



(19)
 Bundesrepublik Deutschland
 Deutsches Patent- und Markenamt

(10) **DE 10 2009 051 862 A1** 2010.07.01

(12)

Offenlegungsschrift

(21) Aktenzeichen: **10 2009 051 862.2**

(22) Anmeldetag: **04.11.2009**

(43) Offenlegungstag: **01.07.2010**

(51) Int Cl.⁸: **G06F 12/02 (2006.01)**
G06F 12/06 (2006.01)

(30) Unionspriorität:
12/347,935 31.12.2008 US

(74) Vertreter:
Samson & Partner, Patentanwälte, 80538 München

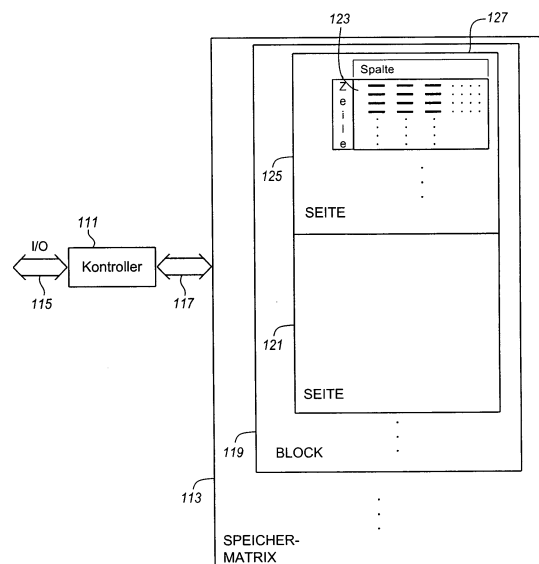
(71) Anmelder:
Chowdhury, Hisham, Fair Oaks, Calif., US;
Edgington, Joseph, Sacramento, Calif., US

(72) Erfinder:
gleich Anmelder

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Speichervorrichtung und Speicherverfahren**

(57) Zusammenfassung: Eine nicht-flüchtige Speichermatrix kann nach einem Spannungsausfall wieder hergestellt werden. Beispielsweise können nach einem Spannungsausfall Seiten einer Speichermatrix abgesucht werden, um eine erste freie Speicherseite zu finden. Die erste freie Speicherseite wird als verfügbar markiert. Und die als verfügbar markierte Speicherseite wird im nächsten Schreibzyklus beschrieben.



Beschreibung

[0001] Nicht-flüchtige Speichereinheiten bieten den Vorteil, Daten ohne eine Spannungsversorgung zum Aufrechterhalten der gespeicherten Daten zu speichern. Jedoch wird eine Spannungsversorgung zum Lesen der gespeicherten Daten und zum anfänglichen Schreiben der Daten benötigt. Bei vielen Typen von nicht-flüchtigem Speicher sind typischerweise die Daten unabhängig von einem Spannungsausfall während Leseoperationen sicher. Jedoch, wenn die Spannungsversorgung während eines Schreibprozesses unterbrochen wird oder ausfällt, dann stoppt nicht nur das Beschreiben des Speichers, sondern es können auch einige Zellen mit instabilen Werten beschrieben worden sein. Nachdem die Spannung wiederhergestellt ist, kann es dann schwierig sein, zu bestimmen, wo das Beschreiben wieder beginnen sollte.

[0002] Weiterhin kann aufgrund einer Prüfung der bereits beschriebenen Werte im Speicher nicht zuverlässig angezeigt werden, an welcher Stelle das Beschreiben wieder beginnen sollte. Beispielsweise wird bei Flash-Speicher der Wert in einer Speicherzelle durch Setzen einer Ladung gespeichert. Um zu bestimmen, ob der gespeicherte Wert eine Null oder eine Eins ist, wird die Spannung an dem Ladungselement gegen eine Schwelle verglichen. Wenn der gespeicherte Wert nahe der Schwelle liegt, weil die Spannung ausgefallen ist, bevor die beabsichtigte Ladung gesetzt wurde, dann kann manchmal eine Null gelesen werden, und bei anderen Gelegenheiten eine Eins. In ähnlicher Weise wird bei einer Phasenwechsel-Speicherzelle der Wert durch Setzen des Widerstandes des Phasenwechsel-Materials gespeichert. Der Widerstand wird mit einer Schwelle verglichen, und wenn der Widerstand in der Nähe der Schwelle liegt, dann kann die Zelle bei einigen Gelegenheiten als Null und bei anderen Gelegenheiten als Eins gelesen werden. Derart instabile oder inkonsistente Speicherwerte können in jedem Typ von nicht-flüchtigem Speicher auftreten, in welchem eine Speicherzelle einen Bereich von Werten annehmen kann.

[0003] Es gibt Verfahren zum Schutz eines Systems vor dem Zusammenbruch aufgrund von in einem Flash-Speicher nach einem Spannungsausfall verbleibenden instabilen Daten. Die Schutzverfahren beinhalten typischerweise das Absuchen der gesamten Speichereinheit oder das Neu-Beschreiben der ganzen Datei von Anfang an. Jedoch stellt bei instabilen Speicherwerten das Absuchen des Speichers keine gute Lösung dar. Auf der anderen Seite kann das Neu-Beschreiben der Daten zeitaufwändig sein.

[0004] Der Gegenstand der Erfindung wird angegeben und klar beansprucht im an die Beschreibung anschließenden Teil, wohingegen die Erfindung, sowohl

hinsichtlich der Ausgestaltung, als auch hinsichtlich des Verfahrens der Ausführung zusammen mit deren Zweckgegenständen, kennzeichnenden Merkmalen und Vorteilen sich am besten aufgrund der folgenden detaillierten Beschreibung zusammen mit den zugehörigen Figuren verstehen lässt, wobei:

[0005] [Fig. 1](#) ein Blockdiagramm einer erfindungsgemäßen Flash-Speichermatrix mit Speicherzellen zeigt, die in Blöcken und in Seiten organisiert sind;

[0006] [Fig. 2](#) ein Prozess-Flussdiagramm des Absuchens von Seiten einer Speichermatrix, um eine erste freie Seite zu finden, gemäß einer Ausführungsform der Erfindung zeigt;

[0007] [Fig. 3](#) ein Prozess-Flussdiagramm des Absuchens von Seiten einer Speichermatrix, um eine erste freie Seite zu finden, gemäß einer anderen Ausführungsform der Erfindung zeigt;

[0008] [Fig. 4](#) ein Prozess-Flussdiagramm des Absuchens von Seiten durch Blöcke einer Speichermatrix, um einen ersten freien Block zu finden, gemäß einer Ausführungsform der Erfindung zeigt;

[0009] [Fig. 5](#) ein Prozess-Flussdiagramm des Absuchens von Seiten durch Blöcke einer Speichermatrix, um einen ersten freien Block zu finden, gemäß einer anderen Ausführungsform der Erfindung zeigt;

[0010] [Fig. 6](#) ein Blockdiagramm eines mobilen Geräts zeigt, in dem die Prozesse der [Fig. 2](#) bis [Fig. 5](#) gemäß einer Ausführungsform implementiert werden können; und

[0011] [Fig. 7](#) ein Blockdiagramm eines Spannungsausfall-Wiederherstellungssystems gemäß einer Ausführungsform zeigt.

[0012] Zur Vereinfachung und Klarheit ist die Darstellung in den Figuren nicht notwendigerweise maßstabsgetreu. Beispielsweise können aus Gründen der klaren Darstellung die Abmessungen einiger Elemente übertrieben im Vergleich zu anderen Elementen dargestellt sein. Weiterhin wurden, wo es angemessen erscheint, in unterschiedlichen Figuren die Bezugszeichen wiederholt, um anzuzeigen, dass es sich um korrespondierende oder analoge Elemente handelt.

Detaillierte Beschreibung

[0013] Mit den Ausführungsformen der vorliegenden Erfindung kann bei einem Spannungsverlust während einer Programmieroperation eines Flash-Speichers oder bei jeder anderen nicht-flüchtigen Speicher-Hardware die Datenintegrität komplett wieder hergestellt werden. Dementsprechend können instabile, beschädigte oder teilweise program-

mierte Datenfragmente sicher von guten Datenfragmenten unterschieden werden. Dies erlaubt eine optimale Durchführung der Wiederherstellung nach einem Spannungsausfall. Dies ermöglicht weiterhin, dass ein teilweises nochmaliges Beschreiben des Speichers gezielt auf einen bestimmten Teil einer Speichereinheit gerichtet werden kann, was in weniger Schreibzyklen resultiert, um eine volle Wiederherstellung nach einem Spannungsausfall zu erhalten. Durch Reduktion der Anzahl der Beschreibungen des Speichers resultieren bessere Speicher-Lebenszyklen. Dies hilft auch, Hardwarefehler, insbesondere für NAND-Typen von Flash-Speichereinheiten, zu vermeiden. Ausführungsformen der vorliegenden Erfindung können auf einen weiten Bereich von unterschiedlichen nicht-flüchtigen Speichern, einschließlich NOR(nicht-oder)-Flash, NAND(nicht-und)-Flash und Phasenwechselspeicher (Phase Change Memory, PCM) angewendet werden. Die Erfindung kann auch auf einen großen Bereich von unterschiedlichen Speicher-Dateisystemen und Speichersektor-Managementsystemen angewendet werden.

[0014] Gemäß einer Ausführungsform der Erfindung wird ein sequentielles Modell zum Schreiben der Daten auf die Speichereinheit auf einer Per-Block-Basis verwendet. In diesem Modell wird immer der erste verfügbare Platz oder der erste freie Platz benutzt, wenn neue Daten in die Einheit geschrieben werden sollen. Während der Initialisierung des Systems nachfolgend einer unerwarteten Abschaltung wird die im schlimmsten Fall auftretende Beschädigung des unbeschriebenen Bereiches des Speichers angenommen. Da beispielsweise in einem Flash-Speicher die Seiten eines Blocks sequentiell geschrieben werden, ist es im Falle eines Spannungsverlusts oder einer Spannungsunterbrechung nicht notwendig, einen gesamten Speicherblock oder eine gesamte Speichermatrix für ungültig zu erklären. Stattdessen können durch intelligentes Bestimmen, welche Speicherseiten eine Beschädigung erfahren haben, diese bestimmten Speicherseiten im Fokus der Reparatur stehen.

[0015] Datenbeschädigung tritt auf, wenn bei einem Spannungsverlust gerade ein Datenfragment in die verwendete Speichereinheit geschrieben wird. Aufgrund dieser Beschädigung sind die Datenbits in der Einheit in einem unbekanntem Zustand. Der unbekanntem Zustand kann den korrekten Wert darstellen, einen unkorrekten Wert oder einen instabilen bzw. unzuverlässigen ungültigen Wert.

[0016] Für eine komplette Wiederherstellung von einem Spannungsverlust muss entweder der gesamte Speicher neu beschrieben werden oder es muss eine Möglichkeit geben, um den Zustand jeder Speicherzelle der Speichermatrix zu bestimmen. In vielen Typen von nicht-flüchtigen Speichersystemen ein-

schließlich Flash-Speicher und PCM kann und wird mit gewisser Wahrscheinlichkeit, wie oben ausgeführt, eine Störung in der Systemspannung während einer Programmier- oder Löschoption einige Speicherzellen in einem unbekanntem Zustand zurücklassen. Derartige Zustände können Null oder einen Zwischenzustand umfassen.

[0017] Ein einfacher Ansatz zur Wiederherstellung sieht vor, den Flash-Speicher auszulesen, und anzunehmen, dass der gelöschte Zustand vorliegt, falls der gelöschte Zustand zurückgelesen wird. Da jeder Wert einer Zelle in einem Flash-Speicher in einem analogen Zustand gehalten wird, kann er jedoch nur nach erfolgreicher Programmierung oder Löschung als einen Binärwert repräsentierend angesehen werden.

[0018] Gemäß eines anderen Ansatzes wird der Speicherbereich zuerst als ungültig (oder verunreinigt, engl. „dirty“), angesehen, bevor Werte in diesen Bereich geschrieben werden. Einige Flash-Speichereinheiten erlauben jedoch nur eine begrenzte Anzahl von Programmieroperationen auf einer Seite, bevor sie gelöscht werden muss. Das Schreiben einer Verunreinigt-Markierung erhöht jedoch die Anzahl der für jeden Teil von Daten auszuführenden Schreiboperationen. Da sich dadurch die Häufigkeit, mit der die Seite gelöscht und mit denselben Werten wieder beschrieben werden muss erhöht, verlangsamt sich somit die Schreibgeschwindigkeit der Schreibsoftware.

[0019] Gemäß eines anderen Ansatzes wird versucht, sicher zurückzuverfolgen, welche Bereiche einer Speichermatrix zu jedem Zeitpunkt in Benutzung sind. Dieser Ansatz reduziert die Schreibgeschwindigkeit zur Laufzeit. Bei einem weiteren Ansatz werden spezifische Schreibsequenzen benutzt, um die Wahrscheinlichkeit des Auftretens eines Versagens zu minimieren.

[0020] Wiederum bei einem anderen Ansatz wird jede Speicherseite gelesen, um ihre Gültigkeit zu bestätigen. Da eine instabile Zelle zu einem Zeitpunkt korrekt, und zu einem anderen Zeitpunkt unkorrekt ausgelesen werden kann, benötigt dies nicht nur Zeit, sondern bietet auch keine vollständige Sicherheit.

[0021] Im Gegensatz dazu wird in den Ausführungsformen der vorliegenden Erfindung durch Annahme des schlimmsten Falles (worst-case) der Mehraufwand, jedes Datenbit lesen zu müssen, eliminiert, und es kann sichergestellt werden, dass nur stabile Speicherzellen benutzt werden. Mit anderen Worten, in den Ausführungsformen der vorliegenden Erfindung wird angenommen, dass die Benutzung jedes Datenbereiches unsicher ist, solange keine andere Indikation vorhanden ist. Wie unten beschrieben, wird im Falle eines plötzlichen Spannungsverlusts während einer Programmier- oder Löschoption

das System sicher aufrecht erhalten und wiederhergestellt. Gleichzeitig wird die Systemleistung nicht beeinträchtigt.

[0022] [Fig. 1](#) zeigt ein Beispiel eines Flash-Speichersystems. Das System umfasst einen Controller (**111**) und eine Speichermatrix (**113**). Der Controller ist an den Input/Output-Bus (**115**) angebunden, um in der Speichermatrix zu speichernde Daten zu empfangen und aus der Speichermatrix gelesene Daten zu senden. Der Controller ist auch an einen Speicher-Bus angebunden, über welchen die Speichermatrix kontrolliert und verwaltet wird. Hierüber werden auch Daten in die Speichermatrix geschrieben und aus der Speichermatrix gelesen.

[0023] Die Speichermatrix (**113**) ist in einen oder mehrere Blöcke (**119**) unterteilt, wobei jeder Block eine oder mehrere Speicherseiten (**121**) enthält. Die Speicherseiten enthalten einzelne Speicherzellen (**123**), um Daten mittels einer Ladung zu speichern. Der Block enthält außerdem die benötigten Schaltkreise zum Lesen, Schreiben und Zurücksetzen (engl: „flashing“) der Speicherzellen. Diese sind in [Fig. 1](#) als Zeilenkontrollierer (**125**) und Spaltenkontrollierer (**127**) gezeigt. Die Anzahl der Komponenten in einem bestimmten System wird von dessen Konfiguration und seiner beabsichtigten Anwendung abhängen. Eine Seite kann Zellen zum Speichern von 512 bis 2 K 8-Bit-Datenbytes umfassen. Alternativ können auch 16-, 32- und 64-Bit-Datenbytes benutzt werden, und die Größe einer Speicherseite kann kleiner oder größer sein. Jeder Block kann 16, 32, 64 oder 128 Seiten umfassen, wobei eine größere oder kleinere Anzahl auch möglich ist. Jede Speichermatrix kann abhängig von der Anwendung 8 oder 16 Speicherblöcke umfassen. Einige Speichermatrizen können zur Vereinfachung der Konstruktion nur einen einzelnen Datenblock umfassen. Eine größere Speichermatrix kann mehrere Speicherblöcke umfassen. Die hier genannten Zahlen sind als Beispiele zu verstehen, und die Erfindung ist nicht auf eine bestimmte Speicherblockgröße limitiert.

[0024] Im typischen Flash-Betrieb wird ein gesamter Block zur gleichen Zeit zurückgesetzt (engl: „flashed“). Hierdurch werden alle Werte, die in einem Block gespeichert sind, gelöscht und alle Bits auf denselben Wert, typischerweise auf Null, gesetzt. Jedoch können die zurückgesetzten Zellen auch derart interpretiert werden, dass sie stattdessen den Wert Eins aufweisen. Jede Speicherzelle wird dann durch Setzen einer Eins entsprechend programmiert, oder sie verbleibt in Null-Zustand. Der Schreibprozess besteht darin, die zurückgesetzten Zellen zu programmieren. Nachdem jede Speicherzelle programmiert ist, kann sie danach nur wieder beschrieben werden, nachdem sie wieder zurückgesetzt wurde. Da dies den gesamten Block löscht, kann für Speicher, in dem häufig geringe Änderungen in den gespeicher-

ten Daten vorzunehmen sind, kleinere Blöcke, bevorzugt werden. Die Verwendung größerer Blöcke kann für Speicher, der nur selten verändert wird, oder bei dem Kosten wichtiger sind, effizienter sein.

[0025] Typischerweise wird jeweils eine Seite mit Daten beschrieben. In einem typischen Beispiel werden die gesamten Daten in einem einzelnen Schreibzyklus auf die gleiche Speicherseite geschrieben. Nach Abschluss des Schreibzyklus wird eine Bestätigungsmarke für diese Seite platziert und die Seite kann nicht von neuem beschrieben werden. Wenn die Spannungsversorgung während des Schreibzyklus unterbrochen wird, können die Werte im Speicher instabil sein. In einer anderen Speicherkonfiguration können zwei oder mehrere Schreibzyklen auf eine einzelne Speicherseite gerichtet werden, bevor ein Bestätigungsmarke geschrieben wird.

[0026] Für das Flash-Speicherbeispiel der [Fig. 1](#) wird eine Speicherseite als die kleinste Granularität eines Datenelements angenommen, das in die Speichereinheit geschrieben werden kann. Somit wird als größte Datenfragmentgröße der Rest des Blocks angesehen. In diesem Beispiel wird angenommen, dass ein Datenfragment nicht die Blockgrenze überschreitet. Andere Speichertypen können in Speicherseiten und Speicherblöcken organisiert sein, oder alternativ können die Ansätze, die hier beschrieben werden, auf andere Typen von Speicherstrukturen angepasst werden.

[0027] Wenn eine Speichereinheit während eines Spannungsunterbruchs betrieben wird, kann sie gerade beim Beschreiben einer Seite der Flashspeicherzellen-Matrix gewesen sein. Nach dem Spannungsunterbruch kann beim Iterieren durch jeden gültigen Block der Speichereinheit dabei jede Kopfzeile von jeder Speicherseite nach einer leeren Kopfzeile abgesehen werden. Diese Speicherseite kann als die erste unbeschriebene, oder als zurückgesetzte, aber nicht programmierte Speicherseite betrachtet werden.

[0028] Bei manchen Flash-Speichermatrizen hat jede Speicherseite zusätzliche Zellen, die als freier Bereich oder Ersatzbereich oder -raum bezeichnet werden. Diese Zellen sind in die Speichermatrix eingebaut, um im Falle des Versagens anderer verwendet zu werden. Sie können auch für Verwaltungs- oder Managementzwecke verwendet werden. In diesem freien Bereich kann auch die Kopfzeile gespeichert werden.

[0029] Beispielsweise weist jede Seite einen fehlerkorrigierenden Code (Error-Correction-Code, ECC) und eine Kopfzeile auf. Die Kopfzeile wird verwendet, um eine Bestätigungsmarke zu schreiben, wenn die Seite erfolgreich vollständig programmiert oder beschrieben wurde. Die Bestätigungsmarke kann ver-

wendet werden, um anzuzeigen, ob eine Programmieroperation erfolgreich abgeschlossen wurde oder nicht. Die Marke wird nach einer erfolgreich abgeschlossenen Programmieroperation geschrieben. Bei Flash-Speicher wird der Prozess des Beschreibens der Zellen nach einem Zurücksetzen oder Löschen als Programmierung bezeichnet.

[0030] Beispielsweise hat jede Seite eine Kopfzeile, die beschrieben wird, wenn die Seite zurückgesetzt oder programmiert ist. Aufgrund der Kopfzeile kann man dann die Anzahl von gültig programmierten Registern nachvollziehen. Wenn die Kopfzeile als voll gelesen wird, dann hat die Speicherseite programmierte Register, dann ist die Seite nicht frei, und da dies der erste Speicherblock ist, bewegt sich der Prozess weiter zur ersten ungeschriebenen Speicherseite. Alternativ kann die Kopfzeile nur einem ganzen Block oder einem anderen Unterbereich der Speichermatrix zugeordnet sein. Die Kopfzeile kann im freien Bereich liegen, in einer separaten Tabelle oder in einem speziell bereitgestellten Teil der Speicherseite oder des Speicherblocks.

[0031] Nach einem Spannungsverlust wird der Speicher abgesucht, um zu bestimmen, welche Seiten gültig sind und welche Seiten ungültig sind. Es ist beim Neustart oder Anbinden (mounting) des Speicher-Managers, wie beispielsweise der Speichermanagement-Software, am sichersten, anzunehmen, dass jeder der Speicherblöcke durch den Spannungsverlust betroffene Daten enthalten kann. Wenn der Speicherblock abgesucht wird, gibt es vier mögliche Spannungsverlust-Zustände, in denen sich ein Speicherblock befinden kann:

- 1) Das letzte gefundene Datenfragment enthält eine Bestätigungsmarke.
- 2) Das letzte gefundene Datenfragment ist ein leeres Fragment.
- 3) Ein unvollständiges Datenfragment wird gefunden. Dies kann durch eine ungültige Kopfzeile oder eine gültige Kopfzeile ohne Bestätigungsmarke angezeigt werden.
- 4) Eine Speicher-Wiedergewinnung (reclaim) oder eine Flash-Löschung (flash erase) war auf dem Datenblock im Gang und es gibt weder Daten noch eine Kopfzeile.

[0032] In den Fällen 1) und 2) bedeutet die Annahme eines Spannungsverlustes, dass zuletzt die erste während des Absuchens des Blocks als freie (FREE) Speicherseite dieses Blocks (welcher den Spannungsverlust erfuhr) gefundene Seite beschrieben wurde. Beispielsweise muss dann diese Speicherseite (erste freie Speicherseite) physikalisch als verunreinigt markiert werden, und die nachfolgende Seite muss als erste verfügbare freie Speicherseite dieses Blocks verwendet werden. Durch die Verunreinigt-Marke kann angezeigt werden, dass die Daten dieser Seite ungültig sind, und dass sie abhängig

vom Typ der Speichermatrix zurückgesetzt, geklärt, wiedergewonnen, oder mit neuen Daten wiederbeschrieben werden sollen. In ähnlicher Weise, wenn die erste gefundene freie Seite die erste Speicherseite des Blocks ist, dann kann die erste freie Seite als verunreinigt markiert werden. Jedoch gibt es in diesem Fall kein vorangehendes gültiges Datenfragment. Die nächste Speicherseite wird dann die erste freie Speicherseite, die neu beschrieben wird.

[0033] Die Verunreinigt-Marke kann auf verschiedene unterschiedliche Arten geschrieben werden. Beispielsweise sind die Identifikationsnummer der Kopfzeile oder eine Kombination von unterschiedlichen Feldern in der Kopfzeile alle als Nullen programmiert. Jedoch können auch stattdessen andere Markierungen verwendet werden. Bei nachfolgenden Anbindungen dieses Speicherblocks wird das Datenfragment mit den ungültigen Feldern der Kopfzeile als verunreinigter Bereich wahrgenommen.

[0034] Die Verunreinigt-Marke stellt einen klaren Vorteil dar. Wenn der Speicherkontroller mit dem Schreiben neuer Daten beginnt, sucht er nach der nächsten Stelle, d. h. nach der Stelle hinter der in der Sequenz zuletzt beschriebenen Stelle. Diese Stelle befindet sich in der ersten freien Speicherseite. Wenn diese Seite nicht als verunreinigt markiert ist, dann wird bei nachfolgenden Anbindungen die übersprungene Seite als die erste freie Seite betrachtet. Dadurch, dass die Seite als verunreinigt markiert wird, wird sie übersprungen, und die nächste Seite wird verwendet. Dadurch kann diese Seite erst geklärt werden, bevor sie wieder verwendet wird.

[0035] Im Fall 3) ist die Kopfzeile ungültig oder das Datenfragment hat keine Bestätigungsmarke. Bei einer derartigen Seite oder einem derartigen Block trat der Spannungsverlust gerade während des Beschreibens auf. Entsprechend sind die gesamten betroffenen Daten einschließlich aller Information, die auf dem Datenkopf geschrieben wurde, wie beispielsweise die Größe des gültigen oder ungültigen Fragments, unzuverlässig. Beispielsweise wird angenommen, dass der Spannungsverlust gerade zu dem Zeitpunkt aufgetreten ist, während irgendeine Seite dieses Datenfragments beschrieben wurde, und somit diese spezielle Seite nicht sicher bestimmt werden kann. Daher wird angenommen, dass das Datenfragment die maximale Größe, d. h. den Rest des Blocks, aufwies. Somit wird der Rest dieses Datenblocks als verunreinigt betrachtet und kann entsprechend markiert werden. Es mag nicht nötig sein, tatsächlich die betroffenen Seiten als verunreinigt zu markieren, wenn zusätzliche Datenfragmente nicht in diese Speicherseiten geschrieben werden sollen. Spätere Anbindungen, bei denen eine ungültige Kopfzeile oder keine Bestätigungsmarkierung vorliegt, werden den Datenblock jederzeit in der gleichen Weise behandeln.

[0036] Wenn jedoch die Kopfzeile instabil ist, dann kann bei einer nachfolgenden Anbindung die erste Speicherseite dieses Datenfragments als freier Bereich gelesen werden. Dies entspricht dem Spannungsverlust im Fall 1). Die instabile Kopfzeile wird als verunreinigt markiert und nicht weiter herangezogen. In jedem Fall wird der als verunreinigt markierte Bereich wiederhergestellt, wenn der Block wiedergewonnen (reclaimed) wird. Anhand dieser Beispiele ist ersichtlich, dass durch die Behebung aller Spannungsverlustfälle während des Anbindens des Speichers jeder Einfluss auf die Schreibleistung nach dem Systemstart unterbunden wird.

[0037] Im Fall 4) wird der Speicher-Wiedergewinnungs-Prozess einfach neu gestartet, falls für den Speicherblock eine Speicher-Wiedergewinnung im Gange war. Die anderen Maßnahmen sind nicht notwendig. Andere Ansätze können verwendet werden, um diesen Fall spezifischer zu adressieren.

[0038] Die oben beschriebenen Ideen können nun unter Benutzung von Flussdiagrammen veranschaulicht werden. [Fig. 2](#) ist ein Beispiel eines Typs des grundlegenden Absuchens einer Speicherseite. Der Prozess gemäß [Fig. 2](#) wird jede Seite eines Speicherblocks mit dem Ziel durchsuchen, die nächste freie Seite zu finden. Die freie Seite ist die nächste Seite, die zum Programmieren verwendet wird. Mit anderen Worten, die freie Seite ist die nächste Seite, die in diesem Block beschrieben wird.

[0039] Bei Box (211) wird der Seitenabsuch-Prozess gestartet. Dies kann zu jeder Zeit geschehen. Allerdings geschieht dies in dem oben beschriebenen Beispiel nach einem Spannungsverlust während einer Schreiboperation in den Speicher. Dies kann außerdem während jedes Systemstarts oder während jedes Systemstarts nach einem Spannungsverlust-Ereignis geschehen. Durch die Initialisierung wird derjenige Block ausgewählt, an welchem zu starten ist. Falls es mehr als einen Block gibt, so werden die Blöcke typischerweise mit numerischen Indikatoren geordnet, beispielsweise von 0 bis 4. Die Nummerierung entspricht der Reihenfolge, in der die Blöcke gefüllt werden. Der erste Block ist typischerweise der einzige Block der Speichermatrix oder der erste Block, und wird mit 0 markiert. Jedoch kann der Prozess bei jedem Datenblock starten.

[0040] Bei Box (213) wird die erste Speicherseite der Sequenz ausgewählt. Wie oben erwähnt, hat der Speicherkontroller typischerweise eine Sequenz, die er zum Beschreiben der Seiten der Speichermatrix benutzt. Die jeweilige Sequenz hängt von dem jeweiligen Design der Speichermatrix ab. Die Nummer 0 entspricht typischerweise einer logischen Reihenfolge anstelle einer physikalischen oder positionsmäßigen Reihenfolge. Bei Box (215) startet das Absuchen Seite für Seite.

[0041] Falls die Seite keine Daten enthält, dann wird bei Box (217) die Seite als verunreinigt markiert. Jedoch wird bei dem Ansatz gemäß [Fig. 2](#) nur die erste leere Speicherseite, die in der verwalteten Speichereinheit gefunden wurde, als verunreinigt markiert, und die nachfolgende Seite wird die erste freie Seite. Bei Box (219) wird festgestellt, ob diese Seite die letzte Seite ist. Falls nein, dann wird bei Box (221) die nächste Seite als freie Seite markiert und der Prozess bei Box (223) beendet. Falls andererseits bei Box (225) diese Seite ohne Daten die letzte Seite ist, dann ist keine freie Seite vorhanden, und Prozess wird bei Box (223) beendet.

[0042] Wenn eine Seite beim Absuchen Daten enthält, dann prüft der Prozess als nächstes bei Box (227) die Kopfzeile auf eine Bestätigungsmarke. In diesem Beispiel wird, nachdem eine Seite erfolgreich und vollständig geschrieben wurde, dann eine Bestätigungsmarke in die Kopfzeile geschrieben. Alternativ gibt es nur eine Bestätigungsmarke für einen vollständigen Block. Dies spart Zeit beim Schreiben jeder Speicherseite und, da ein Spannungsverlust ein seltenes Ereignis darstellt, gibt es nur selten einen Einfluss auf die Wiederherstellung des Systems durch einen Spannungsverlust.

[0043] Gibt es keine Bestätigungsmarkierung für die Speicherseite, dann wird die Speicherseite bei Box (217) wiederum als verunreinigt markiert. Wie schon vorher bei einer verunreinigten Seite wird, falls es bei Box (219) eine nächste Speicherseite gibt, die Seite bei box (221) als frei markiert.

[0044] Gibt es eine Bestätigungsmarke für die Speicherseite mit Daten, dann wird angenommen, dass diese Seite eine gute Seite ist, und das Absuchen wird bei der nächsten Seite fortgesetzt. In [Fig. 2](#) wird bei Box (229), falls die gegenwärtige Speicherseite nicht die letzte Seite ist, bei Box (231) das Absuchen bei der nächsten Speicherseite fortgesetzt. Wenn dies die erste Iteration ist, dann ist die gegenwärtige Speicherseite die erste Seite (Speicherseite 0) des ersten Blocks (Block 0). Wie die Blöcke, sind diese auch typischerweise geordnet, und alle Schreiboperationen werden auf einer Seite ausgeführt, bis diese Seite voll ist. Die Schreiboperation wird dann bei der nächsten Seite in der Sequenz fortgesetzt (Seite 1, Block 0). Entsprechend werden die Seiten während dieses Spannungsausfall-Wiederherstellungsprozesses in derselben Reihenfolge analysiert.

[0045] Wenn die gegenwärtige Seite bei Box (229) die letzte Seite ist, dann gibt es bei Box (225) keine freie Seite und der Prozess endet bei Box (223).

[0046] Wie durch die Rücksprungschleifen in [Fig. 2](#) gezeigt, kann der Prozess gemäß [Fig. 2](#) für jede Speicherseite einzeln durch jeden gültigen Speicherblock der Speichermatrix iterieren. Der Prozess wird

jede Speicherseite absuchen, bis er die erste unbeschriebene Speicherseite findet. Dies ist in dem Ausführungsbeispiel durch die leere Kopfzeile angedeutet.

[0047] [Fig. 3](#) zeigt eine Variation des oben beschriebenen Prozessablaufs. Der Ablauf in [Fig. 3](#) ist gleich dem Ablauf in [Fig. 2](#), bis auf eine Änderung in der Verbindung zwischen der Bestätigungsmarken-Bestimmung in Box (327). Wenn bei Box (315) hier eine Speicherseite Daten enthält, aber keine Bestätigungsmarke bei Box (327) enthält, dann markiert der Prozess den Speicherblock als keine freien Seiten enthaltend, anstelle dass er die Seite bei Box (317) als verunreinigt markiert. Daher wird der Prozess entsprechend den Boxen (315), (327), (329) und (331) jede Seite eines Speicherblocks absuchen. Wenn die nächste freie Speicherseite gefunden wurde, ist diese freie Speicherseite die nächste Speicherseite, die für die Programmierung verwendet wird. Dann verzweigt der Prozess immer noch, um die vorhergehende Seite als verunreinigt zu markieren, und damit die freie Seite anzuzeigen.

[0048] In [Fig. 3](#) wurde eine Änderung bei dem Prozess und dem Dateisystem gemacht, wodurch die Bestätigungsmarke geschrieben wird. In diesem Beispiel wird die Bestätigungsmarke nicht für jede Seite behalten, sondern für größere Fragmente von mehreren Seiten. Ein derartiges Fragment kann ein Block sein oder eine andere Gruppierung von Seiten. In dem Beispiel der [Fig. 2](#) kann bei Box (215) die Gruppen-Bestätigungsmarke analysiert werden. Der Prozess in [Fig. 3](#) jedoch ist weiter verfeinert, so dass ein ganzer Bereich von Seiten als ungültig erklärt wird, wenn für ein Datenfragment die Programmierung begonnen wurde, aber keine Bestätigungsmarke gefunden wurde. Dieser Algorithmus folgt ebenfalls dem Prinzip der Annahme einer Beschädigung der Speicherseiten im schlimmsten Fall. Jedoch erhöht sich im vorliegenden Beispiel die minimale Granularität eines Programmierbereichs von einer Speicherseite auf den Rest eines Datenblocks oder eine Gruppe von Speicherseiten.

[0049] Alternativ kann der Prozess versuchen, zwei nacheinander folgende Seiten zu finden, die keine Daten enthalten und dann die erste Speicherseite als ungültig erklären. Mit dieser Änderung kann immer noch die letzte Seite, die beschrieben sein konnte, gefunden und als ungültig erklärt werden. Wenn in all diesen Variationen eine leere Seite bestätigten Daten folgt, oder sich eine leere Seite am Anfang eines Datenblocks befindet, dann ist im schlimmsten Fall nur eine Seite beschädigt worden. Aus diesem Grund wird gemäß Boxen (217), (219) und (221) die Seite als verunreinigt gekennzeichnet und die nächste Seite als frei markiert.

[0050] Wie oben beschrieben, kann beim Lesen von

einer instabilen Zelle an derselben Stelle eine 0 (ein programmiertes Bit) am Ausgang resultieren oder manchmal eine 1 (entsprechend eines kürzlich gelöschten freien Bits oder einer Leerstelle). Beim Betrieb von Flash-Speichern ist eine Zelle nur dann zuverlässig, wenn sie erfolgreich zurückgesetzt oder erfolgreich programmiert wurde. Wenn einer dieser Prozesse unterbrochen wurde, dann ist die Speicherzelle instabil oder beschädigt. Die einzige Möglichkeit, zu einem stabilen Zustand zurückzukehren ist es, die gesamte Seite, welche die Zelle enthält, zurückzusetzen. Entsprechend wird die Seite mit der leeren Kopfzeile zurückgesetzt und neu beschrieben. Die folgende Seite wird ebenfalls zurückgesetzt, damit sie die nächste freie Seite wird.

[0051] Falls die vorhergehende Seite keine gültigen und bestätigten Daten enthält (Nicht-Spannungsausfall-Fragment), dann wird fortschreitend von dem vorherigen gültigen Datenfragment in diesem Block (oder vom Anfang des Blocks, wenn kein gültiges Datenfragment gefunden wurde) an angenommen, dass ein Datenfragment mit maximaler Größe als ungültig gilt.

[0052] Der durch das Invalidieren von wenigstens einer freien Speicherseite für den ersten freien Block während der Initialisierung verlorengangene Speicherplatz wird als nicht signifikant erachtet, da diese Bedingung nur nach einem unerwarteten Abschalten existiert. Weiterhin wird dieser Speicherplatz zurückgewonnen, wenn die ungültigen Seiten durch eine Speicher-Wiedergewinnungs-Prozedur oder eine Garbage-Collection-Prozedur auf einen bekannten Status zurückgesetzt (gelöscht) werden. Diese Art von Prozess garantiert, dass keine Beschädigung in einer Speicherseite vor einer Schreiboperation in die Seite existiert.

[0053] [Fig. 4](#) zeigt eine grundlegendere Version der vorliegenden Erfindung. Im Beispiel der [Fig. 4](#) beginnt der Prozess bei Box (411). Bei Box (413) ist der erste Block identifiziert und das Absuchen für jede Seite beginnt. Bei Box (415) wird jede Seite des Blocks abgesucht. Falls bei Box (417) der gegenwärtige Block der letzte Block ist, endet der Prozess bei Box (419). Falls der gegenwärtige Block nicht der letzte Block ist, dann fährt der Prozess bei Box (421) für den nächsten Block fort, und wird wiederholt. Der Prozess zum Absuchen einer Speicherseite kann der in den [Fig. 2](#) oder [Fig. 3](#) beschriebene Prozess, oder jeder andere hierin beschriebene Prozess, oder eine Variation eines derartigen Prozesses sein.

[0054] [Fig. 5](#) zeigt eine weitere Variation des Prozesses der [Fig. 4](#). Wie in [Fig. 4](#), beginnt der Prozess bei Box (511) und initialisiert bei Box (513) beim ersten logischen Block. Bei Box (515) werden die Seiten dieses Blocks abgesucht. Der Prozess iteriert wegen Box (517) derart, dass, falls der Block der letzte Block

ist, der Prozess bei Box (519) endet. Wenn bei Box (517) der gegenwärtige Block jedoch nicht der letzte Block ist, fährt der Prozess bei Box (521) für den nächsten Block fort und wiederholt sich bei Box (515).

[0055] Jedoch werden in dem Beispiel der [Fig. 5](#) die Seiten jedes Blocks abgesucht, bis bei Box (523) eine einzelne freie Seite gefunden wird. Mit anderen Worten, bei Box (515) werden die Seiten eines Speicherblocks abgesucht. Wenn keine freie Seite gefunden wird, dann lenkt die Entscheidungsbox (523) den Prozess zu Box (517), und wie bei [Fig. 4](#) wird der Prozess beim nächsten Block zyklisch fortgesetzt oder er endet, falls der gegenwärtige Block der letzte Block ist. Wenn jedoch aufgrund des Absuchens der Seite eine freie Speicherseite gefunden wurde, dann wird aufgrund der Entscheidungsbox (523) der Prozess bei Box (519) beendet. Dadurch kann die Zeit, um das Speichergerät einzubinden, und die beschädigte Seite zu finden, stark verringert werden.

[0056] Diese Alternative funktioniert insbesondere dann gut, wenn die Speicherblöcke immer in derselben Sequenz oder Ordnung verwendet werden. Wenn nur eine Speicherseite zu einer Zeit benutzt wird, dann ist die erste freie Seite eines Blocks die erste freie Seite der gesamten Speichermatrix. Diese Speicherseite ist dann die einzig mögliche beschädigte Speicherseite. Wenn andererseits der Speicher-Controller nicht immer derselben Sequenz durch die Seiten folgt, oder wenn mehr als eine Seite zur gleichen Zeit beschrieben werden kann, dann kann der Prozess zur Bestimmung der ersten freien Seite für jeden Block durch alle Blöcke fortgesetzt werden.

[0057] Für Flash-Speicher gilt, wie oben beschrieben, dass eine unbeschriebene Speicherzelle keine bestätigten gültigen Daten oder bestätigte ungültige Daten enthält. Es gibt eine Vielzahl von unterschiedlichen Möglichkeiten, zu bestimmen, was gültige oder nicht gültige Daten sind. Im obigen Beispiel wurde eine Bestätigungsmarke verwendet, um gültige Datenfragmente nach dem Beschreiben zu markieren. Jedoch kann abhängig von der jeweiligen Anwendung jeder andere Ansatz als Ersatz für die Bestätigungsmarke verwendet werden. Eine Bestätigungsmarke, sowie andere Techniken, können auch für andere Typen von Speicherzellen wie TCM, MRAM und FRAM verwendet werden.

[0058] [Fig. 6](#) zeigt ein Beispielsystem (611), auf das die Ausführungsformen der Erfindung angewendet werden können. Bei dem gezeigten Beispiel ist das System ein mobiles handgehaltenes zelluläres Telefon. Jedoch kann mit geringen Modifikationen das System auch eine Vielzahl von unterschiedlichen Geräten repräsentieren. Das System wird durch eine

zentrale Prozessoreinheit (CPU) (613) gesteuert, welche einen Chipsatz beinhaltet oder auch nicht. Die CPU beinhaltet einen Anwendungsprogramm-Teil (615), der Programme unter Benutzung eines Betriebssystems ausführt, und einen Basisband-Teil (617), der Telefonie-Funktionen handhabt. Beide Teile sind an eine Speicherschnittstelle (619) angebunden, die über einen Bus mit dem Systemspeicher kommuniziert.

[0059] In dem veranschaulichten Beispiel hat der Systemspeicher eine flüchtige Sektion (621), welche zum schnellen Zugriff als Random-Access-Memory (RAM) implementiert sein kann, und eine nicht-flüchtige Sektion (623), welche für Daten, die einen Spannungsverlust überdauern müssen als Flash implementiert werden kann. Typischerweise wird RAM-Speicher als Kurzzeitspeicher für Daten und Befehle benutzt, auf die schnell zugegriffen werden muss, während Flash benutzt wird, um Betriebssysteme, Systemparameter und Anwendungsprogramme zu speichern. Alternativ kann der Speicher als alleiniger Speicher vollständig als Flash-Speicher implementiert werden, wobei die Flash-Sektion mit anderen Typen von nicht-flüchtigem Speicher wie PCM (Phasenwechselspeicher), MRM oder FRM, oder Kombinationen von Speichertypen implementiert sein kann. Die oben im Kontext der [Fig. 2](#) und [Fig. 3](#) beschriebenen Operationen werden auf den nicht-flüchtigen Speicher angewendet. Bei einem Spannungsverlust-Ereignis gehen alle Daten, die im flüchtigen Speicher gespeichert sind, verloren.

[0060] Der Basisband-Teil der CPU ist an eine Benutzerschnittstelle angebunden. Im veranschaulichten Beispiel hat die Benutzerschnittstelle eine Tastatur (625) und einen Kopfhörer (627) mit einem Lautsprecher und einem Mikrofon. Eine Vielzahl von anderen Schnittstellen können verwendet werden, wie beispielsweise ein Touch-Screen, Bluetooth-Geräte, Beschleunigungssensoren, Näherungsschalter und andere Schnittstellen, abhängig von der jeweiligen Anwendung. Der Basisband-Teil ist auch an dem HF(Hochfrequenz)-Schaltkreis (629) angebunden, damit das System mit externen Geräten unter Benutzung einer Funkverbindung kommunizieren kann. Die Funkverbindung kann ein zelluläres Telefon, eine Datenschnittstelle, ein drahtloses Netzwerk oder jede andere gewünschte Schnittstelle darstellen.

[0061] Die CPU kann ebenfalls an eine Vielzahl von Peripheriegeräten (631) wie Kameras, Ortsbestimmungssystemen, Bildschirmen, Druckern, Bluetooth-Geräten oder andere Peripheriegeräte zur Unterstützung von zusätzlichen Funktionen des Systems (611) angebunden werden. [Fig. 6](#) zeigt außerdem einen Leistungsregler (633) zur Regulierung des Leistungsverbrauchs der unterschiedlichen Komponenten, welcher eine Spannungsversorgung, wie beispielsweise eine Batterie, beinhalten kann. Diese

Einheit kann software-basiert sein und durch die CPU oder autonom, oder in Kombination von beiden Möglichkeiten gesteuert werden. Beispielsweise setzt im Falle eines Spannungsverlustes der Leistungsregler ein Flag in einem der CPU zugänglichen Register.

[0062] [Fig. 7](#) zeigt ein Blockdiagramm eines Beispiels eines Spannungsausfall-Wiederherstellungssystems, mit welchem man die Prozesse der oben beschriebenen Flussdiagramme implementieren kann. Das System ist an ein Host-System (**709**) angebunden. Das Host-System kann ein Computer sein, ein zelluläres Telefon, ein mobiles Internetgerät, eine Komponente eines Entertainment-Systems, oder eine Vielzahl von anderen Geräten. Das Host-System ist an einen Speicherschnittstellen-Kontroller (**711**) angebunden, der als spezialisierte Hardware vorliegen kann. Der Speicherschnittstellen-Kontroller ist an einem Seiten-Absucher (**713**) und einem Seiten-Markierer (**715**) angebunden, welche wiederum an einen Programmier-Schaltkreis (**717**) und einen Untersuchungs-Schaltkreis (**719**) angebunden sind. Untersuchungs- und Programmier-Schaltkreis sind an die Flash-Speicherzellen (**720-1**) bis (**720-4**) angebunden. Obwohl nur vier Zellen gezeigt sind, befinden sich in einer tatsächlichen Speichermatrix typischerweise hunderte, tausende oder mehr Speicherzellen.

[0063] Der Seiten-Absucher sucht die Seiten von Speicherzellen wie oben beschrieben ab und bestimmt, welche Seiten gültig, frei und ungültig sind. Der Seiten-Markierer markiert die Seiten sowohl basierend auf den Ergebnissen des Seiten-Absuchers als auch während der Benutzung des Speichers. Der Untersuchungs-Schaltkreis liest die Werte der Zellen und der Marken und stellt diese Werte dem Seiten-Absucher zur Verfügung. Der Programmier-Schaltkreis beschreibt die Speicherzellen und die Marken, falls die Marken in der Speichermatrix abgespeichert werden. Wenn die Marken an einem unterschiedlichen Ort abgespeichert werden, dann können sie zu diesem unterschiedlichen Ort abgespeichert werden. Der Speicherschnittstellen-Kontroller überwacht den Betrieb des Speichers und den der anderen veranschaulichten Komponenten. Er überwacht außerdem den Betrieb der Schnittstelle zum Host-System.

[0064] Die Bezeichnung "computerlesbares Medium" bezeichnet ein geeignetes Medium, das einem Prozessor, einem Speichercontroller oder einem anderen geeigneten Gerät Programminstruktionen zur Ausführung zur Verfügung stellen kann. Ein derartiges Medium kann in vielen Formen vorliegen, einschließlich, aber nicht darauf beschränkt, nicht-flüchtige Medien und flüchtige Medien. Nicht-flüchtige Medien können beispielsweise optische oder magnetische Platten, Solid-State-Storage und anderen Speicher, ROM usw. beinhalten. Flüchtige Medien kön-

nen beispielsweise dynamische Speicher wie Systemspeicher, DRAM, SRAM und andere Typen von flüchtigem Speicher einschließen. Verbreitete Formen von computerlesbaren Medien beinhalten beispielsweise magnetische Medien (z. B. Floppy-Disk, Flexible-Disk, Festplatte, magnetischer Bandspeicher und andere magnetische Medien), optische Medien (z. B. Compact-Disc Read-Only-Memory (CD-ROM) und andere optische Medien), physikalische Medien mit Muster (z. B. Lochkarten, Papierband und andere physikalische Medien), Speicherchips oder Kassetten (z. B. RAM, programmierbare Read-Only-Memory (PROM), löschbare programmierbare Read-Only-Memory (EPROM), Flash-Speicher und andere Speicherchips oder Kassetten) und jedes andere Medium, von dem ein Computer lesen kann.

[0065] In der vorangegangenen Beschreibung wurde die Erfindung im Hinblick auf spezifische exemplarische Ausführungsformen beschrieben. Jedoch kann die Erfindung durch den Fachmann auch ohne diese spezifischen Details ausgeführt werden. An anderen Stellen wurden bekannte Verfahren, Prozeduren, Komponenten und Schaltungen nicht im Detail beschrieben, um nicht den Blick auf die Erfindung zu verstellen.

[0066] Einige Teile der detaillierten Beschreibung wurden in Form von Algorithmen und symbolischen Darstellungen der Operationen auf Datenbits oder binären digitalen Signalen innerhalb eines Computerspeichers dargestellt. Diese algorithmischen Beschreibungen und Wiedergaben sind im Allgemeinen die von Fachleuten der Datenverarbeitung benutzten Techniken, um den Inhalt ihrer Arbeit anderen Fachleuten zu vermitteln.

[0067] Ein Algorithmus wird hier und generell als eine selbst-beinhaltenende Sequenz von Tätigkeiten oder Operationen betrachtet, die zu einem gewünschten Resultat führen. Diese umfassen physikalische Manipulationen von physikalischen Größen. Für gewöhnlich, auch wenn nicht notwendig, nehmen diese Größen die Form von elektrischen oder magnetischen Signalen an, die gespeichert, übertragen, kombiniert, verglichen oder in anderer Art manipuliert werden können. Bislang wurde es wegen der weitverbreiteten Benutzung prinzipiell als nützlich angesehen, diese Signale als Bits, Werte, Elemente, Symbole, Buchstaben, Bezeichnungen, Zahlen o. ä. zu bezeichnen. Es sollte jedoch verstanden werden, dass all diese und ähnliche Bezeichnungen mit der jeweiligen physikalischen Größe in Bezug gesetzt werden müssen, welches zumeist die für gewöhnlich benutzten Bezeichnungen sind, die auf diese Größe angewendet werden.

[0068] Solange es, wie in den vorangegangenen Diskussionen, nicht anders angegeben ist, wird an-

genommen, dass in der Beschreibung Ausführungen, die Begriffe wie „prozessieren“, „berechnen“, „bestimmen“ o. ä. enthalten, sich auf die Aktionen und/oder Prozesse eines Computers oder eines Computersystems oder einer ähnlichen elektronischen Computereinheit beziehen, welche Daten, die als physikalische Daten wie beispielsweise elektronische Größen innerhalb der Register oder Speicher eines Computersystems in andere Daten, die in ähnlicher Weise als physikalische Größen in dem Speicher oder in den Registern eines Computersystems oder anderen Informationsspeichern, Übertragungs- oder Anzeigeeinheiten vorliegen, manipulieren und/oder transformieren.

[0069] Ausführungen der vorliegenden Erfindung können Vorrichtungen zum Ausführen der in der Erfindung beschriebenen Operationen beinhalten. Eine Vorrichtung kann speziell für den vorgesehenen Zweck konstruiert sein oder sie kann eine Computer-Einheit umfassen, die selektiv durch ein Programm aktiviert oder konfiguriert wurde, welches in der Einheit gespeichert ist. Ein derartiges Programm kann auf einem Speichermedium gespeichert sein, welches, jedoch nicht darauf beschränkt, jeden Typ von Platten einschließlich Floppy-Disks, optischen Platten, Compact-Disk Read-Only-Memories (CD-ROM), Magneto-optischen Platten, Read-Only-Memory (ROM), Random-Access-Memory (RAM), elektrisch programmierbaren Read-Only-Memory (EPROM) elektrisch löschbaren und programmierbaren Read-Only-Memory (EPROM), magnetischen oder optischen Karten, oder jede andere Art von Medien, die zum Speichern von elektronischen Anweisungen geeignet sind und an einem Systembus einer Computereinheit angebunden werden können, umfasst.

[0070] Die hier dargestellten Prozesse und Zeichnungen beziehen sich nicht inhärent auf eine bestimmte Computereinheit oder andere Vorrichtung. Unterschiedliche, zu generellem Zweck verwendbare Systeme können mit Programmen im Zusammenhang mit der hier dargestellten Lehre verwendet werden, oder es kann sich als zweckmäßig herausstellen, eine stärker spezialisierte Vorrichtung zur Ausführung des gewünschten Verfahrens zu konstruieren. Die gewünschte Struktur wird aus obiger Beschreibung für eine Vielzahl von derartigen Systemen deutlich. Weiterhin werden Ausführungsformen der vorliegenden Erfindung nicht mit Bezug auf eine bestimmte Programmiersprache beschrieben. Es ist klar, dass eine Vielzahl von Programmiersprachen verwendet werden können, um die Lehre der hier beschriebenen Erfindung zu implementieren. Weiterhin sollte klar sein, dass die in der Beschreibung beschriebenen Operationen, Fähigkeiten und Merkmale mit jeder Kombination von Hardware (diskrete oder integrierte Schaltungen) und Software implementiert werden können.

[0071] Die Bezeichnungen „gekoppelt“ und „verbunden“ nebst deren Derivativen sollen nicht als synonym verstanden werden. Hingegen soll in bestimmten Ausführungen „verbunden“ dann verwendet werden, um anzuzeigen, dass zwei oder mehr Elemente in direktem physikalischem oder elektrischem Kontakt miteinander stehen. „Gekoppelt“ soll verwendet werden, um anzuzeigen, dass zwei oder mehr Elemente entweder in direktem oder indirektem (mit weiter eingreifenden Zwischenelementen) physikalischem oder elektrischem Kontakt miteinander stehen, und/oder dass die zwei oder mehr Elemente miteinander kooperieren oder interagieren (z. B. in einem Ursache-Wirkungszusammenhang).

[0072] Spezifische Ausführungsformen der vorliegenden Erfindung wurden oben beschrieben. Jedoch ist die Erfindung nicht auf die Details derartiger Ausführungsformen beschränkt, sondern nur durch die unten angegebenen Patentansprüche und deren angemessene Äquivalente.

Patentansprüche

1. Verfahren, umfassend
Absuchen von Speicherseiten einer Speichermatrix, um eine erste freie Speicherseite zu finden;
Markieren der ersten freien Speicherseite als verfügbar; und
Beschreiben der als verfügbar markierten Speicherseite im nächsten Schreibzyklus.
2. Verfahren nach Anspruch 1, weiterhin umfassend:
Markieren der Speicherseite vor der ersten freien Speicherseite als ungültig; und
Löschen der als ungültig markierten Speicherseite.
3. Verfahren nach Anspruch 1, wobei das Absuchen von Speicherseiten das Absuchen von Speicherseiten in einer Reihenfolge, in welcher sie beschrieben sind und das Bestimmen, ob in einer Speicherseite Daten gespeichert sind, und, falls in der Speicherseite keine Daten gespeichert sind, dann Bestimmen, dass die nächste Speicherseite die erste freie Speicherseite ist, umfasst.
4. Verfahren nach Anspruch 3, wobei das Absuchen der Speicherseiten weiterhin umfasst dass, falls in der Speicherseite Daten gespeichert sind, dann bestimmen, ob die Speicherseite als gültig markiert ist, und, falls die Speicherseite nicht als gültig markiert ist, dann Bestimmen, dass die nächste Speicherseite die erste freie Speicherseite ist.
5. Verfahren nach Anspruch 4, wobei das Bestimmen, ob eine Speicherseite als gültig markiert ist, das Lesen einer mit der Speicherseite assoziierten Kopfzeile umfasst.

6. Verfahren nach Anspruch 5, wobei das Lesen einer Kopfzeile das Lesen einer Kopfzeile für einen Block von Speicherseiten umfasst.

7. Verfahren nach Anspruch 5, wobei das Lesen einer Kopfzeile das Lesen einer Kopfzeile in einem mit der Speicherseite assoziierten Reservebereich der Speichermatrix umfasst.

8. Verfahren nach Anspruch 1, wobei das Markieren als ungültig das Schreiben einer Marke in eine mit der Speicherseite assoziierten Kopfzeile umfasst.

9. Verfahren nach Anspruch 1, wobei das Markieren als ungültig das Schreiben eines Markers in eine mit einem Block von Speicherseiten assoziierte Kopfzeile umfasst, der die erste freie Speicherseite enthält.

10. Verfahren nach Anspruch 1, wobei das Markieren als gültig das Schreiben in eine mit der ersten freien Speicherseite assoziierten Lookup-Table umfasst.

11. Verfahren nach Anspruch 1, wobei die Speichermatrix in Blöcke unterteilt ist, und jeder Block mehrere Speicherseiten enthält, wobei das Absuchen von Speicherseiten das Absuchen von Speicherseiten eines Blocks umfasst, weiterhin umfassend:
Absuchen der Speicherseiten jedes anderen Blocks, um eine erste freie Speicherseite in jedem Block zu finden;
Markieren der Speicherseite vor der ersten freien Speicherseite in jedem Block als ungültig; und
Markieren der ersten freien Speicherseite in jedem Block als verfügbar.

12. Spannungsausfall-Wiederherstellungssystem für eine Speichermatrix, umfassend:
einen Speicherseiten-Absucher zum Absuchen von Speicherseiten einer Speichermatrix, um eine erste freie Speicherseite zu finden;
einen Speicherseiten-Markierer zum Markieren der ersten freien Speicherseite als verfügbar; und
Programmierschaltkreis zum Schreiben der als verfügbar markierten Speicherseite mit dem nächsten Schreibzyklus.

13. Spannungsausfall-Wiederherstellungssystem nach Anspruch 12, wobei der Speicherseiten-Markierer weiterhin dazu eingereicht ist, die Speicherseite vor der ersten freien Speicherseite als ungültig zu markieren, und wobei der Programmierschaltkreis weiterhin dazu eingerichtet sind, die als ungültig markierte Speicherseite zu löschen.

14. Spannungsausfall-Wiederherstellungssystem nach Anspruch 12, wobei der Speicherseiten-Absucher eine freie Speicherseite findet, indem be-

stimmt wird, ob eine Speicherseite in einer mit der Speicherseite assoziierten Kopfzeile als gültig markiert ist.

15. Spannungsausfall-Wiederherstellungssystem nach Anspruch 5, wobei das Lesen einer Kopfzeile das Lesen einer Kopfzeile für einen Block von Speicherseiten umfasst.

16. Speichereinheit, umfassend:
eine Speicherzellen-Matrix umfassend eine Mehrzahl von Speicherzellen, wobei die Zellen in eine Mehrzahl von Blöcken gruppiert sind; und
einen an die Speicherzellen-Matrix angekoppelter Speicher-Kontroller zum Absuchen von Speicherseiten einer Speichermatrix, um eine erste freie Speicherseite zu finden, die Speicherseite vor der ersten freien Speicherseite als ungültig zu markieren, die erste freie Speicherseite als verfügbar zu markieren, die als ungültig markierte Speicherseite zu löschen, und die als verfügbar markierte Speicherseite zu beschreiben.

17. Speichereinheit gemäß Anspruch 16, wobei der Speicher-Kontroller Speicherseiten in einer Reihenfolge absucht, in der sie beschrieben sind, und bestimmt, ob in einer Speicherseite Daten gespeichert sind, und, falls in der Speicherseite keine Daten gespeichert sind, dann bestimmt, dass die nächste Speicherseite die erste freie Speicherseite ist.

18. Speichereinheit gemäß Anspruch 17, wobei, falls in einer Speicherseite Daten gespeichert sind, der Speicher-Kontroller dann bestimmt, ob die Speicherseite als gültig markiert ist, und, falls die Speicherseite nicht als gültig markiert ist, dann bestimmt, dass die nächste Speicherseite die erste freie Speicherseite ist.

19. Speichereinheit gemäß Anspruch 16, wobei die Speicherzellen-Matrix eine Flash-Speicherzellen-Matrix ist.

20. Speichereinheit gemäß Anspruch 16, wobei die Speicherzellen-Matrix eine Phasenwechsel-Speicherzellen-Matrix ist.

Es folgen 7 Blatt Zeichnungen

Anhängende Zeichnungen

FIG. 1

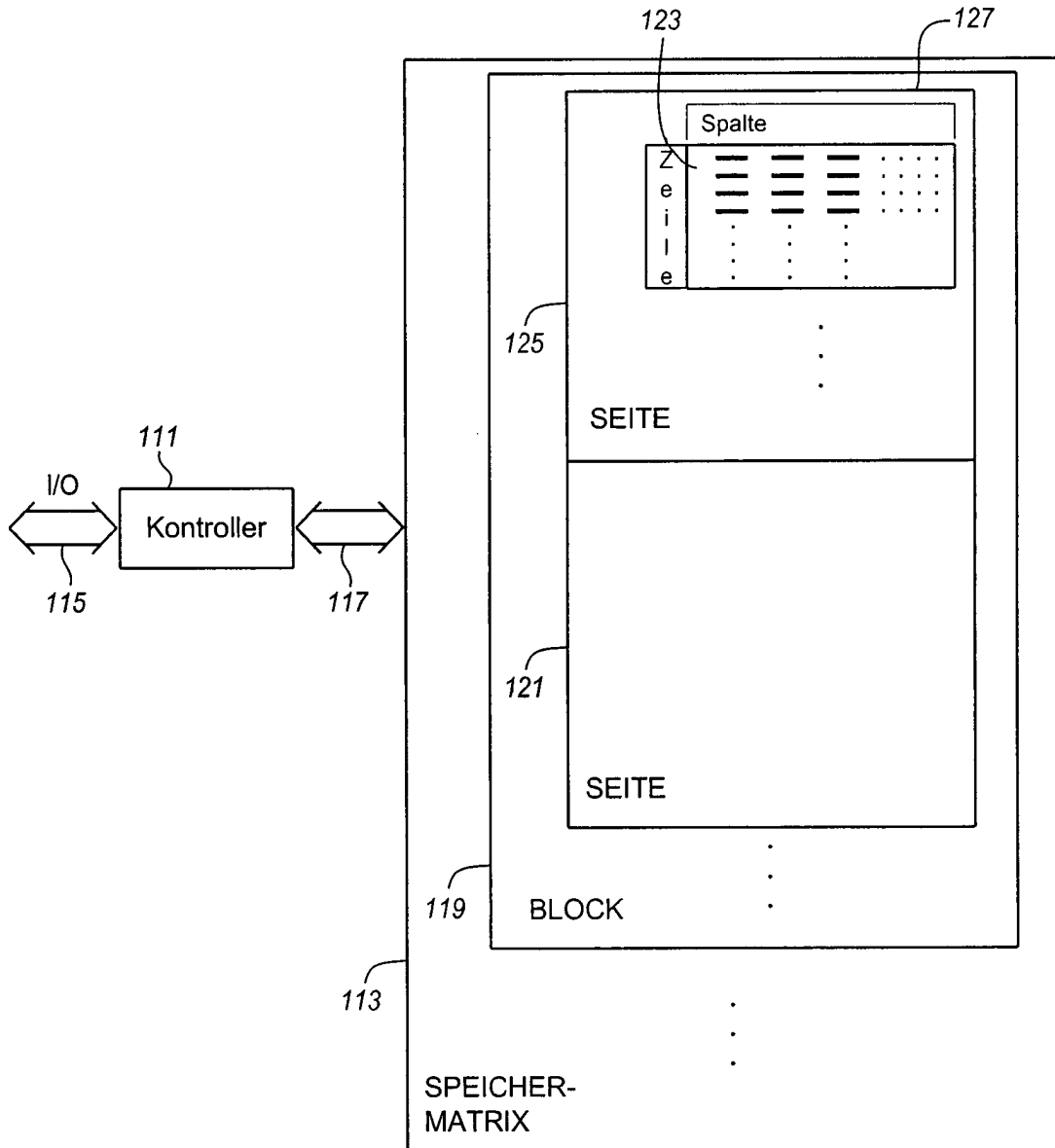


FIG. 2

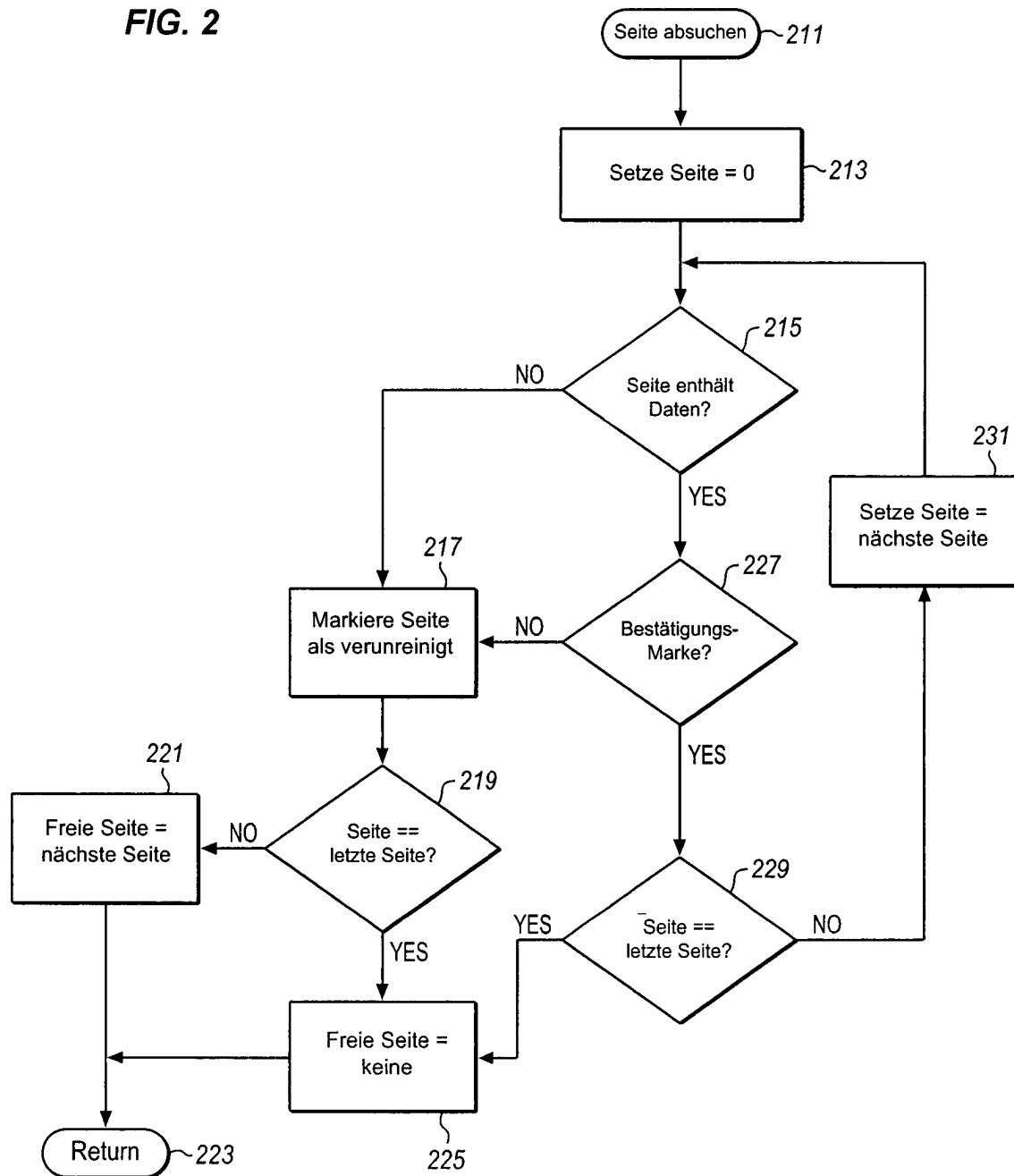


FIG. 3

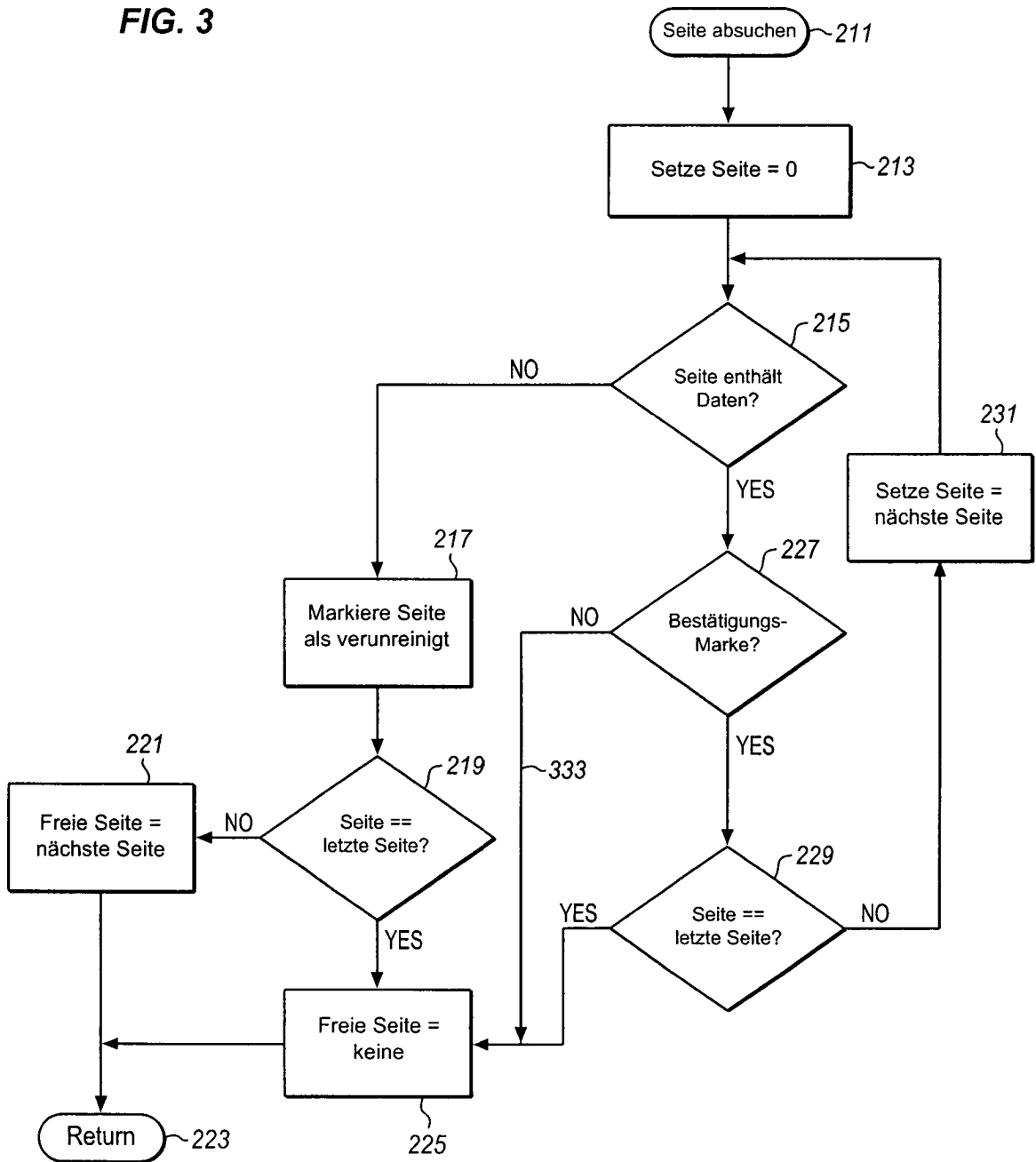


FIG. 4

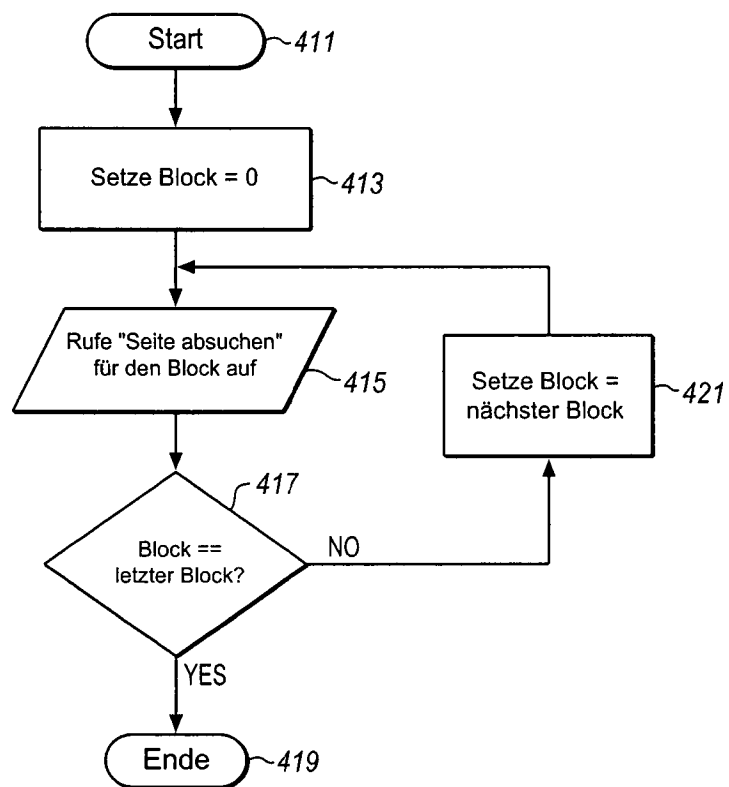


FIG. 5

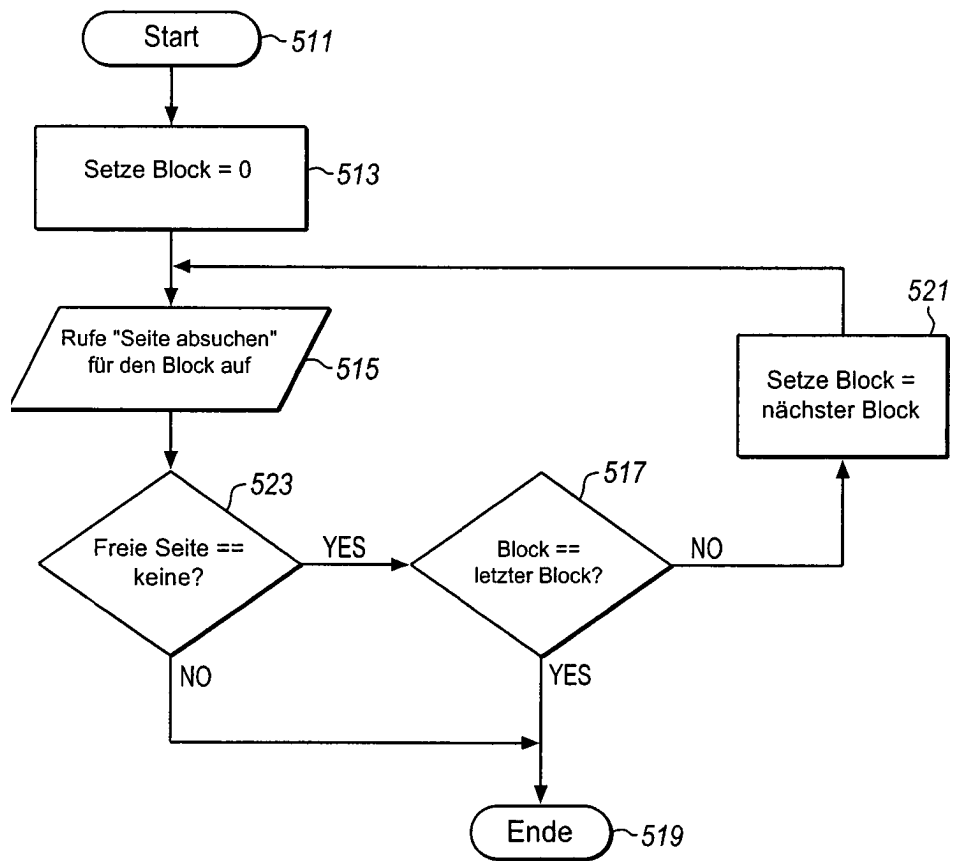


FIG. 6

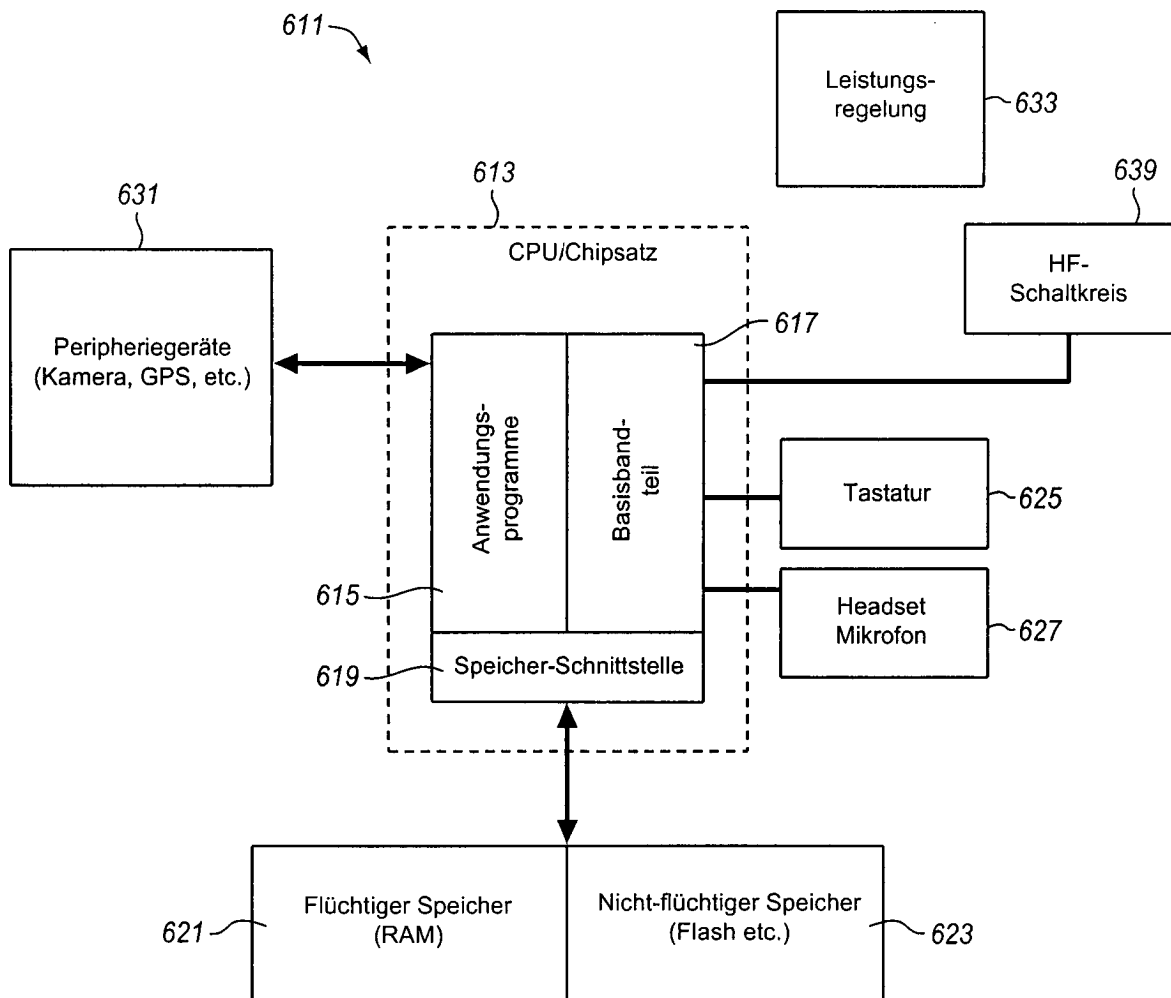


FIG. 7

