



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 602 11 452 T2 2006.12.07**

(12)

Übersetzung der europäischen Patentschrift

(97) **EP 1 313 025 B1**

(51) Int Cl.⁸: **G06F 13/28** (2006.01)

(21) Deutsches Aktenzeichen: **602 11 452.7**

(96) Europäisches Aktenzeichen: **02 257 259.8**

(96) Europäischer Anmeldetag: **18.10.2002**

(97) Erstveröffentlichung durch das EPA: **21.05.2003**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **17.05.2006**

(47) Veröffentlichungstag im Patentblatt: **07.12.2006**

(30) Unionspriorität:

32598 25.10.2001 US

(84) Benannte Vertragsstaaten:

DE, FR, GB

(73) Patentinhaber:

Sun Microsystems, Inc., Santa Clara, Calif., US

(72) Erfinder:

**Butterfield, David A., Broomfield, Colorado 80020,
US**

(74) Vertreter:

**Dr. Weber, Dipl.-Phys. Seiffert, Dr. Lieke, 65183
Wiesbaden**

(54) Bezeichnung: **DMA-Übertragung von Daten und Prüfinformation zu und von einem Datenspeicherungsgerät**

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

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

Beschreibung

GEBIET DER ERFINDUNG

[0001] Diese Erfindung betrifft Computersysteme und insbesondere Verfahren und Vorrichtungen zum Übertragen von Daten und Prüfinformationen zwischen Computerhauptspeicher und Datenspeichereinrichtungen unter Verwendung von direktem Speicherzugriff.

HINTERGRUND DER ERFINDUNG

[0002] Datenspeichereinrichtungen mit direktem Speicherzugriff sind typischerweise in Datenspeicherblöcken fester Größe organisiert. Beispielsweise kann ein Plattenlaufwerk eines Computers Datenspeicher in einer Anzahl von Datenblöcken organisieren, von denen jeder 512 Bytes an Daten enthält und jeder Block eine einzigartige logische Blockadresse hat, wobei die Blöcke beginnend mit Block Null der Reihe nach numeriert sind. Es sind auch andere Blockadressierungsverfahren, einschließlich der Spezifizierung von Zylindern, Spuren und Sektoren; in Gebrauch, jedoch hat auch hier jeder Block eine eindeutige Adresse, und es wird davon ausgegangen, daß die Blöcke in einer wohldefinierten logischen Ordnung auf der Einrichtung erscheinen. Andere Speichermedien, wie Bänder, können in Abhängigkeit von der Anwendung ebenfalls in Blöcken fester Größe organisiert sein.

[0003] Ein wesentlicher Teil der Arbeit, die von einem Computersystem ausgeführt wird, ist das Speichern von Daten in und das Abrufen von Daten aus Datenspeichereinrichtungen (I/O). In frühen Computersystemen wurde diese Arbeit von der Zentralverarbeitungseinheit (CPU) ausgeführt, was die Benutzung eines wesentlichen Teils der verfügbaren Rechenleistung der CPU erforderte, um die einzelnen Schritte der I/O auszuführen. In modernen Systemen dagegen werden die Mechanismen der Datenübertragung zu und von Speichereinrichtungen oft von einer separaten Steuerung für direkten Speicherzugriff (Direct Memory Access, DMA) ausgeführt. [Fig. 1A](#) zeigt ein vereinfachtes Blockdiagramm eines solchen Systems, dargestellt bei **101**. Das System **101** weist eine CPU **103**, einen Hauptspeicher **105**, eine DMA-Steuerung **107**, ein Speichermedium **109** und einen Kommunikationsbus **111** auf. In Systemen wie dem System **101** programmiert die CPU **103**, um die I/O-Datenübertragung zu initiieren, die DMA-Steuerung **107** mit den Informationen, die sie benötigt, und gibt (über den Kommunikationsbus **111**) einen Befehl aus, um die Übertragung zu starten. Die CPU kann sich dann anderen Berechnungen zuwenden, während die DMA-Steuerung die einzelnen Schritte der Übertragung ausführt. Die DMA-Steuerung **107** dient somit als Vermittler zwischen dem Hauptspeicher **105** und dem Speichermedium **109**. Wenn der

I/O-Vorgang abgeschlossen ist, wird die CPU **103** von der DMA-Steuerung **107** über die Beendigung der I/O informiert, woraufhin die CPU die Ergebnisse der Übertragung verwenden kann. Auf diese Weise wird der größte Teil der Arbeit, die erforderlich ist, um die I/O durchzuführen, von der CPU weggenommen, wodurch die CPU frei ist, eine andere Arbeit auszuführen. Beispielsweise kann die CPU Prozesse wie den Austausch mit dem Hauptspeicher **105** ausführen.

[0004] Um eine DMA-Einrichtung zu programmieren, spezifiziert eine CPU solche Informationen typischerweise wie folgt: als Startadresse im Hauptspeicher, die für die Datenübertragung verwendet werden soll, als die Länge der zu übertragenden Daten, ob Daten von der Einrichtung zum Hauptspeicher ("Lesen") oder vom Hauptspeicher zu der Einrichtung ("Schreiben") übertragen werden sollen, und für Einrichtungen mit direktem Zugriff die Adresse des Datenblocks auf der Speichereinrichtung.

[0005] Oft ist es gewünscht, in einem einzigen I/O-Vorgang mehr als einen Datenblock zu oder von einer Speichereinrichtung zu übertragen. In den meisten Fällen ist die Länge solcher Übertragungen auf ganzzahlige Vielfache der Blockgröße der Speichereinrichtung beschränkt. [Fig. 1B](#) zeigt ein Szenarium **113**, bei dem Daten über die DMA-Steuerung **107** von dem Hauptspeicher **105** zu einem Speichermedium **109** übertragen werden. Wenn eine CPU der DMA-Steuerung **107** eine Länge von mehr als einem Datenblock angibt, überträgt der I/O-Vorgang beginnend bei der spezifizierten Startadresse Daten von aufeinanderfolgenden Adressen im Hauptspeicher zu aufeinanderfolgenden Blöcken in dem Speichermedium **109**. Die Daten werden beginnend bei einer spezifizierten Blockadresse oder in einigen Einrichtungen an einem impliziten gegenwärtigen Ort auf dem Speichermedium gespeichert. Die DMA-Steuerung beinhaltet typischerweise Register für die Startadresse des Hauptspeichers, die Datenblocklänge und optional eine Speicheradresse, die mit einem Speichermedium verknüpft ist. In diesem Beispiel wird somit ein Datenblock mit einer Länge von 4096 Bytes (8×512-Byte-Blöcke, bezeichnet mit D0–D7) in einem Vorgang zu dem Speichermedium **109** übertragen (gespeichert in 8×512-Byte-Blöcken, bezeichnet mit B0–B7). Durch das Übertragen einer Mehrzahl von Datenblöcken in einem einzigen I/O-Vorgang wird die CPU von dem Aufwand befreit, den I/O-Vorgang des DMA für jeden der Blöcke D0–D7 einzeln zu initiieren.

[0006] Oft ist es notwendig oder gewünscht, mehrere Datenblöcke zwischen nichtzusammenhängenden Orten im Hauptspeicher zu einem zusammenhängenden Satz von Blöcken auf der Datenspeichereinrichtung zu übertragen. Wenn die DMA-Steuerung für irgendeinen bestimmten I/O-Vorgang nur eine einzi-

ge Hauptspeicher-Startadresse akzeptiert, können nur aufeinanderfolgend adressierbare Orte im Hauptspeicher in einem einzigen I/O-Vorgang übertragen werden. In Fällen, in denen die Granularitätsbeschränkungen der Einrichtungen durch die Organisation der Daten im Hauptspeicher erfüllt werden, kann es möglich sein, für jeden zusammenhängenden Speicherblock separate I/O-Befehle auszugeben. Dies würde jedoch für die CPU den Aufwand mit sich bringen, die zusätzlichen I/O-Befehle zu initiieren und deren Abarbeitung zu handhaben.

[0007] Um dieses Problem zu mindern, implementieren einige DMA-Steuerungen etwas, das als ein "Verstreuen/Einsammeln" (Scatter/Gather)-I/O-Vorgang bezeichnet wurde. Bei solchen Vorgängen erhält die DMA-Steuerung eine Liste von Startadressen und Längen und wird nicht mit einer einzigen Hauptspeicher-Startadresse und -Länge programmiert. Zuerst werden Daten beginnend mit der in dem ersten Scatter/Gather-Listenelement spezifizierten Hauptspeicheradresse übertragen. Wenn dieses Element verbraucht bzw. abgearbeitet ist, d.h. wenn die spezifizierte Datenlänge für das Element übertragen wurde, wird das nächste Element auf der Liste verwendet, um den nächsten Satz von Hauptspeicheradressen zu bestimmen, der bei der Übertragung verwendet werden soll. Auf diese Weise kann ein einziger I/O-Vorgang eine Mehrzahl von Datenblöcken zwischen der Datenspeichereinrichtung und nicht-zusammenhängenden Orten im Hauptspeicher übertragen. Herkömmliche DMA-Steuerungen lassen jedoch für gewöhnlich nur eine kleine, begrenzte Anzahl von Scatter/Gather-Listenelementen zu, oder sie implementieren überhaupt keine Scatter/Gather-Fähigkeit.

[0008] In Rechnerumgebungen, die eine hohe Zuverlässigkeit der Datenspeicherung und des -abrufs erfordern, kann es notwendig oder wünschenswert sein, für jeden Block von gespeicherten Daten Prüfinformationen zu behalten, die verwendet werden können, um zu bestätigen, daß die Daten während der Speicherung nicht korruptiert wurden. Obwohl es viele Arten von Prüfinformationen gibt, wird eine allgemeine Form als Prüfsumme bezeichnet. Eine Prüfsumme ist das Ergebnis einer mathematischen Berechnung auf einem Datenblock, der jedesmal, wenn er berechnet wird, dasselbe Ergebnis liefert. Wenn eine Prüfsumme für einen Datenblock berechnet wird, der gerade gespeichert wird, wird die Prüfsumme typischerweise zusammen mit den Daten gespeichert. Wenn die Daten abgerufen werden, wird die Prüfsumme für den abgerufenen Daten neu berechnet und mit der gespeicherten Prüfsumme verglichen. Wenn die neu berechnete Prüfsumme sich von der gespeicherten Prüfsumme unterscheidet, weiß man, daß die Daten korruptiert wurden, und es können Abhilfemaßnahmen eingeleitet werden.

[0009] In vielen Fällen ist es vorteilhaft, wenn Berechnungen von Prüfinformationen zur Datenintegrität in dem Betriebssystem auf sehr niedriger Ebene durchgeführt werden. Beispielsweise können diese Berechnungen ohne Kenntnis der Softwareeinheiten, die ursprünglich die I/O angefordert hatten, in einem Gerätetreiber durchgeführt werden. Die die I/O anfordernde Einheit kann beispielsweise einfach anfordern, daß beginnend bei einer bestimmten Blockadresse 64 Speicherblöcke in das Speichermedium geschrieben werden sollen. Die Prüfinformationen werden durch den Gerätetreiber berechnet und überprüft und erscheinen nicht in den Daten, die zwischen der anfordernden Einheit und dem Gerätetreiber übermittelt werden.

[0010] Beim Speichern von Daten und ihrer zugehörigen Prüfinformationen ist große Sorgfalt dahingehend erforderlich, daß sie im Falle eines Systemausfalls auf der Speichereinrichtung erhalten bleiben. Eine beträchtliche Vereinfachung dieses Prozesses kann erreicht werden, indem die Prüfinformationen für einen Datenblock in demselben I/O-Vorgang, der die Daten schreibt, auf der Einrichtung gespeichert werden. Eine Möglichkeit, dies durchzuführen, besteht darin, die Speichereinrichtung so zu formatieren, daß jeder Speicherblock auf der Einrichtung lang genug ist, um sowohl die gewünschte Datenblockgröße als auch die gewünschte Prüfinformationsgröße aufzunehmen. Wenn beispielsweise die gewünschte Datenblockgröße 512 Bytes beträgt und die gewünschte Prüfinformationsgröße 8 Bytes beträgt, könnte die Speichereinrichtung in 520-Byte-Blöcken formatiert werden, von denen jeder 512 Bytes an Daten und 8 Bytes an Prüfinformationen aufnehmen könnte. In diesem Beispiel würde die CPU dann I/O-Vorgänge in ganzzahligen Vielfachen von 520 Bytes vornehmen, wobei die ersten 512 Bytes die Daten für den ersten Block in der I/O sind, die nächsten 8 Bytes die Prüfinformationen für den ersten Block in der I/O sind, die nächsten 512 Bytes die Daten für den zweiten Block in der I/O sind, die nächsten 8 Bytes die Prüfinformationen für den zweiten Block in der I/O sind usw.

[0011] Derzeit stellt das Implementieren der Speicherung von Prüfinformationen in längeren Speicherblockgrößen ein Problem dar. Beispielsweise müssen beim Schreiben mehrerer Datenblöcke in einem I/O-Vorgang die durch eine CPU (beispielsweise in einem Gerätetreiber) berechneten Prüfinformationen mit den Datenblöcken verschachtelt werden. Ebenso müssen auch während einer Leseoperation die verschachtelten Prüfinformationen aus den verschachtelten Daten extrahiert bzw. herausgezogen werden, so daß der Gerätetreiber seinen Clients Daten liefern kann, die keine Prüfinformationen enthalten. Selbst wenn nur ein einziger Block geschrieben wird, müssen die Prüfinformationen auch während des Schreibens in den Daten enthalten sein. Die Daten, die von

der anfordernden Einheit bei dem Gerätetreiber ankommen, enthalten diese Prüfinformationen jedoch nicht, noch ist hierfür Raum reserviert.

[0012] Um einen Schreibvorgang auszuführen, wenn die DMA-Steuerung keine Scatter/Gather-Fähigkeit besitzt, müssen die Daten von den Orten im Hauptspeicher, die von der anfordernden Einheit bereitgestellt werden, in einen anderen Puffer kopiert werden, der lang genug ist, um nicht nur die Daten, sondern auch die Prüfinformationen für diese Daten aufzunehmen. Wenn beispielsweise die anfordernde Einheit einen 512-Byte-Puffer übergibt und die Prüfinformationen 8 Bytes ausmachen, muß der Gerätetreiber der DMA-Steuerung folgendes tun: einen zusammenhängenden 520-Byte-Puffer zuweisen, die Daten in diesen hineinkopieren, die Prüfinformationen berechnen und diese hinter den Daten in dem 520-Byte-Puffer plazieren. Der Gerätetreiber kann dann die DMA-Steuerung so programmieren, daß sie den 520-Byte-I/O-Vorgang unter Verwendung des lokalen Puffers ausführt. In einem weiteren Beispiel müssen, wenn dem Gerätetreiber eine 4096 Bytes große Seite übergeben wird, die Daten in Blöcken von je 512 Bytes in einen 4160 Bytes großen lokalen Puffer hineinkopiert werden, wobei jedem 512-Byte-Block unmittelbar seine entsprechenden Prüfinformationen folgen, die von dem Gerätetreiber berechnet wurden. Solche Kopiervorgänge belasten jedoch die CPU, die ansonsten für andere Berechnungen verwendet werden könnte.

[0013] Selbst wenn die DMA-Steuerung Scatter/Gather-Fähigkeit besitzt, beschränken Implementierungen oft die Länge der Scatter/Gather-Liste. Der Versuch, die Scatter/Gather-Fähigkeit zu verwenden, um von dem Gerätetreiber berechnete Prüfinformationen mit Daten für eine bestimmte I/O-Anforderung zu verschachteln, kann die Grenzen der Scatter/Gather-Liste überschreiten, was wiederum das Kopieren von Daten erfordert oder was es notwendig macht, I/O-Anforderungen in eine Mehrzahl kleinerer Anforderungen aufzuteilen. Beispielsweise wären, um 16 je 4096 Bytes große Datenseiten mit Prüfinformationen in eine Speichereinrichtung (formatiert in 520-Byte-Blöcken, die jeweils 512 Bytes an Daten und 8 Bytes an Prüfinformationen beinhalten) zu schreiben, 256 Scatter/Gather-Listenelemente erforderlich anstelle der maximal 16 Elemente, die benötigt werden, um die gleichen Blöcke ohne die Prüfinformationen zu schreiben (8 Blöcke pro Seite \times 16 Seiten \times 2 Elemente pro Block). Somit verfügen standardmäßige DMA-Steuerungen mit Scatter/Gather-Fähigkeit in vielen Fällen nicht über ausreichend Speicher, um eine Scatter/Gather-Liste zu implementieren, die in der Lage ist, Informationen für Daten und ihre zugehörigen Prüfinformationen, die von dem Gerätetreiber berechnet wurden, zu speichern. Die Bereitstellung einer größeren Speicherfähigkeit für die DMA-Steuerung kann dieses Problem

überwinden, bringt jedoch eine größere Belastung für die Rechneffizienz mit sich.

[0014] Die US 5987627 offenbart ein Massenspeichersystem, welches über einen lokalen CPU-Bus mit einer CPU verbunden ist. Das Massenspeichersystem beinhaltet eine Schnittstelle zum Anschließen an den lokalen CPU-Bus, privaten Speicher zum Speichern von Daten und einen Logikschaltkreis. Der Logikschaltkreis erzeugt ECC-Bits für Daten von der Schnittstelle, die in den privaten Speicher geschrieben werden, und bestätigt die ECC-Bits von Daten, die aus dem privaten Speicher auf der Schnittstelle ausgelesen werden.

ZUSAMMENFASSUNG DER ERFINDUNG

[0015] Die vorliegende Erfindung liefert ein Verfahren und eine Vorrichtung, wie sie in den anhängenden Ansprüchen ausgeführt werden.

[0016] Gemäß einer Ausführungsform der Erfindung beschafft eine DMA-Steuerung Prüfinformationen aus einem Computerhauptspeicher, statt die Prüfinformationen in der DMA-Steuerung intern zu berechnen. Beispielsweise können die Prüfinformationen eher durch die CPU als durch die DMA-Steuerung berechnet werden. In einigen Ausführungsformen implementiert die DMA-Steuerung zusätzliche Register, um beispielsweise den Ort von Prüfinformationen im Computerhauptspeicher zu spezifizieren. Die DMA-Steuerung selbst führt das Verschachteln von Daten mit ihren zugehörigen Prüfinformationen aus, während sie Informationen zwischen dem Hauptspeicher und der Speichereinrichtung überträgt. Um dies zu bewerkstelligen, erhält die DMA-Steuerung Befehle dafür, wie die Verschachtelung stattzufinden hat. In einigen Ausführungsformen werden die Länge der Datenblöcke und die Länge der Prüfinformationsblöcke, die für die Speicherung verschachtelt werden sollen, von der CPU programmiert. In anderen Ausführungsformen kann eine oder können beide dieser Längen in anderen programmierten Parametern und/oder der Formatierung der Speichereinrichtung inbegriffen sein.

[0017] Ein solcher Ansatz liefert eine effiziente Architektur und ein effizientes Verfahren zum Verwenden von direktem Speicherzugriff (Direct Memory Access, DMA) für das Speichern und Abrufen von Daten und zugehörigen Prüfinformationen in bzw. aus Blöcken fester Größe auf einer Datenspeichereinrichtung.

[0018] Gemäß einer weiteren Ausführungsform der Erfindung akzeptiert eine DMA-Steuerung in Fällen, in denen es keine Scatter/Gather-Fähigkeit gibt, zwei Hauptspeicheradressen, die den Prüfinformationen und den Daten, die verschachtelt werden sollen, entsprechen. Insbesondere gemäß einigen Ausführungs-

rungsformen werden zwei verschiedene Register von Startadressen implementiert, um Startadressen zu speichern, die den zu verschachtelnden Prüfinformationen und Daten entsprechen. Mit anderen Worten, das erste Register von Startadressen spezifiziert den Start der Daten, und das zweite Register von Startadressen spezifiziert den Start der Prüfinformationen für diese Daten. In einer weiteren Ausführungsform stellt eine DMA-Steuerung zwei zusätzliche Register bereit, die spezifizieren, wie aus den zwei Sätzen von Adressen entnommene Informationen verschachtelt werden sollen. Ein solches Register spezifiziert die Größe der Datenblöcke, die in dem Speicher auf der Speichereinrichtung verschachtelt werden sollen, wohingegen das andere Register die Größe der Prüfinformationsblöcke spezifiziert, die in dem Speicher auf der Speichereinrichtung verschachtelt werden sollen. In letzterem Fall durchläuft die Steuerung, nachdem die DMA-Steuerung programmiert und der Vorgang initiiert wurde, unter Verwendung der zwei Adressenregister die Daten und die Prüfinformationen, wobei sie den ersten Datenblock und seine entsprechenden Prüfinformationen schreibt, dann den Vorgang für den zweiten Datenblock wiederholt usw.

[0019] Gemäß noch einer weiteren Ausführungsform der Erfindung beinhaltet jeder I/O-Vorgang die Übertragung einer Anzahl von Datenblöcken. Die CPU berechnet die Prüfinformationen für jeden Block und plaziert die Prüfinformationen jedes Blockes nacheinander zusammenhängend in einem Speichersegment (z.B. eines Puffers) für Prüfinformationen, ehe die DMA-Steuerung programmiert wird. Indem die Prüfinformationen für alle Datenblöcke in einem zusammenhängenden Speichersegment angeordnet werden, kann eine einzige Adresse die gesamten Prüfinformationen für alle Datenblöcke in dem I/O-Vorgang spezifizieren. Indem die DMA-Steuerung so ausgestaltet wird, daß sie die von dem Hauptspeicher erhaltenen Prüfinformationen mit den Daten verschachtelt, sind keine zusätzlichen Scatter/Gather-Elemente notwendig, und weder die Daten noch die Prüfinformationen müssen in lokale Puffer auf der DMA-Steuerung kopiert werden.

[0020] Gemäß noch einer weiteren Ausführungsform der Erfindung wird die Scatter/Gather-Fähigkeit unterstützt. In einigen Ausführungsformen wird diese Fähigkeit verwendet, um nur auf die Daten zuzugreifen, und eine einzige Startadresse von Prüfinformationen wird verwendet, um für einen bestimmten I/O-Vorgang auf alle Prüfinformationen zuzugreifen. In diesen Fällen plaziert der Gerätetreiber alle Prüfinformationen für den I/O-Vorgang zusammenhängend in einem einzigen Puffer, während unter Verwendung der Scatter/Gather-Liste auf die Daten zugegriffen wird. Im Gegensatz dazu implementieren andere Ausführungsformen eine Scatter/Gather-Fähigkeit sowohl für Daten als auch für Prüfinformationen.

[0021] Gemäß noch einer weiteren Ausführungsform der Erfindung verwendet während Lesevorgängen eine DMA-Steuerung die oben genannten Register, um gespeicherte Daten und zugehörige Prüfinformationen in ihre Bestandteile aufzugliedern bzw. zu entfalten. Die Daten und die zugehörigen Prüfinformationen werden somit unter Verwendung eines Satzes von Verschachtelungskriterien (aus Gründen der Einfachheit wird für die Kriterien, die zum Verschachteln und zum Entfalten von Daten und zugehörigen Prüfinformationen verwendet werden, derselbe Begriff benutzt) aus der Speichereinrichtung extrahiert bzw. herausgezogen. Die getrennten Datenblöcke und zugehörigen Prüfinformationen werden in einen Speicher geschrieben, der zu der DMA-Steuerung extern ist (beispielsweise in einen Hauptspeicher).

KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0022] Die folgende ausführliche Beschreibung ist besser zu verstehen, wenn sie im Zusammenhang mit den Zeichnungen betrachtet wird, in denen:

[0023] [Fig. 1A](#) ein vereinfachtes Blockdiagramm ist, welches die Mechanismen einer Datenübertragung zu und von Speichereinrichtungen unter Verwendung einer separaten Steuerung für direkten Speicherzugriff (DMA) zeigt,

[0024] [Fig. 1B](#) ein vereinfachtes Blockdiagramm ist, welches einen herkömmlichen Mechanismus zum Übertragen von Daten von einem Hauptspeicher zu einem Speichermedium über eine DMA-Steuerung zeigt,

[0025] [Fig. 2A](#) ein vereinfachtes Blockdiagramm ist, welches eine Architektur einer DMA-Steuerung und Aspekte eines Verfahrens für einen Schreibvorgang darstellt,

[0026] [Fig. 2B](#) ein vereinfachtes Blockdiagramm ist, welches ein Beispiel einer Anordnung von verschachtelten Daten und Prüfinformationen in einem Speichermedium darstellt,

[0027] [Fig. 2C](#) ein vereinfachtes Blockdiagramm ist, welches eine weitere Architektur einer DMA-Steuerung und Aspekte eines Verfahrens für einen Schreibvorgang darstellt,

[0028] [Fig. 2D](#) ein Beispiel einer Scatter/Gather-Liste veranschaulicht, die entsprechend der in [Fig. 2C](#) dargestellten DMA-Steuerungsarchitektur verwendet werden kann,

[0029] [Fig. 2E](#) ein vereinfachtes Blockdiagramm ist, welches noch eine weitere DMA-Steuerungsarchitektur und Aspekte eines Verfahrens für einen Schreibvorgang darstellt,

[0030] [Fig. 2F](#) zwei Beispiele von Scatter/Gather-Listen veranschaulicht, die entsprechend der in [Fig. 2E](#) dargestellten DMA-Steuerungsarchitektur verwendet werden können,

[0031] [Fig. 3A](#) einen Arbeitsablauf darstellt, der Aspekte eines Verfahrens zum Übertragen von Daten und Prüfinformationen von einem Hauptspeicher zu einer Speichereinrichtung zeigt,

[0032] [Fig. 3B](#) einen Arbeitsablauf darstellt, der Aspekte eines Verfahrens zeigt, bei dem ein Datenblock und seine zugehörigen Prüfinformationen durch eine DMA-Steuerung gemäß einer Ausführungsform der Erfindung verschachtelt werden,

[0033] [Fig. 4](#) ein typisches Computersystem veranschaulicht, welches, wenn es in geeigneter Weise konfiguriert oder ausgestaltet ist, verwendet werden kann, um eine Ausführungsform der Erfindung zu implementieren.

AUSFÜHRLICHE BESCHREIBUNG

[0034] In der folgenden ausführlichen Beschreibung werden zahlreiche spezifische Ausführungsformen lediglich beispielhaft beschrieben, um ein vollständiges Verständnis der Erfindung zu ermöglichen. Für Fachleute auf dem Gebiet ist es jedoch offensichtlich, daß die vorliegende Erfindung auch ohne diese spezifischen Einzelheiten oder unter Verwendung alternativer Elemente oder Prozesse ausgeführt werden kann.

[0035] Der hierin beschriebene Ansatz liefert effiziente Architekturen und Verfahren zur Verwendung von direktem Speicherzugriff (DMA), um Daten und zugehörige Prüfinformationen in Blöcken fester Größe in einer Datenspeichereinrichtung zu speichern und aus dieser abzurufen. Dies wird durch die Verwendung von Verschachtelungskriterien, die in Registern in der DMA-Steuerung bereitgestellt werden, erreicht. Da Speicherblöcke fester Größe mit einer festen Datengröße und einer festen Prüfinformationsgröße pro Speicherblock spezifiziert sind, können die Verschachtelungskriterien in einer Form dargestellt werden, die bedeutend kompakter ist als es die Darstellung derselben Verschachtelung unter Verwendung einer herkömmlichen Scatter/Gather-Liste wäre. Bei diesem Ansatz ist die Darstellung der Verschachtelungskriterien auf Blöcke fester Größe beschränkt, die zwei Segmente beinhalten (z.B. eines für Daten und eines für Prüfinformationen). Die Größe der Daten- und der Prüfinformationsblöcke kann verschieden sein, ihre jeweiligen Größen variieren jedoch nicht. Somit sind für das spezifische Beispiel der Verschachtelung von Daten und Prüfinformationen unabhängig von der Gesamtgröße der Übertragung die Verschachtelungskriterien für eine vollständige Übertragung in einer kleinen Anzahl von Regis-

tern dargestellt. Die Verwendung einer herkömmlichen Scatter/Gather-Liste, um die gleiche beschränkte Verschachtelung auszudrücken, würde für jeden einzelnen verschachtelten Block (Daten und Prüfinformationen), der an der Übertragung beteiligt ist, zwei Scatter/Gather-Listenelemente erfordern – umfangreiche Übertragungen würden eine übermäßige Anzahl von Scatter/Gather-Listenelementen erfordern.

[0036] [Fig. 2A](#) ist ein vereinfachtes Blockdiagramm, welches die Architektur **201** einer DMA-Steuerung und Aspekte eines Verfahrens für einen Schreibvorgang gemäß einer Ausführungsform der Erfindung darstellt. Während einer Schreib-"DMA-Übertragung" überträgt die DMA-Steuerung **205** Daten von dem Hauptspeicher **203** zu dem Speichermedium **207**. In diesem Beispiel enthält der Hauptspeicher **203** zwei separate jeweils zusammenhängende Reihen von Speichersegmenten, die Daten und ihren zugehörigen Prüfinformationen entsprechen. Insbesondere beinhaltet, wie gezeigt ist, das erste Speichersegment die Datenblöcke D0–D7, während das zweite Speichersegment die zugehörigen Prüfinformationsblöcke C0–C7 beinhaltet. In einer Ausführungsform berechnet die CPU die zugehörigen Prüfinformationen und speichert sie in dem Hauptspeicher zusammenhängend als C0–C7. Die DMA-Steuerung **205** überträgt die Datenblöcke D0–D7 (4096 Bytes an Daten) und ihre zugehörigen Prüfinformationen C0–C7 (64 Bytes an Daten) von dem Hauptspeicher **203** zu dem Speichermedium **207**. Bei dem Übertragungsvorgang werden die Daten und die Prüfinformationen vorzugsweise verschachtelt und als ein zusammenhängender Satz von Datenblöcken B0–B7 (4160 Bytes an Daten) gespeichert. Der Verschachtelungsvorgang wird unten unter Bezugnahme auf die [Fig. 3A](#) und [Fig. 3B](#) ausführlicher diskutiert.

[0037] Wie es in [Fig. 2A](#) dargestellt ist, speichert die DMA-Steuerung **205** vorzugsweise Startadreßdaten, die eine Datenstartadresse der zu übertragenden Daten und eine Prüfinformations-Startadresse für die zu übertragenden Prüfinformationen angeben, und eine Speicheradresse, die angibt, wo die verschachtelten Daten und Prüfinformationen gespeichert werden. Zusätzlich speichert die DMA-Steuerung **205** innerhalb einer Speichereinrichtung vorzugsweise auch Verschachtelungskriterien, die die Länge des Datenblocks und die Länge des Prüfinformationsblocks, die verschachtelt werden sollen, angeben. Wie gezeigt ist, weist die DMA-Steuerung **205** gemäß einer Ausführungsform vorzugsweise Register zum Speichern der Startadreßdaten und der Verschachtelungskriterien auf. Insbesondere werden vorzugsweise separate Register verwendet, um beide Startadressen zu speichern: die Datenadresse und die Prüfinformationsadresse. In ähnlicher Weise werden die Verschachtelungskriterien vorzugsweise in anderen bzw. separaten Registern gespeichert als die Daten-

blockgröße (512 Bytes) der zu übertragenden Daten und die Größe der zugehörigen Prüfinformationsblöcke (8 Bytes). Ein Register wird auch für die Speicherlänge (4160 Bytes) der verschachtelten Daten, die in die Speichereinrichtung geschrieben werden sollen, verwendet. Alternativ könnte dieses Register die Datenlänge (4096 Bytes) spezifizieren, wobei die Prüflänge (64 Bytes) sich implizit auf Basis der Verschachtelungskriterien ergibt. Ebenso könnte das Register alternativ spezifizieren, daß ein Datenblock (512 Bytes) und seine zugehörige Prüfinformationslänge (8 Bytes) in der Verschachtelung achtmal übertragen werden sollen, d.h. in diesem Beispiel einfach die Zahl "8".

[0038] In diesem Beispiel eines Schreibvorgangs werden die Register der Datenadresse und der Prüfinformationsadresse zum Speichern der Hauptspeicher-Startadressen verwendet, wie angegeben. Indem die DMA-Steuerung die Register für Datenadressen und Prüfinformationen verwendet, kann sie einen Datenblock und seine zugehörigen Prüfinformationen aus dem Hauptspeicher **203** beschaffen. Die Größe des Datenblocks und die Größe des Prüfinformationsblocks (die Verschachtelungskriterien) werden so verwendet, daß die DMA-Steuerung die Anzahl an Bytes kombinierter Daten und Prüfinformationen, die nacheinander übertragen werden sollen, bestimmen kann. Die Speicherlänge kann von der DMA-Steuerung zusammen mit der Länge des Datenblocks und der Länge des Prüfinformationsblocks verwendet werden, um die Menge an zu übertragenden Daten zu berechnen, indem sie eine Zählung durchführt, wenn jeder Datenblock und seine zugehörigen Prüfinformationen übertragen werden. Wie es oben beschrieben wurde, gibt das Speicheradressenregister anfangs die Startposition an, wo die zusammenhängenden Blöcke von verschachtelten Daten gespeichert werden sollen.

[0039] Gemäß einer Ausführungsform der Erfindung berechnet die DMA-Steuerung keine Prüfinformationen; vielmehr verschachtelt sie die Daten und die im Vorfeld berechneten Prüfinformationen, während sie sie zu einem Speichermedium überträgt. Somit beschafft unter Bezugnahme auf [Fig. 2A](#) als Teil eines Übertragungsprozesses die DMA-Steuerung **205** den Datenblock D0 (512 Bytes) und die zugehörigen Prüfinformationen C0 (8 Bytes). In diesem Beispiel werden D0 und C0 verschachtelt und in einen einzigen Block B0 (520 Bytes Länge) des Speichermediums **207** geschrieben. Dieser Vorgang wird wiederholt, wobei D1 und C1 in B1 umgewandelt werden, der mit B0 zusammenhängt, und so weiter, bis alle Daten übertragen wurden. Wie es oben erwähnt wurde, werden die Register für Speicherlänge, Datenblockgröße und Prüfinformationsblockgröße verwendet, um Zähler und Zeiger zurückzusetzen, so daß die DMA-Steuerung die Quelladresse, von der die Daten und Prüfinformationen beschafft werden

sollen, und die Zieladresse der verschachtelten Daten/Prüfinformationen bestimmen kann, sowie bestimmen kann, wann die Übertragung der Daten und Prüfinformationen gestoppt werden soll. Es können auch andere Verfahren zum Verfolgen und/oder Berechnen der Datenblocklänge und/oder der Prüfinformationsblocklänge verwendet werden. In diesem Beispiel waren die Datenblocklänge (512 Bytes) und die Prüfinformationsblocklänge (8 Bytes) vorgegeben. In einem alternativen Beispiel können die Datenblocklänge (512 Bytes) und die verschachtelte Blocklänge (520 Bytes), aus denen die Länge des Prüfinformationsblocks berechnet werden kann, beschafft werden. Somit werden Daten in dem Speichermedium **207** unter Verwendung von Verschachtelungskriterien, wie beispielsweise der Länge von Daten- und Prüfinformationsblöcken, verschachtelt.

[0040] [Fig. 2B](#) ist ein vereinfachtes Blockdiagramm, welches darstellt, wie die übertragenen Daten in dem Speichermedium **207** angeordnet sind. Die Blöcke B0–B7, die jeweils 520 Bytes lang sind, werden zusammenhängend gespeichert. Jeder Block beinhaltet einen Datenblock und seine zugehörigen Prüfinformationen. Beispielsweise umfaßt B0, wie dargestellt, D0 und C0, die zusammenhängend gespeichert sind, und B3 umfaßt D3 und C3, die zusammenhängend gespeichert sind. Wie es in [Fig. 2B](#) gezeigt ist, werden die Datenblöcke D0–D7 und C0–C7 abwechselnd eingefügt oder verschachtelt, wenn sie in das Speichermedium **207** geschrieben werden.

[0041] In einem analogen Lesevorgang verwendet die DMA-Steuerung **205** die oben genannten Register, um die gespeicherten Informationen B0–B7 in ihre Bestandteile, nämlich D0–D7 und C0–C7, zu entfalten bzw. aufzulösen und sie beispielsweise in den Hauptspeicher **203** zurückzuschreiben. Vorzugsweise werden sie während eines Lesevorgangs, wenn jeder der einzelnen Blöcke (Daten und zugehörige Prüfinformationen) aus den verschachtelten Daten extrahiert wird, in einen Speicher geschrieben, der zu der DMA-Steuerung extern ist, vorzugsweise in den Hauptspeicher. Anders ausgedrückt werden während des Lesevorgangs die Daten aus der Speichereinrichtung gelesen und in zwei Informationsströme (Daten und Prüfinformationen) entfaltet bzw. aufgliedert, die an Speicherorte geschrieben werden, die durch ihre jeweilige Startadresse spezifiziert sind. Dies ist ein Beispiel dafür, wie ein Leseprozess ablaufen könnte. Für einen Fachmann auf dem Gebiet versteht es sich, daß ein solcher Leseprozess eine Umkehrung des oben beschriebenen Schreibprozesses mit sich bringt.

[0042] [Fig. 2C](#) ist ein vereinfachtes Blockdiagramm, welches eine weitere Architektur **209** einer DMA-Steuerung und Aspekte eines Verfahrens für einen Schreibvorgang darstellt. In diesem Beispiel werden die Datenblöcke D0–D7 im Hauptspeicher **211**

verstreut. Insbesondere gibt es zusammenhängende Blöcke D0–D2, D3–D4, D6–D7 sowie den Block D5. Die zugehörigen Prüfinformationen C0–C7 werden (wie in [Fig. 2A](#)) in einem zusammenhängenden Format an einem separaten Ort im Hauptspeicher **211** gespeichert.

[0043] In diesem Beispiel ist die Architektur der DMA-Steuerung **213** anders konfiguriert als in dem vorherigen Beispiel. Das heißt, die Datenadressregister berücksichtigen eine Datenadresse und eine Datenblocklänge für jeden der drei zusammenhängenden Blöcke D0–D2, D3–D4, D6–D7 sowie Block D5.

[0044] Um eine DMA-Übertragung für diese Art der Datenanordnung auszuführen, wird eine Scatter/Gather-Liste verwendet. [Fig. 2D](#) zeigt ein Beispiel einer Scatter/Gather-Liste **221**, die verwendet wird, um eine DMA-Übertragung der Datenblöcke D0–D7 und der zugehörigen Prüfinformationen C0–C7 aus dem Hauptspeicher **211** zu dem Speichermedium **207** zu implementieren. In der Scatter/Gather-Liste **221** sind vier mit 0–3 numerierte (wie ein Computer sie numerieren würde) Elemente enthalten. Die Elemente 0–3 enthalten jeweils eine Datenblock-Startadresse und die entsprechende Blocklänge, was das Verteilungsmuster und die Länge der drei zusammenhängenden Sätze von Datenblöcken D0–D2, D3–D4, D6–D7 sowie Block D5 in dem Hauptspeicher **211** liefert. Während einer DMA-Übertragung, wie sie oben zu den [Fig. 2A](#) und [Fig. 2B](#) beschrieben wurde, verwendet die DMA-Steuerung **213** in diesem Fall die Scatter/Gather-Liste **221**, um über das Datenadressregister auf die Datenblöcke D0–D7 zuzugreifen. Während eines DMA-Schreibvorgangs werden die Datenblöcke D0–D7 und die Prüfinformationen C0–C7 in der gleichen Weise, wie es oben beschrieben wurde, übertragen und in dem Prozeß verschachtelt und als Blöcke B0–B7 in dem Speichermedium **207** gespeichert. Ein analoger Lesevorgang bringt wiederum das Entfalten der verschachtelten Daten aus dem Speichermedium **207** durch die DMA-Steuerung **213** unter Verwendung ihrer Register und der Scatter/Gather-Liste **221** mit sich. Der Lesevorgang in Bezug auf die Scatter/Gather-Liste wird unten unter Bezugnahme auf die [Fig. 2E](#) und [Fig. 2F](#) ausführlicher beschrieben.

[0045] Statt Prüfinformationen zusammenhängend zu speichern, kann es vorteilhaft sein, auf Prüfinformationen zuzugreifen zu können, die im Hauptspeicher verteilt sind. [Fig. 2E](#) ist ein vereinfachtes Blockdiagramm, welches noch eine weitere Architektur einer DMA-Steuerung **219** und Aspekte eines Verfahrens für einen Schreibvorgang darstellt. In diesem Beispiel sind die Datenblöcke D0–D7 in dem Hauptspeicher **217** verstreut. Insbesondere gibt es zusammenhängende Sätze von Datenblöcken D0–D2, D3–D4, D6–D7 sowie Block D5. Die zugehörigen Prüfinformationen C0–C7 sind ebenfalls in dem Hauptspei-

cher **217** verstreut. Insbesondere gibt es zusammenhängende Sätze von Prüfinformationsblöcken C0–C2, C3–C4, C6–C7 sowie Block C5. In diesem Beispiel spiegelt das Verteilungsmuster der Prüfinformationsblöcke C0–C7 dasjenige der Datenblöcke D0–D7 wider. Die Prüfinformationen können auf diese Weise erzeugt und im Hauptspeicher **217** gespeichert werden, so daß der Prozeß der Zuweisung von Speicher, der die Prüfinformationen der verschiedenen Datenblöcke enthalten soll, in überaus effizienter Weise ablaufen kann. Wie in dem vorherigen Beispiel sind die Datenadressregister so konfiguriert, daß sie nicht nur eine Datenblock-Startadresse, sondern auch eine entsprechende Blocklänge beinhalten. Auch enthalten in diesem Beispiel die Prüfinformations-Adressregister nicht nur eine Prüfinformationsblock-Startadresse, sondern auch eine entsprechende Blocklänge. Somit arbeitet die DMA-Steuerung **219** in sehr ähnlicher Weise, wie es oben zu den [Fig. 2C](#) und [Fig. 2D](#) beschrieben wurde, nur sind in diesem Fall sowohl das Datenadressregister als auch das Prüfinformations-Adressregister auf eine Scatter/Gather-Liste angewiesen, um Daten aus dem Hauptspeicher **217** zu beschaffen.

[0046] [Fig. 2F](#) zeigt beispielhafte Scatter/Gather-Listen **223** und **225**, die verwendet werden, um eine DMA-Übertragung der Datenblöcke D0–D7 und der zugehörigen Prüfinformationen C0–C7 aus dem Hauptspeicher **217** zu dem Speichermedium **207** zu implementieren. Es gibt vier Elemente in der Scatter/Gather-Liste **223**, die von 0–3 numeriert sind, und vier analoge Elemente in der Scatter/Gather-Liste **225**. In diesem Beispiel enthält jedes der Elemente 0–3 in den Listen entweder die Startadresse eines Datenblocks und seine entsprechende Blocklänge oder die Startadresse eines Prüfinformationsblocks und seine entsprechende Blocklänge. Die Scatter/Gather-Listen **223** und **225** stellen der DMA-Steuerung **219** somit das Verteilungsmuster und die Längen der sechs zusammenhängenden Blöcke D0–D2, D3–D4, D6–D7, C0–C2, C3–C4, C6–C7 sowie der Blöcke D5 und C5 im Hauptspeicher **217** zur Verfügung. Während einer DMA-Übertragung, wie sie oben zu den [Fig. 2C](#) und [Fig. 2D](#) beschrieben wurde, verwendet in diesem Fall die DMA-Steuerung die Scatter/Gather-Listen **223** und **225**, um über das Datenadressregister auf die Datenblöcke D0–D7 bzw. über das Prüfinformations-Adressregister auf die Datenblöcke C0–C7 zuzugreifen. Die Datenblöcke D0–D7 und die Prüfinformationen C0–C7 werden in der gleichen Weise übertragen, wie es oben beschrieben wurde, und werden in dem Vorgang verschachtelt und als zusammenhängende Blöcke B0–B7 in dem Speichermedium **207** gespeichert.

[0047] Bei einem typischen Lesevorgang verwendet die DMA-Steuerung **219** die oben genannten Register, um die gespeicherten Informationen B0–B7 in

ihre Bestandteile zu entfalten. Bei diesem Lesevorgang werden die zusammenhängenden Blöcke D0–D2, D3–D4, D6–D7, C0–C2, C3–C4, C6–C7 sowie die Blöcke D5 und C5 beispielsweise in den Hauptspeicher **217** zurückgeschrieben. In diesem Fall werden, da die Scatter/Gather-Listen **223** und **225** verwendet wurden, weil der Entfaltungsvorgang zwei Informationsströme (Daten und korrespondierende Prüfinformationen) liefert, diese jeweils an die in ihrer jeweiligen Scatter/Gather-Liste spezifizierten Orte "verstreut". Unter Bezugnahme auf [Fig. 2C](#) (wo nur eine Scatter/Gather-Liste verwendet wurde) wird nur einer der beiden Informationsströme an die in der Scatter/Gather-Liste spezifizierten Orte verstreut.

[0048] [Fig. 3A](#) stellt einen Arbeitsablauf **301** dar, der Aspekte eines Verfahrens zum Übertragen von Daten und Prüfinformationen aus einem Hauptspeicher zu einer Speichereinrichtung gemäß einer Ausführungsform der Erfindung zeigt. In diesem Beispiel berechnet eine CPU bei **303** die Prüfinformationen für einen Datenblock im Hauptspeicher. Die Prüfinformationen werden dann bei **305** in einem separaten Teil des Hauptspeichers gespeichert. Wie es in den obigen Beispielen angemerkt wurde, müssen die Prüfinformationen nicht zusammenhängend gespeichert werden. Als nächstes wird eine Entscheidung dahingehend getroffen, ob alle Prüfinformationen berechnet wurden. Falls die Antwort Nein ist, werden bei **309** die Prüfinformationen für den nächsten spezifizierten Datenblock berechnet. Der Ablauf **303–309** wird wiederholt, bis für den gesamten Satz von gewünschten Datenblöcken die zugehörigen Prüfinformationen erzeugt wurden. Unter Rückkehr zum Entscheidungsblock **307** kann eine DMA-Steuerung, wenn alle Prüfinformationen berechnet wurden, bei **311** eine Speicherstartadresse beschaffen, an der die Daten und die zugehörigen Prüfinformationen in dem Speichermedium gespeichert werden sollen. Es sei angemerkt, daß die Speicheradresse nicht in der DMA-Steuerung liegen muß. Beispielsweise kann die Speichereinrichtung diese bei einigen Anwendungen direkt von einer CPU erhalten, oder die Speichereinrichtung kann beginnend an einem impliziten gegenwärtigen Ort einfach in das Speichermedium schreiben. Als nächstes kann die DMA-Steuerung bei **313** auch Speicherlängeninformationen für die zu verschachtelnden und zu speichernden Daten beschaffen. Bei Block **315** werden ein Datenblock und seine zugehörigen Prüfinformationen durch die DMA-Steuerung und die Speichereinrichtung verschachtelt und zusammenhängend in einem Speichermedium gespeichert. Bei Block **319** wird eine Entscheidung dahingehend getroffen, ob alle Daten und die zugehörigen Prüfinformationen zu dem Speichermedium übertragen wurden. Falls nicht, wird die Speicherstartadresse inkrementiert bzw. erhöht. Die Blöcke **315–321** werden für jeden Datenblock und seine zugehörigen Prüfinformationen, die von dem Hauptspeicher zu dem Speichermedium übertragen wer-

den sollen, wiederholt. Unter Rückkehr zu Block **319** ist das Verfahren abgeschlossen, wenn alle Daten (und die zugehörigen Prüfinformationen) übertragen wurden.

[0049] [Fig. 3B](#) stellt Aspekte des Ablaufschritts **315** aus [Fig. 3A](#) dar. Startadreßdaten, die Adressen im Hauptspeicher für die Datenblöcke und ihre zugehörigen Prüfinformationen angeben, werden bei **323** beschafft. Dann werden bei **325** der Datenblock und seine Prüfinformationen gemäß den Verschachtelungskriterien zu einer Speichereinrichtung übertragen. Als nächstes werden bei **327** die Adreßdaten, die die Startadressen der Datenblöcke und der Prüfinformationen angeben, inkrementiert, so daß jegliche nachfolgenden Übertragungen an die korrekten Adressen im Hauptspeicher gerichtet werden. Während der in [Fig. 3A](#) dargestellte Ablauf durch die durch die Blöcke **315–321** dargestellte Schleife fortgesetzt wird, werden jeder nachfolgende Datenblock und die zugehörigen Prüfinformationen verschachtelt, und die Startadreßdaten werden inkrementiert, wie es in [Fig. 3B](#) dargestellt ist.

[0050] Ein Verfahren und ein System gemäß der vorliegenden Erfindung werden typischerweise in einer Computervorrichtung implementiert, die speziell für die benötigten Zwecke ausgestaltet sein kann, oder es kann sich dabei um einen Vielzweckcomputer handeln, der durch ein Computerprogramm und/oder eine in dem Computer gespeicherte Datenstruktur selektiv aktiviert oder neu konfiguriert werden kann. Die hierin dargestellten Abläufe sind nicht notwendigerweise auf irgendeinen bestimmten Computer bezogen. Insbesondere können verschiedene Allzweckmaschinen mit gemäß den hierin angegebenen Lehren geschriebenen Programmen verwendet werden, oder es kann geeigneter sein, eine spezialisiertere Einrichtung zu konstruieren, um die erforderlichen Verfahrensschritte auszuführen.

[0051] Zusätzlich können bei dem hierin beschriebenen Ansatz computerlesbare Medien oder Computerprogrammprodukte, die Programmbefehle und/oder Daten (einschließlich Datenstrukturen) zum Ausführen verschiedener computerimplementierter Prozesse beinhalten, verwendet werden. Beispiele computerlesbarer Medien beinhalten magnetische Medien, wie Festplatten, Disketten und Magnetbänder, optische Medien, wie CD-ROMs, magnetooptische Medien, Halbleiterspeichereinrichtungen und Hardwareeinrichtungen, die speziell dafür konfiguriert sind, Programmbefehle zu speichern und auszuführen, wie Nur-Lese-Speichereinrichtungen (ROM) und Speicher mit wahlfreiem Zugriff (RAM), sind jedoch nicht hierauf beschränkt. Die Daten- und Programmbefehle können auch auf einer Trägerwelle oder einem anderen Transportmedium ausgeführt sein. Beispiele von Programmbefehlen beinhalten sowohl Maschinencode, wie er z.B. durch einen

Compiler produziert wird, als auch Dateien, die höheren Code bzw. höhere Sprache enthalten, die durch den Computer unter Verwendung eines Übersetzungsprogramms kompiliert bzw. übersetzt oder ausgeführt werden können.

[0052] **Fig. 4** veranschaulicht ein typisches Computersystem, in dem, wenn es in geeigneter Weise konfiguriert oder ausgestaltet ist, ein Verfahren und ein System gemäß einer Ausführungsform der Erfindung implementiert werden können. Das Computersystem **400** beinhaltet irgendeine Anzahl von Prozessoren **402** (auch als Zentralverarbeitungseinheiten oder CPUs bezeichnet), die mit Speichereinrichtungen verbunden sind, die einen Primärspeicher **406** (typischerweise einen Speicher mit wahlfreiem Zugriff oder RAM) und einen Primärspeicher **404** (typischerweise einen Nur-Lese-Speicher oder ROM) beinhalten. Die CPU **402** kann in unterschiedlichen Formen vorliegen, einschließlich Mikrocontroller und Mikroprozessoren, wie z.B. programmierbare Einrichtungen (z.B. CPLDs und FPGAs) und nicht-programmierbare Einrichtungen, wie ACISs aus Gatearrays oder Vielzweckmikroprozessoren. Wie es im Stand der Technik gut bekannt ist, funktioniert der Primärspeicher **404** so, daß er Daten und Befehle unidirektional an die CPU überträgt, und der Primärspeicher **406** wird typischerweise verwendet, um Daten und Befehle bidirektional zu übertragen. Diese beiden Primärspeichereinrichtungen können jegliche geeignete computerlesbare Medien beinhalten, wie beispielsweise diejenigen, die oben beschrieben wurden. Eine Massenspeichereinrichtung **408** kann ebenfalls bidirektional mit der CPU **402** verbunden sein und stellt eine zusätzliche Datenspeicherkapazität bereit und kann jegliche der oben beschriebenen computerlesbaren Medien beinhalten. Die Massenspeichereinrichtung **408** kann verwendet werden, um Programme, Daten und dergleichen zu speichern, und ist typischerweise ein Sekundärspeichermedium, wie z.B. eine Festplatte. Es versteht sich, daß die in der Massenspeichereinrichtung **408** gespeicherten Informationen, wo dies geeignet ist, standardmäßig als Teil des Primärspeichers **406** als virtueller Speicher enthalten sein können. Eine bestimmte Massenspeichereinrichtung, wie z.B. eine CD-ROM **414**, kann Daten auch unidirektional an die CPU übermitteln.

[0053] Die CPU **402** kann auch mit einer Schnittstelle **410** verbunden sein, an die eine oder mehrere Eingabe/Ausgabe-Vorrichtungen, wie beispielsweise Videomonitore, Rollkugeln (Trackballs), Mäuse, Tastaturen, Mikrofone, berührungsempfindliche Anzeigen, Transducer-Kartenlesegeräte, Magnetband- oder Lochstreifenleser, Schreibtafeln, Stifte, Sprach- oder Handschrifterkennungseinrichtungen oder andere wohlbekanntes Eingabegeräte, wie natürlich beispielsweise auch andere Computer, angeschlossen ist bzw. sind. Zusätzlich kann die CPU **402** optional

unter Verwendung eines externen Anschlusses, wie er allgemein bei **412** gezeigt ist, mit einer externen Einrichtung, wie einer Datenbank oder einem Computer oder einem Telekommunikationsnetz, verbunden sein. Man kann sich vorstellen, daß die CPU mit einem solchen Anschluß bzw. einer solchen Verbindung Informationen von dem Netzwerk empfangen oder Informationen an das Netzwerk ausgeben kann, während sie die hierin beschriebenen Verfahrensschritte ausführt.

[0054] Eine DMA-Steuerung, wie sie hierin beschrieben ist, kann in vielen Anordnungen und Konfigurationen, wie sie für einen Fachmann auf dem Gebiet auf der Hand liegen, in das Computersystem **400** integriert sein. Darüber hinaus können die oben beschriebenen Ablaufschritte durch Schaltkreise ausgeführt werden, die dafür ausgestaltet sind, die obigen Schritte auszuführen. Zusätzlich kann für das Übertragen von Daten und Prüfinformationen zwischen dem Hauptspeicher und dem Speichermedium ein Bus verwendet werden.

[0055] Obwohl die Erfindung im Hinblick auf einige bevorzugte Ausführungsformen beschrieben wurde, soll sie nicht auf die bestimmten oben angegebenen Formen beschränkt sein. Viele Variationen der oben beschriebenen bevorzugten Ausführungsformen können verwendet werden. Daher sollte die Erfindung unter Bezugnahme auf die nachfolgenden Ansprüche allgemein interpretiert werden.

Patentansprüche

1. Verfahren zum Übertragen eines Datensatzes und eines zugehörigen Satzes von Prüfinformation, wobei das Verfahren die folgenden, von einer Steuerung für direkten Speicherzugriff (Direct Memory Access Controller) (**107**) durchgeführten Schritte aufweist:

- (a) Beschaffen des Datensatzes aus einem Speicher (**105**), wobei der Speicher extern zu der Steuerung für direkten Speicherzugriff ist;
- (b) Beschaffen des zugehörigen Satzes von Prüfinformation aus dem Speicher;
- (c) Verschachteln des Datensatzes und des zugehörigen Satzes von Prüfinformation zum Kreieren eines verschachtelten Datensatzes in einem Speichermedium (**109**);

wobei der Verschachtelungsschritt mittels eines Satzes von Verschachtelungskriterien durchgeführt wird, wobei die Verschachtelungskriterien entweder direkt oder indirekt eine Datenblockgröße für jeden Block des Datensatzes und eine Prüfinformationsblockgröße für jeden Block des zugehörigen Satzes von Prüfinformation angeben.

2. Verfahren nach Anspruch 1, wobei der Speicher einem Hauptspeicher entspricht, und der verschachtelte Datensatz in einem Speichermedium se-

parat von dem Hauptspeicher gespeichert ist.

3. Verfahren nach Anspruch 1 oder 2, wobei der verschachtelte Datensatz zusammenhängend in dem Speichermedium gespeichert ist.

4. Verfahren nach einem der vorstehenden Ansprüche, wobei das Speichermedium Blöcke fester Größe aufweist.

5. Verfahren nach einem der vorstehenden Ansprüche, wobei der Datensatz zusammenhängende Blöcke von Daten aufweist, die sich in dem Speicher befinden, der an einer ersten Speicherposition beginnt, und die zugehörige Prüfinformation zusammenhängende Blöcke von Prüfinformation aufweist, die sich in dem Speicher befinden, der an einer zweiten Speicherposition beginnt.

6. Verfahren nach Anspruch 5, wobei das Beschaffen des Datensatzes und des zugehörigen Satzes von Prüfinformation das Verwenden eines ersten Registers aufweist, das eine Startadresse des Datensatzes angibt, und eines zweiten Registers, das die Startadresse des zugehörigen Satzes von Prüfinformation angibt.

7. Verfahren nach einem der Ansprüche 1 bis 4, wobei der Datensatz eine Mehrzahl von verstreut liegenden Datenblöcken aufweist, und die zugehörige Prüfinformation einen Satz von zusammenhängenden Prüfinformationsblöcken aufweist.

8. Verfahren nach Anspruch 7, wobei das Beschaffen des Datensatzes und des Satzes von zugehöriger Prüfinformation das Verwenden aufweist von: einer Verstreutensammelliste, wobei jedes Element der Verstreutensammelliste eine Adresse und eine Länge eines zusammenhängenden Satzes von einem oder mehreren Datenblöcken beinhaltet; und eines Registers, welches die Startadresse des zugehörigen Satzes von Prüfinformation angibt.

9. Verfahren nach einem der Ansprüche 1 bis 4, wobei der Datensatz eine erste Mehrzahl von verstreuten Datenblöcken aufweist, und die zugehörige Prüfinformation eine zweite Mehrzahl von verstreuten Prüfinformationsblöcken aufweist.

10. Verfahren nach Anspruch 9, wobei das Beschaffen des Datensatzes und des zugehörigen Satzes von Prüfinformation das Verwenden einer ersten Verstreutensammelliste zur Angabe von Information zu Startadresse und Länge für jeden Datenblock oder irgendwelche zusammenhängenden Sätze von Datenblöcken der ersten Mehrzahl von verstreuten Datenblöcken und einer zweiten Verstreutensammelliste zur Angabe von Information zu Startadresse und Länge für jeden Prüfinformationsblock oder irgendwelche zusammenhängenden Sätze von Prüfinfor-

mationsblöcken der zweiten Mehrzahl von verstreuten Prüfinformationsblöcken aufweist.

11. Verfahren nach Anspruch 9, wobei das Beschaffen des Datensatzes und des zugehörigen Satzes von Prüfinformation das Verwenden einer Verstreutensammelliste aufweist, wobei jedes Element der Verstreutensammelliste eine Adresse und eine Länge eines zusammenhängenden Satzes von einem oder mehreren Datenblöcken aus der ersten Mehrzahl von verstreuten Datenblöcken und eine Adresse und eine Länge eines entsprechenden zusammenhängenden Satzes von einem oder mehreren Prüfinformationsblöcken aus der zweiten Mehrzahl von verstreuten Prüfinformationsblöcken angibt.

12. Verfahren nach einem der vorstehenden Ansprüche, das ferner das Verwenden eines oder mehrerer Register zur Angabe mindestens einer der folgenden Größen aufweist: einer Speicheradresse, die eine Adresse in dem Speichermedium angibt, in dem der verschachtelte Datensatz zu speichern ist, einer Speicherlänge, der Datenblockgröße und der Prüfinformationsblockgröße.

13. Verfahren nach einem der vorstehenden Ansprüche, das ferner aufweist:

- (a) Berechnen eines Prüfinformationsblocks für einen von einer Mehrzahl von Datenblöcken in dem Hauptspeicher;
- (b) Speichern des Prüfinformationsblocks in dem Hauptspeicher;
- (c) Wiederholen von (a) und (b) für jeden Block der Mehrzahl von Datenblöcken im Hauptspeicher, wodurch ein Satz von Prüfinformationsblöcken in dem Hauptspeicher kreiert wird, wobei jeder Block aus dem Satz von Prüfinformationsblöcken einem Block von der Mehrzahl der Datenblöcken im Hauptspeicher entspricht, wobei die Mehrzahl von Datenblöcken und der Satz von Prüfinformationsblöcken gemeinsam einen Datensatz bilden;
- (d) Zusammenhängendes Speichern eines Blocks der Mehrzahl von Datenblöcken und eines Blocks aus dem Satz von diesem Datenblock entsprechenden Prüfinformationsblöcken in dem Speichermedium an einer Speicheradresse in dem Speichermedium mittels der Verschachtelungskriterien mit Hilfe der Steuerung für direkten Speicherzugriff;
- (e) Inkrementieren der Speicheradresse mittels der Verschachtelungskriterien; und
- (f) Wiederholen von (d)-(e), bis der gesamte Datensatz übertragen wurde.

14. Verfahren nach Anspruch 13, wobei das Speichern eines Blockes von der Mehrzahl von Datenblöcken und eines Blockes von dem Satz von dem Datenblock entsprechenden Prüfinformationsblöcken mittels eines Satzes von Verschachtelungskriterien aufweist:

Beschaffen einer Datenstartadresse, von der ein Da-

tenblock oder ein zusammenhängender Satz von Datenblöcken von der Mehrzahl von Datenblöcken zu übertragen ist;

Beschaffen einer Prüfinformationsblock-Startadresse, von der ein zugehöriger Prüfinformationsblock oder ein zusammenhängender Satz von zugehörigen Prüfinformationsblöcken zu übertragen ist;

Übertragen eines Datenblocks und seines zugehörigen Prüfinformationsblocks in einem zusammenhängenden Format zu dem Speichermedium; und Inkrementieren der Datenstartadresse und der Prüfinformationsblock-Startadresse im Hauptspeicher; wobei die Verschachtelungskriterien die Längeninformation eines oder mehrerer von der Mehrzahl der zu übertragenden Datenblöcke und Längeninformation eines oder mehrerer von dem Satz von zu übertragenden, zugehörigen Prüfinformationsblöcken angeben, und das Inkrementieren der Datenstartadresse und der Prüfinformationsblock-Startadresse das Addieren jeweils der Längeninformation eines oder mehrerer Blöcke von der Mehrzahl von Datenblöcken und der Längeninformation eines oder mehrerer Blöcke des Satzes der zugehörigen Prüfinformationsblöcke zu jeweils der Datenstartadresse und der Prüfinformation-Startadresse aufweist.

15. Verfahren nach Anspruch 13, wobei (a)–(c) von einer CPU (**103**) durchgeführt werden.

16. Verfahren zum Übertragen eines Datensatzes und eines zugehörigen Satzes von Prüfinformation, wobei das Verfahren die folgenden von einer Steuerung (**107**) des direkten Speicherzugriffs durchgeführten Schritte aufweist:

(a) Beschaffen eines verschachtelten Datensatzes aus einem Speichermedium (**109**);

(b) Auflösen bzw. Entfalten des verschachtelten Datensatzes, um einen Datensatz und einen zugehörigen Satz von Prüfinformation zu erhalten;

(c) Schreiben des Datensatzes und des zugehörigen Satzes von Prüfinformation in einen Speicher (**105**); wobei der Schritt des Entfaltens mittels eines Satzes von Verschachtelungskriterien durchgeführt wird, wobei die Verschachtelungskriterien entweder direkt oder indirekt eine Datenblockgröße für jeden Block des Datensatzes und eine Prüfinformationsblockgröße für jeden Block des zugehörigen Satzes von Prüfinformation angeben.

17. Verfahren nach Anspruch 16, wobei der Speicher einem Hauptspeicher entspricht und das Speichermedium separat von dem Hauptspeicher ist.

18. Verfahren nach Anspruch 16 oder 17, wobei der verschachtelte Datensatz zusammenhängend in dem Speichermedium gespeichert wird.

19. Verfahren nach einem der Ansprüche 16 bis 18, wobei das Speichermedium Blöcke fester Größe aufweist.

20. Verfahren nach einem der Ansprüche 16 bis 19, wobei der Schreibrschritt das Schreiben des Datensatzes als zusammenhängende Blöcke von Daten beginnend bei einer ersten Speicherposition und das Schreiben der zugehörigen Prüfinformation als zusammenhängende Blöcke von Prüfinformation beginnend bei einer zweiten Speicherposition aufweist.

21. Verfahren nach Anspruch 20, das ferner das Verwenden eines ersten Registers zur Angabe einer Startadresse des Datensatzes und eines zweiten Registers zur Angabe der Startadresse des zugehörigen Satzes von Prüfinformation aufweist.

22. Verfahren nach einem der Ansprüche 16 bis 19, wobei der Schreibrschritt das Schreiben des Datensatzes als eine Mehrzahl von verstreuten Datenblöcken und das Schreiben der zugehörigen Prüfinformation als ein zusammenhängender Satz von Prüfinformationsblöcken aufweist.

23. Verfahren nach Anspruch 22, wobei der Schreibrschritt das Verwenden aufweist von: einer Verstreutensammelliste, wobei jedes Element der Verstreutensammelliste eine Adresse und eine Länge eines zusammenhängenden Satzes von einem oder mehreren Datenblöcken beinhaltet; und eines Registers zur Angabe der Startadresse des zugehörigen Satzes von Prüfinformation.

24. Verfahren nach einem der Ansprüche 16 bis 19, wobei der Schreibrschritt das Schreiben des Datensatzes als eine erste Mehrzahl von verstreuten Datenblöcken und der zugehörigen Prüfinformation als eine zweite Mehrzahl von verstreuten Prüfinformationsblöcken aufweist.

25. Verfahren nach Anspruch 24, wobei der Schreibrschritt das Verwenden einer ersten Verstreutensammelliste zur Angabe von Startadresse und Längeninformation für jeden Datenblock oder jedwede zusammenhängende Sätze von Datenblöcken der ersten Mehrzahl von verstreuten Datenblöcken und einer zweiten Verstreutensammelliste zur Angabe von Startadresse und Längeninformation für jeden Prüfinformationsblock oder jedwede zusammenhängende Sätze von Prüfinformationsblöcken der zweiten Mehrzahl von verstreuten Prüfinformationsblöcken aufweist.

26. Verfahren nach Anspruch 24, wobei der Schreibrschritt das Verwenden einer Verstreutensammelliste aufweist, wobei jedes Element der Verstreutensammelliste eine Adresse und eine Länge eines zusammenhängenden Satzes von einem oder mehreren Datenblöcken aus der ersten Mehrzahl von verstreuten Datenblöcken und eine Adresse und eine Länge eines zugehörigen, zusammenhängenden Satzes von einem oder mehreren Prüfinformationsblöcken aus der zweiten Mehrzahl von verstreuten

Prüfinformationsblöcken angibt.

27. Verfahren nach einem der Ansprüche 16 bis 26, das ferner das Verwenden eines oder mehrerer Register zur Angabe mindestens einer der folgenden Größen aufweist: einer Speicheradresse, die eine Adresse in dem Speichermedium angibt, wo der verschachtelte Datensatz gespeichert ist, einer Speicherlänge, der Datenblockgröße und der Prüfinformationsblockgröße.

28. Verfahren nach einem der Ansprüche 16 bis 27, das ferner aufweist:

(a) Beschaffen eines zusammenhängend gespeicherten Datensatzes mit einem Satz von Datenblöcken und einem Satz von zugehörigen Prüfinformationsblöcken, die auf dem Speichermedium verschachtelt sind.

(b) Entfalten des zusammenhängend gespeicherten Datensatzes mittels eines Satzes von Verschachtelungskriterien, um den Satz von Datenblöcken und den Satz von Prüfinformationsblöcken zu erhalten; und

(c) Schreiben des Satzes von Datenblöcken und des Satzes von zugehörigen Prüfinformationsblöcken in den Hauptspeicher;

wobei (b) in einer Steuerung des direkten Speicherzugriffs durchgeführt wird, die eine Mehrzahl von Registern umfaßt, wobei die Mehrzahl von Registern entweder direkt oder indirekt mindestens eine Datenblockgröße für jeden Block des Satzes von Datenblöcken und eine Prüfinformationsblockgröße für jeden Block des Satzes von zugehörigen Prüfinformationsblöcken angibt.

29. Verfahren nach Anspruch 28, wobei (b)–(c) aufweisen:

(d) Beschaffen einer Datenstartadresse, zu der ein Datenblock oder ein zusammenhängender Satz von Datenblöcken der Mehrzahl von Datenblöcken zu übertragen ist;

(e) Beschaffen einer Prüfinformationsblock-Startadresse, zu der ein zugehöriger Prüfinformationsblock oder ein zusammenhängender Satz von zugehörigen Prüfinformationsblöcken zu übertragen ist;

(f) Übertragen eines Datenblocks und seines zugehörigen Prüfinformationsblocks in den Hauptspeicher;

(g) Inkrementieren der Datenstartadresse und Prüfinformationsblock-Startadresse im Hauptspeicher; und
(h) Wiederholen von (d) – (g), bis der gesamte Datensatz übertragen wurde.

wobei der Satz von Verschachtelungskriterien Längeninformation eines oder mehrerer Blöcke von der Mehrzahl von zu übertragenden Datenblöcken und Längeninformation eines oder mehrerer Blöcke von dem Satz von zu übertragenden, zugehörigen Prüfinformationsblöcken angibt, und das Inkrementieren der Datenstartadresse und der Prüfinformationsblock-Startadresse das Addieren jeweils der Längen-

informationen eines oder mehrerer Blöcke von der Mehrzahl von zu übertragenden Datenblöcken und der Längeninformation eines oder mehrerer Blöcke von dem Satz von zu übertragenden, zugehörigen Prüfinformationsblöcken zu jeweils der Datenstartadresse und der Prüfinformation-Startadresse aufweist.

30. Verfahren nach Anspruch 29, wobei die Prüfinformation für jeden in (f) übertragenen Datenblock neu berechnet und mit dem entsprechenden, in (f) übertragenen Prüfinformationsblock verglichen wird, um die Korrektheit der Übertragung zu überprüfen.

31. Verfahren nach Anspruch 30, wobei die Neuberechnung der Prüfinformation und der Vergleich mit den übertragenen Prüfinformationsblöcken von einer CPU (**103**) durchgeführt werden.

32. Ein Computerprogramm zum Implementieren des Verfahrens nach einem der vorstehenden Ansprüche.

33. Ein Computerprogrammprodukt, das ein computerlesbares Medium aufweist, das Anweisungen speichert, die das Computerprogramm nach Anspruch 32 repräsentieren.

34. Eine Steuerung (**107**) für direkten Speicherzugriff (Direct Memory Access, DMA) zum Übertragen von Daten zwischen einem Hauptspeicher (**105**) und einer Datenspeichereinrichtung (**109**), wobei die DMA-Steuerung aufweist:

ein erstes Register zum Speichern einer Startadresse, von der Daten aus dem Hauptspeicher zu der Datenspeichereinrichtung zu übertragen sind oder zu der Daten in den Hauptspeicher aus der Datenspeichereinrichtung zu schreiben sind;

ein zweites Register zum Speichern einer Startadresse, von der Prüfinformation aus dem Hauptspeicher zu der Datenspeichereinrichtung zu übertragen ist oder zu der Prüfinformation in den Hauptspeicher aus der Datenspeichereinrichtung zu schreiben ist;

Mechanismen zur Angabe oder Ermittlung einer Länge von Daten in jedem aus dem Hauptspeicher zu der Datenspeichereinrichtung oder von der Datenspeichereinrichtung zum Hauptspeicher zu übertragenden Datenblock;

Mechanismen zur Angabe oder Ermittlung einer Länge von Prüfinformation in jedem aus dem Hauptspeicher zu der Datenspeichereinrichtung oder von der Datenspeichereinrichtung zum Hauptspeicher zu übertragenden Prüfinformationsblock.

35. DMA-Steuerung nach Anspruch 34, die ferner aufweist:

ein Quellbit, das anzeigt, ob eine aktuelle DMA-Adreßquelle aus dem ersten Register oder dem zweiten Register ist.

36. DMA-Steuerung nach Anspruch 34 oder 35,

die ferner eine Verstreutensammelliste aufweist, in der jedes Element der Liste eine Adresse und eine Länge eines zusammenhängenden Satzes von Daten, die aus dem Hauptspeicher zu der Datenspeichereinrichtung oder aus der Datenspeichereinrichtung in den Hauptspeicher zu übertragen sind.

37. DMA-Steuerung nach Anspruch 34 oder 35, die ferner aufweist:

eine erste Verstreutensammelliste, in der jedes Element der ersten Verstreutensammelliste eine Adresse und eine Länge einer zusammenhängenden Sammlung von Daten angibt, die vom Hauptspeicher zur Datenspeichereinrichtung oder von der Datenspeichereinrichtung zum Hauptspeicher zu übertragen sind; und

eine zweite Verstreutensammelliste, in der jedes Element der zweiten Verstreutensammelliste eine Adresse und eine Länge einer zusammenhängenden Sammlung von Prüfinformation angibt, die vom Hauptspeicher zur Datenspeichereinrichtung oder von der Datenspeichereinrichtung zum Hauptspeicher zu übertragen sind.

38. DMA-Steuerung nach Anspruch 34, die ferner eine Verstreutensammelliste aufweist, in der jedes Element der Liste eine Adresse und eine Länge eines zusammenhängenden Satzes von Daten angibt, die aus dem Hauptspeicher zu der Datenspeichereinrichtung oder aus der Datenspeichereinrichtung in den Hauptspeicher zu übertragen sind, sowie eine Adresse und eine Länge eines zugehörigen, zusammenhängenden Satzes von Prüfinformation, die aus dem Hauptspeicher zu der Datenspeichereinrichtung oder aus der Speichereinrichtung in den Hauptspeicher zu übertragen sind.

Es folgen 8 Blatt Zeichnungen

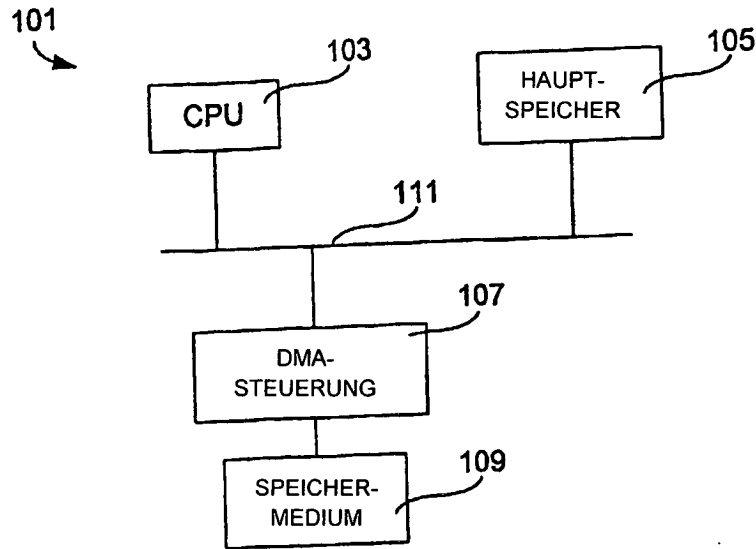


FIG. 1A
(STAND DER TECHNIK)

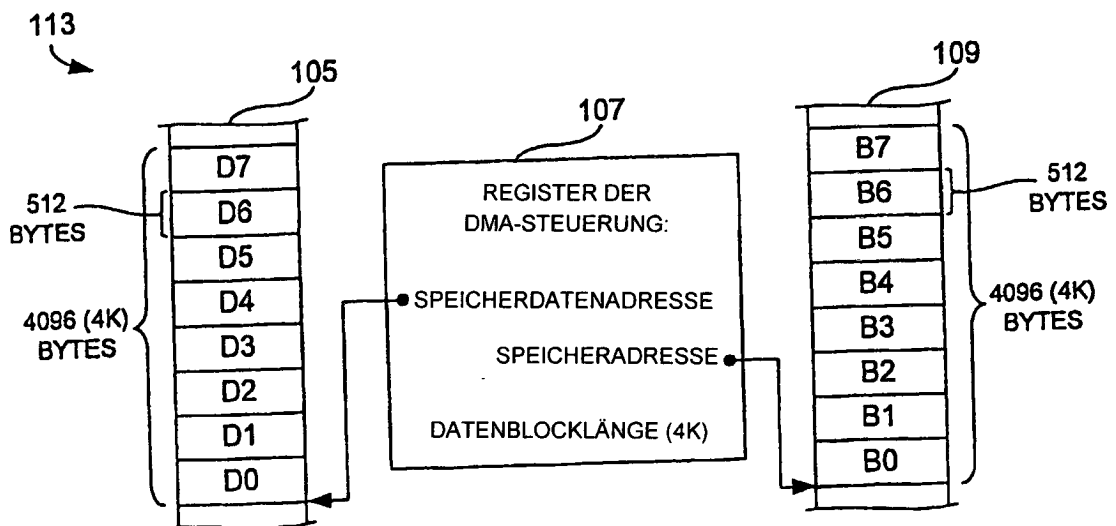
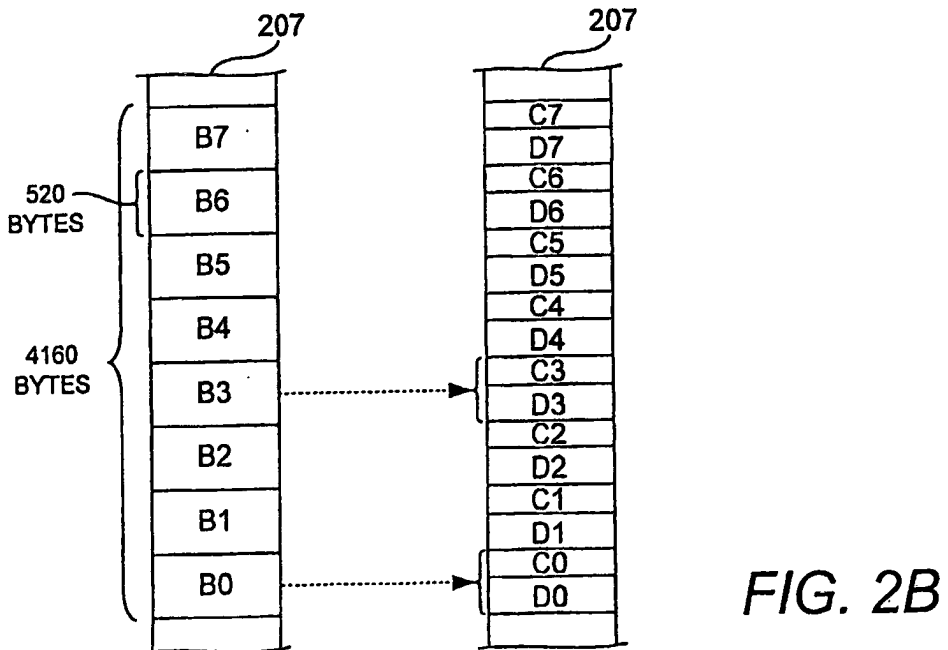
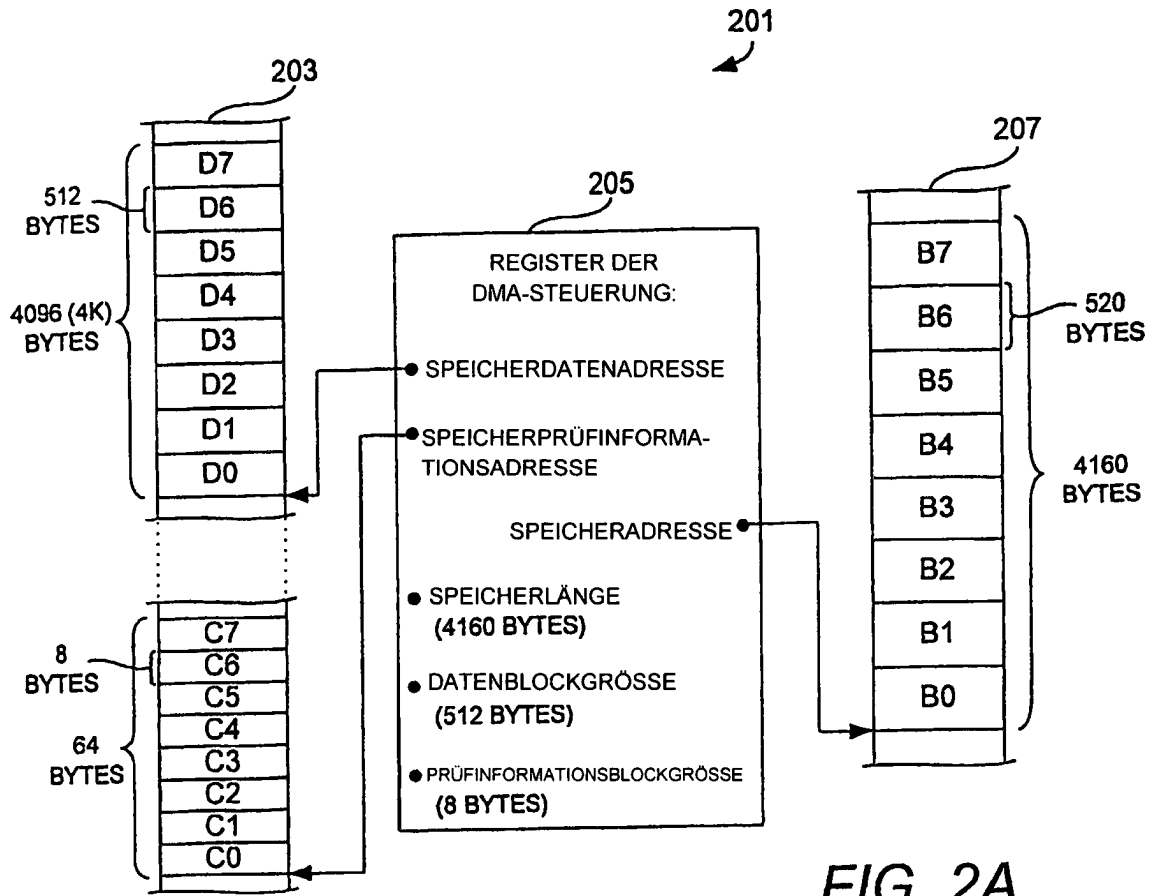


FIG. 1B
(STAND DER TECHNIK)



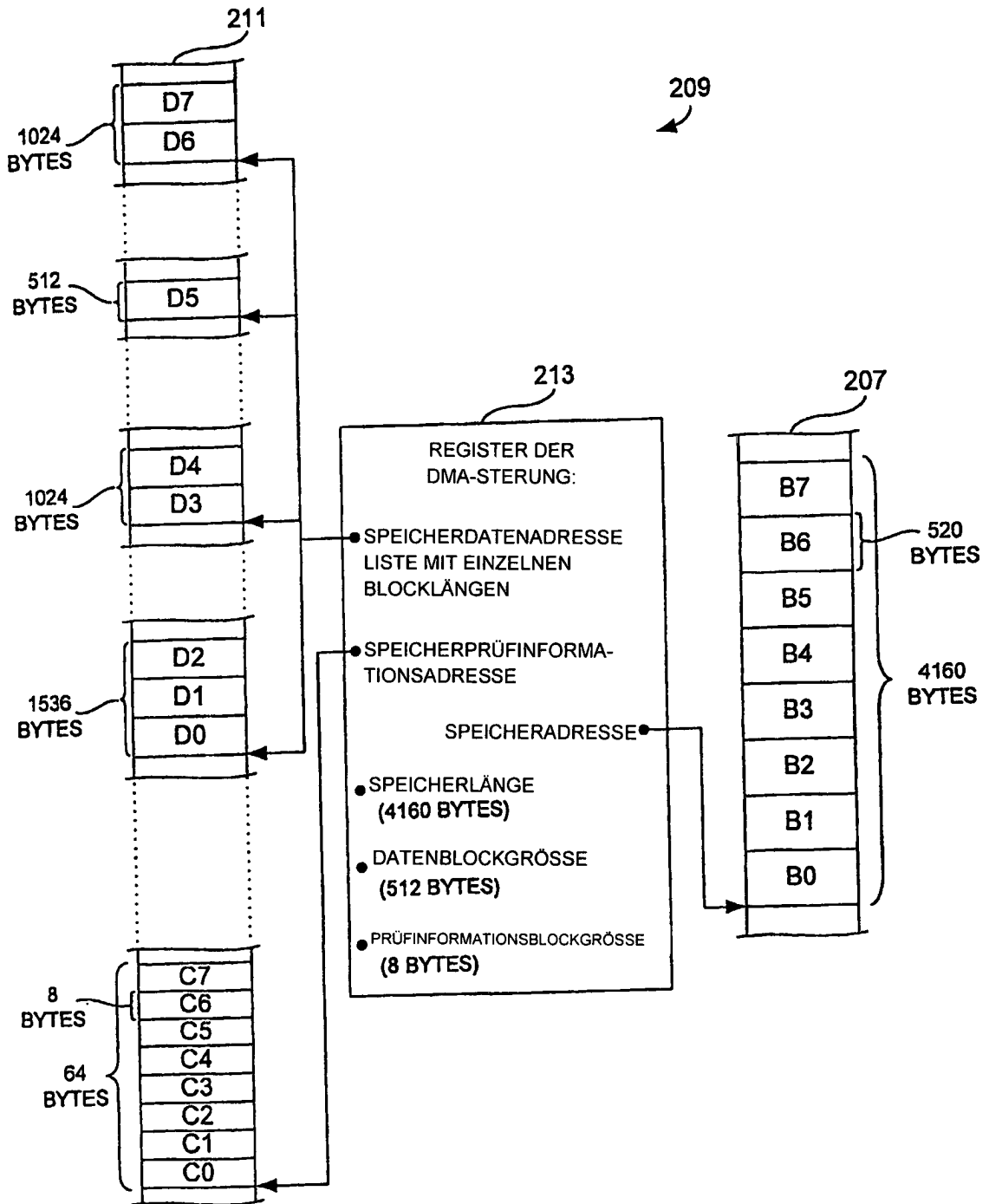


FIG. 2C

221
↙

0	D0	1536 bytes
1	D3	1024 bytes
2	D5	512 bytes
3	D6	1024 bytes

FIG. 2D

223
↙

0	D0	1536 bytes
1	D3	1024 bytes
2	D5	512 bytes
3	D6	1024 bytes

225
↙

0	C0	24 bytes
1	C3	16 bytes
2	C5	8 bytes
3	C6	16 bytes

FIG. 2F

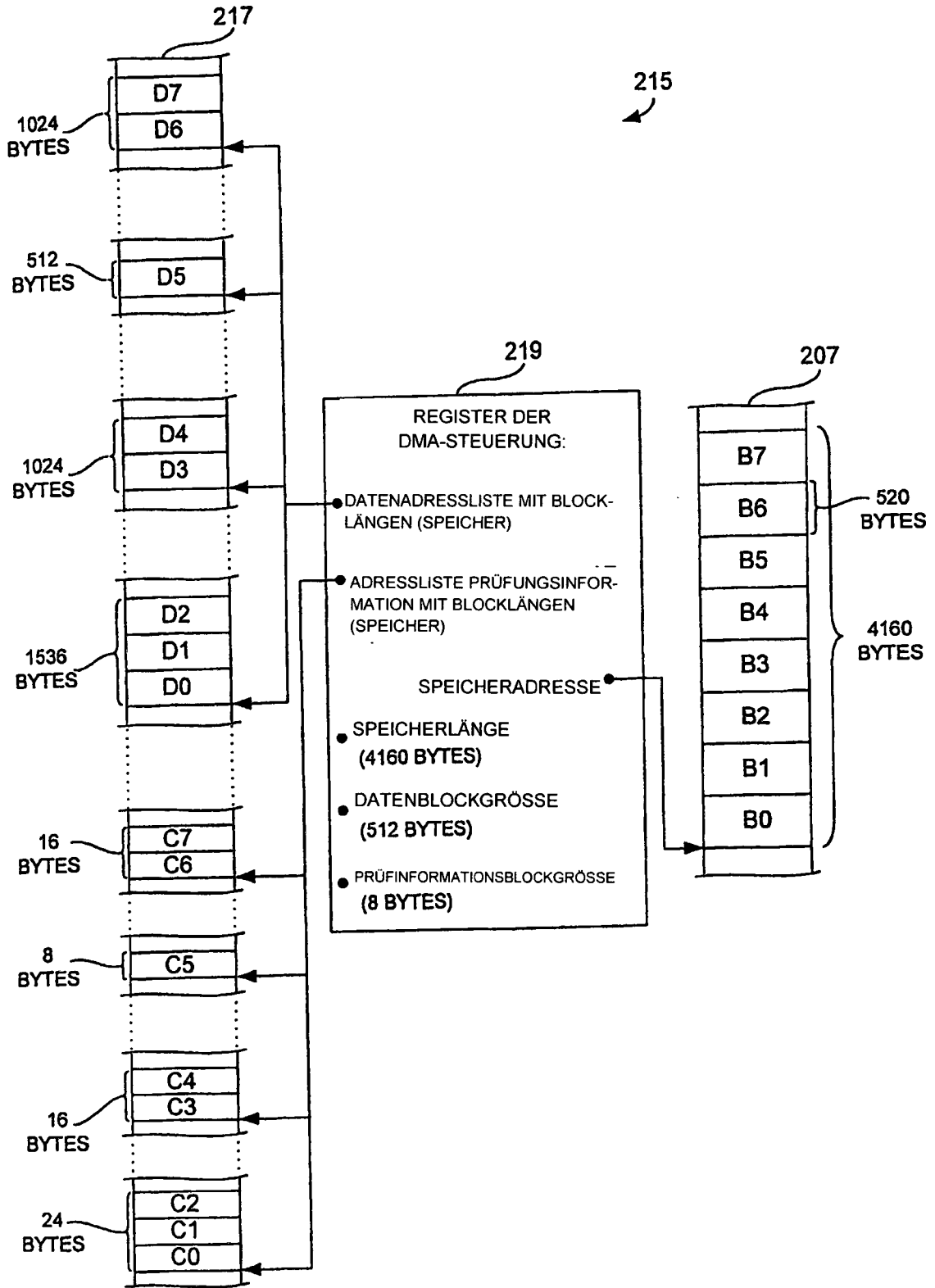


FIG. 2E

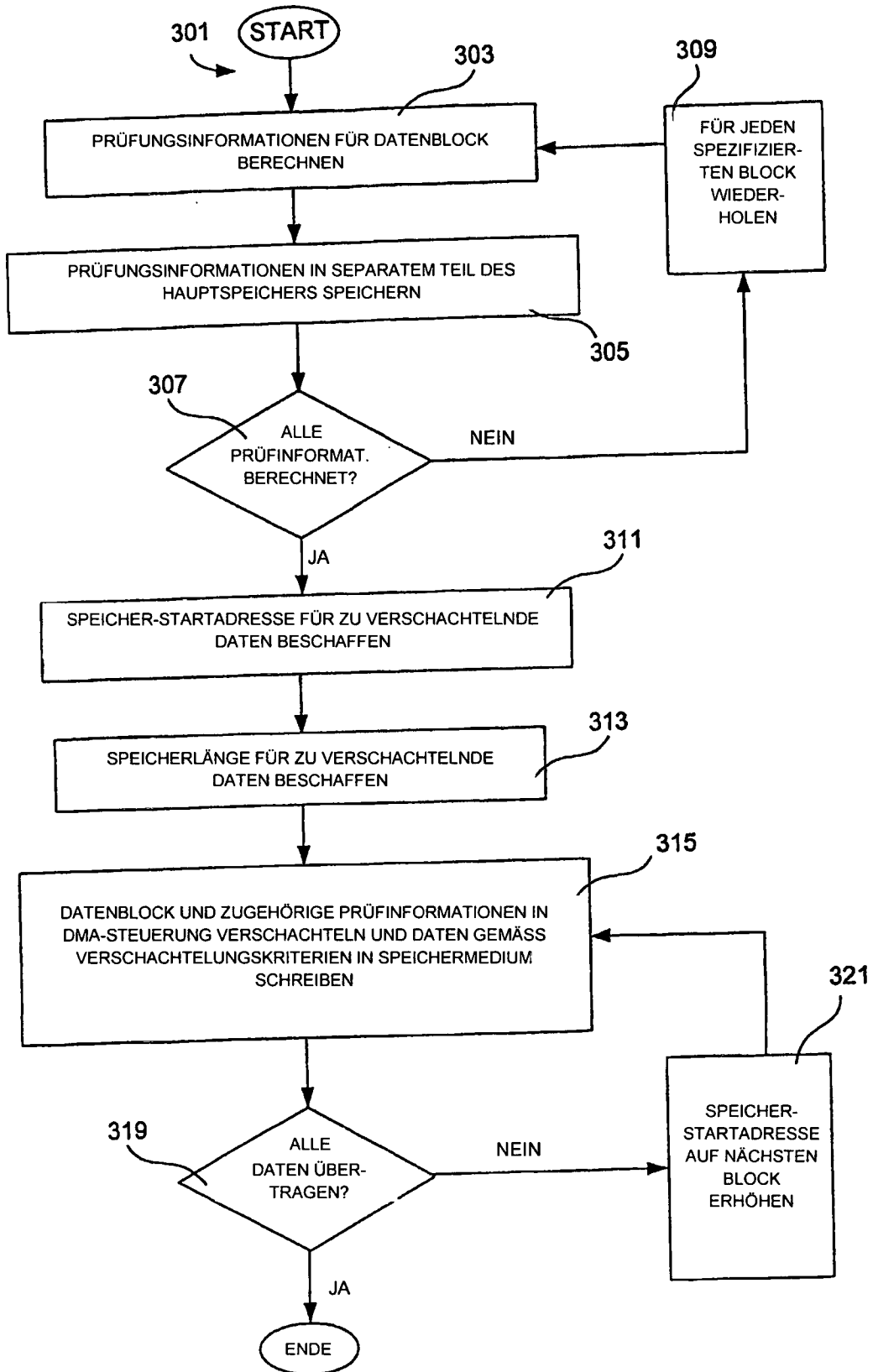


FIG. 3A

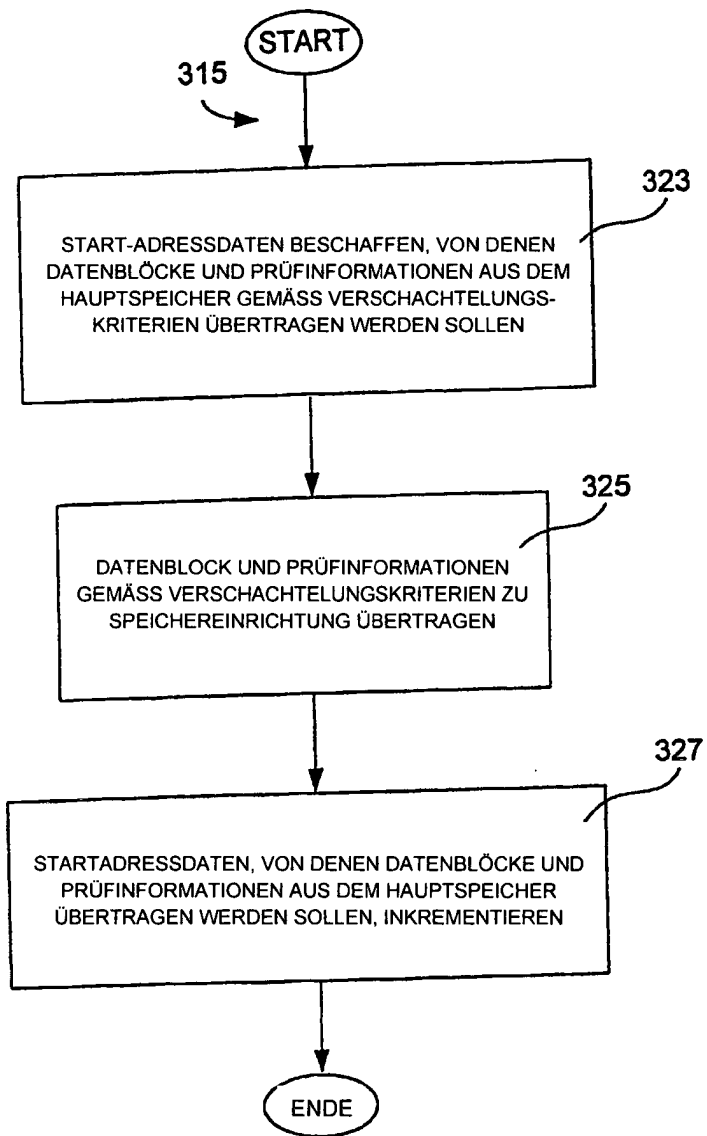


FIG. 3B

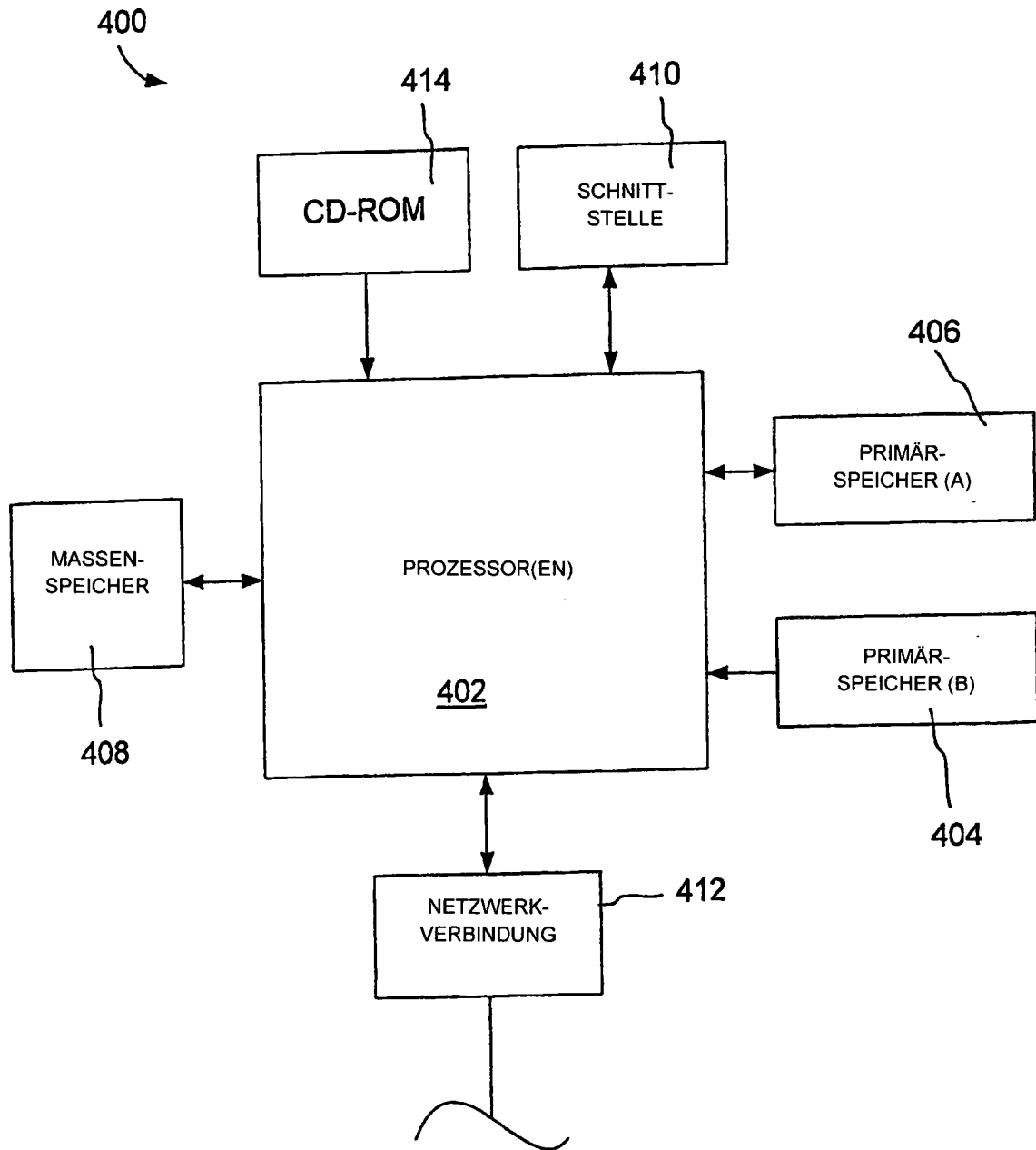


FIG. 4