen an, zum Beispiel von NAND-Flash-Speichermodulen mit Mehr-Bit-Zellen (MLC), mit Drei-Bit-Zellen (TLC) oder mit Vier-Bit-Zellen (QLC). Die effektive Speicherkapazität, die durch Flash-Speichereinheiten 152 bereitgestellt wird, kann durch die Implementierung einer Datenkomprimierung zum Beispiel durch die Flash-Steuereinheiten 140 und/oder übergeordnete Steuereinheiten wie zum Beispiel die GPPs 132, die RAID-Steuereinheiten 124 oder die SMCs 123 erhöht werden.

[0021] Unter Bezugnahme auf **Fig. 2** wird ein Blockschaubild eines beispielhaften Flash-Speichermoduls 200 dargestellt, das dazu genutzt werden kann, eine beliebige der NAND-Flash-Speichereinheiten 152 von **Fig. 18** zu implementieren. Das Flash-Speichermodul 200 enthält einen oder mehrere Speicher-Chips, die jeweils zumindest ein Speicher-Array 202 implementieren, das aus einem zweidimensionalen oder dreidimensionalen Array von NAND-Flash-Speicherzellen ausgebildet ist. Wie in **Fig. 2** angegeben, sind die Speicherzellen innerhalb des Speicher-Arrays 202 physisch in mehreren physischen Blöcken 204 angeordnet, die jeweils wiederum mehrere physische Seiten 206 enthalten. Diese Seiten können in Seitengruppen verwaltet werden, die zum Beispiel jeweils aus sämtlichen der Seiten, die mit einer gemeinsamen Wortleitung verbunden sind, aus sämtlichen Seiten in einer oder mehreren Schichten in einem 3D-NAND-Flasch, aus einem Satz von Seiten in einer oder mehreren Schichten oder allgemein aus Seiten mit ähnlichen Merkmalen ausgebildet sein können.

[0022] Wie Fachleuten bekannt ist, muss ein NAND-Flash-Speicher wie zum Beispiel derjenige, der in dem Speicher-Array 202 eingesetzt wird, gelöscht werden, bevor er programmiert wird. Des Weiteren kann ein NAND-Flash-Speicher (muss jedoch nicht) durch seinen Aufbau so eingeschränkt sein, dass das kleinste Speicherelement, das gelöscht werden kann, ein Block 204 ist und das kleinste Speicherelement, auf das durch eine Lese- oder Schreibanforderung zugegriffen werden kann, auf die Größe einer einzelnen physischen Seite 206 festgelegt ist. Es versteht sich in dieser Hinsicht, dass die durch Host-Einheiten bereitgestellten LBAs logischen Seiten innerhalb eines logischen Adressraums entsprechen, in dem jede logische Seite typischerweise eine Größe von 4 Kilobyte (kB) aufweist. Physische Seiten 206 weisen demgegenüber typischerweise eine größere Größe von zum Beispiel 16 kB auf und können daher mehrere logische Seiten aufnehmen.

[0023] Das Flash-Speichermodul 200 enthält des Weiteren einen Zeilen-Decodierer 210, durch den Wortleitungen des Speicher-Arrays 202 adressiert werden können, und einen Spalten-Decodierer 212, durch den Bit-Leitungen des Speicher-Arrays 202 adressiert werden können. Darüber hinaus enthält das Flash-Speichermodul 200 eine Lese-/Schreibschaltung 214, die ermöglicht, dass die Speicherzellen einer physischen Seite 206 parallel programmiert oder gelesen werden. Die Flash-Steuereinheit 200 enthält zusätzlich eine Steuerschaltung 205, die eine Steuerung des Betriebs des Speicher-Arrays 202 auf der Chip-Ebene bereitstellt, einschließlich von Lese- und Schreibzugriffen, die auf physische Seiten 206 in dem Speicher-Array 202 erfolgen, Löschen von Blöcken 204 und der Amplitude, der Dauer und der Polarität von verwandten Spannungen, die an das Speicher-Array 202 angelegt werden.

[0024] Nachdem die allgemeine physische Struktur einer beispielhaften Ausführungsform eines Datenspeichersystems 120 beschrieben worden ist, werden im Folgenden bestimmte Betriebsaspekte des Datenspeichersystems 120 unter Bezugnahme auf **Fig. 3** beschrieben, die ein Übersichtsablaufplan der Flash-Verwaltungsfunktionen und Datenstrukturen ist, die durch einen GPP 132 und/oder eine Flash-Steuereinheit 140 gemäß einer Ausführungsform eingesetzt werden.

[0025] Das Datenspeichersystem 120 ermöglicht im Allgemeinen nicht, dass externe Einheiten (z.B. Hosts) die physischen Speicherpositionen innerhalb der NAND-Flash-Speichersysteme 150 direkt adressieren und/oder auf diese zugreifen. Stattdessen ist das Datenspeichersystem 120 allgemein dazu konfiguriert, Host-Einheiten ein oder mehrere logische Datenträger zur Verfügung zu stellen, die jeweils einen zusammenhängenden logischen Adressraum aufweisen, so dass ermöglicht wird, dass die Host-Einheiten Daten aus logischen Blockadressen (LBAs) innerhalb des logischen Adressraums lesen und in diese schreiben, während zugelassen wird, dass eine oder mehrere der verschiedenen Ebenen von Steuereinheiten (z.B. die Systemverwaltungs-Steuereinheit 123, die RAID-Steuereinheiten 124, die Flash-Steuereinheiten 140 und der GPP 132) steuern, wo sich die Daten, die den verschiedenen LBAs zugehörig sind, tatsächlich an den physischen Speicherpositionen befinden, die die NAND-Flash-Speichersysteme 150 aufweisen. Auf diese Weise können die Leistungsfähigkeit und Langlebigkeit der NAND-Flash-Speichersysteme 150 intelligent verwaltet und optimiert werden.

[0026] Bei der veranschaulichten Ausführungsform führt jede Flash-Steuereinheit 140 eine Umsetzung von logischen in physische Adressen für einen zugehörigen Satz von LBAs mithilfe einer Datenstruktur für eine Umsetzung von logischen in physische Adressen wie zum Beispiel einer Tabelle 300 für eine logische-physische Umsetzung (LPT) durch, die beispielsweise in dem zugehörigen Flash-Steuereinheitenspeicher 142 gespeichert sein kann. Es ist zu beachten, dass die logische Adresse, die an die Flash-Steuereinheit(en) 140 geliefert wird, sich von der logischen Adresse unterscheiden kann, die ursprünglich an das Datenspeichersystem 120 geliefert wurde, da verschiedene Komponenten innerhalb des Datenspeichersystems 120 Adressumsetzungsoperationen zwischen den externen Einheiten und der/den Flash-Steuereinheit(en) 140 durchführen können.

[0027] Der Flash-Verwaltungscode, der auf dem GPP 132 ausgeführt wird, verfolgt gelöschte Blöcke

des NAND-Flash-Speichersystems 150, die sich gebrauchsfertig in Gebrauchsfertigkeits(ready-to-use, RTU)-Warteschlangen 306 befinden, die zum Beispiel in dem GPP-Speicher 134 gespeichert sein können. Bei der dargestellten Ausführungsform erhält der auf dem GPP 132 ausgeführte Flash-Verwaltungscode eine RTU-Warteschlange 306 pro Kanal oder Ebene (d.h., pro Datenbus) aufrecht, und eine Kennung jedes gelöschten Blocks, der wiederzuverwenden ist, wird in die RTU-Warteschlange 306 entsprechend seinem Kanal oder seiner Ebene eingestellt. Darüber hinaus können getrennte RTU-Warteschlangen 306 für verschiedene Zustandsgrade, Wärmeklassen/Schreibdatenströme oder Blockoperationsmodi, d.h., den SLC- oder den QLC-Operationsmodus, aufrechterhalten werden. Eine Funktion 320 zum Erstellen von Blockstreifen (block stripes), die durch den Flash-Verwaltungscode durchgeführt wird, der auf dem GPP 132 ausgeführt wird, baut neue Blockstreifen zum Speichern von Daten und zugehörigen Paritätsinformationen aus den gelöschten Blöcken auf, die in die RTU-Warteschlangen 306 eingestellt sind. Die neuen Blockstreifen werden dann zur Datenplatzierung an die Flash-Steuereinheit 140 übergeben. Blockstreifen werden bevorzugt aus Blöcken ausgebildet, die sich in verschiedenen Kanälen befinden, was bedeutet, dass die Funktion 320 zum Erstellen von Blockstreifen einen Blockstreifen in geeigneter Weise aufbauen kann, indem sie jeden Block des neuen Blockstreifens aus einer anderen RTU-Warteschlange 306 zieht. Im Allgemeinen versucht die Funktion 320 zum Erstellen von Blockstreifen Streifen aus Blöcken in etwa gleichem Zustand (d.h., der erwarteten verbleibenden Nutzungsdauer) aufzubauen.

[0028] Als Reaktion auf eine Schreibanforderung, die von einem Host wie zum Beispiel einem Prozessorsystem 102 empfangen wird, ermittelt eine Datenplatzierungsfunktion 310 der Flash-Steuereinheit 140 unter Bezugnahme auf die LPT-Tabelle 300, ob die in der Schreibanforderung angegebene(n) Ziel-LBA(s) aktuell (einer) physischen Speicherseite(n) in dem NAND-Flash-Speichersystem 150 zugeordnet ist/sind, und ändert, wenn dies der Fall ist, den Status jeder Datenseite, die aktuell einer Ziel-LBA zugehörig ist, um anzugeben, dass die zugehörigen Daten nicht mehr gültig sind. Darüber hinaus weist die Datenplatzierungsfunktion 310 gegebenenfalls einen Seitenstreifen (page stripe) zu, um die Schreibdaten der Schreibanforderung und alle nicht aktualisierten Daten (d.h., bei Schreibanforderungen, die kleiner als eine logische Seite sind, die verbleibenden gültigen Daten aus einem vorherigen Schreibvorgang an dieselbe logische Adresse, die nicht überschrieben werden und die in einer Lese-Änderungs-Schreib-Weise verarbeitet werden müssen) aus einem vorhandenen Seitenstreifen, falls vorhanden, auf den die Schreibanforderung abzielt, zu speichern, und/oder speichert die Schreibdaten der Schreibanforderung und alle nicht aktualisierten (d.h., noch gültigen) Daten aus einem vorhandenen Seitenstreifen, falls vorhanden, auf den die Schreibanforderung abzielt, in einem bereits zugewiesenen Seitenstreifen, der noch freien Platz aufweist. Der Seitenstreifen kann entweder aus einem Blockstreifen, der bereits zum Speichern von Daten zugewiesen ist, oder aus einem neuen Blockstreifen zugewiesen werden, der durch die Funktion 320 zum Erstellen von Blockstreifen erstellt worden ist. Bei einer bevorzugten Ausführungsform kann die Zuweisung von Seitenstreifen auf dem Zustand der zur Zuweisung verfügbaren Blöcke und der „Wärme“ (d.h., der geschätzten oder gemessenen Schreibzugriffshäufigkeit) der LBA der Schreibdaten beruhen. Anschließend schreibt die Datenplatzierungsfunktion 310 die Schreibdaten, zugehörige Metadaten (z.B. Werte von zyklischen Redundanzcodes (cyclic redundancy code, CRC) und Fehlerkorrekturcodes (error correcting code, ECC)) und Paritätsinformationen für den Seitenstreifen in den zugewiesenen Seitenstreifen. Die Flash-Steuereinheit 140 aktualisiert darüber hinaus die LPT-Tabelle 300 so, dass die physische(n) Seite(n), die dazu genutzt wird/werden, die Schreibdaten zu speichern, der/den LBA(s) zugeordnet wird/werden, die durch die Host-Einheit angegeben wird/werden. Anschließend kann die Flash-Steuereinheit 140 auf die Daten zugreifen, um Host-Leseanforderungen durch Bezugnahme auf die LPT-Tabelle 300 zu bearbeiten, wie des Weiteren in **Fig. 3** veranschaulicht.

[0029] Nachdem alle Seiten in einem Blockstreifen geschrieben worden sind, platziert die Flash-Steuereinheit 140 den Blockstreifen in einer der belegten Blockwarteschlangen 302, die der Flash-Verwaltungscode, der auf dem GPP 132 ausgeführt wird, nutzt, um die Garbage Collection zu vereinfachen. Wie oben angemerkt, werden Seiten durch den Schreibprozess ungültig gemacht, und daher werden Abschnitte des NAND-Flash-Speichersystems 150 frei. Die zugehörige Flash-Steuereinheit 140 (und/oder der GPP 132) muss diesen Platz letztendlich durch eine Garbage Collection zurückfordern, die durch einen Garbage Collector 312 durchgeführt wird. Der Garbage Collector 312 wählt bestimmte Blockstreifen für eine Garbage Collection auf Grundlage einer Anzahl von Faktoren aus, darunter zum Beispiel des Zustands des Blocks 204 innerhalb der Blockstreifen und wie viele der Daten innerhalb der Löschblöcke 204 ungültig sind. In dem veranschaulichten Beispiel wird eine Garbage Collection an vollständigen Blockstreifen durchgeführt, und ein Flash-Verwaltungscode, der auf dem GPP 132 ausgeführt wird, protokolliert die Blockstreifen, die zur Wiederverwendung bereit sind, in einer Verlagerungswarteschlange 304, die in geeigneter Weise in dem zugehörigen Flash-Steuereinheitenspeicher 142 oder dem GPP-Speicher 134 implementiert sein kann.

[0030] Zu den Flash-Verwaltungsfunktionen, die durch den GPP 132 oder die Flash-Steuereinheit 140 durchgeführt werden, zählt darüber hinaus eine Verlagerungsfunktion 314, die die Daten verlagert, die in Blockstreifen gespeichert sind, die in die Verlagerungswarteschlange 304 eingestellt sind. Um solche Daten zu verlagern, gibt die Verlagerungsfunktion 314 Verlagerungsschreibanforderungen an die Datenplatzierungsfunktion 310 aus, um anzufordern, dass die gültigen Daten des alten Blockstreifens in einen neuen Blockstreifen in dem NAND-Flash-Speichersystem 150 geschrieben werden. Darüber hinaus aktualisiert die Verlagerungsfunktion 314 die LPT-Tabelle 300, um die aktuelle Zugehörigkeit zwischen den logischen und den physischen Adressen der Daten zu entfernen. Nachdem alle verbleibenden gültigen Daten aus dem alten Blockstreifen verschoben worden sind, wird der alte Blockstreifen an eine Funktion 316 zum Auflösen von Blockstreifen übergeben, die den alten Blockstreifen in seine einzelnen Blöcke 204 zerlegt, wodurch die Zuordnung der Blöcke 204 aufgehoben wird. Jeder der Blöcke 204, die zuvor den aufgelösten Blockstreifen ausgebildet hatten, wird anschließend unter der Steuerung der Flash-Steuereinheit 140 und/oder der Steuerschaltung 205 des entsprechenden Flash-Speichermoduls 200 gelöscht, und eine entsprechende Programmier-/Lösch(P/E)-Zyklusanzahl 334 für jeden gelöschten Block wird inkrementiert. Auf Grundlage der Zustandsmessgrößen 332 jedes gelöschten Blocks 204 (z.B. Bit-Fehlerraten(bit error rate, BER)-Messgrößen, unkorrigierbare Fehler, P/E-Zyklusanzahl usw.) wird jeder gelöschte Block 204 entweder durch eine Blockzurückziehungsfunktion 318 aus den Flash-Verwaltungsfunktionen, die auf dem GPP 132 ausgeführt werden, zurückgezogen (d.h., aus der Verwendung genommen) oder alternativ zur Wiederverwendung vorbereitet, indem eine Kennung des Blocks 204 in der entsprechenden Gebrauchsfertigkeits(RTU)-Warteschlange 306 in dem zugehörigen GPP-Speicher 134 platziert wird.

[0031] Wie des Weiteren in **Fig. 3** dargestellt, enthalten die Flash-Verwaltungsfunktionen, die auf dem GPP 132 und/oder der Flash-Steuereinheit 140 ausgeführt werden, darüber hinaus ein Hintergrund-Zustandsprüfprogramm 330. Das Hintergrund-Zustandsprüfprogramm 330, das unabhängig von den Lese- und Schreibanforderungen von Hosts wie zum Beispiel dem Prozessorsystem 102 arbeitet, ermittelt fortlaufend eine oder mehrere Blockzustands-Messgrößen 332 für Blöcke, die zu Blockstreifen gehören, die in belegten Blockwarteschlangen 302 aufgezeichnet sind. Auf Grundlage der einen oder mehreren Blockzustands-Messgrößen 332 platziert das Hintergrund-Zustandsprüfprogramm 330 Blockstreifen in der Verlagerungswarteschlange 304 zum Verarbeiten durch die Verlagerungsfunktion 314. Die wichtigsten Blockzustands-Messgrößen 332, die bevorzugt durch das Hintergrund-Zustandsprüfprogramm überwacht und aufgezeichnet werden, beziehen sich auf die Bit-Fehlerraten(BER)-Messgrößen, die für gültige Blöcke und physische Seiten beobachtet werden, und können zum Beispiel die schlechteste Seiten-BER jedes Blocks, die mittlere Seiten-BER jedes Blocks, die Änderungsraten der schlechtesten Seiten-BER und der mittleren Seiten-BER jedes Blocks usw. enthalten. Um die genaueste Einschätzung des Zustands zu gewinnen, kann der Zustand anhand einer Analyse von gültigen und ungültigen Daten ermittelt werden, wodurch sichergestellt wird, dass Blöcke, die fast ausschließlich ungültige Daten enthalten, vollständig analysiert werden, um den Gesamtzustand des Blocks zu ermitteln.

[0032] Eine Funktion des GPP 132 und/oder der Flash-Steuereinheit 140, die in das Hintergrund-Zustandsprüfprogramm 330 integriert sein kann oder die getrennt implementiert sein kann, ist die regelmäßige Anpassung (Kalibrierung) von Lesespannungs-Schwellenwerten 336, die zum Decodieren der in den Speicherzellen der Speicher-Arrays 202 gespeicherten Daten-Bits genutzt werden. Diese Lesespannungs-Schwellenwerte 336, die einzeln auf jeder gewünschten Granularitätsebene (z.B. pro Seite, pro Seitengruppe innerhalb eines Blocks, pro Block usw.) definiert sein können, werden bevorzugt so ausgewählt, dass sie eine oder mehrere Bit-Fehlermessgrößen verbessern. Wie des Weiteren in **Fig. 3** dargestellt, verfolgen der GPP 132 und/oder die Flash-Steuereinheit 140 bevorzugt zusätzliche Statistiken, um eine intelligente Anpassung der Lesespannungs-Schwellenwerte 336 zu vereinfachen. Zu diesen zusätzlichen Statistiken können zum Beispiel die P/E-Zyklusanzahlen 334, die eine Anzahl von Programmier-/Löschzyklen angeben, denen jeder Block unterzogen worden ist, und Anzahlen von Lesevorgängen 338 zählen, die angeben, wie oft eine jeweilige Speichereinheit (z.B. eine Seite, eine Seitengruppe und/oder ein Block) seit ihrer Programmierung gelesen worden ist. Darüber hinaus können diese Statistiken Kalibrierungsauslöser 340 enthalten, z.B. Zähler pro Speichereinheit, die eine verstrichene Zeitspanne angeben, seitdem Lesespannungs-Schwellenwerte dieser Speichereinheit angepasst wurden, und/oder Zähler pro Block, die eine Anzahl von Programmier-/Löschzyklen angeben, denen jeder Block unterzogen worden ist, seit die Lesespannungs-Schwellenwerte für diesen Block zuletzt angepasst wurden.

[0033] Unter Bezugnahme auf **Fig. 4** wird eine beispielhafte Speicherarchitektur dargestellt, die in einer nichtflüchtigen Speichereinheit implementiert sein kann. In diesem Beispiel konfiguriert ein GPP 132 und/oder eine Flash-Steuereinheit 140 (die hierin allgemein als „Steuereinheit“ bezeichnet wird) einige oder sämtliche der Flash-Speichereinheiten 152 in einem NAND-Flash-Speichersystem 150, um

mehrere Pools von physischen Blöcken 204 zu implementieren, wobei die physischen Blöcke 204 in den verschiedenen Block-Pools dazu konfiguriert sind, in verschiedenen Modi zu arbeiten, in denen verschiedene Anzahlen von Bits pro Zelle gespeichert werden. Obwohl ein gesamtes Speicher-Array 202 zum Beispiel einen QLC-Flash-Speicher aufweist, der in der Lage ist, vier Bits pro Zelle zu speichern, weist eine Steuereinheit daher in diesem Beispiel einige der physischen Blöcke 204 in dem Speicher-Array 202 einer Flash-Speichereinheit 152 einem QLC-Pool 400 zu, in dem vollständig programmierte Blöcke vier Bits pro Zelle speichern, und weist sonstige physische Blöcke 204 des Speicher-Arrays 202 einem SLC-Pool 402 zu, in dem vollständig programmierte Blöcke nur ein einzelnes Bit pro Zelle speichern. Informationen über eine Pool-Zugehörigkeit können zum Beispiel in dem GPP-Speicher 124 und/oder in dem Flash-Steuereinheitenspeicher 142 als Blockmodus-Metadaten 335 gespeichert werden. Wie oben angemerkt, kann ein Betreiben von Blöcken in mehreren verschiedenen Modi auf diese Weise eine verbesserte Leistungsfähigkeit und/oder Lebensdauer bereitstellen, wenn häufig geschriebene LBAs physischen Blöcken 204 innerhalb des SLC-Pools 402 zugeordnet werden.

[0034] Wenngleich dies nicht erforderlich ist, organisiert die Steuereinheit bei einigen Ausführungsformen den QLC-Pool 400 und den SLC-Pool 402 in einer mehrschichtigen Anordnung, bei der die in Host-Schreibanforderungen enthaltenen Daten zunächst in physische Blöcke 204 geschrieben werden, die aus dem SLC-Pool 402 ausgewählt werden, wie durch Pfeil 404 angegeben. Wenn der SLC-Pool 402 durch die Verwendung belegt wird und mehr Kapazität in dem SLC-Pool 402 benötigt und/oder gewünscht wird, überträgt die Steuereinheit Daten durch Verlagerungsschreibvorgänge aus dem SLC-Pool 402 in den QLC-Pool 400, wie durch Pfeil 406 angegeben. Um eine angemessene Belieferung mit physischen Blöcken 204 in den beiden Block-Pools 400 und 402 zu gewährleisten, während die Blöcke einem Verschleiß ausgesetzt sind, überträgt die Steuereinheit zusätzlich Blöcke 204 zwischen den Pools 400 und 402 (weist diese neu zu) nach Bedarf und/oder Wunsch, um Speicherkapazität und/oder Lebensdauer bereitzustellen. Die Übertragung von physischen Blöcken 204 (im Gegensatz zu Daten) zwischen den Block-Pools 400 und 402 wird in **Fig. 4** durch Pfeil 408 dargestellt.

[0035] Die vorliegende Offenbarung erkennt an, dass der Prozess zum Übertragen von Blöcken zwischen Block-Pools 400 und 402, wenn er nicht sorgfältig konzipiert wird, in unerwünschter Weise zu einer verringerten Leistungsfähigkeit (aufgrund erhöhter Fehlerbehebungsaktivitäten) oder sogar zu Datenverlust in Fällen führen kann, in denen die Fehlerbehebung nicht erfolgreich ist (z.B. wenn Blöcke mit einer hohen Bit-Fehlerrate von dem Block-Pool 402 an den Block-Pool 400 übertragen werden). Im Allgemeinen überträgt die Steuereinheit Blöcke zwischen den Block-Pools 400 und 402, indem sie in schlechterem/schlechtesten Zustand befindliche Blöcke aus dem Block-Pool 400 dazu auswählt, an den Block-Pool 402 übertragen zu werden, und indem sie in besserem/besten Zustand befindliche Blöcke aus dem Block-Pool 402 dazu auswählt, an den Block-Pool 400 übertragen zu werden.

[0036] Die vorliegende Offenbarung erkennt an, dass diese allgemeine Modusumwandlungsstrategie, wenngleich sie für einige Systeme geeignet ist, in Systemen suboptimal ist, die eine Wärmetrennung einsetzen oder eine sehr hohe Einheitennutzung aufweisen, was direkt zu einem beschleunigten Verschleiß und sogar zu einer vorzeitigen Außerbetriebnahme der Einheiten führen kann. Ein Grund für diese negativen Auswirkungen besteht darin, dass der Verschleiß (wie zum Beispiel durch durchschnittliche P/E-Zyklen gemessen) der Blöcke 204 in dem QLC-Pool 400 unter bestimmten Bedingungen zum Teil aufgrund der Schreibverstärkung im Allgemeinen erheblich höher sein kann als der Verschleiß der Blöcke 204 in dem SLC-Pool 402. **Fig. 5** ist ein Graph einer Schreibverstärkung gegenüber einer Einheitennutzung für eine beispielhafte mehrschichtige Speicherarchitektur, wie in **Fig. 4** dargestellt. Wie dargestellt, kann bei hoher Einheitennutzung (z.B. wenn über 90 % einer Flash-Speichereinheit mit Daten beschrieben werden) die Schreibverstärkung des QLC-Pools 400 exponentiell zunehmen, was zu einem immer schnelleren Verschleiß von Blöcken in dem QLC-Pool 400 führt. Wie im Folgenden ausführlicher erörtert wird, berücksichtigt die Modusumwandlungsstrategie daher bevorzugt den Blockverschleiß, wenn Blöcke ausgewählt werden, die zwischen Block-Pools zu übertragen sind.

[0037] Unter Bezugnahme auf **Fig. 6** wird ein Übersichtsplan eines logischen Ablaufs eines beispielhaften Prozesses zum Konfigurieren eines nichtflüchtigen Speichers gemäß einer Ausführungsform dargestellt. Der veranschaulichte Prozess kann zum Beispiel durch eine Steuereinheit (z.B. den GPP 132 und/oder die Flash-Steuereinheit 140) in Hardware, Firmware, Software oder einer Kombination von diesen beim Start des Datenspeichersystems 120 durchgeführt werden. Wie bei den sonstigen hierin dargestellten logischen Ablaufplänen werden Operationen in logischer statt streng chronologischer Reihenfolge dargestellt, und bei einigen Ausführungsformen können Operationen in einer anderen Reihenfolge als der dargestellten oder gleichzeitig durchgeführt werden.

[0038] Der Konfigurationsprozess von **Fig. 6** beginnt in Block 600 und geht anschließend zu Block 602 über, der eine Steuereinheit einer Flash-

Speichereinheit 152 veranschaulicht, die eine gewünschte Anzahl von Block-Pools konfiguriert. In dem Beispiel von **Fig. 4** implementiert die Steuereinheit zwei Block-Pools in Block 602, und zwar den QLC-Block-Pool 400 und den SLC-Block-Pool 402. Bei sonstigen Ausführungsformen kann eine größere Anzahl von Block-Pools und/oder Block-Pools, die in verschiedenen Blockmodi (z.B. TLC oder MLC) arbeiten, implementiert werden. Die Steuereinheit weist darüber hinaus jedem Block-Pool eine anfängliche Anzahl von physischen Blöcken 204 zu und zeichnet auf Grundlage der Block-Pool-Zugehörigkeit jedes physischen Blocks 204 den entsprechenden Blockmodus jedes physischen Blocks 204 in Blockmodus-Metadaten 335 auf.

[0039] Der Prozess geht von Block 602 zu Block 604 über, der veranschaulicht, dass die Steuereinheit in eine Verarbeitungsschleife eintritt, in der jeder der in Block 602 implementierten Block-Pools verarbeitet wird. Wenn die Steuereinheit in Block 604 ermittelt, dass noch nicht alle Block-Pools verarbeitet worden sind, geht der Prozess zu Block 610 über, der darstellt, dass die Steuereinheit einen nächsten Block-Pool (z.B. den QLC-Pool 400 oder den SLC-Pool 402) zum Verarbeiten auswählt. In Block 612 konfiguriert die Steuereinheit anschließend eine gewünschte Anzahl von Zustandsgraden für die Blöcke 204, die zu dem ausgewählten Block-Pool gehören. Bei verschiedenen Ausführungsformen kann die Anzahl von Zustandsgraden variieren, jedoch hat sich in Versuchen eine Anzahl von Zustandsgraden zwischen zwei und einschließlich acht als wirksam erwiesen. Beispielsweise konfiguriert die Steuereinheit bei einer in **Fig. 4** dargestellten Ausführungsform vier Zustandsgrade 410a bis 410d oder 412a bis 412d in Block 612. Die hierin eingesetzten Zustandsgrade sind ausdrücklich so definiert, dass sie auf zumindest einem anderen Kriterium als nur der Anzahl von Programmier-/Löschzyklen (P/E-Zyklen) beruhen, zum Beispiel auf einer oder mehreren Bit-Fehlermessgrößen für jeden Block (und/oder den physischen Seiten innerhalb jedes Blocks) des physischen Speichers. Bei einer Ausführungsform werden die verschiedenen Zustandsgrade beispielsweise durch jeweilige Bereiche der schlechtesten Seiten-BER definiert, die innerhalb einer geringen Anzahl von Lesezugriffen im Anschluss an die Programmierung ermittelt werden, da diese Bit-Fehlermessgröße die dauerhaften Auswirkungen des Verschleißes auf die Blöcke genau widerspiegelt. Auf Grundlage der gemessenen schlechtesten Seiten-BER sind die im besten Zustand befindlichen Blöcke diejenigen mit der niedrigsten schlechtesten Seiten-BER kurz nach der Programmierung, und die im schlechtesten Zustand befindlichen Blöcke sind diejenigen mit der höchsten schlechtesten Seiten-BER kurz nach der Programmierung.

[0040] In Block 614 konfiguriert die Steuereinheit zusätzlich eine gewünschte Anzahl von Wärmeklassen für LBAs, die durch die Flash-Speichereinheit 152 bearbeitet werden. Jede Wärmeklasse entspricht einem bestimmten Bereich der Schreibwärme (d.h., der Schreibhäufigkeit) für die LBAs. Bei verschiedenen Ausführungsformen kann die Anzahl der Wärmeklassen variieren, jedoch hat sich in Versuchen eine Anzahl von Wärmeklassen zwischen zwei und einschließlich acht als wirksam erwiesen. Bei einer Ausführungsform konfiguriert die Steuereinheit zum Beispiel in Block 614 vier Wärmeklassen, die jeweils den am häufigsten geschriebenen heißen LBAs, den weniger häufig geschriebenen warmen LBAs, den noch seltener geschriebenen kühlen LBAs und den am seltensten geschriebenen kalten LBAs entsprechen. Der Satz von Host-Schreibanforderungen, die auf LBAs innerhalb einer jeweiligen Wärmeklasse abzielen, wird hierin als „Schreibdatenstrom“ bezeichnet. Die Anzahl der in Block 614 konfigurierten Wärmeklassen (und damit der Schreibdatenströme) kann, muss aber nicht, der Anzahl der in Block 612 konfigurierten Zustandsgrade entsprechen.

[0041] In Block 616 erstellt die Steuereinheit eine Zuordnung zwischen den in Block 612 konfigurierten Wärmeklassen und den in Block 614 konfigurierten Zustandsgraden. Bei einer besonders bevorzugten Ausführungsform sind Wärme und Zustand direkt miteinander korreliert, wobei die heißesten LBAs dem besten Zustandsgrad und die kältesten LBAs einem schlechteren Zustandsgrad zugeordnet werden. Selbstverständlich können bei sonstigen Ausführungsformen sonstige Zuordnungen zwischen Schreibwärme und Zustandsgraden eingesetzt werden. Beispielsweise wird bei einer Ausführungsform, in der numerische Schreibwärme und Zustandsgrade eingesetzt werden, die in Block 616 erstellte Zuordnung zum Beispiel durch eine mathematische Funktion ermittelt, die so ausgewählt wird, dass sie ein annähernd konstantes Ergebnis aus den einander zugeordneten Zustandsgraden und Schreibwärmen ergibt (z.B. $C \approx f$ (Wärme, Zustand)). Bei einer weiteren Ausführungsform ordnet die Zuordnung von Schreibwärme zu Zustandsgrad jede Wärmeklasse einer geordneten Prioritätenliste von Zustandsgraden zu. Zu dem Zeitpunkt, zu dem Blockstreifen aus Blöcken in den RTU-Warteschlangen 306 erstellt werden, ist möglicherweise kein Block in der bevorzugten RTU-Warteschlange 306 mit einem jeweiligen Zustandsgrad vorhanden. Die geordnete Prioritätsliste ermöglicht dann eine Zuweisung eines Blocks aus einer weiteren RTU-Warteschlange 306 zu einem Zustandsgrad, der dem bevorzugten so nahe wie möglich kommt.

[0042] Im Anschluss an Block 616 kehrt der in **Fig. 6** dargestellte Prozess zu Block 604 zurück. Wenn eine negative Ermittlung in Block 604 erfolgt, die angibt,

dass noch ein oder mehrere zusätzliche Block-Pools zu verarbeiten sind, kehrt der Prozess zu den Blöcken 610 bis 616 zurück, die beschrieben worden sind. Wenn jedoch in Block 604 ermittelt wird, dass alle in Block 602 konfigurierten Block-Pools verarbeitet worden sind, endet der Konfigurationsprozess von **Fig. 6** bei Block 606.

[0043] Unter Bezugnahme auf **Fig. 7** wird ein Übersichtsplan eines logischen Ablaufs eines beispielhaften Prozesses zum dynamischen Übertragen von Blöcken zwischen Block-Pools gemäß einer Ausführungsform veranschaulicht. Der veranschaulichte Prozess kann zum Beispiel durch eine Steuereinheit (z.B. den GPP 132 und/oder die Flash-Steuereinheit 140) in Hardware, Firmware, Software oder einer Kombination von diesen während eines Betriebs des Datenspeichersystems 120 durchgeführt werden.

[0044] Der Prozess von **Fig. 7** beginnt bei 700 und geht anschließend zu Block 702 über, der darstellt, dass die Steuereinheit ermittelt, ob die Größe eines der Block-Pools 400 und 402 zu ändern ist. Beispielsweise kann die Steuereinheit auf Grundlage eines oder mehrerer Kriterien ermitteln, ob die Größe eines jeweiligen Block-Pools zu ändern ist, einschließlich der Belegung von Blöcken, die zu dem jeweiligen Block-Pool innerhalb einer oder mehrerer RTU-Warteschlangen 306 gehören, die einen Schwellenwert nicht erfüllen, der Belegung von Blöcken, die zu einem weiteren Block-Pool innerhalb einer oder mehrerer RTU-Warteschlangen 306 gehören, der P/E-Zyklusanzahlen von Blöcken innerhalb des jeweiligen Block-Pools, der Anzahl von Blöcken innerhalb des jeweiligen Block-Pools, die einem oder mehreren der Zustandsgrade zugeordnet sind, usw. Die in Block 702 durchgeführte Ermittlung kann zum Beispiel durch ein(e, en) oder mehrere des Hintergrund-Zustandsprüfprogramms 330, des Garbage Collectors 312, der Blockzurückziehungsfunktion 318 und/oder der Funktion 320 zum Erstellen von Blockstreifen ausgelöst werden. Als Reaktion darauf, dass die Steuereinheit in Block 702 ermittelt, dass die Größe eines Block-Pools nicht zu ändern ist, iteriert der Prozess in Block 702. Als Reaktion auf eine Ermittlung in Block 702, dass die Größe eines Block-Pools zu ändern ist, geht der Prozess zu Block 704 über.

[0045] Block 704 veranschaulicht, dass die Steuereinheit den Block-Pool auswählt, an den physische Blöcke 204 zu übertragen sind. Darüber hinaus wählt die Steuereinheit in Block 706 die Wärmeklasse oder den Schreibdatenstrom (z.B. heiß, warm, kühl oder kalt) aus, für die/den ein physischer Block 204 zu dem jeweiligen Block-Pool hinzuzufügen ist. Anschließend ermittelt die Steuereinheit in Block 708 eine Datenstrom-Verschleißmessgröße für Blöcke 204, die aktuell zum Speichern von Daten zugewiesen sind, die LBAs in der ausgewählten Wärmeklasse zugehörig sind. Bei einer bevorzugten Ausführungsform verfolgt die Steuereinheit die Blöcke 204, die jeder/jedem Wärmeklasse/-Schreibdatenstrom zugeordnet sind, und ermittelt (z.B. aus den P/E-Zyklusanzahlen 334) die Gesamtanzahl der P/E-Zyklen, denen diese Blöcke 204 unterzogen worden sind. Aus diesen Anzahlen kann die Steuereinheit die Datenstrom-Verschleißmessgröße für die ausgewählte Wärmeklasse als die Gesamtanzahl von Blöcken 204 berechnen, die Daten in der Wärmeklasse speichern, geteilt durch die Gesamtanzahl von P/E-Zyklen, denen diese Blöcke 204 unterzogen worden sind.

[0046] Auf Grundlage der in Block 708 ermittelten Datenstrom-Verschleißmessgröße wählt die Steuereinheit anschließend zumindest einen unprogrammierten Block 204 aus dem sonstigen Block-Pool zum Übertragen an den ausgewählten Block-Pool aus, der einen individuellen Blockzustand aufweist, der für die ausgewählte Wärmeklasse geeignet ist (Block 710). Diese Auswahl gewährleistet, dass die im besten Zustand befindlichen Blöcke 204 der größten Anzahl von künftigen P/E-Zyklen unterzogen wird und die im schlechtesten Zustand befindlichen Blöcke 204 der geringsten Anzahl von künftigen P/E-Zyklen unterzogen wird. Infolgedessen wird eine Lebensdauer von Einheiten für alle Typen von Verarbeitungsprozessen und alle Einheitennutzungspunkte verbessert. In Block 712 überträgt die Steuereinheit den/die in Block 710 ausgewählten Block/Blöcke 204 an den ausgewählten Block-Pool, aktualisiert die Blockmodus-Metadaten 335 mit dem neuen Betriebsmodus (z.B. QLC oder SLC) und aktualisiert gegebenenfalls den Blockzustand. Anschließend kehrt der Prozess von **Fig. 7** zu Block 702 zurück, der beschrieben worden ist.

[0047] Wie beschrieben worden ist, enthält ein nichtflüchtiger Speicher bei zumindest einer Ausführungsform eine Mehrzahl von physischen Speicherblöcken, die jeweils eine entsprechende Mehrzahl von Zellen enthalten, wobei jede der Mehrzahl von Zellen einzeln in der Lage ist, mehrere Daten-Bits zu speichern. Eine Steuereinheit weist physische Blöcke aus der Mehrzahl von physischen Blöcken einem ersten Pool zu, der physische Blöcke enthält, die in einem ersten (z.B. QLC-) Modus arbeiten, um eine größere Anzahl von Bits pro Zelle zu speichern, und weist sonstige physische Blöcke aus der Mehrzahl von physischen Blöcken einem zweiten Pool zu, der physische Blöcke enthält, die in einem zweiten (z.B. SLC-) Modus arbeiten, um eine geringere Anzahl von Bits pro Zelle zu speichern. Die Steuereinheit überträgt physische Blöcke zwischen dem ersten Pool und dem zweiten Pool zumindest auf Grundlage von Bit-Fehlerraten, die für die übertragenen physischen Blöcke gemessen werden.

[0048] Bei der vorliegenden Erfindung kann es sich um ein System, ein Verfahren und/oder ein Computerprogrammprodukt handeln. Das Computerprogrammprodukt kann (ein) durch einen Computer lesbare(s) Speichermedium (oder -medien) enthalten, auf dem/denen durch einen Computer lesbare Programmanweisungen gespeichert ist/sind, um einen Prozessor dazu zu veranlassen, Aspekte der vorliegenden Erfindung auszuführen.

[0049] Bei dem durch einen Computer lesbaren Speichermedium kann es sich um eine physische Einheit handeln, die Anweisungen zur Verwendung durch eine Einheit zur Ausführung von Anweisungen behalten und speichern kann. Bei dem durch einen Computer lesbaren Speichermedium kann es sich zum Beispiel um eine elektronische Speichereinheit, eine magnetische Speichereinheit, eine optische Speichereinheit, eine elektromagnetische Speichereinheit, eine Halbleiterspeichereinheit oder jede geeignete Kombination daraus handeln, ohne auf diese beschränkt zu sein. Zu einer nicht erschöpfenden Liste spezifischerer Beispiele des durch einen Computer lesbaren Speichermediums gehören die Folgenden: eine tragbare Computerdiskette, eine Festplatte, ein Direktzugriffsspeicher (RAM), ein Festwertspeicher (ROM), ein löschbarer programmierbarer Festwertspeicher (EPROM bzw. Flash-Speicher), ein statischer Direktzugriffsspeicher (SRAM), ein tragbarer Kompaktspeicherplatten-Festwertspeicher (CD-ROM), eine DVD (digital versatile disc), ein Speicher-Stick, eine Diskette, eine mechanisch codierte Einheit wie zum Beispiel Lochkarten oder erhabene Strukturen in einer Rille, auf denen Anweisungen gespeichert sind, und jede geeignete Kombination daraus. Ein durch einen Computer lesbares Speichermedium soll in der Verwendung hierin nicht als flüchtige Signale an sich aufgefasst werden, wie zum Beispiel Funkwellen oder andere sich frei ausbreitende elektromagnetische Wellen, elektromagnetische Wellen, die sich durch einen Wellenleiter oder ein anderes Übertragungsmedium ausbreiten (z.B. durch ein Glasfaserkabel geleitete Lichtimpulse) oder durch einen Draht übertragene elektrische Signale.

[0050] Hierin beschriebene, durch einen Computer lesbare Programmanweisungen können von einem durch einen Computer lesbaren Speichermedium auf jeweilige Datenverarbeitungs-/Verarbeitungseinheiten oder über ein Netzwerk wie zum Beispiel das Internet, ein lokales Netzwerk, ein Weitverkehrsnetz und/oder ein drahtloses Netzwerk auf einen externen Computer oder eine externe Speichereinheit heruntergeladen werden. Das Netzwerk kann Kupferübertragungskabel, Lichtwellenübertragungsleiter, drahtlose Übertragung, Router, Firewalls, Vermittlungseinheiten, Gateway-Computer und/oder Edge-Server aufweisen. Eine Netzwerkadapterkarte oder Netzwerkschnittstelle in jeder Datenverarbeitungs-/Verarbeitungseinheit empfängt durch einen Computer lesbare Programmanweisungen aus dem Netzwerk und leitet die durch einen Computer lesbaren Programmanweisungen zur Speicherung in einem durch einen Computer lesbaren Speichermedium innerhalb der entsprechenden Datenverarbeitungs-/Verarbeitungseinheit weiter.

[0051] Bei durch einen Computer lesbaren Programmanweisungen zum Ausführen von Arbeitsschritten der vorliegenden Erfindung kann es sich um Assembler-Anweisungen, ISA-Anweisungen (Instruction-Set-Architecture), Maschinenanweisungen, maschinenabhängige Anweisungen, Mikrocode, Firmware-Anweisungen, zustandssetzende Daten oder entweder Quellcode oder Objektcode handeln, die in einer beliebigen Kombination aus einer oder mehreren Programmiersprachen geschrieben werden, darunter objektorientierte Programmiersprachen wie Smalltalk, C++ o.ä. sowie herkömmliche prozedurale Programmiersprachen wie die Programmiersprache „C“ oder ähnliche Programmiersprachen. Die durch einen Computer lesbaren Programmanweisungen können vollständig auf dem Computer des Benutzers, teilweise auf dem Computer des Benutzers, als eigenständiges Software-Paket, teilweise auf dem Computer des Benutzers und teilweise auf einem entfernt angeordneten Computer oder vollständig auf dem entfernt angeordneten Computer oder Server ausgeführt werden. In letzterem Fall kann der entfernt angeordnete Computer mit dem Computer des Benutzers durch eine beliebige Art Netzwerk verbunden sein, darunter ein lokales Netzwerk (LAN) oder ein Weitverkehrsnetz (WAN), oder die Verbindung kann mit einem externen Computer hergestellt werden (zum Beispiel über das Internet unter Verwenden eines Internet-Dienstanbieters). In einigen Ausführungsformen können elektronische Schaltungen, darunter zum Beispiel programmierbare Logikschaltungen, feldprogrammierbare Gate-Arrays (FPGA) oder programmierbare Logik-Arrays (PLA) die durch einen Computer lesbaren Programmanweisungen ausführen, indem sie Zustandsdaten der durch einen Computer lesbaren Programmanweisungen nutzen, um die elektronischen Schaltungen zu personalisieren, um Aspekte der vorliegenden Erfindung durchzuführen.

[0052] Aspekte der vorliegenden Erfindung sind hierin unter Bezugnahme auf Ablaufpläne und/oder Blockschaubilder von Verfahren, Vorrichtungen (Systemen) und Computerprogrammprodukten gemäß Ausführungsformen der Erfindung beschrieben. Es wird darauf hingewiesen, dass jeder Block der Ablaufpläne und/oder der Blockschaubilder sowie Kombinationen von Blöcken in den Ablaufplänen und/oder den Blockschaubildern mittels durch einen Computer lesbare Programmanweisungen implementiert werden können.

[0053] Diese durch einen Computer lesbaren Programmanweisungen können einem Prozessor eines Universalcomputers, eines Spezialcomputers oder einer anderen programmierbaren Datenverarbeitungsvorrichtung bereitgestellt werden, um eine Maschine zu erzeugen, so dass die über den Prozessor des Computers bzw. der anderen programmierbaren Datenverarbeitungsvorrichtung ausgeführten Anweisungen Mittel zur Umsetzung der in dem Block bzw. den Blöcken des Ablaufplans und/oder des Blockschaubildes festgelegten Funktionen/-Schritte erzeugen. Diese durch einen Computer lesbaren Programmanweisungen können auch auf einem durch einen Computer lesbaren Speichermedium gespeichert sein, das einen Computer, eine programmierbare Datenverarbeitungsvorrichtung und/oder andere Einheiten so steuern kann, dass sie auf eine bestimmte Art funktionieren, so dass das durch einen Computer lesbare Speichermedium, auf dem Anweisungen gespeichert sind, ein Herstellungsprodukt aufweist, darunter Anweisungen, welche Aspekte der/des in dem Block bzw. den Blöcken des Ablaufplans und/oder des Blockschaubildes angegebenen Funktion/Schritts implementieren.

[0054] Die durch einen Computer lesbaren Programmanweisungen können auch auf einen Computer, eine andere programmierbare Datenverarbeitungsvorrichtung oder eine andere Einheit geladen werden, um das Ausführen einer Reihe von Prozessschritten auf dem Computer bzw. der anderen programmierbaren Vorrichtung oder anderen Einheit zu verursachen, um einen auf einem Computer ausgeführten Prozess zu erzeugen, so dass die auf dem Computer, einer anderen programmierbaren Vorrichtung oder einer anderen Einheit ausgeführten Anweisungen die in dem Block bzw. den Blöcken des Ablaufplans und/oder des Blockschaubildes festgelegten Funktionen/Schritte umsetzen.

[0055] Der Ablaufplan und die Blockschaubilder in den Figuren veranschaulichen die Architektur, die Funktionalität und den Betrieb möglicher Implementierungen von Systemen, Verfahren und Computerprogrammprodukten gemäß verschiedenen Ausführungsformen der vorliegenden Erfindung. In diesem Zusammenhang kann jeder Block in dem Ablaufplan oder den Blockschaubildern ein Modul, ein Segment oder einen Teil von Anweisungen darstellen, die eine oder mehrere ausführbare Anweisungen zum Implementieren der bestimmten logischen Funktion(en) aufweisen. In einigen alternativen Implementierungen können die in dem Block angegebenen Funktionen in einer anderen Reihenfolge als in den Figuren gezeigt auftreten. Zwei nacheinander gezeigte Blöcke können zum Beispiel in Wirklichkeit im Wesentlichen gleichzeitig ausgeführt werden, oder die Blöcke können manchmal je nach entsprechender Funktionalität in umgekehrter Reihenfolge ausgeführt werden. Es ist ferner anzumerken, dass jeder Block der Blockschaubilder und/oder des Ablaufplans sowie Kombinationen aus Blöcken in den Blockschaubildern und/oder dem Ablaufplan durch spezielle auf Hardware beruhende Systeme implementiert werden können, welche die festgelegten Funktionen oder Schritte durchführen, oder Kombinationen aus Spezial-Hardware und Computeranweisungen ausführen.

Patentansprüche

1. Verfahren zum Betreiben eines nichtflüchtigen Speichers, der eine Mehrzahl von physischen Speicherblöcken (204) aufweist, die jeweils eine entsprechende Mehrzahl von Zellen aufweisen, wobei jede der Mehrzahl von Zellen einzeln in der Lage ist, mehrere Daten-Bits zu speichern, wobei das Verfahren aufweist:
Zuweisen durch eine Steuereinheit (120) von physischen Blöcken aus der Mehrzahl von physischen Blöcken (204) zu einem ersten Pool (400), der physische Blöcke (204) aufweist, die in einem ersten Modus arbeiten, um eine größere Anzahl von Bits pro Zelle zu speichern, und Zuweisen von sonstigen physischen Blöcken (204) aus der Mehrzahl von physischen Blöcken zu einem zweiten Pool (402), der physische Blöcke aufweist, die in einem zweiten Modus arbeiten, um eine geringere Anzahl von Bits pro Zelle zu speichern; und
Übertragen von physischen Blöcken (204) zwischen dem ersten Pool (400) und dem zweiten Pool (402) durch die Steuereinheit (120) zumindest auf Grundlage von Bit-Fehlerraten, die für die übertragenen physischen Blöcke (204) gemessen werden.

2. Verfahren nach Anspruch 1, wobei das Übertragen aufweist:
Übertragen von physischen Blöcken (204) mit einer höheren Bit-Fehlerrate von dem ersten Pool an den zweiten Pool; und
Übertragen von physischen Blöcken (204) mit einer niedrigeren Bit-Fehlerrate von dem zweiten Pool (402) an den ersten Pool (400).

3. Verfahren nach Anspruch 1, wobei:
das Verfahren des Weiteren aufweist:
jeweils für den ersten (400) und den zweiten Pool (402):
Verfolgen der Anzahl von physischen Blöcken (204), die zum Speichern von Daten für jeden einer Mehrzahl von Schreibdatenströmen mit verschiedenen Schreibwärmen zugewiesen worden sind, und der Anzahl von Programmier-/Lösch(P/E)-Zyklen (334) in jedem der Mehrzahl von Schreibdatenströmen; und
Berechnen für jeden der Mehrzahl von Schreibdatenströmen einer entsprechenden Datenstrom-Verschleißmessgröße auf Grundlage der Anzahl von dem Schreibdatenstrom zugewiesenen physischen Blöcken (204) und einer Gesamtanzahl von P/E-

Zyklen (334) der dem Schreibdatenstrom zugewiesenen physischen Blöcke; und
wobei das Übertragen ein Auswählen von physischen Blöcken (204) zum Übertragen auf Grundlage (1) eines ausgewählten Schreibdatenstroms aus der Mehrzahl von Schreibdatenströmen, (2) der Datenstrom-Verschleißmessgröße des ausgewählten Schreibdatenstroms und (3) des Blockzustands von physischen Blöcken (204), die zum Speichern von Daten für den ausgewählten Schreibdatenstrom zugewiesen worden sind, aufweist.

4. Verfahren nach Anspruch 3, und das des Weiteren ein Ermitteln eines Blockzustands von physischen Blöcken (204) in sämtlichen der Mehrzahl von Schreibdatenströmen aus einer Bit-Fehlerrate, die kurz nach einem Programmieren der physischen Blöcke gemessen wird, durch die Steuereinheit (120) aufweist.

5. Verfahren nach Anspruch 3, wobei das Auswählen aufweist:
auf Grundlage des ausgewählten Schreibdatenstroms, der eine Datenstrom-Verschleißmessgröße aufweist, die einen größten Verschleiß unter sämtlichen der Mehrzahl von Schreibdatenströmen widerspiegelt, Auswählen von im besten Zustand befindlichen physischen Blöcken aus den physischen Blöcken (204), die dem ausgewählten Schreibdatenstrom zugewiesen worden sind, zum Übertragen; und
auf Grundlage des ausgewählten Schreibdatenstroms, der eine Datenstrom-Verschleißmessgröße aufweist, die einen geringsten Verschleiß unter sämtlichen der Mehrzahl von Schreibdatenströmen widerspiegelt, Auswählen von im schlechtesten Zustand befindlichen physischen Blöcken aus den physischen Blöcken (204), die dem ausgewählten Schreibdatenstrom zugewiesen worden sind, zum Übertragen.

6. Verfahren nach Anspruch 1, wobei:
es sich bei dem ersten Modus um einen Vier-Bit-Zellen(QLC)-Modus handelt; und
es sich bei dem zweiten Modus um einen Ein-Bit-Zellen(SLC)-Modus handelt.

7. Verfahren nach Anspruch 1, wobei das Übertragen ein Übertragen der physischen Blöcke auf Grundlage einer verfügbaren Anzahl von unprogrammierten physischen Blöcken aus einem jeweiligen des ersten (400) und des zweiten Pools (402) aufweist, die einen Schwellenwert nicht erfüllen.

8. Datenspeichersystem (104), das aufweist:
eine Steuereinheit (120) für einen nichtflüchtigen Speicher, der eine Mehrzahl von physischen Speicherblöcken aufweist, die jeweils eine entsprechende Mehrzahl von Zellen aufweisen, wobei jede der Mehrzahl von Zellen einzeln in der Lage ist, mehrere Daten-Bits zu speichern, wobei die Steuereinheit (120) dazu konfiguriert ist das Verfahren gemäß einem der Ansprüche 1 bis 7 auszuführen.

9. Computerprogrammprodukt, wobei das Computerprogrammprodukt ein durch einen Computer lesbares Speichermedium aufweist, in dem Programmanweisungen verkörpert sind, wobei die Programmanweisungen durch eine Steuereinheit eines nichtflüchtigen Speichers ausführbar sind, der eine Mehrzahl von Zellen aufweist, die jeweils einzeln in der Lage sind, mehrere Daten-Bits zu speichern, wobei die Programmanweisungen, wenn sie ausgeführt werden, bewirken, dass die Steuereinheit das Verfahren gemäß einem der Ansprüche 1 bis 7 ausführt.

Es folgen 6 Seiten Zeichnungen

Anhängende Zeichnungen

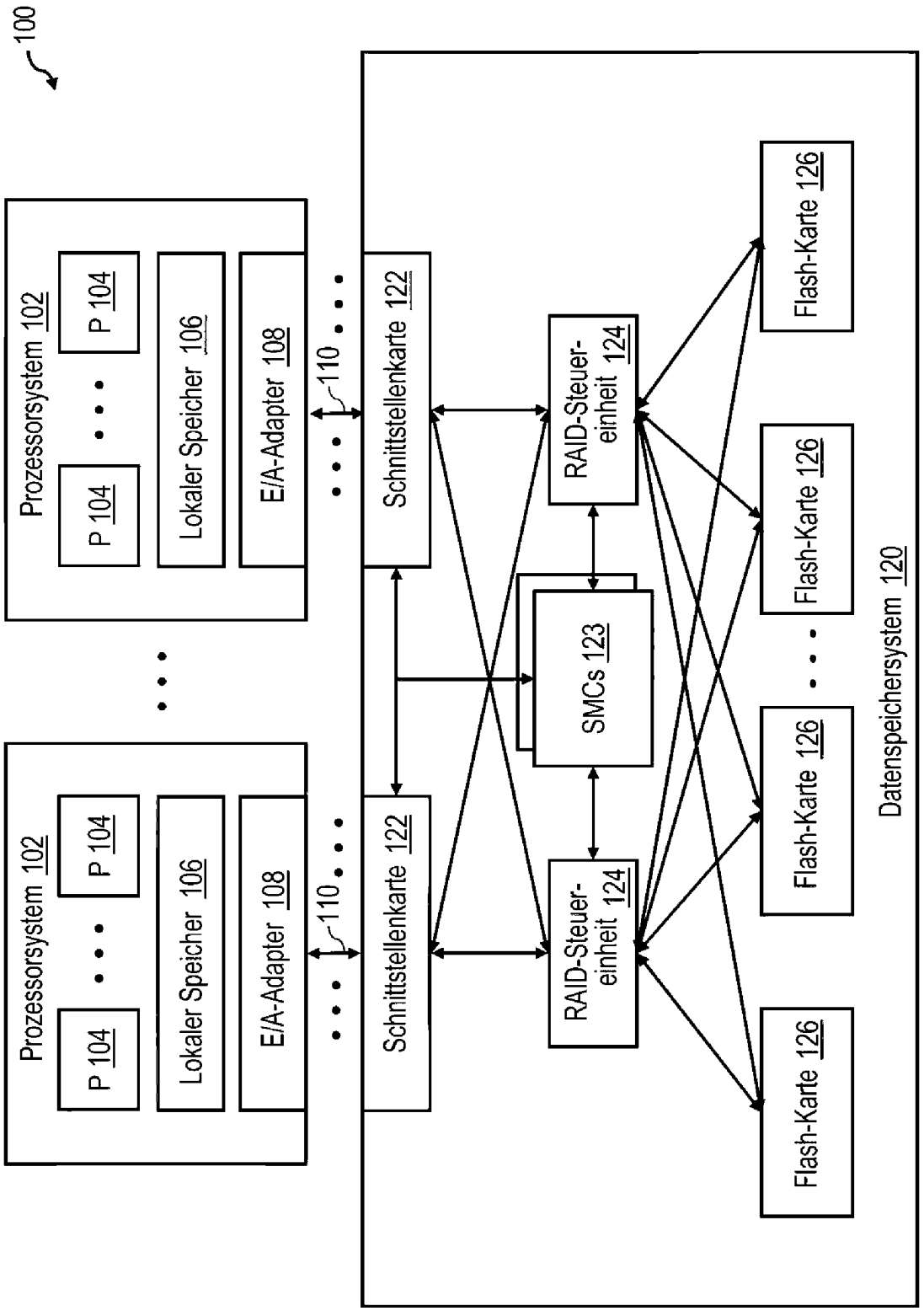

Fig. 1A

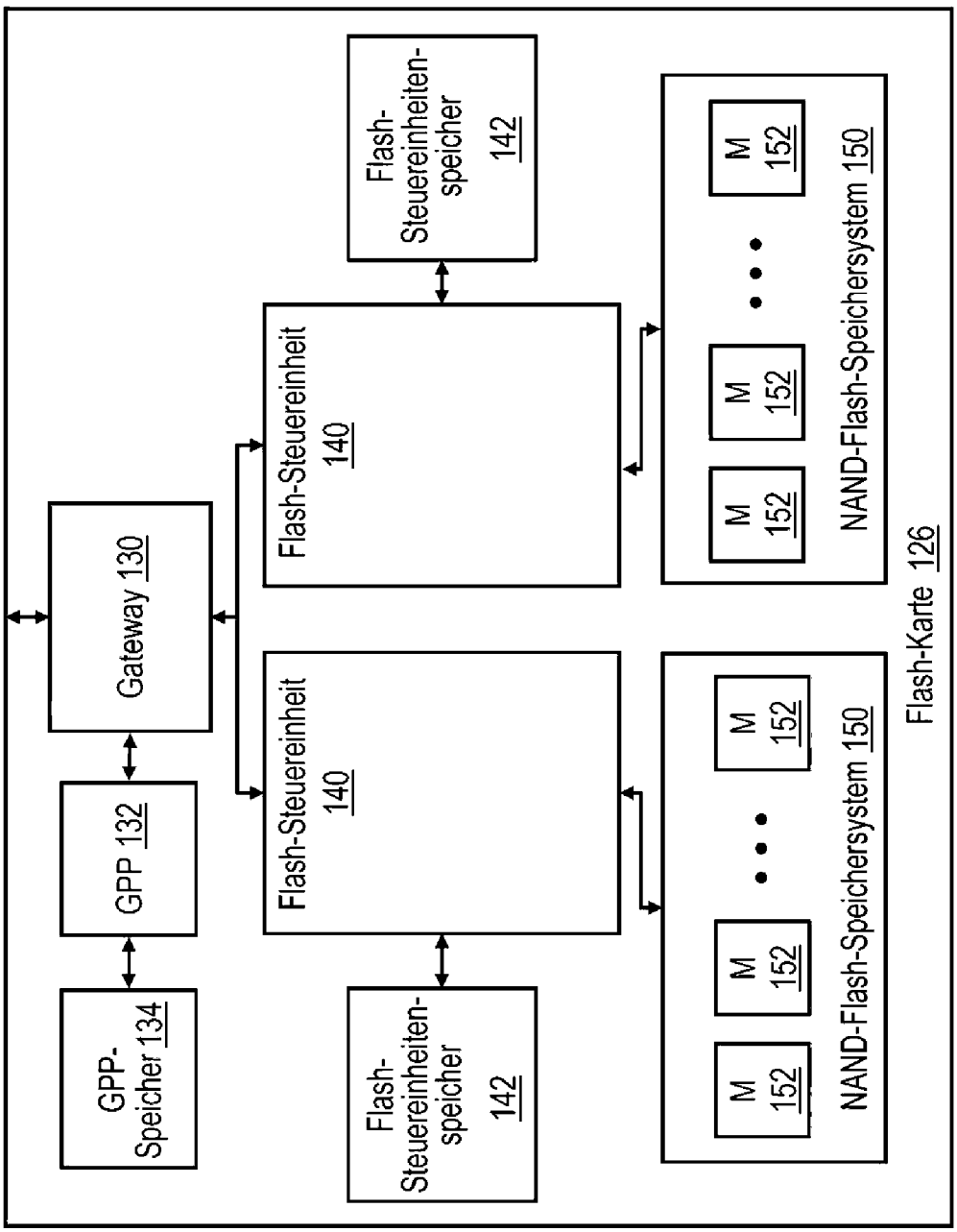
Flash-Karte 126
Gateway 130
GPP 132
GPP-Speicher 134
Flash-Steuereinheit 140
Flash-Steuereinheit 140
Flash-Steuereinheiten-speicher 142
Flash-Steuereinheiten-speicher 142
NAND-Flash-Speichersystem 150
NAND-Flash-Speichersystem 150
M 152
M 152
M 152
M 152
M 152
M 152

Fig. 1B

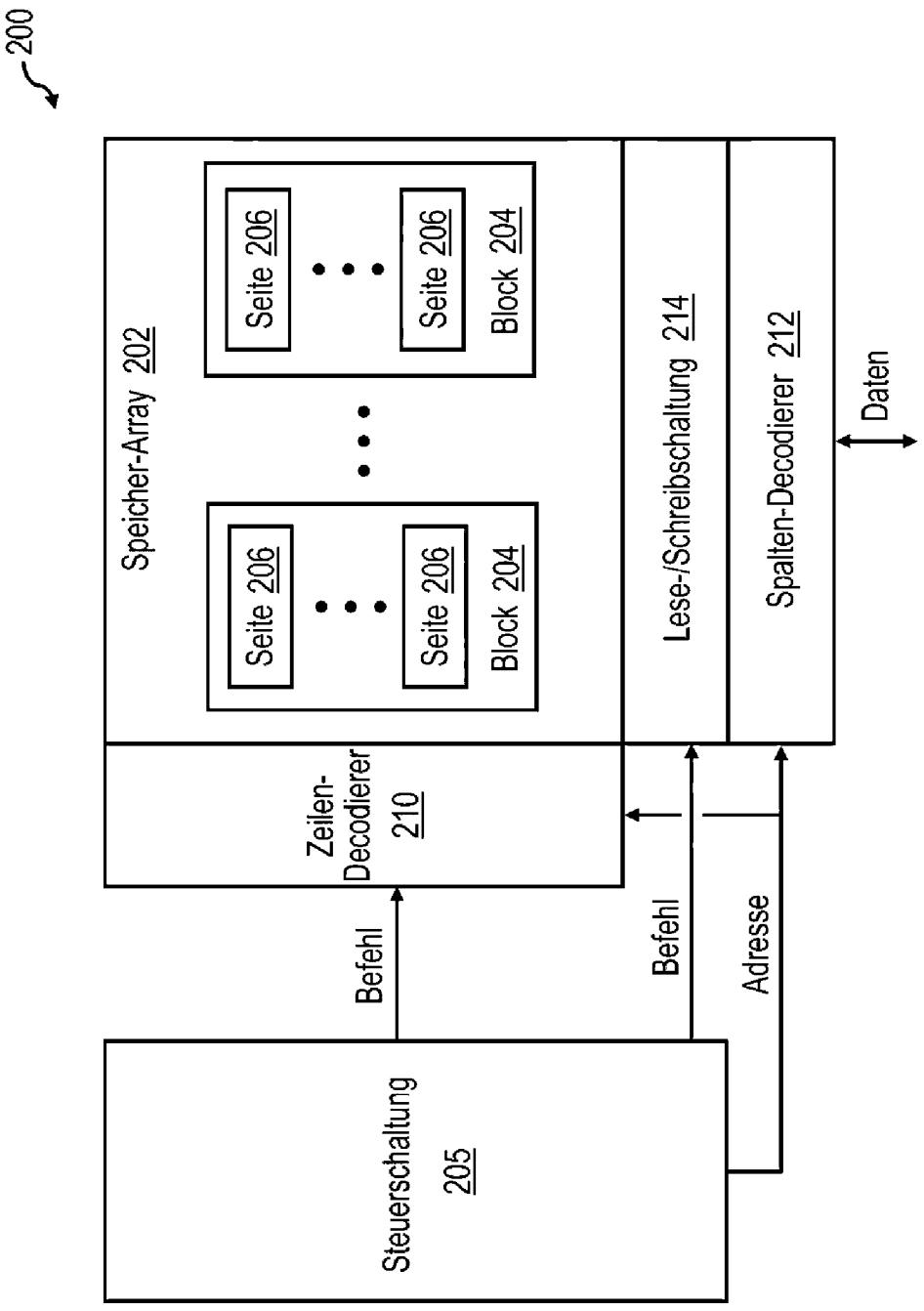
200
Speicher-Array 202
Seite 206
Seite 206
Block 204
Seite 206
Seite 206
Block 204
Lese-/Schreibschaltung 214
Spalten-Decodierer 212
Daten
Zeilen-Decodierer 210
Befehl
Befehl
Adresse
Steuerschaltung 205

Fig. 2

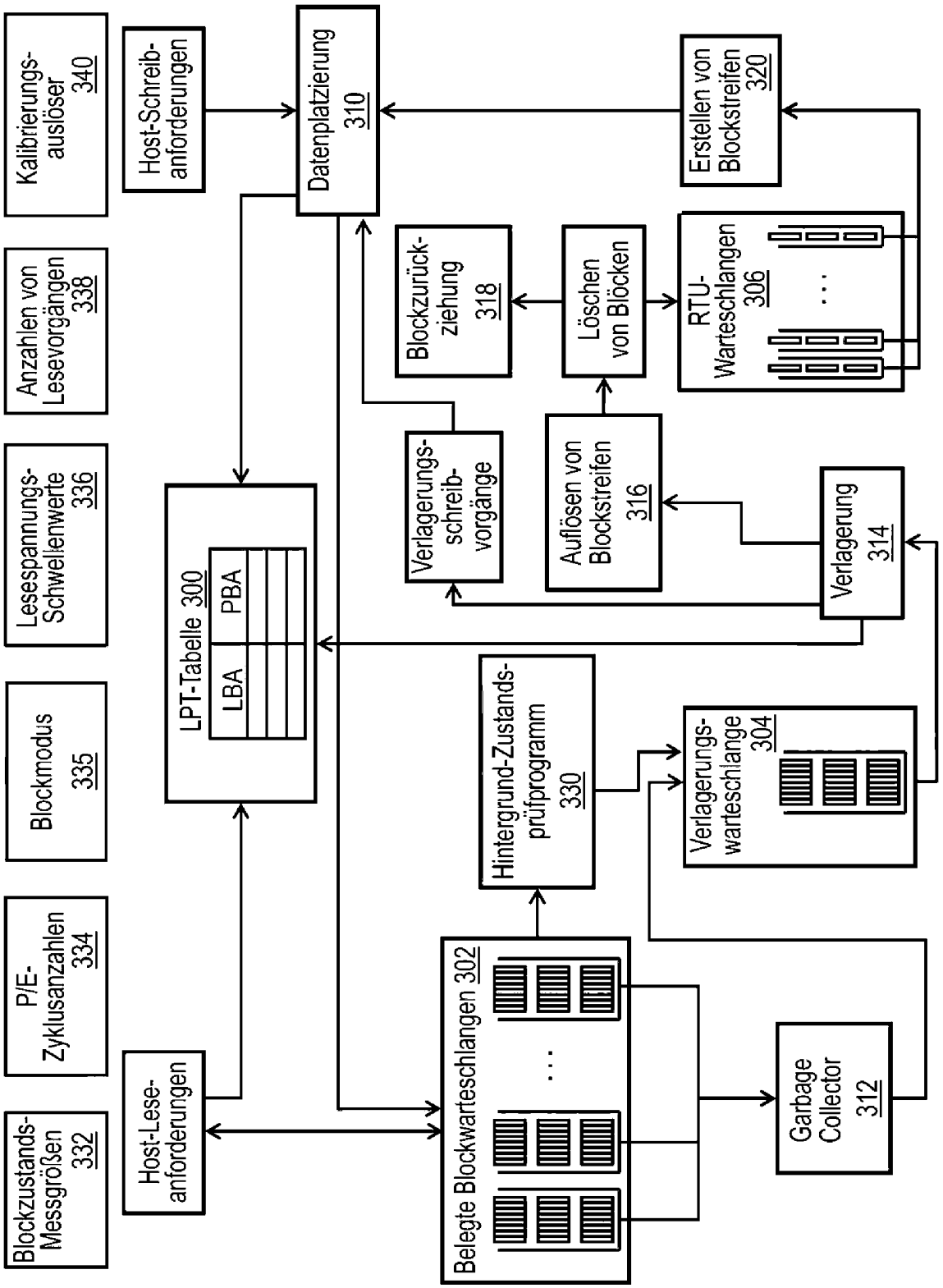
Kalibrierungs-auslöser 340
Host-Schreib-anforderungen
Datenplatzierung 310
Erstellen von Blockstreifen 320
Anzahlen von Lesevorgängen 338
Blockzurück-ziehung 318
Löschen von Blöcken
RTU-Warteschlangen 306
Lesespannungs-Schwellenwerte 336
Verlagerungs-schreib-vorgänge
Auflösen von Blockstreifen 316
Verlagerung 314
LPT-Tabelle 300
LBA
PBA
Blockmodus 335
Hintergrund-Zustands-prüfprogramm 330
Verlagerungs-warteschlange 304
P/E-Zyklusanzahlen 334
Host-Lese-anforderungen
Belegte Blockwarteschlangen 302
Garbage Collector 312
Blockzustands-Messgrößen 332

Fig. 3

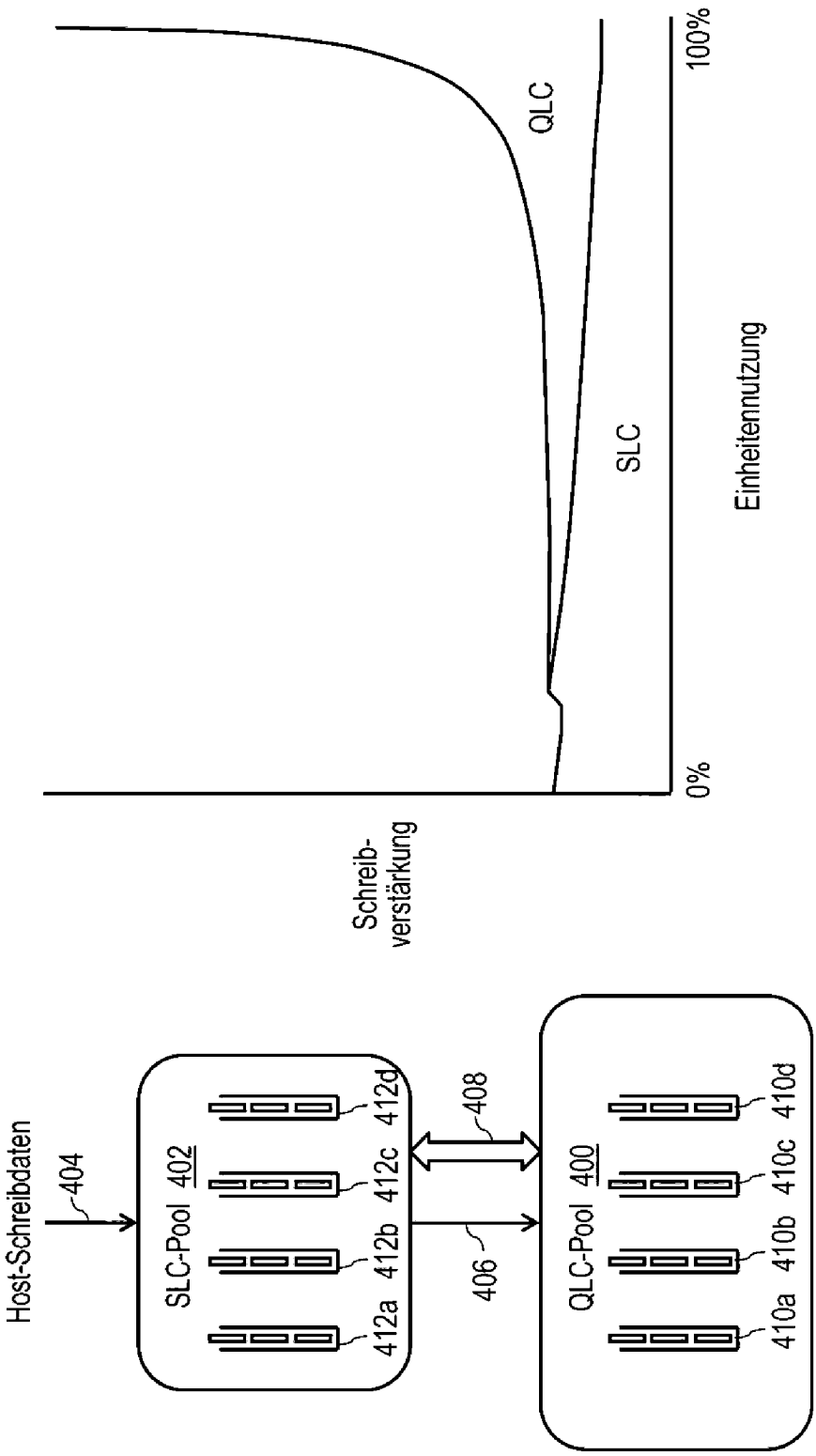
Host-Schreibdaten
404
SLC-Pool 402
412a
412b
412c
412d
406
408
QLC-Pool 400
410a
410b
410c
410d

Fig. 4

QLC
SLC
100%
0%
Einheitennutzung
Schreib-
verstärkung

Fig. 5

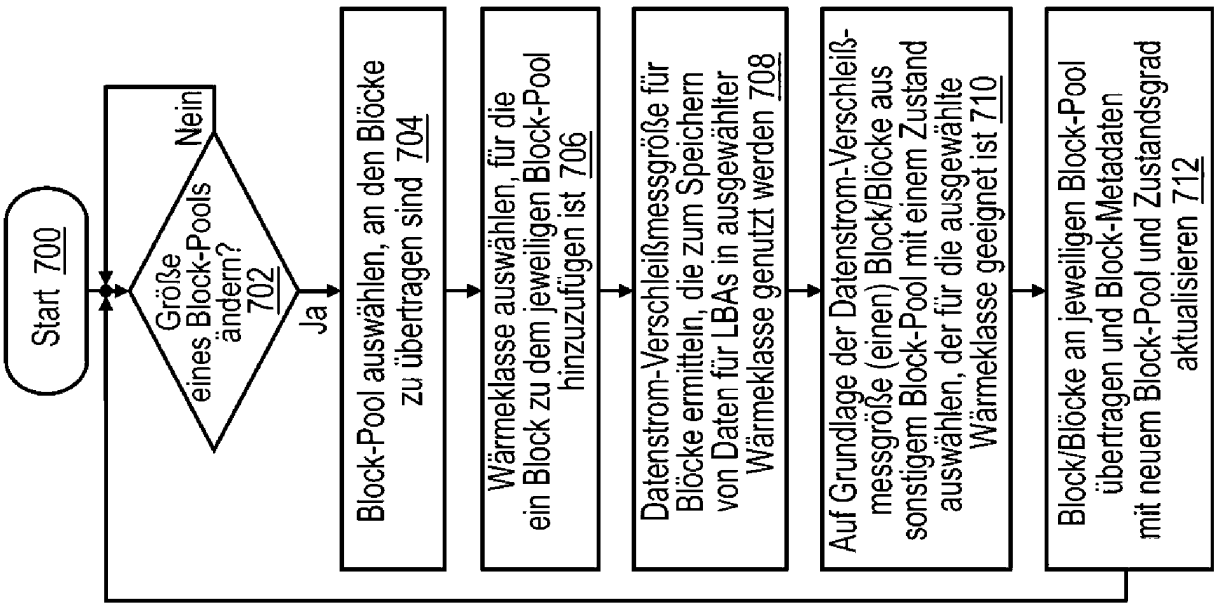

Fig. 7

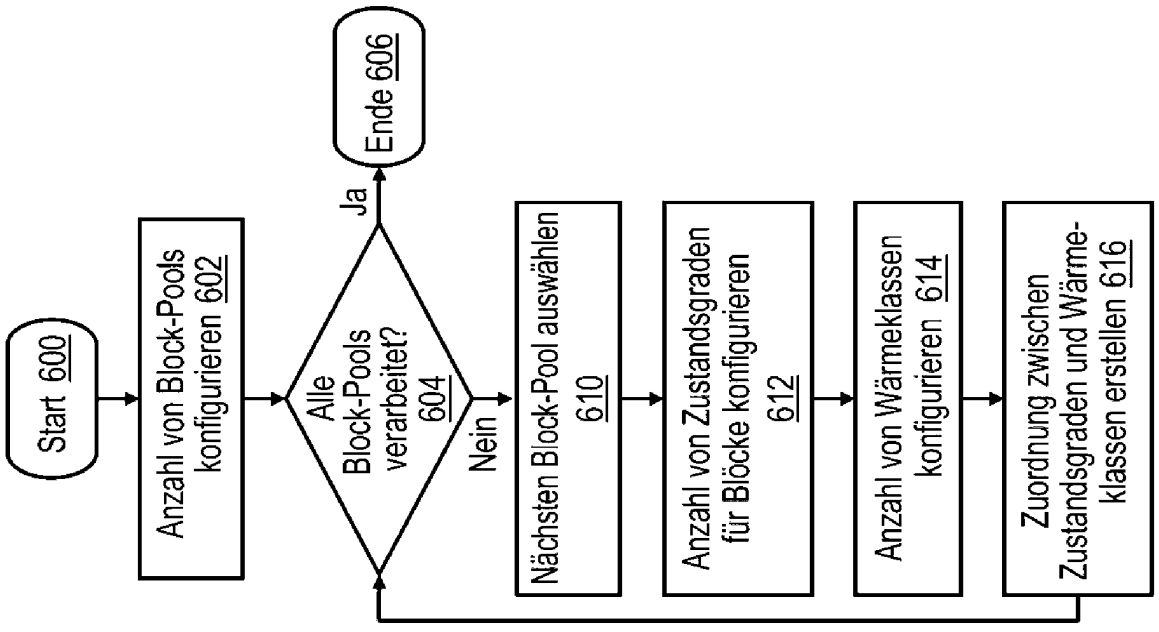

Fig. 6

 Das Dokument wurde durch die Firma Luminess hergestellt.