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

(10) **DE 698 19 853 T2** 2004.09.02

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 924 596 B1**
(21) Deutsches Aktenzeichen: **698 19 853.0**
(96) Europäisches Aktenzeichen: **98 123 844.7**
(96) Europäischer Anmeldetag: **15.12.1998**
(97) Erstveröffentlichung durch das EPA: **23.06.1999**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **19.11.2003**
(47) Veröffentlichungstag im Patentblatt: **02.09.2004**

(51) Int Cl.[7]: **G06F 3/06**

(30) Unionspriorität:
35051097 19.12.1997 JP

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:
Hitachi, Ltd., Tokio/Tokyo, JP

(74) Vertreter:
Strehl, Schübel-Hopf & Partner, 80538 München

(72) Erfinder:
Kitamura, Manabu, Yokohama-shi, JP; Yamamoto, Akira, Sagamihara-shi, JP; Honma, Shigeo, Odawara-shi, JP

(54) Bezeichnung: **Speichersystem mit einer Mehrzahl von sich an mehrere Datenformate anpassenden Schnittstellen**

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

HINTERGRUND DER ERFINDUNG

[0001] Diese Anmeldung steht in Zusammenhang mit der US-Anmeldung mit der laufenden Nr. 08/774.947 vom 30. Dezember 1996 und der US-Anmeldung mit der laufenden Nr. 08/912.872 vom 19. August 1997.
[0002] Die vorliegende Erfindung bezieht sich auf ein Speicher-Subsystem in einem Datenverarbeitungssystem und insbesondere auf ein Speicher-Subsystem mit mehreren Interfaces (Schnittstellen) für mehrere Datenformate.
[0003] In jüngster Zeit ist ein weitgehendes „Downsizing" bzw. eine Verkleinerung von Computersystemen erreicht worden, indem Funktionen, die üblicherweise von einem Mainframe-Rechner ausgeführt werden, von kleineren Computern wie z. B. Personalcomputern und Workstations übernommen werden. Auf dem Mainframe haben sich durch die in der Vergangenheit ausgeführten Aufgaben große Mengen von Daten angesammelt. In dieser Situation ist es wünschenswert, dass kleinere Computer auf die auf dem Mainframe angesammelten Daten zugreifen können. Zu diesem Zweck ist ein technisches Verfahren für den Austausch oder das Konvertieren von Dateien zwischen verschiedenen Betriebssystemen bekannt, bei dem zum Beispiel ein Betriebssystem einer Workstation auf eine Datei in einem Dateisystem zugreift, das von einem Betriebssystem eines Personalcomputers angelegt worden ist. Bei einigen Personalcomputer-Dateisystemen sind Daten bzw. Informationen wie z. B. Dateiformate und Positionen von Dateimanagementinformationen allgemein offen. Daher kann das Betriebssystem der Workstation ein Dateiformat eines Personalcomputers in der Workstation interpretieren, um auf eine Datei in einem Dateisystem zuzugreifen, das von dem Betriebssystem des Personalcomputers angelegt worden ist.
[0004] Auf dem Mainframe erfolgt ein Plattenzugriff üblicherweise im Count-Key-Format (im Folgenden abgekürzt als CKD), während für den Plattenzugriff in dem offenen System ein Fixed-Length-Format (im Folgenden abgekürzt als FBA) benutzt wird. Folglich muss zum Austauschen einer Datei zwischen dem Mainframe und dem offenen System der Unterschied zwischen den Datenformaten aufgehoben bzw. absorbiert werden. JP-A-9-258908 (entsprechend der US-Patentanmeldung mit der laufenden Nr. 08/774.447) beschreibt ein Verfahren zur Aufhebung der Datenformatabweichungen. Dort ist ein Speicher vorgesehen, der eine Funktion hat, bei der, wenn das offene System einen Datensatz im CKD-Format in dem Speicher liest, die Daten als Daten in einem Format mit fester Länge dargestellt werden, d. h. die Daten erhalten ein bestimmtes Format. Zusätzlich ist auf der Seite des offenen Systems eine Einrichtung zum Interpretieren der so in das bestimmte Format gebrachten Daten im Fixed-Length-Format vorgesehen, um auf diese Weise daraus ein Datenfeld zu erzeugen. Dadurch ist es möglich, dass das offene System einen Datensatz lesen kann, der von dem Mainframe in den Speicher geschrieben worden ist.
[0005] EP-A-0 811 905 beschreibt ein Speichersystem, das Schnittstellen (Interfaces) sowohl mit einem Host-Computer unter Verwendung eines CKD-Datenformats und mit einem Host-Computer unter Verwendung eines FBA-Datenformats (mit fester Länge) aufweist, wobei dieser Speicher eine Einrichtung zum Konvertieren zwischen diesen CKD- und FBA-Datenformaten aufweist.

ZUSAMMENFASSUNG DER ERFINDUNG

[0006] Obwohl JP-A-9-258908 (entsprechend der US-Patentanmeldung mit der laufenden Nr. 08/774.447) ein Verfahren für das offene System beschreibt, um einen Datensatz aus dem Mainframe zu lesen, wird dort kein Verfahren zum Schreiben von Daten in ein Dateisystem auf dem Mainframe beschrieben. Auf der Seite des offenen Systems existiert weder ein Konzept des Zählfelds der CKD-Daten noch ein Konzept für den Zylinder und die Kopfspur, die darin beschrieben sind. Um daher auf der Seite des offenen Systems Daten im CKD-Format schreiben zu können, ist eine Einrichtung nötig, um auf der Seite des offenen Systems ein Zählfeld zu erzeugen. Um auf der Mainframe-Seite einen Datensatz zu lesen, der auf der Seite des offenen Systems geschrieben worden ist, muss darüber hinaus zum Beispiel das Managementsystem des auf der Mainframe-Seite generierten Dateisystems aktualisiert werden.
[0007] Daher ist es ein Ziel der vorliegenden Erfindung, ein Interface für Daten im CKD-Format und ein Interface für Daten im FBA-Format in einem Plattengerät bereitzustellen, damit das offene System Daten in das Dateisystem auf dem Mainframe schreiben kann.
[0008] Um das obige Ziel zu erreichen, wird ein Verfahren zum Steuern eines Computersystems bereitgestellt und ein entsprechendes Computersystem nach der vorliegenden Erfindung gemäß Definition in den anliegenden voneinander unabgängigen Ansprüchen 1 bzw. 5. Das Computersystem nach der vorliegenden Erfindung umfasst einen ersten Computer, einen zweiten Computer und ein Speicher-Subsystem mit mehreren Interfaces zur Herstellung von Verbindungen zu den ersten und zweiten Computern. Der erste Computer greift auf das Speicher-Subsystem mit Daten im CKD-Format zu, und der zweite Computer greift auf das Speicher-Subsystem mit Daten im FBA-Format zu. Der zweite Computer weist eine Zugriffseinrichtung auf, um über ein In-

terface für Daten im FBA-Format auf einen Datensatz im CKD-Format zuzugreifen, der von dem ersten Computer in das Speicher-Subsystem geschrieben worden, eine Aufrufeinrichtung zum Aufrufen ausschließlich von Anwenderdaten aus dem mit den Zugriffseinrichtungen gelesenen Datensatz sowie eine Interpretier- und Nutzungseinrichtung zum Interpretieren und Benutzen von Managementinformationen eines Dateisystems und dergleichen, die in einen Datensatz geschrieben sind, der von dem ersten Computer in dem Speicher-Subsystem erzeugt worden ist. Zusätzlich weist der zweite Computer eine Schreibeinrichtung auf, um Schreibdaten in einen Datensatz im CKD-Format zu konvertieren und den Datensatz über ein Interface für Daten im FBA-Format in das Speicher-Subsystem zu schreiben.

[0009] Weiterhin bezieht sich bei dem Datenschreibvorgang die Schreibeinrichtung des zweiten Computers auf Managementinformationen in einer von dem ersten Computer in dem Speicher-Subsystem erzeugten Datei; sie interpretiert ein Datenformat der Datei und schreibt darin Datensätze in einem der Datei entsprechenden Datenformat. Darüber hinaus ist es durch Rückschreiben der Managementinformationen in der Datei nach Abschluss des Schreibvorgangs des zweiten Computers möglich, dass der erste Computer den von dem zweiten Computer geschriebenen Datensatz lesen kann.

[0010] Darüber hinaus weist das Speicher-Subsystem des Computersystems nach der vorliegenden Erfindung eine Eins-zu-Eins-Entsprechung zwischen einer durch eine Zylindernummer und eine Spurnummer im CKD-Format angegebenen Adresse und einer durch eine logische Blocknummer im FBA-Format angegebenen Adresse auf. Die Schreibeinrichtung des zweiten Computers behandelt angeforderte Daten als einen Datensatz im CKD-Format, erzeugt ein Zählfeld eines Datensatzes im CKD-Format entsprechend einer Adresse einer Schreibposition, um so durch Konvertierung einen Datensatz im CKD-Format zu erhalten, und berechnet eine logische Blocknummer im FBA-Format in Zusammenhang mit einer Zylindernummer und einer Spurnummer für einen Schreibvorgang in dem Speicher-Subsystem, um entsprechend der logischen Blocknummer auf das Speicher-Subsystem zuzugreifen.

[0011] Zusätzlich weist das Speicher-Subsystem des Computersystems nach der vorliegenden Erfindung einen Speicher zum Speichern von Daten im FBA-Format auf, und der erste Computer besitzt eine Schreibeinrichtung, die einen von dem ersten Computer über das Interface für Daten im CKD-Format in eine Datei geschriebenen Datensatz in eine Folge von Daten im FBA-Format in einem vorbestimmten Format konvertiert, um so die Folge von Daten in den Speicher zu schreiben. Weiterhin umfasst die Schreibeinrichtung des zweiten Computers eine ähnliche Einrichtung wie das Speicher-Subsystem, um einen Datensatz in eine Folge von Daten im FBA-Format zu konvertieren, so dass die konvertierte Folge von Daten im FBA-Format über ein Interface für Daten im FBA-Format in das Speicher-Subsystem geschrieben wird.

[0012] Darüber hinaus unterteilt die Schreibeinrichtung den Datensatz im CKD-Format zum Schreiben in Blöcke fester Länge. Wenn der letzte der unterteilten Datensätze eine Länge hat, die kleiner als die feste Länge ist, wird eine vorbestimmte Folge von Daten beginnend am Ende der Daten hinzugefügt, um im Ergebnis einen Block fester Länge zu erhalten, damit der Block in das Speicher-Subsystem geschrieben werden kann.

KURZBESCHREIBUNG DER ZEICHNUNGEN

[0013] Die Ziele und Merkmale der vorliegenden Erfindung werden aus der folgenden detaillierten Beschreibung besser verständlich, wenn diese in Verbindung mit den zugehörigen Zeichnungen gelesen wird.

[0014] **Fig.** 1 ist ein Konfigurationsdiagramm, das den Aufbau eines Plattenspeichersystems nach der vorliegenden Erfindung zeigt.

[0015] **Fig.** 2 ist ein Diagramm, das ein Layout von in einem Speicher-Subsystem gespeicherten Daten im CKD-Format zeigt.

[0016] **Fig.** 3 ist ein Diagramm, das ein Datenlayout von auf einem Plattengerät gespeicherten Datensätzen zeigt.

[0017] **Fig.** 4 ist ein Ablaufdiagramm, das die Verarbeitungsabläufe beim Öffnen eines Datensatzes zeigt.

[0018] **Fig.** 5 ist ein Ablaufdiagramm, das die Verarbeitungsabläufe beim Schreiben eines Datensatzes zeigt.

[0019] **Fig.** 6 ist ein Ablaufdiagramm, das die Verarbeitungsabläufe beim Schließen eines Datensatzes zeigt.

BESCHREIBUNG DER BEVORZUGTEN AUSFÜHRUNGSFORMEN

[0020] **Fig.** 1 zeigt ein Konfigurationsbeispiel eines Computersystems nach der vorliegenden Erfindung. Das Computersystem umfasst einen Host-Computer **1**, einen Host-Computer **2** und ein Speicher-Subsystem **3**, das mit den Host-Computern **1** und **2** verbunden ist.

[0021] Der Host-Computer **1** weist ein Interface **11** für Daten im Count-Key-Format (im Folgenden abgekürzt als CKD) als eine Schnittstelle für das Speicher-Subsystem **3** auf und arbeitet entsprechend dem CKD-Format, wobei der Host-Computer **1** durch ein Betriebssystem **12** gesteuert wird. Zusätzlich weist das System ein Anwendungsprogramm **13** auf, das unter der Steuerung des Betriebssystems **12** arbeitet. Obwohl es weitere unvermeidliche Bestandteile wie z. B. eine Zentraleinheit (CPU) in dem Computer gibt, sind diese für die Be-

schreibung der Ausführungsform der Erfindung nicht von besonderer Bedeutung und werden daher in dieser Anmeldung nicht beschrieben.

[0022] Auf der anderen Seite weist der Host-Computer **2** ein Interface **21** für Daten im FBA-Format als eine Schnittstelle für das Speicher-Subsystem **3** auf und arbeitet entsprechend dem FBR-Format, wobei der Host-Computer **2** durch ein Betriebssystem **22** gesteuert wird. Außerdem gibt es ein Anwendungsprogramm **23** und ein Formatkonvertierungsprogramm **24**, die unter der Steuerung des Betriebssystems arbeiten.

[0023] Das Speicher-Subsystem **3** weist eine Platte **31**, ein CKD-Interface **32** für die Verbindung zum Host-Computer **1** und ein FBA-Interface **33** für die Verbindung zum Host-Computer **2** auf.

[0024] Beim Fixed-Length-Format werden alle Daten in einem als „Block" bezeichneten Bereich gespeichert. Jeder Block hat eine feste Länge von z. B. 512 Bytes und ist mit einer Blocknummer versehen (im Folgenden abgekürzt als LBA), wobei die Blocknummer für den Zugriff auf den Block verwendet wird.

[0025] Andererseits erfolgt beim CKD-Format der Zugriff auf einen Datensatz anhand der Angabe einer Zylindernummer (CC), einer Spurnummer (HH) und einer Datensatznummer (R). Die kleinste Zugriffseinheit ist ein Datensatz. In der nachfolgenden Beschreibung wird ein durch die Zylinder-, Spur- und Datensatznummer angegebener Datensatz als CCHHR bezeichnet, und eine durch eine Zylindernummer und eine Spurnummer angegebene Spuradresse wird als CCHH bezeichnet. Im CKD-Format umfasst ein Datensatz ein Zählfeld (im Folgenden als Feld C bezeichnet), ein Schlüsselfeld (im Folgenden als Feld K bezeichnet) und ein Datenfeld (im Folgenden als Feld D bezeichnet). Feld C umfasst CCHHR und die Längen der Felder K und D und hat in jedem Fall eine feste Länge. Jedes der Felder K und D hat eine variable Länge und eine in das Feld C geschriebene Länge. Eine Spur besteht aus mehreren Datensätzen. Weil die Spur jedoch eine feste Länge aufweist, während die Datensätze gegenseitig unterschiedliche Längen haben, variiert die Anzahl der Datensätze pro Spur zwischen den Spuren. Informationen über die Anzahl der Datensätze und deren Positionen in der Spur werden nicht verwaltet. Wenn daher ein Datensatz an einem zwischenliegenden Punkt auf der Spur aufgerufen werden soll, müssen die Datensätze ab der ersten Position gelesen werden, d. h. das Zählfeld muss gelesen werden, um die Position des nächsten Datensatzes zu erhalten.

[0026] Darüber hinaus weist ein Zylinder mehrere Spuren auf. Die Anzahl der Spuren pro Zylinder ist fest.

[0027] **Fig.** 2A zeigt die Datenanordnung in einer Spur auf einem allgemeinen CKD-Plattengerät, während **Fig.** 2B ein Datenformat zum Speichern von Daten im CKD-Format in dem Speicher-Subsystem nach der erfindungsgemäßen Ausführungsform zeigt. Die Platte **31** entspricht einem Fixed-Length-Format (FBA), und Daten im CKD-Format werden in Daten im FBA-Format konvertiert, um sie in dem Speicher zu speichern.

[0028] In **Fig.** 2A bezeichnet HA **51** eine Spuradresse, die z. B. eine Position einer Spur angibt. R0C **52** ist ein Zählfeld des Datensatzes 0, und R0D **53** bezeichnet ein Datenfeld des Datensatzes 0, in dem keine Anwenderdaten gespeichert werden dürfen. R1C **54**, R1K **55** und R1D **56** sind ein Zählfeld, ein Schlüsselfeld bzw. ein Datenfeld in Datensatz 1. Zwischen den einzelnen Feldern gibt es einen als „Lücke" bezeichneten Bereich fester Länge, in dem keine Daten gespeichert sind, d. h. die Lücke ist eine Abgrenzung, um die Felder voneinander zu trennen. Darüber hinaus ist an einem Ende jedes der Felder C, K und D zusätzlich ein Fehlerkorrekturcode (ECC) **60** vorgesehen. Der Inhalt, der ECC und die Lücke von Feld C werden in dem Plattengerät erzeugt.

[0029] In dem Speicher-Subsystem werden HA **51**, R0C **52** und R0D **53** für jede Spur zusammengestellt, um so in einem Bereich gespeichert zu werden, dass HA **51**, R0C **52** und R0D **53** über das Interface **33** für Daten im FBA-Format nicht sichtbar sind. Beim Zugriff auf LBA0 von dem Interface **33** erfolgt der Zugriff auf die erste Position gemäß der Angabe durch Zylindernummer 0, Spurnummer 0 und Datensatznummer 1. Überschreitet ein Datensatz die Größe eines Blocks wie R1D in **Fig.** 2, werden die Daten fortlaufend in den nächsten Block geschrieben, d. h. LBA1 **62**. In dem Datensatz gibt es keine Lücke zwischen den Feldern C, K und D. Weil die Größe jedes Datensatzes nicht unbedingt mit der Grenze des Blocks fester Länge übereinstimmt, wird ein Bereich vom Ende des Datensatzes bis zum Ende des Blocks mit einem Leerbereich **65** gefüllt, d. h. 0 wird darin als Lücke angeordnet. Datensatz 2 beginnt dann mit dem nächsten Block, d. h. LBA2 **63**. Schreibt der Host-Computer **1** einen Datensatz in das Speicher-Subsystem **3**, erzeugt das CKD-Interface **32** Fehlerkorrekturcodes ECC, die in die Felder C und D an deren Endposition eingefügt werden, und der Leerbereich **65** wird hinzugefügt, um einen Block fester Länge zu erhalten, um so den Datensatz auf die Platte **31** zu schreiben. Auch wenn der Inhalt des Leerbereichs keinen Einschränkungen unterliegt, wird bei dieser Ausführungsform 0 verwendet. Bei einem Lesevorgang liest das CKD-Interface **32** einen Block fester Länge, extrahiert die Felder K und D und gibt diese Felder an den Host-Computer **1** weiter. Daher braucht der Host-Computer **1** das Schreibdatenformat und die Position des Datensatzes auf der Platte **31** nicht zu berücksichtigen. Weil hingegen bei einem Vorgang, bei dem der Host-Computer **2** einen Datensatz aus dem Speicher-Subsystem **3** liest, z. B. LBA0 in **Fig.** 2B, der Lesevorgang nur in der Einheit eines Blocks ausgeführt werden kann, muss der gesamte Inhalt von LBA0, d. h. der Inhalt einschließlich Feld C und ECC an den Host-Computer **2** weitergegeben werden. Bei einem Datenschreibvorgang werden die von dem Host-Computer **2** weitergegebenen Daten direkt in LBA0 geschrieben.

[0030] Die Spurlänge ist ein ganzzahliges Vielfaches der Blocklänge, und daher weist jeder Block eine iden-

tische Anzahl von Blöcken auf. Stimmt das Ende des Datensatzes in der Spur nicht mit dem Ende der Spur überein, wird die Spur bis zum Ende mit 0 aufgefüllt, so dass alle Spuren dieselbe Anzahl von Blöcken aufweisen. Darüber hinaus besitzt jeder Zylinder eine identische Anzahl von Spuren.

[0031] Um daher auf die erste Position von CCHHR = (C, H, 1) zuzugreifen, muss folglich nur

$$LBA = (C \times \text{Anzahl der Spuren pro Zylinder} + H) \times \text{Anzahl der Blöcke pro Spur} \quad (1)$$

berechnet werden.

[0032] **Fig.** 3 zeigt Daten und Dateien (Datenmengen) eines auf der Platte **3** durch den Host-Computer **1** erzeugten Dateisystems. An einer bestimmten Position auf der Platte **31** befindet sich ein Datenträgeretikett **71**, in das eine Schreibadresse eines Datenträger-Inhaltsverzeichnisses (VTOC **72**) in der Form von CCHHR geschrieben ist. Das VTOC **72** enthält ausschließlich aufgezeichnete Managementinformationen für die auf die Platte **31** geschriebenen Dateien, zum Beispiel eine Gruppe von Datensätzen mit je einem 44 Byte großen Schlüsselfeld und einem 96 Byte großen Datenfeld. Das Schlüsselfeld enthält einen Dateinamen, und das Datenfeld enthält Informationen wie z. B. Positionsinformationen für die Datei. Als Positionsinformationen sind dort eine Datensatzlänge, eine Position eines Bereichs und eine Position eines zuletzt in den Bereich geschriebenen Datensatzes eingeschrieben. Eine Gruppe dieser Datenelemente wird als DSCB bezeichnet.

[0033] Eine Datei befindet sich in einem als „Bereich" (englisch „extent) bezeichneten Bereich. Ein Bereich ist ein zusammenhängender Bereich mit mehreren Spuren. Je nach Situation ist ein Bereich bzw. sind zwei oder mehr Bereiche für eine Datei vorgesehen. Wenn mehrere Bereiche vorgesehen sind, müssen diese Bereiche nicht in einem zusammenhängenden Bereich liegen.

[0034] Wenn das Anwendungsprogramm **13** des Host-Computers **1** eine Datei liest, werden eine Zugriffsstartposition und eine letzte Datensatzposition aus den Positionsinformationen der zugehörigen Daten in der VTOC **72** erhalten; damit wird die CCHHR einer Leseposition bestimmt, um die Datei zu lesen. Diese Verarbeitungsschritte werden von dem Betriebssystem **12** ausgeführt. Auch wenn die Daten auf der Platte **31** Informationen für Feld C und ECC enthalten, die für den Host-Computer **1** nicht benötigt werden, wird dieser Teil von den CKD-Interfaces **11** und **32** entfernt. Dies bedeutet, dass nur die eigentlichen Daten in den Feldern K und D an das Betriebssystem **12** des Host-Computers **1** weitergegeben werden, die nicht benötigten Informationen jedoch nicht an diesen weitergegeben werden.

[0035] Andererseits kennt das Betriebssystem **22** des Host-Computers **2** weder die Struktur des Dateisystems auf dem Host-Computer **1** noch die Konfiguration der im CKD-Format auf der Platte **31** gespeicherten Daten; daher ist es in diesem Fall nicht möglich, das Verfahren zum Zugriff auf ein Dateisystem auf dem Host-Computer **2** zu benutzen. Zur Überwindung dieser Schwierigkeit ist ein Formatkonvertierungsprogramm **24** in der Art vorgesehen, dass das Anwendungsprogramm **23** des Host-Computers **2** über das Programm **24** auf eine Datei auf dem Host-Computer **1** zugreift. Das Programm **24** arbeitet mit einem ähnlichen Verfahren wie das Speicher-Subsystem **3**, um Daten im CKD-Format auf die Platte **31** zu schreiben und CKD-Daten von der Platte **31** zu lesen. Das Programm **24** umfasst ein Verfahren zum Lesen von Daten in einer Blockeinheit aus dem Speicher-Subsystem **3**, um den eigentlichen Datenteil aus den Daten in einem in **Fig.** 2B gezeigten Format zu erhalten, und ein Verfahren, bei dem die von dem Anwendungsprogramm **23** weitergegebenen Daten als ein Datensatz behandelt werden, der in einen CKD-Datensatz konvertiert werden soll, so dass der CKD-Datensatz weiter in einen Datensatz im FBA-Format umgewandelt wird, der auf die Platte **31** geschrieben werden soll. Darüber hinaus umfasst das Formatkonvertierungsprogramm ein Verfahren zum Interpretieren von Managementinformationen einer in das VTOC **72** geschriebenen Datei, um eine Position zu erhalten, an der die Datei geschrieben wird. Als Nächstes wird anhand von Ablaufdiagrammen der Ablauf eines von dem Konvertierungsprogramm **24** an einer Datei durchgeführten Schreibvorgangs in drei Abschnitten der Verarbeitungsabläufe beim Öffnen, Schreiben und Schließen beschrieben.

[0036] **Fig.** 4 ist ein Ablaufdiagramm der Verarbeitungsschritte beim Öffnen einer Datei durch das Programm **24**.

[0037] Das Programm **24** hat einen Spurpuffer zur Aufnahme von Ein-Spur-Daten, einen Blockpuffer zur Aufnahme der einem DSCB entsprechenden Datenblöcke, ein Set (einen sog. Datensatzzeiger) aus einer Spurnummer und einer Datensatznummer einer aufzurufenden Datei und einen Zeiger (einen sog. Pufferzeiger), der eine Position in dem Spurpuffer für die durch den Datensatzzeiger angegebene Datensatznummer angibt. Bei der Öffnungsverarbeitung wird zuerst eine Spur einschließlich des Datenträgeretiketts **71** gelesen, um die Position der VTOC zu kontrollieren. Das Etikett **71** befindet sich immer an einer festen Position, und das Konvertierungsprogramm **24** kennt im Voraus die durch CCHHR auf der Platte **31** angegebene Position des Datenträgeretiketts **71**. Es wird zum Beispiel angenommen, dass sich das Datenträgeretikett **71** an einer Position befindet, die durch Zylindernummer 0, Spurnummer 0 und Datensatznummer 3 angegeben ist. Das Programm **24** berechnet entsprechend dem Ausdruck (1) den LBA für Zylinder Nr. 0 und Kopf Nr. 0, d. h. LBA Nr. 0 (Schritt **101**). Danach liest es Blöcke in einer Spur, beginnend an der Position, und speichert die Blöcke im Spurpuffer (Schritt **102**). Anschließend wird durch Kontrollieren von Feld C in Datensatz 1 an einer ersten Position des

erhaltenen Blocks und danach der Länge von Feld D eine Blockposition erhalten, in der Datensatz 2 gespeichert ist (Schritt **103**). Der obige Vorgang wird nun wiederholt, d. h. das erste Feld von Datensatz 2 wird gelesen, um die Länge von Feld D entsprechend Feld C zu ermitteln, um dadurch eine Blockposition von Datensatz 3 zu kontrollieren (Schritt **104**). Als Ergebnis ist Feld D von Datensatz 3 bekannt.

[0038] Als Nächstes wird eine Speicherposition von VTOC **72** aus Feld D in Datensatz 3 gelesen (Schritt **105**), um anhand von Ausdruck (**1**) den der Position entsprechenden LBA zu berechnen (Schritt **106**). Danach werden beginnend an der LBA Blöcke in einer Spur gelesen, die in den Spurpuffer gegeben werden (Schritt **107**). Im Anschluss daran wird über den Spurpuffer ein Abruf vorgenommen, um einen Dateinamen für den Öffnungsvorgang zu erhalten. In einem vorbereiteten Zähler r wird ein Ausgangswert von eins auf r eingestellt (Schritt **108**). Das Schlüsselfeld von Datensatz r der erhaltenen Spur wird kontrolliert, um festzustellen, ob das Schlüsselfeld mit dem Namen der Zieldatei übereinstimmt (Schritt **109**). Wird ein zugehöriger Datensatz gefunden, wird eine Speicherposition der Datei aus Feld D erhalten, um die Felder C, K und D des Datensatzes vollständig in einem internen Puffer zu speichern (Schritt **110**). Fehlt der Datensatz, wird der Wert von r um eins erhöht (Schritt **115**), und die Steuerung wird an Schritt **109** übergeben, um einen nächsten Datensatz abzurufen. Zusätzlich wird, wenn der Datensatz fehlt, auch wenn die Prüfung bis zum Ende der Spur der zugehörigen Spur durchgeführt wird, eine nachfolgende Spur von der Platte gelesen, um einen Dateiabruf beginnend mit Datensatz 1 durchzuführen (Schritte **113** und **114**).

[0039] Weiterhin werden anhand von **Fig.** 5 die Verarbeitungsschritte beim Schreiben beschrieben. Das Formatkonvertierungsprogramm **24** erhält eine Datensatzlänge und Dateninhalte von dem Anwendungsprogramm **23** und führt daraufhin den Schreibvorgang in einem Speichergerät aus. Weil der Schreibvorgang auf der Platte **31** in einer Ein-Spur-Einheit erfolgt, bei der es sich um ein ganzzahliges Vielfaches von Blöcken handelt, bleiben, wenn in der Vergangenheit Schreibanforderungen erteilt worden sind, die für die Schreibanforderungen erhaltenen Daten im Spurpuffer gespeichert. Beim Schreibvorgang wird zuerst geprüft, ob die erhaltenen zugehörigen Daten in den Spurpuffer geschrieben werden können oder nicht (Schritt **201**). Können die Daten nicht in den Puffer geschrieben werden, wird der aktuelle Inhalt des Spurpuffers von der Schreibposition bis zum Pufferende gelöscht und auf null gesetzt (Schritt **202**). Danach wird der Spurpuffer in das Speicher-Subsystem **3** geschrieben (Schritt **203**), der Pufferzeiger auf die erste Position des Spurpuffers (Schritt **204**) zurückgesetzt und der Datensatzzeiger auf Datensatz 1 der nächsten Spur eingestellt (Schritt **205**). Der im Spurpuffer erzeugte Inhalt wird direkt auf die Platte **31** geschrieben.

[0040] In Schritt **206** und den folgenden Schritten erfolgt die Verarbeitung zum Schreiben von Daten im CKD-Format in den Spurpuffer. Zuerst werden die Informationen in Feld C anhand des Datensatzzeigers und der Datensatzlänge erzeugt, die dann an einer Position in den Spurpuffer geschrieben werden, die durch den Pufferzeiger angegeben ist (Schritt **206**), und die erhaltenen Daten werden darin hinter Feld C geschrieben (Schritt **207**). Der nächste Datensatz muss darin beginnend an der ersten Position des nächsten Blocks geschrieben werden. Wenn der Block nicht mit Daten gefüllt ist, wird folglich bis zur ersten Position des nächsten Blocks 0 darin eingetragen (Schritt **208**). Als Letztes wird der Pufferzeiger auf die erste Position des nächsten Blocks eingestellt (Schritt **209**), ehe der Datensatzzeiger auf den nächsten Datensatz eingestellt wird (Schritt **210**).

[0041] Im Folgenden werden anhand von **Fig.** 6 die Verarbeitungsschritte des Formatkonvertierungsprogramms **24** beim Schließen beschrieben. Zuerst wird kontrolliert, ob mit dem bis zu diesem Punkt durchgeführten Schreibverfahren Daten in den Spurpuffer geschrieben worden sind (Schritt **301**). Sind Daten in den Puffer geschrieben worden, wird der Inhalt des Puffers beginnend ab der durch den aktuellen Pufferzeiger angegebenen Position bis zum Ende des Spurpuffers gelöscht und auf null zurückgesetzt (Schritt **302**), ehe der Spurpuffer auf die Platte geschrieben wird (Schritt **303**). Als Nächstes werden zum Rückschreiben der letzten Datensatzposition in der VTOC **72** Informationen über die letzte Datensatzposition in den Inhalt des gespeicherten Blockpuffers zurückgeschrieben (Schritt **304**), ehe der Blockpuffer auf die Platte geschrieben wird (Schritt **305**). Nach diesem Vorgang ist die Verarbeitung beim Schließen abgeschlossen.

[0042] Mit dem Computersystem nach der vorliegenden Erfindung können Daten zwischen Systemen ausgetauscht werden, die mit Speicherverfahren für unterschiedliche Datenformate, nämlich dem CKD- und dem FBA-Format, arbeiten. Dies ermöglicht auf vorteilhafte Weise einen flexiblen Datenaustausch mit einer höheren Leistung.

[0043] Auch wenn die vorliegende Erfindung anhand bestimmter, der Illustration dienender Ausführungsformen beschrieben worden ist, ist sie nicht durch diese Ausführungsformen, sondern nur durch die anliegenden Ansprüche eingeschränkt.

Fig. 1

Bezugszeichenliste

1 Host 1
13 Anwendungsprogramm
12 Betriebssystem
11 Interface für Daten im CKD-Format
2 Host 2
23 Anwendungsprogramm
24 Formatkonvertierungsprogramm
22 Betriebssystem
21 Interface für Daten im FBA-Format
32 Interface für Daten im CKD-Format
33 Interface für Datenim FBA-Format
3 Speicher-Subsystem

FIG. 2

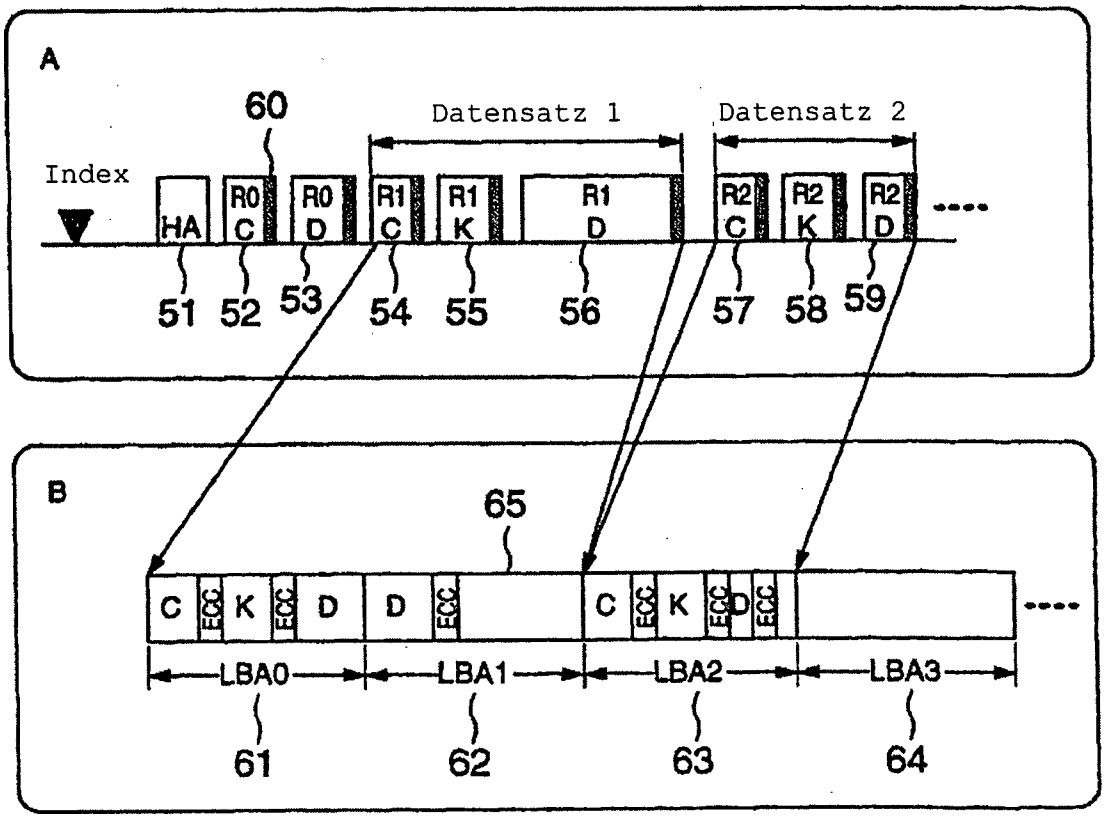


Fig. 3

Bezugszeichenliste

71 Datenträgeretikett
72 Datenträger-Inhaltsverzeichnis (VTOC)
Bereich
Bereich
Bereich
Bereich

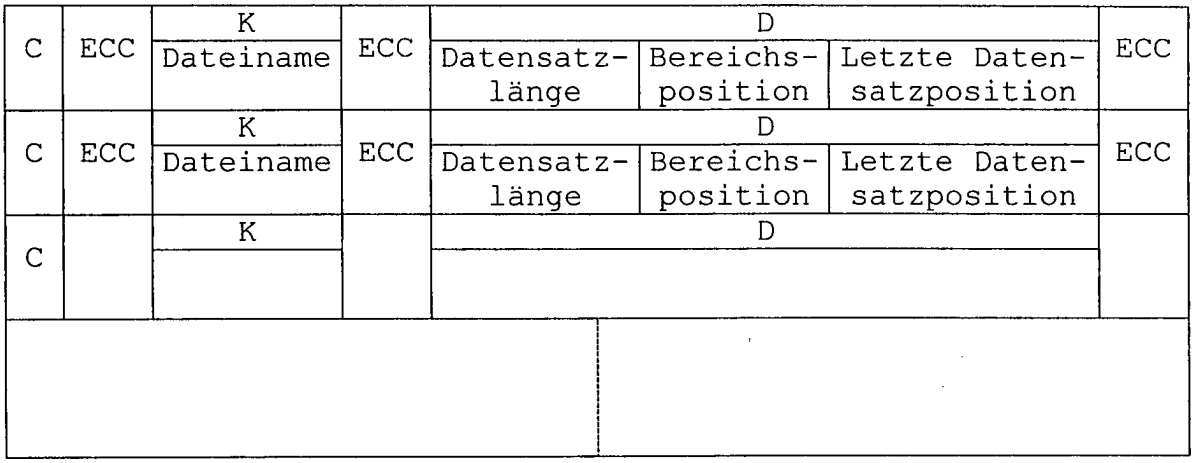


Fig. 4

Bezugszeichenliste

Start der Verarbeitung beim Öffnen
101 Berechnen der LBA anhand der Adresse (CCHH)der Datenträgeradresse
102 Lesen von Blöcken in einer Spur beginnend ab der berechneten LBA
103 Ermittelnder Position von R2 entsprechend Feld C des ersten Blocks (R1)
104 Ermitteln der Position von R3 entsprechend Feld C von R2
105 Ermitteln der Adresse der VTOC entsprechend Feld D von R3

106 Berechnen der LBA anhand der Adresse (CCHH) der VTOC

107 Lesen von Blöcken in einer Spur beginnend ab der berechneten LBA
108 r = 1
109 Feld K von Datensatz R stimmt mit dem Dateinamen überein?
JA NEIN
110 Ermitteln der Dateispeicherposition anhand von Feld D
111 Einstellen des internen Zeigers auf die relative Spur R1
Ende der Verarbeitung beim Öffnen
112 Letzter Datensatz in der Spur geprüft?
JA NEIN
113 Nächste Spur lesen
114 r = 1
115 r ← + 1
Ende der Verarbeitung beim Öffnen

Fig. 5

Bezugszeichenliste

Start der Verarbeitung beim Schreiben
201 Zieldatensatz kann in den Spurpuffer geschrieben werden?
JA NEIN
202 Feld bis zum Ende des Spurpuffers löschen und auf null setzen
203 Inhalt des Spurpuffers auf Platte schreiben
204 Pufferzeiger auf Ausgangspunkt des Spurpuffers zurücksetzen
205 Datensatzzeiger auf R1 der nächsten Spur einstellen
206 Feld C anhand von CCHHR und Datensatzlänge erzeugen und Feld C in den Spurpuffer schreiben
207 Feld D in den Spurpuffer schreiben

208 Feld bis zum ersten Punkt des nächsten Blocks löschen und auf null setzen
209 Pufferzeiger auf den nächsten Block einstellen
210 Datensatzzeiger auf den ersten Punkt des nächsten Datensatzes einstellen
Ende der Verarbeitung beim Schreiben

Fig. 6

Bezugszeichenliste

Start der Verarbeitung beim Schließen
301 Spurpuffer enthält Daten, die auf die Platte geschrieben werden sollen?
JA NEIN
302 Feld bis zum Ende des Spurpuffers löschen und auf null setzen
303 Inhalt des Spurpuffers auf Platte schreiben
304 Letzte Datensatzposition in DSCB schreiben
305 Geänderten Inhalt von DSCB auf Platte schreiben
Ende der Verarbeitung beim Schließen

Patentansprüche

1. Verfahren zum Steuern eines Computersystems mit einem ersten Computer (**1**), der ein erstes Interface (**11**) für Daten im Count-Key-Format (CKD) aufweist, einem zweiten Computer (**2**), der ein zweites Interface (**21**) für Daten im Fixed-Length-Format (FBA) aufweist, und einem Speicher-Subsystem (**3**), das ein drittes, an das erste Interface angeschlossenes Interface (**32**) und ein viertes, an das zweite Interface angeschlossenes Interface (**33**) aufweist, **dadurch gekennzeichnet**, daß der zweite Computer (**2**):
ein Zählfeld auf Grundlage von durch ein Anwendungsprogramm (**23**) erzeugten Daten erzeugt wird;
ein Datensatz im CKD-Format erzeugt wird, der das Zählfeld und die von dem Anwendungsprogramm (**23**) erzeugten Daten enthält;
der Datensatz in einen Block fester Länge konvertiert wird;
der konvertierte Datensatz über das zweite Interface (**21**) in das Speicher-Subsystem (**3**) geschrieben wird; und
den Datensatz betreffende Managementinformation in eine Tabelle (**72**) geschrieben wird, die in dem Speicher-Subsystem (**3**) gespeichert und von dem ersten Computer (**1**) beim Lesen des Datensatzes aus dem Speicher-Subsystem (**3**) verwendet wird.

2. Verfahren nach Anspruch 1, wobei
mehrere Blöcke fester Länge in einem in dem zweiten Computer (**2**) angeordneten Puffer gespeichert werden, und
der Datensatz als Block fester Länge so gespeichert wird, daß eine Startposition eines der mehreren Blöcke fester Länge eine Startposition des Datensatzes wird.

3. Verfahren nach Anspruch 2, wobei, wenn ein Ende des in dem einen Block fester Länge gespeicherten Datensatzes nicht mit einem Ende des einen Blocks fester Länge übereinstimmt, eine Lücke zwischen beiden Enden mit Pseudodaten gefüllt wird.

4. Verfahren nach einem der vorstehenden Ansprüche, wobei der Datensatz Information enthält, die die Länge von in dem Datensatz enthaltenen Daten angibt.

5. Computersystem mit:
einem ersten Computer (**1**), der ein erstes Interface (**11**) für Daten im Count-Key-Format (CKD) aufweist;
einem zweiten Computer, der ein zweites Interface (**21**) für Daten im Fixed-Length-Format (FBA) aufweist; und
einem Speicher-Subsystem (**3**), das ein drittes, an das erste Interface angeschlossenes Interface (**32**) und ein viertes, an das zweite Interface angeschlossenes Interface (**33**) aufweist,
dadurch gekennzeichnet, daß der zweite Computer (**2**) Einrichtungen aufweist, um
ein Zählfeld auf Grundlage von durch ein Anwendungsprogramm (**23**) erzeugten Daten zu erzeugen,
einen Datensatz im CKD-Format zu erzeugen, der das Zählfeld und die von dem Anwendungsprogramm (**23**) erzeugten Daten enthält,
den Datensatz in einen Block fester Länge zu konvertieren,
den konvertierten Datensatz in das Speicher-Subsystem (**3**) zu schreiben, und
den Datensatz betreffende Managementinformation in eine Tabelle (**72**) zu schreiben, die in dem Speicher-Subsystem (**3**) gespeichert ist und von dem ersten Computer (**1**) beim Lesen des Datensatzes aus dem Speicher-Subsystem (**3**) verwendet wird.

Es folgen 6 Blatt Zeichnungen

Anhängende Zeichnungen

FIG.1

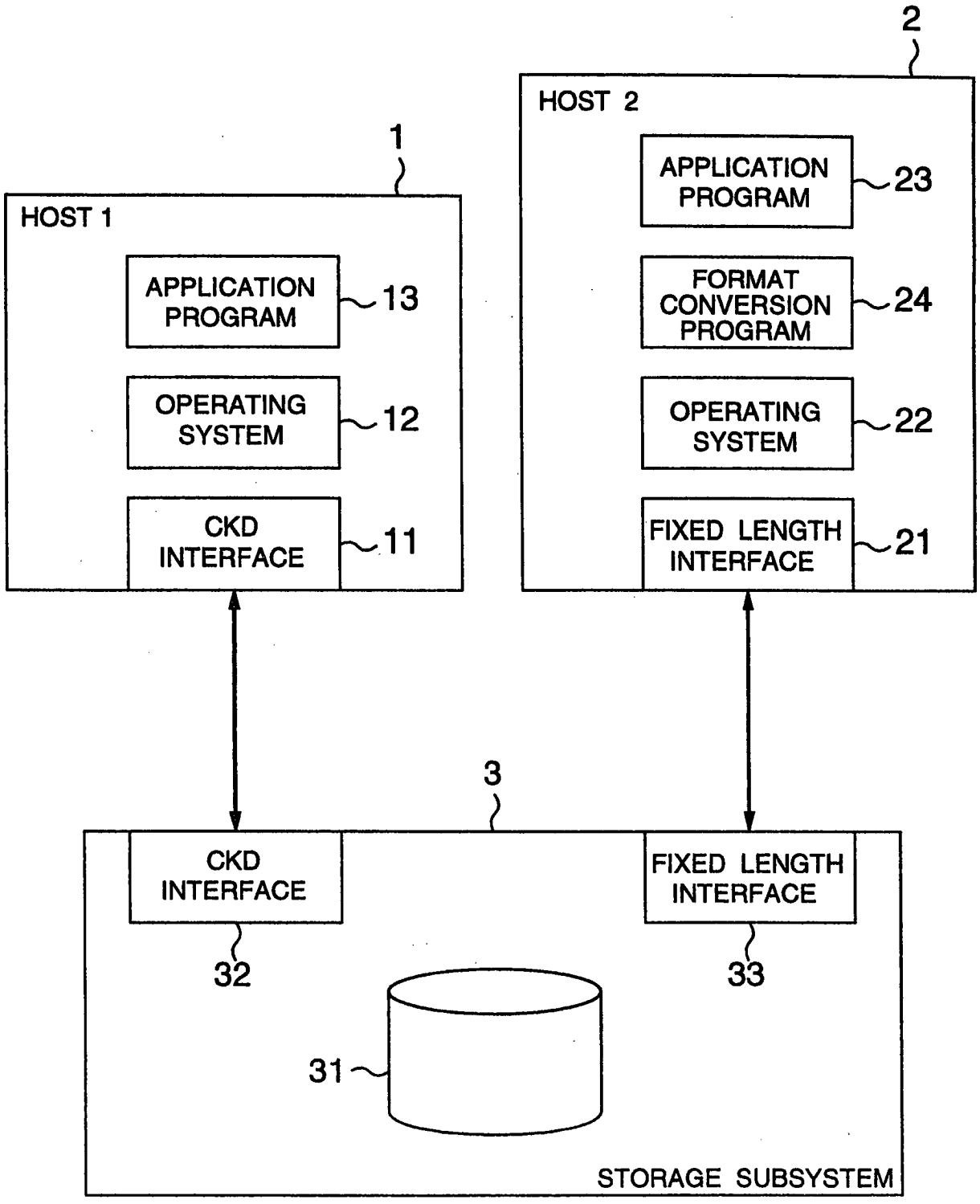


FIG.2

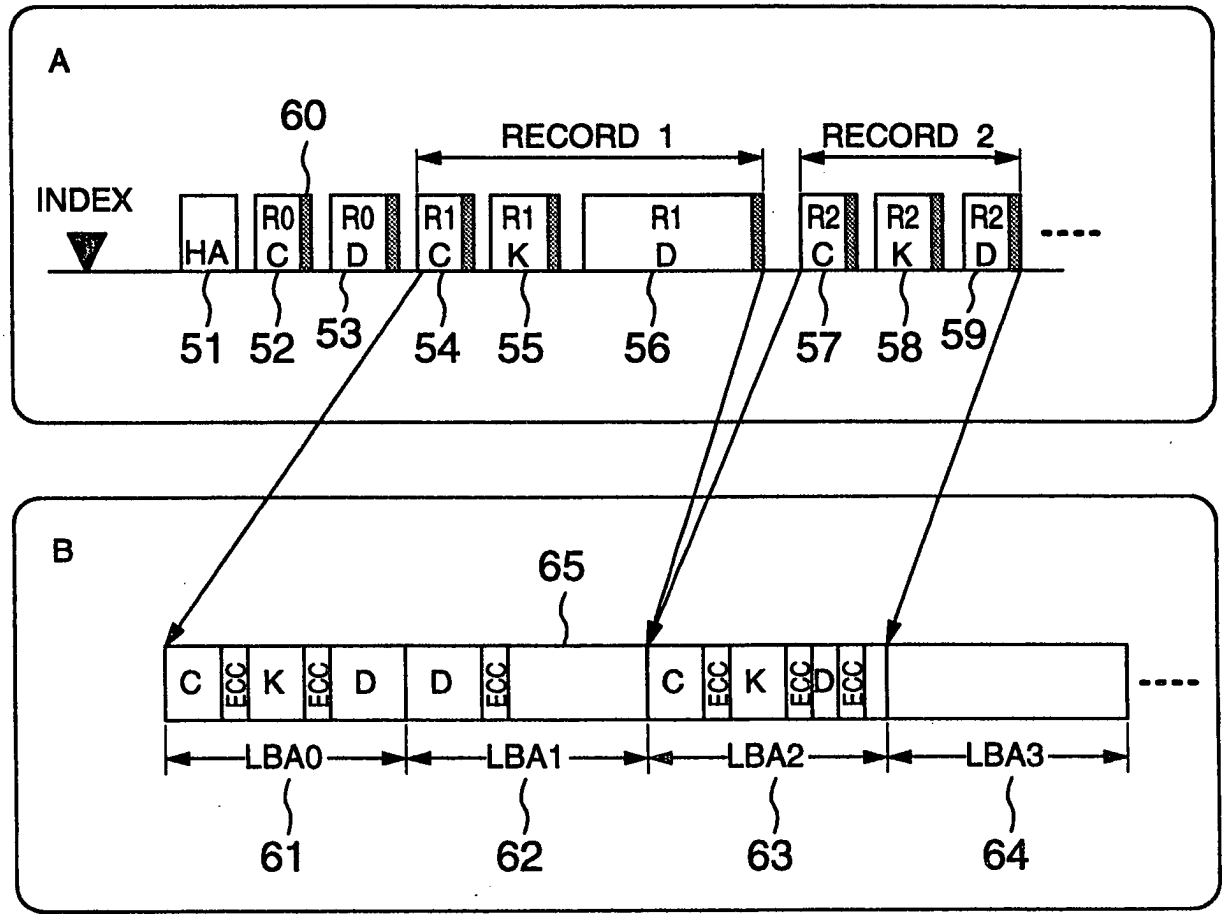

A
60
RECORD 1
RECORD 2
INDEX
HA
R0 C
R0 D
R1 C
R1 K
R1 D
R2 C
R2 K
R2 D
51
52
53
54
55
56
57
58
59
B
65
C
ECC
K
ECC
D
D
ECC
C
ECC
K
ECC
D
ECC
LBA0
LBA1
LBA2
LBA3
61
62
63
64

FIG.3

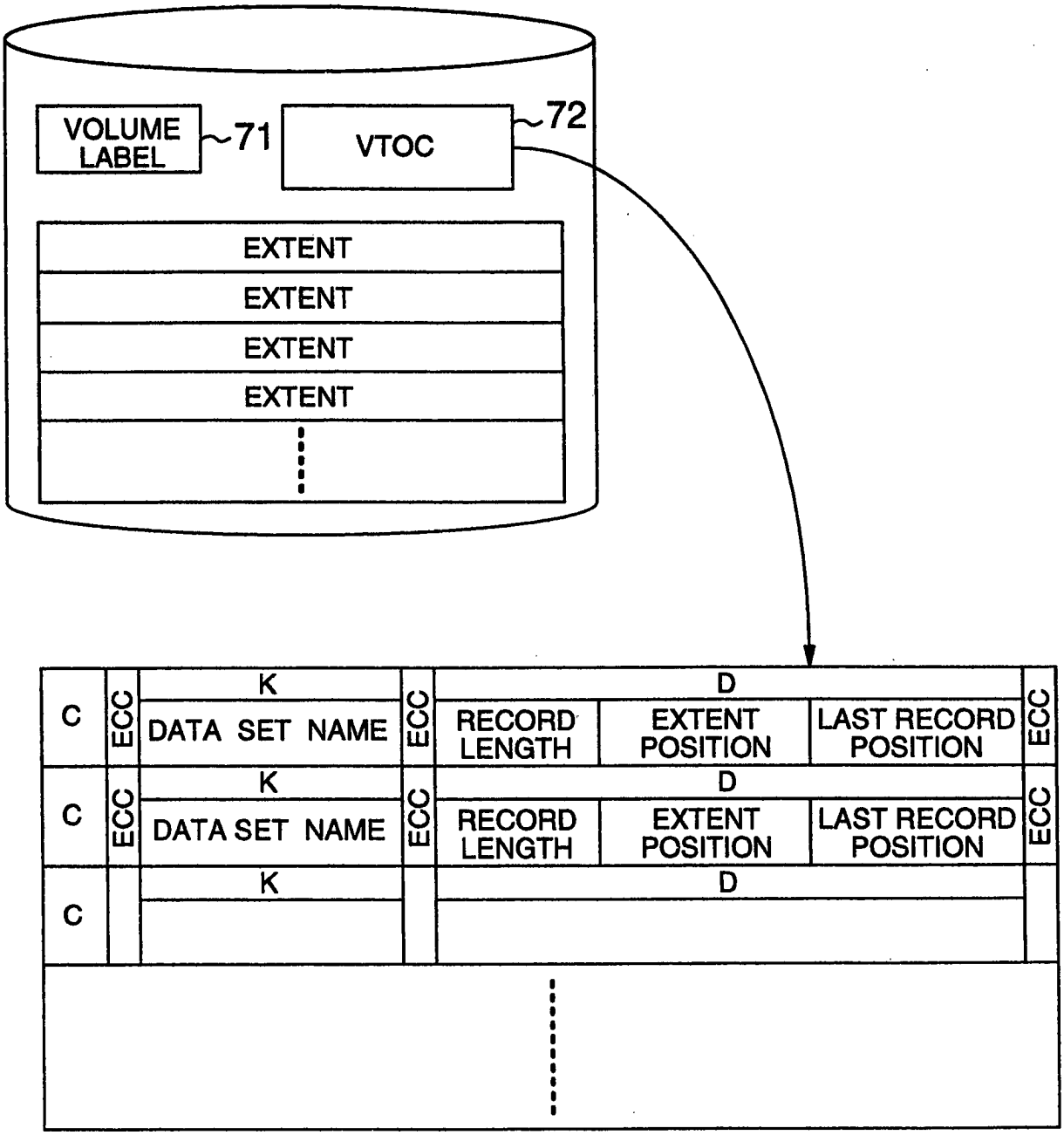

VOLUME LABEL
71
VTOC
72
EXTENT
EXTENT
EXTENT
EXTENT
C
ECC
K
DATA SET NAME
ECC
D
RECORD LENGTH
EXTENT POSITION
LAST RECORD POSITION
ECC

FIG.4

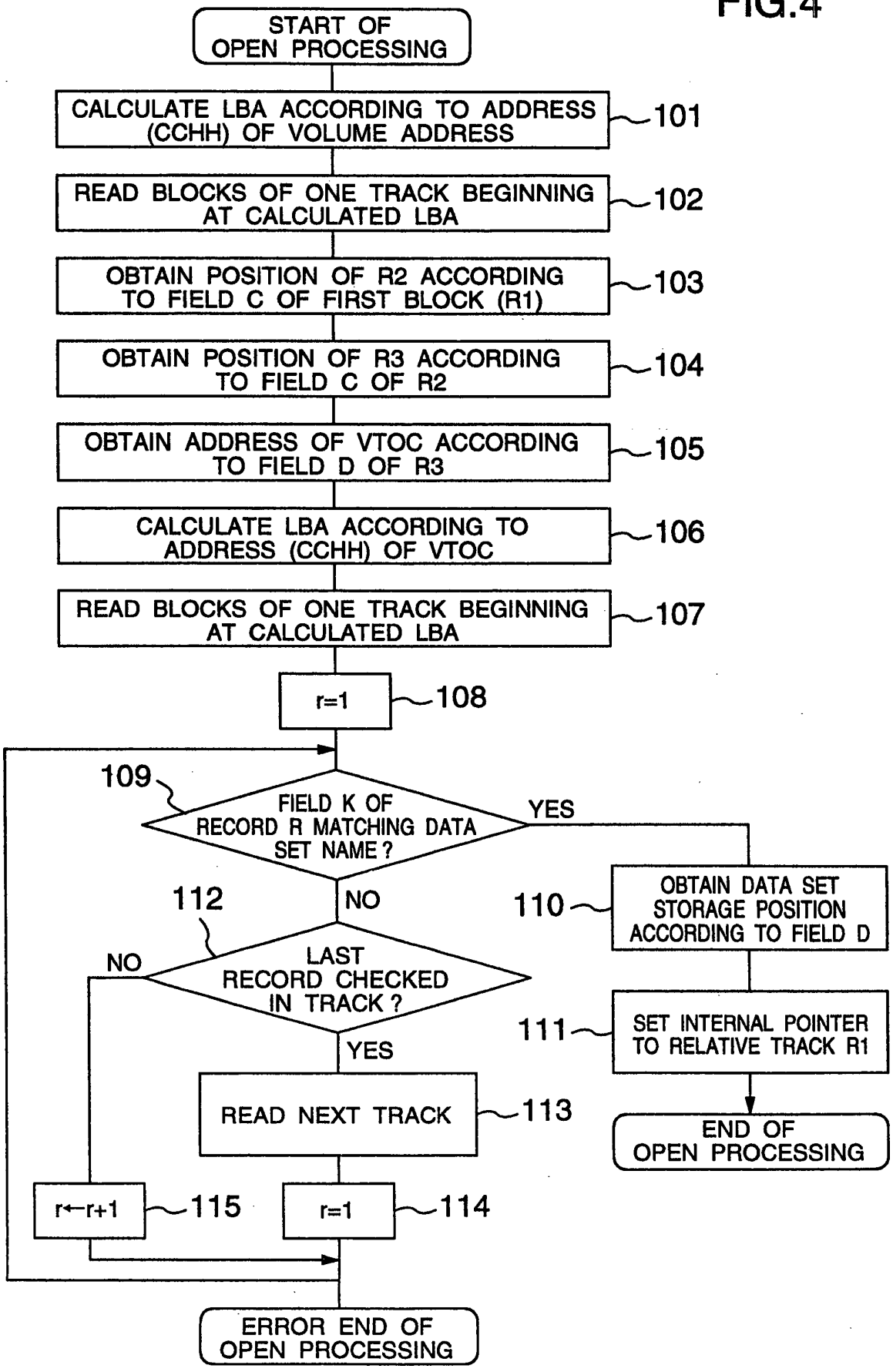

START OF OPEN PROCESSING
CALCULATE LBA ACCORDING TO ADDRESS (CCHH) OF VOLUME ADDRESS
101
READ BLOCKS OF ONE TRACK BEGINNING AT CALCULATED LBA
102
OBTAIN POSITION OF R2 ACCORDING TO FIELD C OF FIRST BLOCK (R1)
103
OBTAIN POSITION OF R3 ACCORDING TO FIELD C OF R2
104
OBTAIN ADDRESS OF VTOC ACCORDING TO FIELD D OF R3
105
CALCULATE LBA ACCORDING TO ADDRESS (CCHH) OF VTOC
106
READ BLOCKS OF ONE TRACK BEGINNING AT CALCULATED LBA
107
r=1
108
109
FIELD K OF RECORD R MATCHING DATA SET NAME?
YES
NO
110
OBTAIN DATA SET STORAGE POSITION ACCORDING TO FIELD D
112
NO
LAST RECORD CHECKED IN TRACK?
YES
111
SET INTERNAL POINTER TO RELATIVE TRACK R1
READ NEXT TRACK
113
END OF OPEN PROCESSING
r←r+1
115
r=1
114
ERROR END OF OPEN PROCESSING

FIG.5

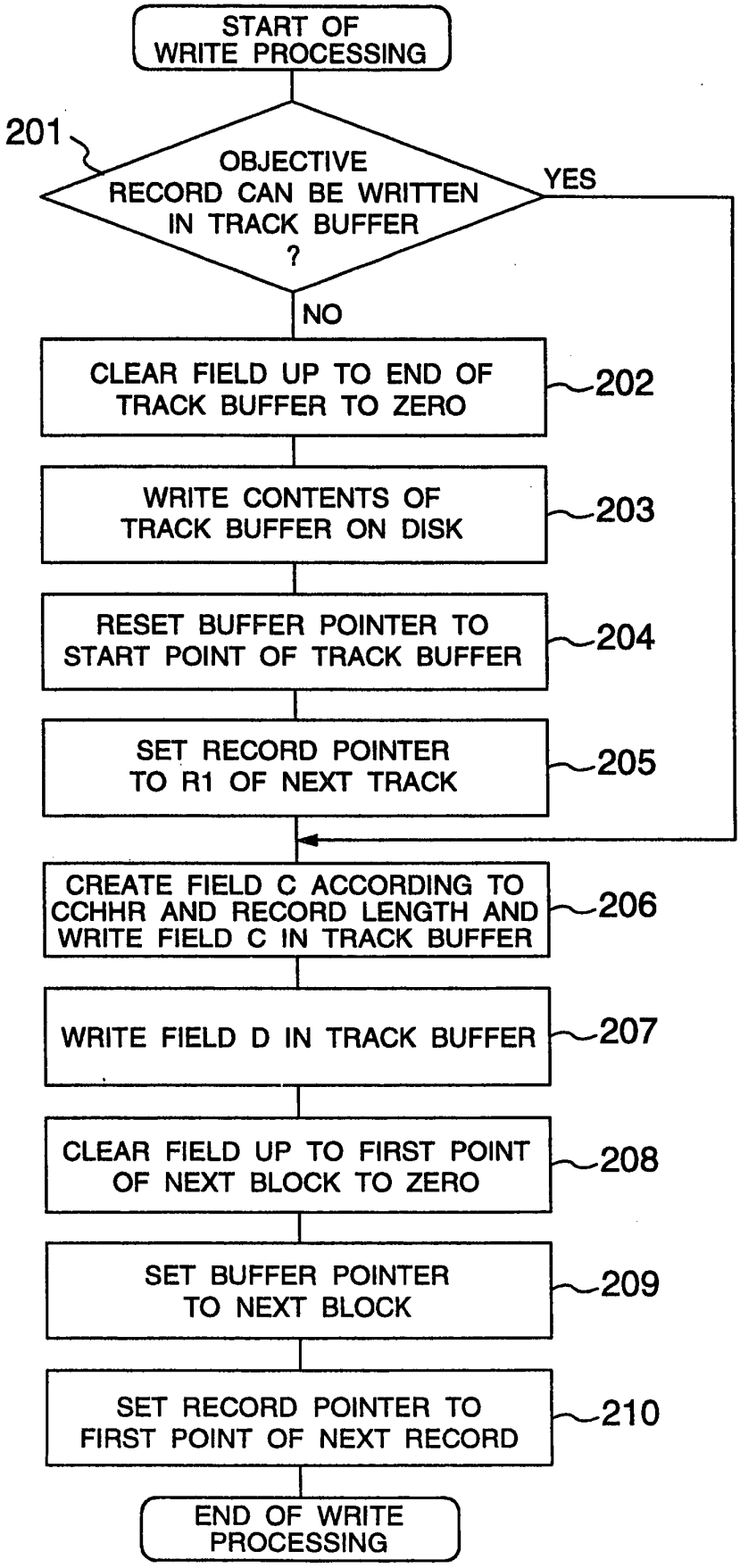
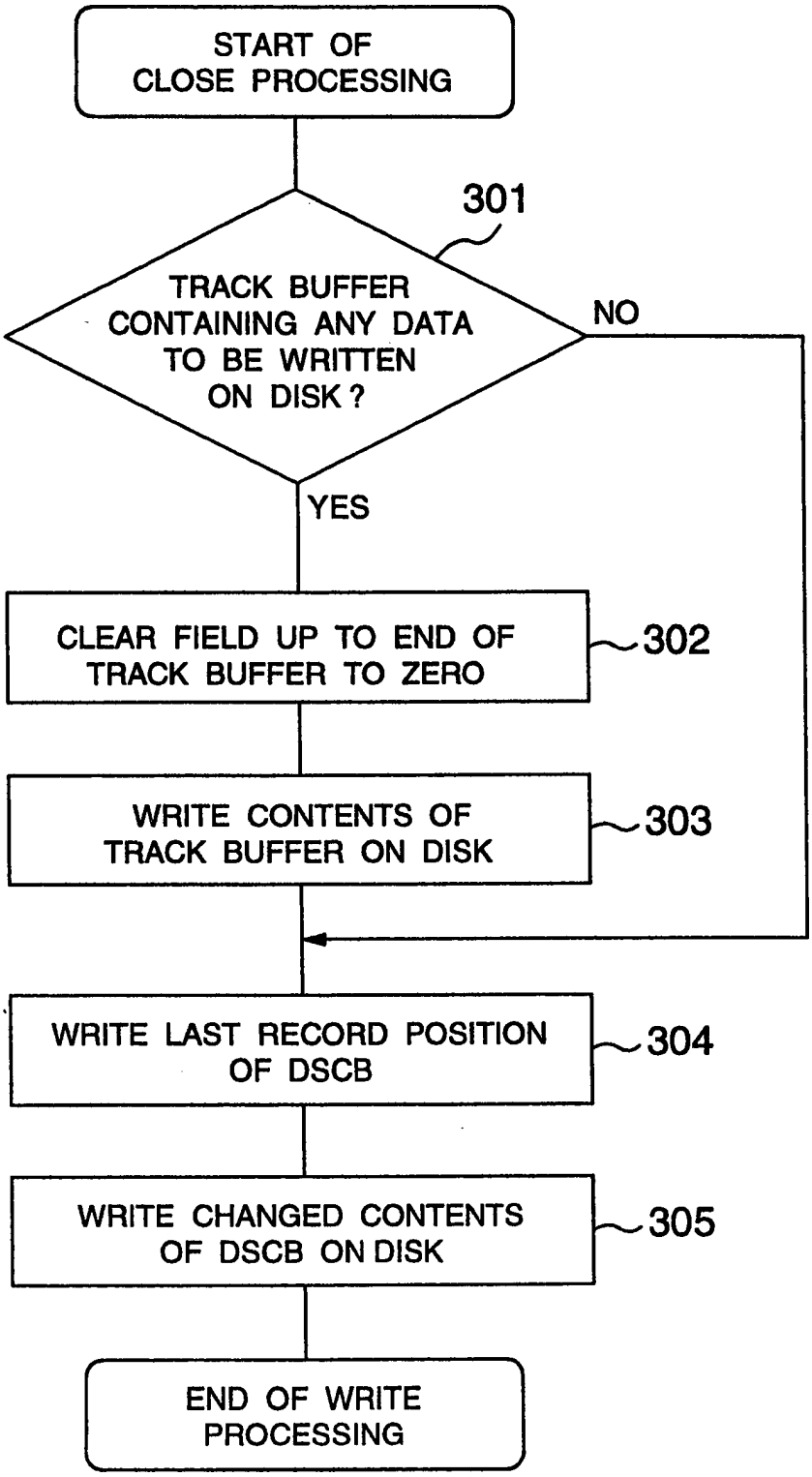

START OF WRITE PROCESSING
201
OBJECTIVE RECORD CAN BE WRITTEN IN TRACK BUFFER ?
YES
NO
CLEAR FIELD UP TO END OF TRACK BUFFER TO ZERO
202
WRITE CONTENTS OF TRACK BUFFER ON DISK
203
RESET BUFFER POINTER TO START POINT OF TRACK BUFFER
204
SET RECORD POINTER TO R1 OF NEXT TRACK
205
CREATE FIELD C ACCORDING TO CCHHR AND RECORD LENGTH AND WRITE FIELD C IN TRACK BUFFER
206
WRITE FIELD D IN TRACK BUFFER
207
CLEAR FIELD UP TO FIRST POINT OF NEXT BLOCK TO ZERO
208
SET BUFFER POINTER TO NEXT BLOCK
209
SET RECORD POINTER TO FIRST POINT OF NEXT RECORD
210
END OF WRITE PROCESSING

FIG.6


START OF CLOSE PROCESSING
301
TRACK BUFFER CONTAINING ANY DATA TO BE WRITTEN ON DISK?
NO
YES
CLEAR FIELD UP TO END OF TRACK BUFFER TO ZERO
302
WRITE CONTENTS OF TRACK BUFFER ON DISK
303
WRITE LAST RECORD POSITION OF DSCB
304
WRITE CHANGED CONTENTS OF DSCB ON DISK
305
END OF WRITE PROCESSING