



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) **DE 697 30 449 T2** 2005.09.15

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 000 397 B1**

(21) Deutsches Aktenzeichen: **697 30 449.3**

(86) PCT-Aktenzeichen: **PCT/US97/16206**

(96) Europäisches Aktenzeichen: **97 941 564.3**

(87) PCT-Veröffentlichungs-Nr.: **WO 98/012642**

(86) PCT-Anmeldetag: **16.09.1997**

(87) Veröffentlichungstag  
der PCT-Anmeldung: **26.03.1998**

(97) Erstveröffentlichung durch das EPA: **17.05.2000**

(97) Veröffentlichungstag  
der Patenterteilung beim EPA: **25.08.2004**

(47) Veröffentlichungstag im Patentblatt: **15.09.2005**

(51) Int Cl.7: **G06F 13/00**  
**G06F 11/20**

(30) Unionspriorität:  
**714255 17.09.1996 US**

(73) Patentinhaber:  
**Marathon Technologies Corp., Boxborough,  
Mass., US**

(74) Vertreter:  
**v. Fünér Ebbinghaus Finck Hano, 81541 München**

(84) Benannte Vertragsstaaten:  
**DE, FR, GB, IT, NL, SE**

(72) Erfinder:  
**FITZGERALD, J., Martin, Medway, US; TREMBLAY,  
A., Glenn, Upton, US**

(54) Bezeichnung: **ERZEUGUNG EINER SPIEGELDATENKOPIE (BILD) UNTER VERWENDUNG VON REFERENZZETTELN**

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]** Die Erfindung bezieht sich auf die Technik der Erzeugung einer gespiegelten Kopie eines Plattenlaufwerks oder einer anderen Speichereinrichtung.

**[0002]** In vielen Computersystemen wird ein Maß an Fehlertoleranz dadurch erreicht, dass identische Daten auf jedem von mehreren Speichern gespeichert werden. Speicher mit identischen Daten werden als gespiegelte Geräte bezeichnet und gehören zu einem „mirror-set“ (gespiegelte Gruppe). Wenn ein gespiegeltes Gerät in einem Mirror-Set ausfällt oder sonst nicht zugreifbar wird, erlaubt das andere gespiegelte Gerät oder Geräte im Mirror-Set einen weiteren Zugriff auf die Daten.

**[0003]** Um auf jedem Gerät in einem Mirror-Set identische Daten zu halten, muss jedes Gerät jede Anforderung zum Speichern von Daten auf dem Mirror-Set empfangen und verarbeiten (d. h. jede Schreib-Anforderung). Ein Gerät in einem Mirror-Set weicht von den anderen Geräten im Mirror-Set ab, wenn es eine solche Schreib-Anforderung nicht verarbeiten kann. Divergieren die Mitglieder eines Mirror-Set, so muss eine Mirror-Set-Kopie ausgeführt werden, um die Daten von einem gespiegelten Gerät auf ein anderes gespiegeltes Gerät zu kopieren. Eine Möglichkeit, eine Mirror-Set-Kopie auszuführen, besteht darin, das Computersystem herunterzufahren und alle Daten von einem gespiegelten Gerät auf das andere gespiegelte Gerät zu kopieren.

**[0004]** Aus EP-A-0 672 984 ist ein Transaktions-Verarbeitungssystem bekannt, das während des Ausfalls und der Wiederherstellung jeder von zwei Duplikat-Datenbanken fortfährt, einlaufende Transaktionen zu verarbeiten. Einer der beiden Duplikat-Datenbanken wird der Status „aktiv“ zugeordnet, während die andere im Status „redundant“ arbeitet. Alle einlaufenden Abfragen werden nur der aktiven Datenbank zugesandt und alle einlaufenden Updates werden der aktiven und der redundanten Datenbank zugeführt. Fällt eine Datenbank aus, so wird der anderen sofort der aktive Zustand zugeordnet, wenn sie nicht bereits aktiv ist, und sie fährt während des Ausfalls fort, einlaufende Abfragen und Updates zu verarbeiten. Wenn die ausgefallene Datenbank repariert ist und neu gestartet wird, werden alle Aufzeichnungen von dieser gelöscht und die ausgefallene Datenbank wird unter Benutzung eingeschobener Kopier- und Update-Operationen in einem einzigen Lauf durch die Datenbank rekonstruiert. Einlaufende Transaktions- und Kopier-Operationen werden entsprechend einem Queue-Schwellenverfahren zum Drosseln der Kopier-Operationen zwischengeschaltet, das auf eine erhöhte Anzahl einlaufender Trans-

aktionen anspricht. Das Transaktions-Verarbeitungssystem bleibt sowohl während des Ausfalls als auch während der Wiederherstellungsaktivitäten betriebsbereit und erreicht eine volle Wiederherstellung ohne Ausfallzeit.

## Zusammenfassung der Erfindung

**[0005]** Nach einem Aspekt der vorliegenden Erfindung wird bereitgestellt ein Verfahren zum Ausführen einer gespiegelten Kopie von einer ersten Speichervorrichtung auf eine zweite Speichervorrichtung in einem Computersystem, bei dem jeder Schreibrequest durch ein Bezugslabel identifiziert wird, wobei das Verfahren umfasst:

Empfangen von Schreibrequests an der ersten Speichervorrichtung,  
Empfangen von Schreibrequests an der zweiten Speichervorrichtung,  
Verarbeiten von Schreibrequests an der ersten Speichervorrichtung,  
Lesen von Spiegeldaten von der ersten Speichervorrichtung als Reaktion auf einen Spiegel-Leserequest,  
Senden der Spiegeldaten zur zweiten Speichervorrichtung zusammen mit wenigstens einen Schreibrequest bezeichnender Information, dass die zweite Speichervorrichtung arbeiten darf, wobei die Information die von der ersten Speichervorrichtung empfangenen Bezugslabels der Schreibrequests enthält,  
Schreiben der Spiegeldaten auf die zweite Speichervorrichtung, und,  
nach dem Schreiben der Spiegeldaten, Verarbeiten jeglicher von der mit den Spiegeldaten gelieferten Information bezeichneten Schreibrequests an der Speichervorrichtung.

**[0006]** Ausführungsformen der Erfindung können eines oder mehrere der folgenden Merkmale umfassen: Die mit den Daten zum zweiten Speicher gesandte Information kann ein Bezugslabel eines Schreibrequests sein, das vor dem Senden der Daten zum zweiten Speicher am ersten Speicher empfangen wurde. Danach kann der zweite Speicher die Schreibrequests verarbeiten, bis der zweite Speicher einen Schreibrequest antrifft, der das gleiche Bezugslabel hat wie das mit den Daten gesendete.

**[0007]** Das mit den Daten zum zweiten Speicher gesandte Bezugslabel kann einem Schreibrequest zugeordnet werden, der am ersten Speicher vor dem Senden der Daten empfangen und verarbeitet wurde. Insbesondere kann das Bezugslabel einem Schreibrequest zugeordnet werden, der am ersten Speicher vor dem Lesen der Daten vom ersten Speicher auf den Spiegel-Leserequest empfangen und verarbeitet wurde. Diese Lösung erlaubt es dem ersten Speicher, Schreibrequests und andere Spiegel-Leserequests unmittelbar nach dem Lesen von Daten auf den Spiegel-Leserequest auszuführen.

**[0008]** Alternativ kann das mit den Daten gesendete Bezugslabel einem Schreibrequest zugeordnet werden, der am ersten Speicher vor dem Senden der Daten empfangen aber noch nicht verarbeitet wurde. Bei dieser Lösung muss der erste Speicher so konfiguriert werden, dass der erste Speicher keine anderen Spiegel-Leserequests ausführt, bis der erste Speicher den Leserequest verarbeitet, der dem mit den Daten gesendeten Referenzlabel zugeordnet ist. Hätte der erste Speicher einen zusätzlichen Spiegel-Leserequest ausführen sollen, bevor er den Schreibrequest ausführt, der dem mit den Daten gesendeten Bezugslabel zugeordnet ist, besteht die Gefahr, dass der zweite Speicher Daten vom Schreibrequest mit überholten Daten vom zusätzlichen Spiegel-Leserequest überschreiben könnte.

**[0009]** Der Spiegel-Leserequest kann vom zweiten Speicher oder von einem zentralen Prozessor übertragen werden. Der Spiegel-Leserequest kann auch am ersten Speicher erzeugt werden. Eine Reihe von Spiegel-Leserequests kann zusammen den gesamten Inhalt des ersten Speichers anfordern.

**[0010]** Jeder Speicher muss die Schreibrequests in der gleichen Reihenfolge abarbeiten. Schreibrequests können in einer Reihenfolge abgearbeitet werden, in der die Schreibrequests empfangen werden; sie können auch in einer Reihenfolge abgearbeitet werden, die den zugehörigen Bezugslabels entspricht. Die Schreibrequests brauchen nicht in einer den zugeordneten Bezugslabels entsprechenden Reihenfolge empfangen zu werden. Die Bezugslabel können z. B. sequenzielle oder nicht sequenzielle Zahlen oder Text sein.

**[0011]** Nach einem weiteren Aspekt der vorliegenden Erfindung wird bereitgestellt ein Speichersystem für gespiegelte Daten, mit:

- einer ersten Speichervorrichtung;
- einer zweiten Speichervorrichtung;
- einem der ersten Speichervorrichtung zugeordneten ersten Controller;
- einem der zweiten Speichervorrichtung zugeordneten zweiten Controller; wobei:
- der erste Controller so konfiguriert ist, dass er Schreibrequests empfängt, die je durch ein Bezugslabel identifiziert sind,
- die Schreibrequests verarbeitet, indem er Daten auf die erste Speichervorrichtung schreibt, auf einen Spiegel-Leserequest Spiegeldaten von der ersten Speichervorrichtung liest, und
- die Spiegeldaten zusammen mit Information, die wenigstens einen Schreibrequest, den der zweite Controller verarbeiten darf, zum zweiten Controller sendet, wobei die Information die Bezugslabels der von der ersten Speichervorrichtung empfangenen Schreibrequests erhält; und
- wobei der zweite Controller so konfiguriert ist, dass er die Schreibrequests empfängt,

die Spiegeldaten vom ersten Controller empfängt, die Spiegeldaten zur zweiten Speichervorrichtung schreibt und nach dem Schreiben der Spiegeldaten jegliche Schreibrequests verarbeitet, die durch die mit den Spiegeldaten gelieferte Information bezeichnet sind.

**[0012]** Die mit den Daten zum zweiten Speicher gesandte Information kann ein Bezugslabel sein, das einem vom ersten Controller vor dem Senden der Daten empfangenen Schreibrequest zugeordnet ist. Der zweite Controller kann so konfiguriert sein, dass er Schreibrequests verarbeitet, indem er Schreibdaten zum zweiten Speicher sendet, bis ein Schreibrequest festgestellt wird, der dem gleichen Bezugslabel zugeordnet ist wie das mit den Daten vom ersten Controller gesandte.

**[0013]** Weitere Merkmale und Vorteile der Erfindung ergeben sich aus der folgenden detaillierten Beschreibung in Zusammenhang mit den Zeichnungen sowie aus den Ansprüchen.

#### Kurzbeschreibung der Zeichnung

**[0014]** [Fig. 1](#) ist ein Blockschaltbild eines gespiegelten Plattenlaufwerks;

**[0015]** [Fig. 2](#) und [Fig. 3](#) sind Zeitablaufdiagramme;

**[0016]** [Fig. 4](#) ist ein Flussdiagramm einer Prozedur zum Herstellen einer Mirror-Set-Kopie.

#### Beschreibung der bevorzugten Ausführungsformen

**[0017]** [Fig. 1](#) zeigt ein Blockschaltbild eines Mirror-Set **100** mit zwei Plattenlaufwerken: einem ersten Plattenlaufwerk **10** und einem zweiten Plattenlaufwerk **20**. Eines der beiden Plattenlaufwerke sei als Master-Plattenlaufwerk bezeichnet; es dient als primärer Datenspeicher. Das andere Plattenlaufwerk sei als Slave-Plattenlaufwerk bezeichnet; es dient als redundantes Backup. Wenn beide Plattenlaufwerke aktiv sind und die gleichen Daten enthalten, kann der Master/Slave-Status willkürlich einem der beiden Plattenlaufwerke zugeordnet werden. In [Fig. 1](#) ist das Plattenlaufwerk **10** als Master-Plattenlaufwerk und das Laufwerk **20** als Slave-Plattenlaufwerk bezeichnet.

**[0018]** Dem ersten Plattenlaufwerk **10** ist ein erster Controller **30** und dem zweiten Plattenlaufwerk **20** ein zweiter Controller **40** zugeordnet. Die Controller **30**, **40** steuern das Lesen und Schreiben von Daten auf den Laufwerken.

**[0019]** Ein Prozessor **45** sendet gleichzeitig Schreibrequests **50** an beide Controller. Jeder Schreibrequest enthält Daten. Zusätzlich ist jedem Schreibrequest ein Bezugslabel, z. B. eine sequenzi-

elle Bezugsnummer zugeordnet. Die Controller schreiben die Daten von den Schreibrequests auf ihr zugehöriges Laufwerk, so dass unter normalen Bedingungen beide Plattenlaufwerke identische Daten enthalten. Jeder Controller muss die Schreibrequests in der gleichen Reihenfolge bearbeiten. Bearbeiten die Controller die Schreibrequests in der Reihenfolge der Bezugslabel, brauchen die Controller die Schreibrequests nicht in der gleichen Reihenfolge erhalten.

**[0020]** Der Prozessor **45** sendet ferner Leserequests **55** zu den Controllern. Enthalten beide Laufwerke die gleichen Daten, antwortet nur das Master-Plattenlaufwerk auf die Leserequests **55**. Wenn das Master-Plattenlaufwerk ausfällt oder unzugänglich wird, wird das Slave-Plattenlaufwerk wieder als Master-Plattenlaufwerk bezeichnet und es fährt fort, Daten zum Prozessor **45** zu liefern. Wenn also das Plattenlaufwerk **10** ausfällt, dann wird das Plattenlaufwerk **20** das Master-Laufwerk.

**[0021]** Ein Plattenlaufwerk in einem Mirror-Set **100** enthält von denen seines Partners divergente Daten, wenn das Laufwerk für eine gewisse Zeitperiode keine Schreibrequests verarbeiten kann. Wird beispielsweise das Slave-Plattenlaufwerk für eine gewisse Zeit abgeschaltet, unterscheiden sich die Daten des Slave-Plattenlaufwerks von denen des Master-Plattenlaufwerks. Werden die Plattenlaufwerke in einem Mirror-Set divergent, so muss eine Mirror-Set-Kopie implementiert werden, um die Daten vom Plattenlaufwerk mit den „guten Daten“ auf das Plattenlaufwerk mit den divergenten Daten zu kopieren.

**[0022]** Die folgende Diskussion geht davon aus, dass das Slave-Plattenlaufwerk **20** ausgefallen war und wieder hochgefahren wurde, so dass das Slave-Plattenlaufwerk **20** Daten enthält, die von den „guten Daten“ auf dem Master-Plattenlaufwerk **10** divergieren. Der Mirror-Set **100** führt eine Mirror-Set-Kopie der Daten vom Master-Plattenlaufwerk **10** auf das Slave-Plattenlaufwerk **20** aus, während das Master-Plattenlaufwerk **10** weiter Schreibrequests **50** und Leserequests **55** verarbeitet.

**[0023]** Der Slave-Controller **40** initiiert eine Mirror-Set-Kopie durch Übertragen eines Spiegel-Leserequest **60** zum Master-Controller **30**. Der Master-Controller **30** setzt bei der gezeigten Ausführungsform den Spiegel-Leserequest **60** in eine Queue **70**, die auch Schreibrequests **50** und Leserequests **55** enthalten kann. Der Master-Controller **30** verarbeitet dann die Eingaben in der Queue **70** in der Reihenfolge, in der die Eingaben in die Queue gesetzt wurden. Nach Erreichen des Spiegel-Leserequests **60** in der Queue **70** liest der Master-Controller **30** die im Request identifizierten Daten und überträgt die Daten zum Slave-Controller **40** als Spiegel-Schreibrequest **75**. Der Master-Controller **30** ent-

hält mit dem Spiegel-Schreibrequest **75** die Bezugszahl (LW), die dem letzten Schreibrequest **50** zugeordnet war und die der Master-Controller **30** verarbeitet hat, bevor der Spiegel-Leserequest **60** verarbeitet wurde.

**[0024]** Nachdem er den Spiegel-Schreibrequest **75** empfangen hat, setzt der Slave-Controller **40** den Spiegel-Schreibrequest **75** in eine Queue **80**, die sich von einer Queue **85** unterscheidet, die Schreibrequests **50** und Leserequests **55** enthalten kann. Der Slave-Controller **45** schreibt dann die Daten vom Spiegel-Schreibrequest **75** auf das Plattenlaufwerk **20**.

**[0025]** Weil der Master-Controller **30** während der Ausführung der Mirror-Set-Kopie Schreibrequests **50** bearbeitet, können im Spiegel-Schreibrequest **75** enthaltene Daten durch einen nachfolgenden Schreibrequest **50** geändert werden. Dies kann zur Aufzeichnung veralteter Daten auf dem Slave-Plattenlaufwerk **20** führen. Um diese potentielle Schwierigkeit zu vermeiden, verarbeitet der Slave-Controller **40** auch Schreibrequests **50**, während er die Mirror-Set-Kopie ausführt. Dies geschieht in einer Weise, die gewährleistet, dass Daten von einem Schreibrequest **50** nicht durch überholte Daten von einem Spiegel-Schreibrequest **75** überschrieben werden.

**[0026]** [Fig. 2](#) zeigt ein Zeitablaufdiagramm für den Betrieb des Master-Controllers **30** und des Slave-Controllers **40** während einer Mirror-Set-Kopie. Die Pfeile an der Oberseite des Diagramms zeigen die Zeiten an, zu denen Schreibrequests **50** an den Controllern empfangen werden, und die Zeiten, zu denen am Master-Controller **30** Spiegel-Leserequests empfangen werden. Die Kästchen zeigen die Perioden, während derer Schreibrequests, Spiegel-Leserequests und Spiegel-Schreibrequests von den Controllern verarbeitet werden. Zur Erleichterung der Darstellung und Diskussion sind die den Schreibrequests zugeordneten Bezugslabel als aufeinander folgende Zahlen dargestellt, die in einer Reihenfolge empfangen werden. Bei tatsächlichen Anwendungen brauchen die Bezugslabel nicht aufeinander folgend oder sequenziell zu sein. Auch brauchen die Bezugslabel keine Zahlen zu sein.

**[0027]** Wie in [Fig. 2](#) gezeigt, empfangen die Controller Schreibrequests **220** (W1) und **230** (W2), bevor der Slave-Controller **40** einen Spiegel-Leserequest **240** (MRI) erzeugt. Entsprechend bearbeitet der Master-Controller **30** in der gezeigten Ausführungsform die Schreibrequests W1 und W2, bevor er den Spiegel-Leserequest MR1 verarbeitet.

**[0028]** Wie durch einen Pfeil **250** dargestellt, lässt der Master-Controller **30** einen Spiegel-Schreibrequest **260** (MW1) zum Slave-Controller **40** durch,

nachdem der Spiegel-Leserequest MR1 ausgeführt wurde. Der Spiegel-Schreibrequest MW1 enthält die Spiegel-Lesedaten und die Bezugszahl **270** (LW) des letzten Schreibrequest, den der Master-Controller **30** verarbeitet hat, bevor er den Spiegel-Leserequest MR1 verarbeitete. Da W2 der zuletzt verarbeitete Schreibrequest war, zeigt der Spiegel-Schreibrequest an, dass LW gleich zwei ist.

**[0029]** Nach Empfang des Spiegel-Schreibrequest MW1 verarbeitet ihn der Slave-Controller **40**. Der Slave-Controller **40** erzeugt dann einen zweiten Spiegel-Leserequest **280** (MR2). Danach verarbeitet der Slave-Controller die Schreibrequests W1 und W2. Da der Schreibrequest W2 der Zahl LW im Spiegel-Schreibrequest entspricht, stoppt der Slave-Controller **40** die Verarbeitung und wartet auf die Ergebnisse des Spiegel-Leserequest MR2.

**[0030]** Bei Fortsetzung der Mirror-Set-Kopie verarbeitet der Master-Controller **30** Schreibrequests **300** (W3), **310** (W4) und **320** (W5). Der Master-Controller **30** führt dann eine Spiegel-Lesung MR2 aus. Wie durch den Pfeil **330** dargestellt, lässt der Master-Controller **30** dann einen Spiegel-Schreibrequest **340** (MW2) durch zum Slave-Controller **40**. Der Spiegel-Schreibrequest MW2 enthält die Spiegel-Lesedaten und die Bezugszahl **350** des letzten Schreibrequest, den der Master-Controller **30** vor der Verarbeitung des Spiegel-Leserequest MR2 verarbeitet hat. Da W5 der letzte bearbeitete Schreibrequest war, zeigt der Spiegel-Schreibrequest an, dass die Bezugszahl des letzten Schreibrequest **350** (LW) gleich fünf ist.

**[0031]** Nach Empfang des Spiegel-Schreibrequest MW2 verarbeitet ihn der Slave-Controller **40**. Der Slave-Controller **40** erzeugt dann einen Spiegel-Leserequest **360** (MR3). Darauf verarbeitet der Slave-Controller **40** die Schreibrequests W3, W4 und W5. Da der Schreibrequest W5 der Bezugszahl LW im Spiegel-Schreibrequest entspricht, stoppt der Slave-Controller **40** die Verarbeitung und wartet auf das Ergebnis des Spiegel-Leserequest LR3. Dieser Prozess läuft weiter, bis der Slave-Controller **40** Spiegel-Schreibrequests verarbeitet hat, die dem gesamten Inhalt des Master-Plattenlaufwerks **10** entsprechen (d. h. bis die Plattenlaufwerke keine divergenten Daten mehr enthalten).

**[0032]** **Fig. 3** zeigt ein Zeitablaufdiagramm für eine andere Ausführungsform der Erfindung. Diese Ausführungsform funktioniert wie oben beschrieben, mit der Ausnahme, dass der Master-Controller im Spiegel-Schreibrequest die Bezugszahl des letzten Schreibrequest enthält, den er zu der Zeit empfing (aber nicht notwendigerweise verarbeitet hat), zu der der Master-Controller den Spiegel-Schreibrequest erzeugt, statt den letzten zu der Zeit empfangenen und verarbeiteten Schreibrequest, zu der der Mas-

ter-Controller begann, den Spiegel-Leserequest zu verarbeiten.

**[0033]** Wie in **Fig. 3** gezeigt, empfangen die Controller Schreibrequests W1 und W2, bevor der Slave-Controller **40** einen Spiegel-Leserequest MR1 erzeugt. Daher verarbeitet der Master-Controller **30** die Schreibrequests W1 und W2, bevor er den Spiegel-Leserequest MR1 verarbeitet.

**[0034]** Wie durch den Pfeil **250** dargestellt, lässt der Master-Controller **30** einen Spiegel-Schreibrequest MW1 zum Slave-Controller **40** passieren, nachdem er den Spiegel-Leserequest MR1 ausgeführt hat. Der Spiegel-Leserequest MW1 enthält die Spiegel-Lesedaten und die Bezugszahl LW des letzten Schreibrequest, den der Master-Controller **30** vor Erzeugung des Spiegel-Schreibrequest MW1 empfing. Da W3 der letzte empfangene Schreibrequest war, zeigt der Spiegel-Schreibrequest an, dass LW gleich drei ist.

**[0035]** Nach Empfang des Spiegel-Schreibrequest MW1 verarbeitet ihn der Slave-Controller **40**. Der Slave-Controller **40** erzeugt dann einen zweiten Spiegel-Leserequest MR2. Danach verarbeitet der Slave-Controller **40** die Schreibrequests W1, W2 und W3. Da der Schreibrequest W3 der Zahl LW im Spiegel-Schreibrequest entspricht, stoppt der Slave-Controller **40** die Verarbeitung und wartet auf die Ergebnisse des Spiegel-Leserequest MR2.

**[0036]** Bei der Fortsetzung der Mirror-Set-Kopie verarbeitet der Master-Controller **30** die Schreibrequests W3, W4 und W5. Der Master-Controller **30** führt dann die Spiegel-Lesung MR2 aus. Wie durch den Pfeil **330** dargestellt, lässt der Master-Controller **30** dann einen Spiegel-Schreibrequest MW2 zum Slave-Controller **40** passieren. Der Spiegel-Schreibrequest MW2 enthält die Spiegel-Lesedaten und die Bezugszahl LW des letzten Schreibrequest, den der Master-Controller **30** vor Erzeugung des Spiegel-Schreibrequest MW2 empfing. Da der Schreibrequest **370** (W6) der letzte empfangene Schreibrequest war, zeigt der Spiegel-Schreibrequest an, dass die Bezugszahl des letzten Schreibrequest LW gleich sechs ist.

**[0037]** Nach Empfang des Spiegel-Schreibrequest MW2 verarbeitet ihn der Slave-Controller **40**. Der Slave-Controller **40** erzeugt dann einen Spiegel-Leserequest MR3. Darauf verarbeitet der Slave-Controller **40** die Schreibrequests W4, W5 und W6. Da der Schreibrequest W6 der Bezugszahl LW im Spiegel-Schreibrequest entspricht, stoppt der Slave-Controller **40** die Verarbeitung und wartet auf die Ergebnisse des Spiegel-Leserequest MR3. Dieser Prozess läuft weiter, bis der Slave-Controller **40** Spiegel-Schreibrequests verarbeitet hat, die dem gesamten Inhalt des Master-Plattenlaufwerks **10** entsprechen (d. h. so lange, bis die Plattenlaufwerke keine

divergenten Daten mehr enthalten).

[0038] Die Ausführungsform der [Fig. 3](#) kann effizienter sein als die der [Fig. 2](#). Insbesondere ist es bei der Ausführungsform der [Fig. 3](#) weniger wahrscheinlich, dass Perioden entstehen, in denen das Slave-Plattenlaufwerk **20** inaktiv ist. Eine solche Periode ist in [Fig. 2](#) durch den Abstand **400** angedeutet. Bei der Ausführungsform der [Fig. 3](#) ist es aber erforderlich, dass der Master-Controller keine zusätzlichen Spiegel-Leserequests verarbeitet, bis der Master-Controller den Schreibrequest verarbeitet hat, der in einem vorherigen Spiegel-Schreibrequest identifiziert wurde. Im Gegensatz dazu erlaubt es die Ausführungsform der [Fig. 2](#), dass der Master-Controller jegliche Aktion ausführt, einschließlich der Verarbeitung eines zusätzlichen Spiegel-Leserequest, unmittelbar nach Erzeugung eines Spiegel-Schreibrequest.

[0039] Zu [Fig. 4](#): Der Slave-Controller **40** kann die Mirror-Set-Kopie nach einer Prozedur **500** ausführen. Um eine Mirror-Set-Kopie zu starten, löscht der Slave-Controller **40** zunächst seine Lese/Schreibrequest-Queue **85** (Schritt **510**). Der Controller kann dies ausführen, weil jegliche Schreibrequests in der Queue (d. h. jegliche vor der Initialisierung der Spiegelset-Kopie empfangenen Schreibrequests) vom Master-Controller **30** verarbeitet wurden und in den Daten in den Spiegel-Schreibrequests wiedergegeben sind. Darauf sendet der Slave-Controller **40** einen Spiegel-Leserequest **60** zum Master-Controller **30** (Schritt **520**). Danach wartet der Slave-Controller **40** auf den Empfang eines Spiegel-Schreibrequest **75** (Schritt **530**).

[0040] Sobald der Slave-Controller einen Spiegel-Schreibrequest empfängt, bestimmt er, ob die Daten der letzte Datenblock sind, die vom Master-Plattenlaufwerk **10** kopiert werden müssen (Schritt **540**). Zutreffendenfalls schreibt der Slave-Controller die Spiegel-Schreibdaten auf das Slave-Plattenlaufwerk **20** (Schritt **550**) und die Mirror-Set-Kopie ist vollständig (Schritt **560**).

[0041] Sind die vom Master-Plattenlaufwerk **10** empfangenen Daten nicht der letzte vom Master-Plattenlaufwerk **10** zu kopierende Datenblock (Schritt **540**), sendet der Slave-Controller **40** einen weiteren Spiegel-Leserequest **60** (Schritt **570**) und schreibt die gerade empfangenen Daten auf das Slave-Plattenlaufwerk **20** (Schritt **580**).

[0042] Als nächstes bestimmt der Slave-Controller **40**, ob das Bezugslabel des nächsten Schreibrequest **50** in der Queue **85** dem vom Master-Controller im Spiegel-Schreibrequest identifizierten Schreibrequest (LW) entspricht (Schritt **590**). Falls nicht, verarbeitet der Slave-Controller **40** einen weiteren Schreibrequest **50** aus der Queue **85** (Schritt **600**)

und prüft den nächsten Schreibrequest **50** in der Queue **85** (Schritt **590**). Der Slave-Controller **40** verarbeitet weiter die Schreibrequests **50** von der Queue **85**, bis das Bezugslabel eines Schreibrequests **50** dem vom Master-Controller im Spiegel-Schreibrequest identifizierten Bezugslabel (LW) entspricht. An diesem Punkt verarbeitet der Slave-Controller den Schreibrequest **50** (Schritt **610**) und wartet auf den nächsten Spiegel-Schreibrequest (Schritt **530**).

[0043] Andere Ausführungsformen liegen innerhalb des Rahmens der folgenden Ansprüche: Statt vom Slave-Controller Spiegel-Leserequests zu empfangen, kann der Master-Controller die Spiegel-Leserequests erzeugen. Zusätzlich kann der Master-Controller andere Information senden statt eines Bezugslabels mit Daten in Reaktion auf einen Spiegel-Leserequest. Wenn z. B. die Bezugslabel Zeiten entsprechen, zu denen die Schreibrequests erzeugt werden, kann der Master-Controller Information senden, entsprechend der Zeit, zu der der Spiegel-Schreibrequest erzeugt wird, oder zu der Zeit, zu der die Verarbeitung des Spiegel-Leserequest vollendet wird.

### Patentansprüche

1. Verfahren zum Ausführen einer gespiegelten Kopie von einer ersten Speichervorrichtung (**10**) auf eine zweite Speichervorrichtung (**20**) in einem Computersystem (**100**), bei dem jeder Schreibrequest (W) durch ein Bezugslabel identifiziert wird, wobei das Verfahren umfasst:

Empfangen von Schreibrequests (W) an der ersten Speichervorrichtung (**10**),  
Empfangen von Schreibrequests (W) an der zweiten Speichervorrichtung (**20**),  
Verarbeiten von Schreibrequests (W) an der ersten Speichervorrichtung (**10**),  
Lesen von Spiegeldaten von der ersten Speichervorrichtung (**10**) als Reaktion auf einen Spiegel-Leserequest (MR),  
Senden der Spiegeldaten zur zweiten Speichervorrichtung (**20**) zusammen mit wenigstens einen Schreibrequest (W) bezeichnender Information, dass die zweite Speichervorrichtung (**20**) arbeiten darf, wobei die Information die von der ersten Speichervorrichtung (**10**) empfangenen Bezugslabels der Schreibrequests (W) enthält,  
Schreiben der Spiegeldaten auf die zweite Speichervorrichtung (**20**), und,  
nach dem Schreiben der Spiegeldaten, Verarbeiten jeglicher von der mit den Spiegeldaten gelieferten Information bezeichneten Schreibrequests (W) an der zweiten Speichervorrichtung (**20**).

2. Verfahren nach Anspruch 1, wobei der Schritt des Sendens von Information mit den Spiegeldaten zur zweiten Speichervorrichtung (**20**) das Senden der Bezugslabels der von der ersten Speichervorrichtung (**10**) empfangenen Schreibrequests (W) umfasst, be-

vor sie beginnt, den Spiegel-Leserequest (MR) zu verarbeiten.

3. Verfahren nach Anspruch 1, wobei die mit den Daten gesendete Information einem Bezugslabel eines Schreibrequests (W) zugeordnet ist, der vor dem Senden der Daten zur zweiten Speichervorrichtung (20) an der ersten Speichervorrichtung (10) empfangen wurde.

4. Verfahren nach Anspruch 3, wobei der Schritt des Verarbeitens jeglichen Schreibrequests (W) an der zweiten Speichervorrichtung (20), der durch die von der ersten Speichervorrichtung (10) gelieferte Information bezeichnet wurde, das Verarbeiten von Schreibrequests (W) umfasst, bis ein Schreibrequest (W2; W3), der dem mit den Daten gesendeten Bezugslabel zugeordnet ist, angetroffen wird.

5. Verfahren nach Anspruch 4, wobei der Schritt des Verarbeitens jeglichen Schreibrequests (W) an der zweiten Speichervorrichtung (20), der durch die von der ersten Speichervorrichtung (10) gelieferte Information bezeichnet wurde, das Verarbeiten von Schreibrequests (W) umfasst, bis ein Schreibrequest (W2; W3), der dem mit den Daten gesendeten Bezugslabel zugeordnet ist, angetroffen und verarbeitet wird.

6. Verfahren nach Anspruch 3, wobei das mit den Daten gesendete Bezugslabel einem Schreibrequest (W1, W2) zugeordnet ist, der an der ersten Speichervorrichtung (10) vor dem Senden der Daten empfangen und verarbeitet wurde.

7. Verfahren nach Anspruch 6, wobei das mit den Daten gesendete Bezugslabel einem Schreibrequest (W1, W2) zugeordnet ist, der an der ersten Speichervorrichtung (10) vor dem Schritt des Lesens von Daten von der ersten Speichervorrichtung (10) als Reaktion auf den Spiegel-Leserequest (MR) empfangen und verarbeitet wurde.

8. Verfahren nach Anspruch 3, wobei das mit den Daten gesendete Bezugslabel einem Schreibrequest (W3) zugeordnet ist, der vor dem Senden der Daten an der ersten Speichervorrichtung (10) empfangen, jedoch noch nicht verarbeitet wurde.

9. Verfahren nach Anspruch 1, ferner umfassend den Schritt des Erzeugens des Spiegel-Leserequests (MR) an der ersten Speichervorrichtung (10).

10. Verfahren nach Anspruch 1, ferner umfassend den Schritt des Übertragens des Spiegel-Leserequests (MR) von der zweiten Speichervorrichtung (20).

11. Verfahren nach Anspruch 1, wobei der Schritt des Verarbeitens von Schreibrequests (W) an der

ersten Speichervorrichtung (10) das Verarbeiten der Schreibrequests (W) in einer Reihenfolge umfasst, in der die Schreibrequests (W) empfangen werden, und wobei der Schritt des Verarbeitens der Schreibrequests (W) an der zweiten Speichervorrichtung (20) das Verarbeiten der Schreibrequests (W) in einer Reihenfolge umfasst, in der die Schreibrequests (W) empfangen werden.

12. Verfahren nach Anspruch 1, wobei der Schritt des Verarbeitens der Schreibrequests (W) an der ersten Speichervorrichtung (10) das Verarbeiten der Schreibrequests (W) in einer Reihenfolge umfasst, die den zugehörigen Bezugslabels entspricht, und wobei der Schritt des Verarbeitens der Schreibrequests (W) an der zweiten Speichervorrichtung (20) das Verarbeiten der Schreibrequests (W) in einer Reihenfolge umfasst, die den zugehörigen Bezugslabels entspricht.

13. Verfahren nach Anspruch 12, wobei die Bezugslabels Zahlen umfassen, wobei der Schritt des Verarbeitens der Schreibrequests (W) an der ersten Speichervorrichtung (10) das Verarbeiten der Schreibrequests (W) in der Reihenfolge der numerischen Werte der zugehörigen Bezugslabels umfasst, und wobei der Schritt des Verarbeitens der Schreibrequests (W) an der zweiten Speichervorrichtung (20) das Verarbeiten der Schreibrequests in der Reihenfolge der numerischen Werte der zugehörigen Bezugslabels umfasst.

14. Speichersystem für gespiegelte Daten, mit:  
 einer ersten Speichervorrichtung (10);  
 einer zweiten Speichervorrichtung (20),  
 einem der ersten Speichervorrichtung (10) zugeordneten ersten Controller (30);  
 einem der zweiten Speichervorrichtung (20) zugeordneten zweiten Controller (40); wobei:  
 der erste Controller (30) so konfiguriert ist, dass er Schreibrequests (W) empfängt, die je durch ein Bezugslabel identifiziert sind,  
 die Schreibrequests (W) verarbeitet, indem er Daten auf die erste Speichervorrichtung (10) schreibt,  
 auf einen Spiegel-Leserequest (MR) Spiegeldaten von der ersten Speichervorrichtung (10) liest, und  
 die Spiegeldaten zusammen mit Information, die wenigstens einen Schreibrequest (W), den der zweite Controller (40) verarbeiten darf, zum zweiten Controller (40) sendet,  
 wobei die Information die Bezugslabels der von der ersten Speichervorrichtung (10) empfangenen Schreibrequests (W) enthält; und  
 wobei der zweite Controller (40) so konfiguriert ist, dass er  
 die Schreibrequests (W) empfängt,  
 die Spiegeldaten vom ersten Controller (30) empfängt,  
 die Spiegeldaten zur zweiten Speichervorrichtung (20) schreibt und

nach dem Schreiben der Spiegeldaten jegliche Schreibrequests (W) verarbeitet, die durch die mit den Spiegeldaten gelieferte Information bezeichnet sind.

15. System nach Anspruch 14, wobei die mit den Spiegeldaten zum zweiten Controller (40) gesendete Information die vom ersten Controller (30) vor seinem Beginn der Verarbeitung des Spiegel-Leserequests (MR) empfangenen Bezugslabels der Schreibrequests (W) enthält.

16. System nach Anspruch 14, wobei der erste Controller (30) so konfiguriert ist, dass die mit den Daten gesendete Information einem Bezugslabel eines Schreibrequests (W) zugeordnet ist, der vor dem Senden der Daten zur zweiten Speichervorrichtung (20) an der ersten Speichervorrichtung (10) empfangen wurde.

17. System nach Anspruch 16, wobei der zweite Controller (40) so konfiguriert ist, dass er Schreibrequests (W) verarbeitet, bis ein Schreibrequest (W2; W3) angetroffen wird, der dem mit den Daten gesendeten Bezugslabel zugeordnet ist.

18. System nach Anspruch 16, wobei der erste Controller (30) so konfiguriert ist, dass das mit den Daten gesendete Bezugslabel einem Schreibrequest (W1, W2) zugeordnet ist, der vor dem Senden der Daten vom ersten Controller (30) empfangen und verarbeitet wurde.

19. System nach Anspruch 16, wobei der erste Controller (30) so konfiguriert ist, dass das mit den Daten gesendete Bezugslabel einem Schreibrequest (W1, W2) zugeordnet ist, der vom ersten Controller (30) empfangen und verarbeitet wurde, bevor der erste Controller (30) auf den Spiegel-Leserequest (MR) Daten von der ersten Speichervorrichtung (10) liest.

20. System nach Anspruch 18, wobei der erste Controller (30) so konfiguriert ist, dass das mit den Daten gesendete Bezugslabel einem Schreibrequest (W3) zugeordnet ist, den der erste Controller (30) vor dem Senden der Daten empfangen, jedoch noch nicht verarbeitet hat.

21. System nach Anspruch 14, wobei der erste Controller (30) so konfiguriert ist, dass er die Spiegel-Leserequests (MR) erzeugt.

22. System nach Anspruch 14, wobei der zweite Controller (40) so konfiguriert ist, dass er den Spiegel-Leserequest (MR) zum ersten Controller (30) überträgt.

23. System nach Anspruch 14, wobei der erste (30) und der zweite Controller (40) so konfiguriert

sind, dass sie die Leserequests (W) in einer Reihenfolge verarbeiten, in der die Schreibrequests (W) empfangen werden.

24. System nach Anspruch 14, wobei der erste (30) und der zweite Controller (40) so konfiguriert sind, dass sie die Schreibrequests (W) in einer den zugehörigen Bezugslabels entsprechenden Reihenfolge verarbeiten.

25. System nach Anspruch 24, wobei die Bezugslabels Zahlen umfassen und der erste (30) und der zweite Controller (40) so konfiguriert sind, dass sie die Schreibrequests (W) in der Reihenfolge der numerischen Werte der zugeordneten Bezugslabels verarbeiten.

Es folgen 4 Blatt Zeichnungen

Anhängende Zeichnungen

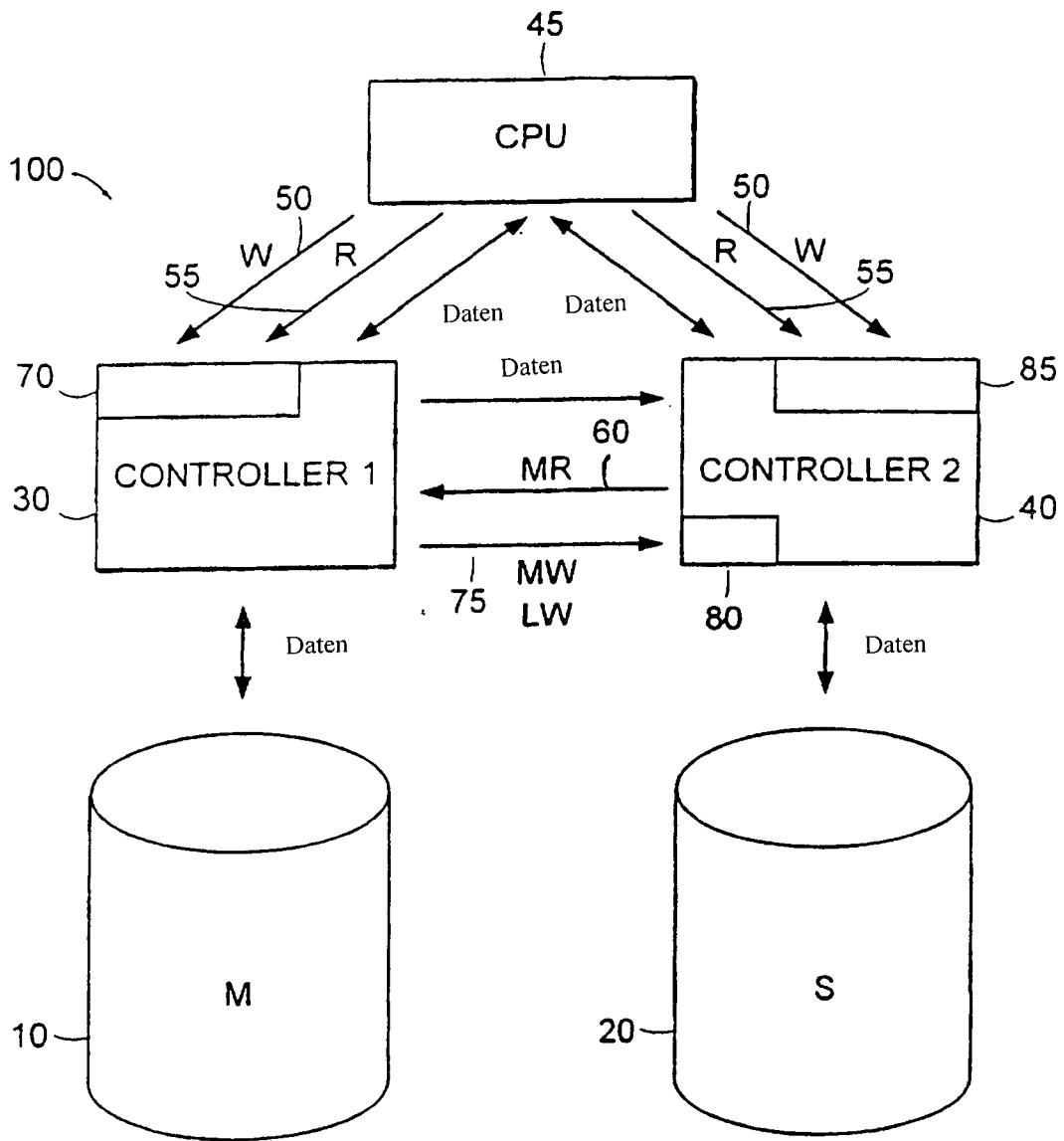


FIG. 1

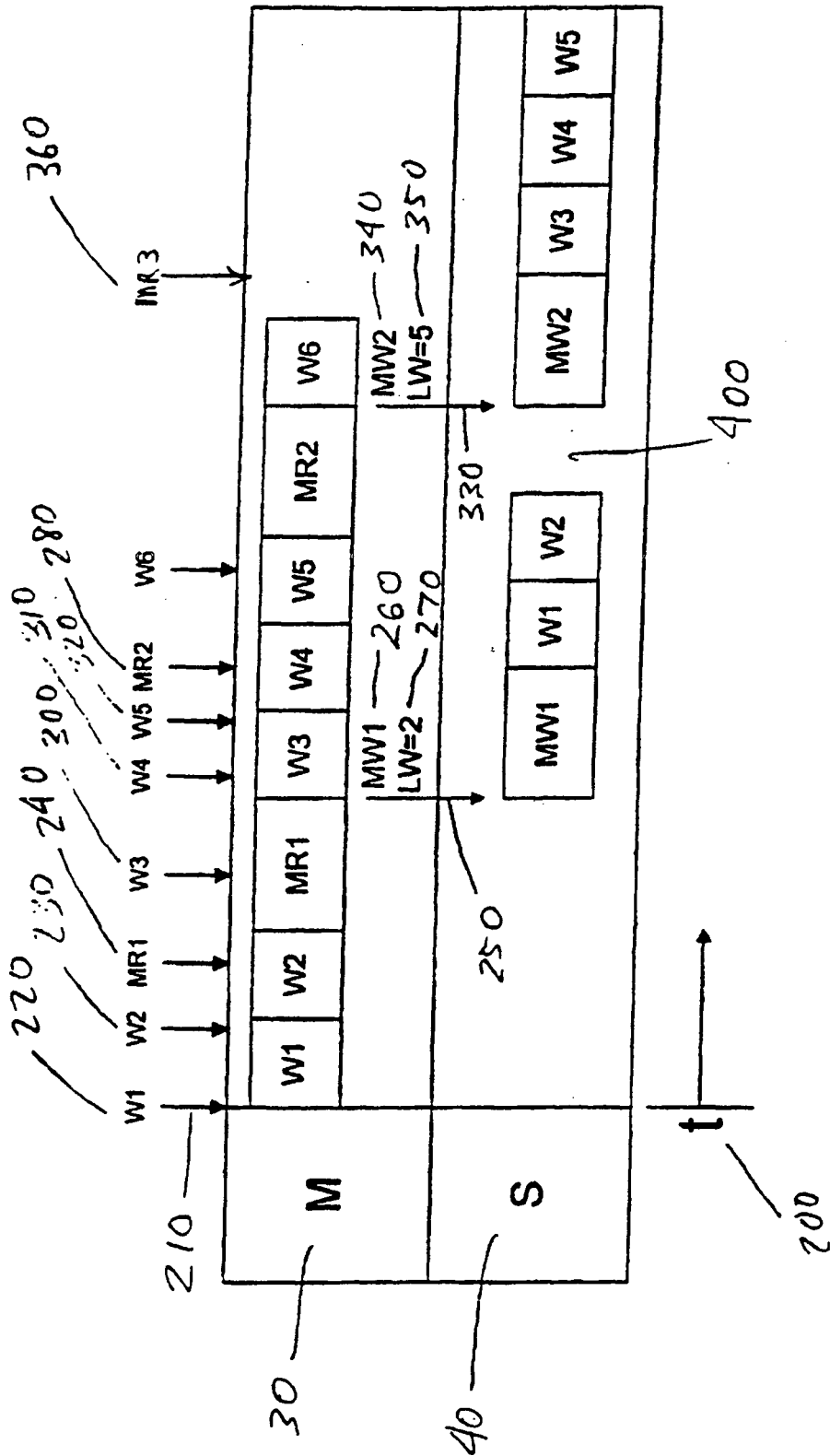


FIG. 2

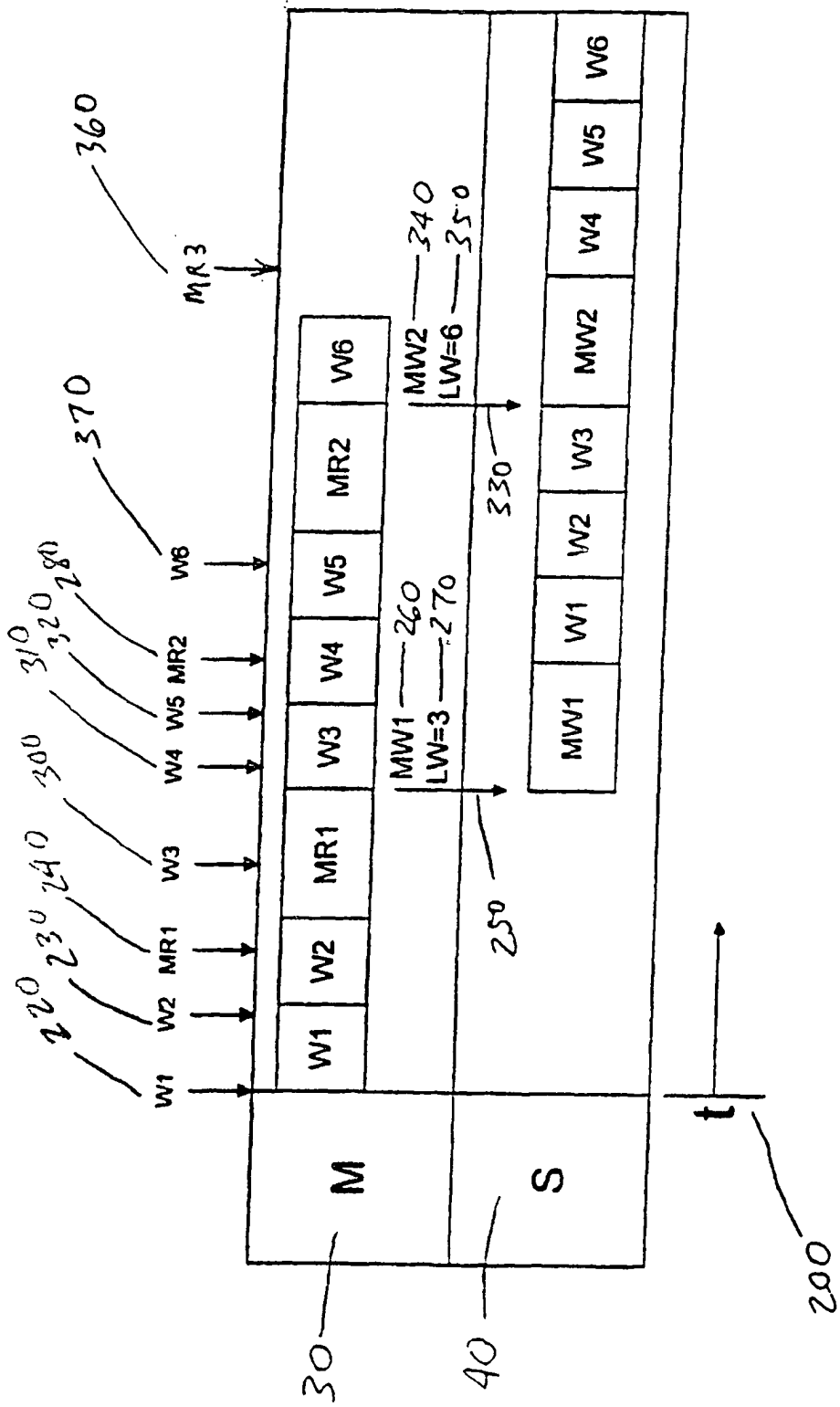


FIG. 3

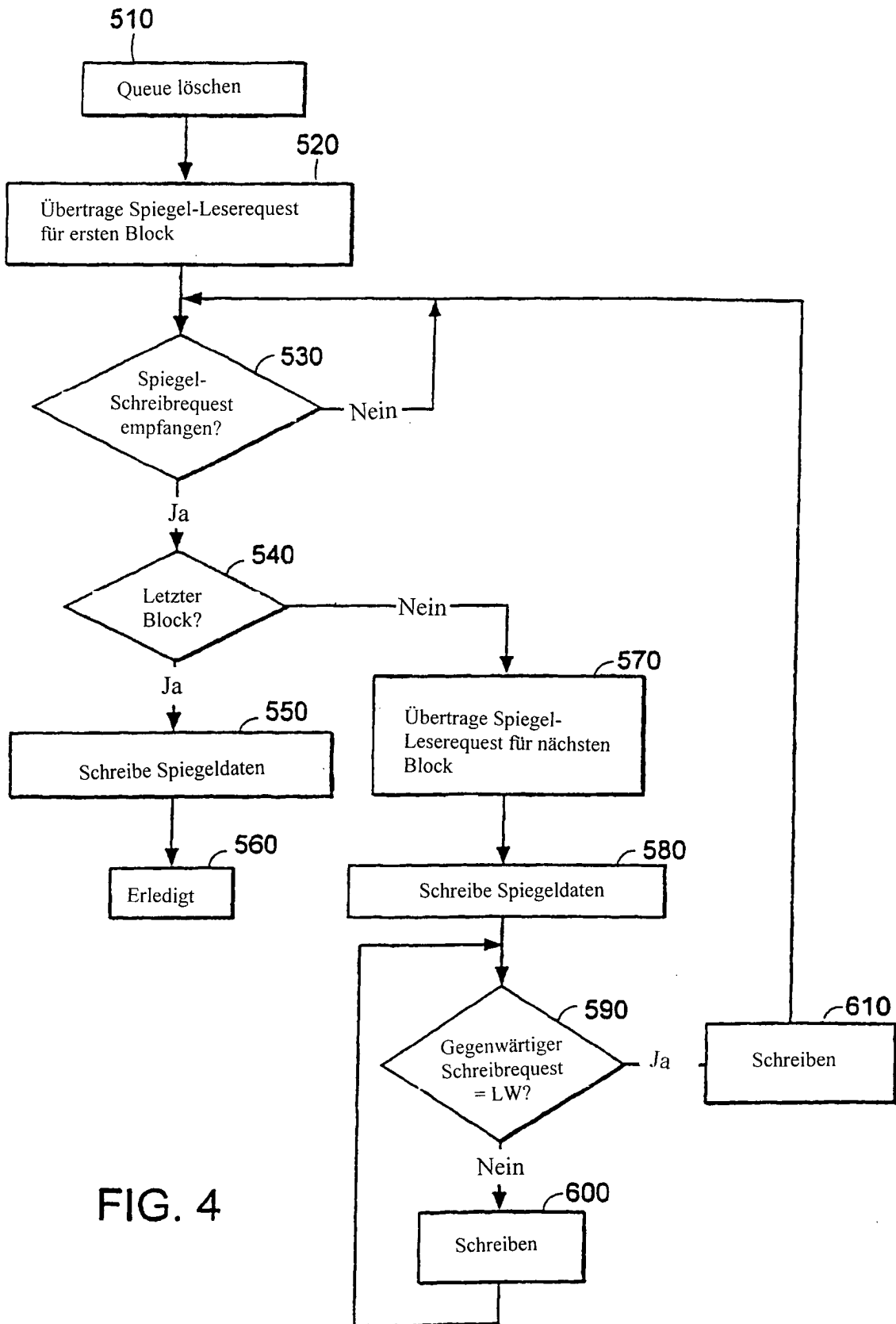


FIG. 4