



(12) **Offenlegungsschrift**

(21) Aktenzeichen: **10 2015 200 808.8**

(51) Int Cl.: **G11C 16/06 (2006.01)**

(22) Anmeldetag: **20.01.2015**

(43) Offenlegungstag: **21.07.2016**

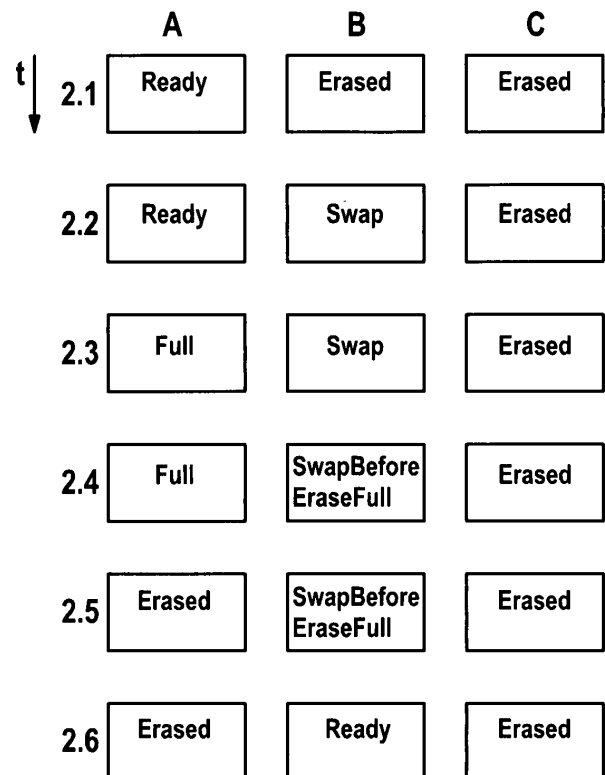
<p>(71) Anmelder: Continental Automotive GmbH, 30165 Hannover, DE</p> <p>(72) Erfinder: Srinivasan, Dilip, Bangalore, IN; Govindan, Naveen, Bangalore, IN</p>	<p>(56) Ermittelter Stand der Technik:</p> <table> <tr> <td>US</td> <td>6 834 331</td> <td>B1</td> </tr> <tr> <td>US</td> <td>7 478 270</td> <td>B2</td> </tr> <tr> <td>US</td> <td>2005 / 0 144 368</td> <td>A1</td> </tr> </table>	US	6 834 331	B1	US	7 478 270	B2	US	2005 / 0 144 368	A1
US	6 834 331	B1								
US	7 478 270	B2								
US	2005 / 0 144 368	A1								

Rechercheantrag gemäß § 43 Abs. 1 Satz 1 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Verfahren zur Behandlung von Unterbrechungen der Energieversorgung eines sektorweise löschbaren nichtflüchtigen Datenspeichers, elektronisches System und Computerprogrammprodukt**

(57) Zusammenfassung: Die Erfindung betrifft ein Verfahren zur Behandlung von Unterbrechungen der Energieversorgung eines löschbaren nichtflüchtigen Datenspeichers, wobei der Datenspeicher einzeln löschbare Speichersektoren (A, B, C) aufweist, die eine Vielzahl von Speichereinheiten enthalten, welche individuell beschrieben werden können, wobei sich das Verfahren dadurch auszeichnet, dass nach einem Einschalten der Energieversorgung des Datenspeichers von wenigstens drei aufeinanderfolgenden Speichersektoren (A, B, C) ein Vergleich von Zustandsindikatoren (Ready, Swap, Erased, SwapBeforeEraseFull, Full) mit definierten Zustandsindikatoren erfolgt und in Abhängigkeit einer Konstellation der Zustandsindikatoren dieser wenigstens drei Speichersektoren, dieser festgestellten Konstellation zugeordnete Handlungen vorgenommen werden. Weiterhin betrifft die Erfindung ein korrespondierendes elektronisches System und ein Computerprogrammprodukt.



Beschreibung

[0001] Die vorliegende Erfindung betrifft ein Verfahren gemäß Oberbegriff von Anspruch 1, ein korrespondierendes elektronisches System und ein entsprechendes Computerprogrammprodukt.

[0002] In elektronischen Steuergeräten, insbesondere in Kraftfahrzeugen, wird dazu übergegangen externe EEPROM-Speicher (elektrisch-löschbare Nur-Lese-Speicher) zu vermeiden, um die Kosten und den Platzbedarf zu reduzieren. Stattdessen werden häufig in einen Mikrocontroller des Steuergeräts integrierte lediglich sektorweise löschbare Flash-EEPROMs (elektrisch löschbarer programmierbarer Nur-Lese-Speicher) verwendet, welche in der Weise adaptiert sind, dass sie sich wie gewöhnliche EEPROM-Speicher verhalten. Diese Adaption wird als EEPROM Emulation bezeichnet.

[0003] Flash-Speicher sind in Speichersektoren und diese wiederum in kleineren Speichereinheiten (z.B. Bytes, Worte, etc.) organisiert, wobei ein Sektor z.B. aus 256 Speichereinheiten mit je 512 Bytes bestehen kann. Die Speicher besitzen die Eigenschaft, dass die kleineren Speichereinheiten beim Schreiben in den Flash-Speicher einzeln adressiert werden können, das Löschen jedoch jeweils nur für einen gesamten Sektor und somit nur für alle durch diesen umfasste Speichereinheiten erfolgen kann. Das Schreiben in den Speicher dauert üblicherweise länger als das Lesen und eine Löschoperation eines Sektors ist vergleichsweise noch zeitaufwändiger, z.B. mehrere Millisekunden. Bei konventionellen EEPROM-Speichern können der Speicherzugriff (Lesen und Schreiben) und die Löschung auf niedrigster Ebene erfolgen. Vor jedem Speichern neuer Daten in den einzelnen Speicherzellen muss für Flash-Speichern sowie konventionellen EEPROM-Speichern eine Löschung bereits vorhandener Daten durchgeführt werden. Da bei Flash-Speichern lediglich ein sektorweises Löschen möglich ist, ergibt sich zugleich eine wesentliche Einschränkung dieses Speichertyps gegenüber den konventionellen EEPROMs und zwar, dass die Lebensdauer bzw. die Anzahl der durchführbaren Löschungen geringer ausfällt, was für Anwendungen mit häufigen Speicherprozessen zu berücksichtigen ist. Um dennoch Flash-EEPROMs verwenden zu können, erfolgt üblicherweise eine Emulation des konventionellen EEPROMs in einem Flash-Speicher, wofür wenigstens eine Emulationssabstraktionsebene zwischen Applikationssoftware und Flash-Treiber Software vorgesehen ist. Dafür wird beispielsweise ein Sektor des emulierten EEPROM Flash-Speichers, von dem Daten während der Verarbeitung verändert wurden, in einen anderen Sektor dieses Speichers geschrieben, wofür entsprechend mehr Speichervolumen im Vergleich zum konventionellen EEPROM vorgesehen werden muss, wodurch die Lebensdauer allerdings entsprechend vergrößert wer-

den kann. Während ein Sektor mit Daten beschrieben wird, erfolgt die Vorbereitung des nächsten zu beschreibenden Sektors. Im Anschluss wird der vorher aktive Sektor gelöscht.

[0004] Die Energieversorgung eines Flash-Speichers, des Mikrocontrollers bzw. einer diesen umfassenden elektronischen Kontrolleinheit kann im laufenden Betrieb aufgrund unterschiedlicher Ursachen plötzlich unterbrochen werden. Die Inhalte der Flash-Speicher können in solchen Fällen undefinierte Zustände einnehmen. Für ein mittels der zugrundeliegenden Kontrolleinheit betriebenes System, beispielsweise eines Kraftfahrzeugs, ist die Datenintegrität jedoch häufig eine sicherheitsrelevante Voraussetzung.

[0005] Die DE 10 2008 002 494 A1 beschreibt ein artverwandtes Verfahren zum Aktualisieren eines Flash-Speichers mit einem Originaldatensatz-Speicherbereich, der mit einer Mehrzahl von Datensätzen beschrieben ist, und einem Datensatzinstanzen-Speicherbereich in dem eine aktuelle Fassung bzw. Instanz des wenigstens einen Datensatzes speicherbar ist.

[0006] Aus der DE 103 49 595 B3 geht ebenfalls ein artverwandtes Verfahren zum Schreiben von Speichersektoren in einzeln löschbaren Speicherblöcken, die eine Vielzahl von Speichersektoren enthalten, hervor.

[0007] Insbesondere im Falle eines Energieversorgungsabbrisses während eines Wechsels von einem gefüllten und im Löschungsprozess befindlichen Sektor zu einem nächsten Sektor eines Flash-Speichers, kann es vorkommen, dass dieser zumindest teilweise gelöschte Sektor keine Speicheranfragen auf Inhalte desselben akzeptiert, selbst der Inhalt des Sektors einen gelöschten Zustand aufweist.

[0008] Aufgabe der Erfindung ist es, ein Verfahren bereitzustellen, mittels welchem plötzliche Unterbrechungen der Energieversorgung für emulierte EEPROM Implementierungen behandelt werden können, welches ressourcenschonend ist und eine möglichst kurze Startzeit ermöglicht.

[0009] Diese Aufgabe wird durch ein Verfahren gemäß Anspruch 1, ein elektronisches System gemäß Anspruch 8 und ein Computerprogrammprodukt gemäß Anspruch 9 gelöst.

[0010] Die Erfindung beschreibt ein Verfahren zur Behandlung von Unterbrechungen der Energieversorgung eines löschbaren nichtflüchtigen Datenspeichers, wobei der Datenspeicher einzeln löschbare Speichersektoren aufweist, die eine Vielzahl von Speichereinheiten enthalten, welche individuell beschrieben werden können, wobei sich das Verfah-

ren dadurch ausgezeichnet, dass nach einem Einschalten der Energieversorgung des Datenspeichers von wenigstens drei aufeinanderfolgenden Speichersektoren ein Vergleich von Zustandsindikatoren mit definierten Zustandsindikatoren erfolgt und in Abhängigkeit einer Konstellation der Zustandsindikatoren dieser wenigstens drei Speichersektoren, dieser festgestellten Konstellation zugeordnete Handlungen vorgenommen werden.

[0011] In vorteilhafter Weise kann nach einer plötzlichen Unterbrechung der Energieversorgung somit sehr schnell der Speichersektor identifiziert werden, auf welchen Speicherzugriffe beispielsweise einer Applikationssoftware oder einer Emulierungsebene erfolgen sollen bzw. können, wodurch die Ausfallzeit eines zugrundeliegenden elektronischen Systems verringert sowie fehlerhafte Speicherzugriffe vermieden werden können. Das Behandeln von Szenarien der Unterbrechungen der Energieversorgung ist für die Bereitstellung einer zuverlässigen EEPROM-Emulation eine wichtige Voraussetzung. Mit anderen Worten wird durch die Erfindung in vorteilhafter Weise eine determinierte, robuste und zuverlässige EEPROM-Emulation bereitgestellt, welche die Erkennung und Korrektur undefinierter Flash-Speicher Zustände ermöglicht. Aufeinanderfolgende Speichersektoren sind unter Berücksichtigung der Sektorgröße bevorzugt in ihrer physischen Adresse des Datenspeichers aufeinanderfolgende Sektoren. Dadurch kann eine schnelle Abarbeitung ermöglicht werden. Alternativ können Speichersektoren diesbezüglich auch in einem linear abgebildeten virtuellen Speicherbereich aufeinanderfolgend sein, wobei die Information der Reihenfolge der Aufreihung auch im Falle einer Unterbrechung der Energieversorgung erhalten bleiben sollte.

[0012] Bevorzug erfolgt vor dem Vergleich der Zustandsindikatoren mit den definierten Zustandsindikatoren eine Suche nach wenigstens einem Daten Sektor, welcher einen Zustandsindikator aufweist, der auf einen nicht gelöschten Zustand des Datensektors hindeutet. Die drei aufeinanderfolgenden Speichersektoren, deren Konstellation festgestellt wird, werden bevorzugt von dem Datensektor, welcher einen Zustandsindikator aufweist, der auf einen nicht gelöschten Zustand des Datensektors hindeutet, sowie zumindest zwei weiteren – insbesondere mittelbar oder unmittelbar angrenzenden – Datensektoren gebildet. Eine Wahl einer Vorgehensweise zum Auffinden des Ausgangsdatensektors erfolgt dabei bevorzugt entsprechend einer kürzesten Suchdauer.

[0013] Gemäß einer bevorzugten Ausführungsform der Erfindung werden die Zustandsindikatoren bzw. die Headerinformationen dieser wenigstens drei Sektoren vor dem Vergleich der Zustandsindikatoren mit den definierten Zustandsindikatoren auf Plausibilität überprüft. Datensektoren mit undefinierten bzw. feh-

lerhaften Zuständen können somit korrigiert werden. Eine Plausibilitätsprüfung aller oder zumindest einer Vielzahl der Sektoren des Datenspeichers kann beispielsweise als separater Vorgang oder im Rahmen der Suche des Datensektors erfolgen, welcher als Ausgangspunkt für die Feststellung der Konstellation der wenigstens drei aufeinanderfolgenden Speichersektoren dient.

[0014] Besonders bevorzugt sind im laufenden Betrieb des Datenspeichers wenigstens vier Zustandsindikatoren vorgesehen. Bevorzugt sind genau fünf Zustandsindikatoren vorgesehen. Vorzugsweise unterscheiden die Zustandsindikatoren Speichersektoren, welche sich in einem gelöschten Zustand befinden, einen Abschluss eines Kopiervorgangs anzeigen, welche bereit zum Beschreiben von Daten sind, welche gefüllt sind und welche aktuell einem Kopierprozess unterliegen bzw. in welche gerade geschrieben wird. Eine Erkennung und Korrektur von beschädigten Datensektoren und somit ein Erhalt der Datenintegrität von Flash-Datenspeichern kann somit mittels eines sehr geringen Ressourceneinsatzes realisiert werden.

[0015] Entsprechend einer vorteilhaften Weiterbildung ist wenigstens ein Zustandsindikator vorgesehen, welcher einen Abschluss eines Kopiervorgangs eines Inhalts von einem Speichersektor auf einen weiteren Speichersektor anzeigt.

[0016] Bei einer Konstellation der Speichersektoren in der Reihenfolge aufeinanderfolgender Sektoren: Speichersektor in einem gelöschten Zustand, Speichersektor in dem einen Abschluss eines Kopiervorgangs anzeigenden Zustand und Speichersektor in einem gelöschten Zustand, wird bevorzugt der Speichersektor gelöscht, der den einen Abschluss eines Kopiervorgangs anzeigenden Zustand aufweist.

[0017] Der Inhalt eines in einem undefinierten Zustand befindlichen Sektors wird zweckmäßigerweise gelöscht. Vorteilhafterweise wird somit ein fehlerhaftes Verhalten des Datensektors, in Folge eines unterbrochenen Löschmodus vor der Unterbrechung der Energieversorgung, insbesondere für anschließende Speicheroperationen verhindert.

[0018] Das erfindungsgemäße Verfahren wird vorzugsweise während einer Initialisierungsphase des Datenspeichers und/oder eines zugrundeliegenden elektronischen Systems ausgeführt.

[0019] Die Erfindung betrifft weiterhin ein elektronisches System umfassend einen löschbaren nichtflüchtigen Datenspeicher, wobei der Datenspeicher einzeln löschbare Speichersektoren aufweist, die eine Vielzahl von Speichereinheiten enthalten, welche individuell beschrieben werden können, sowie Mittel

zum Schreiben von Daten in den Datenspeicher aufweist, wobei das System ausgestaltet ist das erfindungsgemäße Verfahren auszuführen.

[0020] Zudem betrifft die Erfindung ein Computerprogrammprodukt, welches eingerichtet ist, dass es, wenn es auf einem elektronischen System ausgeführt wird, das erfindungsgemäße Verfahren ausführt.

[0021] Bevorzugt sind die Zustandsindikatoren in einem Header eines jeweiligen Sektors hinterlegt.

[0022] Jeder Speichersektor des Datenspeichers umfasst bevorzugt den Header sowie die zu speichernden Informationen (Daten) als solche. Dadurch wird in vorteilhafter Weise eine effiziente Ausnutzung der Speicherressourcen für die zu speichernden Informationen erreicht.

[0023] Weitere bevorzugte Ausführungsformen ergeben sich aus der nachfolgenden Beschreibung von Ausführungsbeispielen an Hand von Figuren.

[0024] In schematischer Darstellung zeigen:

[0025] Fig. 1 eine erfindungsgemäße Datenstruktur eines Sektors eines Flash-Speichers,

[0026] Fig. 2 ein beispielesgemäßer Ablauf eines Wechselsvorgangs zwischen Sektoren,

[0027] Fig. 3 ein beispielhaftes Ablaufdiagramm als Teil einer Initialisierungssequenz des Flash-Speichers und

[0028] Fig. 4. Szenarien der Zustände von Sektoren während einer Initialisierungssequenz.

[0029] Die Fig. 1 zeigt eine Prinzipdarstellung einer beispielesgemäßen Datenstruktur Data_Record der Sektoren eines Flash-Speichers. Diese umfasst den Identifikator Id, mittels dem jede Datenstruktur eindeutig identifiziert werden kann, die zu speichernden Daten Data[] und weitere Elemente, wie beispielsweise eine Sicherstellung der Datenintegrität durch Einhaltung vorgegebener Bit-Grenzen (z.B. 64-bit) der Datensätze Align_Byte[] und/oder zumindest einen Prüfsummenwert Checksum zur Sicherstellung der Datenintegrität.

[0030] Der Ablauf während eines Wechsels zu beschreibender Sektoren eines Flash-Speichers im schrittweisen Zeitablauf t wird nachfolgend anhand von Fig. 2 erläutert. Um Daten in einem Sektor eines Flash-Speichers speichern zu können, muss sich dieser in einem gelöschten Zustand befinden, wobei das Potential abhängig von der jeweiligen Implementierung abhängig ist. Der zuerst zu beschreibende Sektor wird in der Weise vorbereitet, dass darin Daten gespeichert werden können. Zu diesem Zweck wird die

Headerinformation des Sektors A aktualisiert. Dieser Zustand ist in Fig. 2 durch Ready repräsentiert. Alle weiteren Speicherzugriffe der Applikationssoftware erfolgen auf diesen Sektor. Daten können in Schritt 2.1 in Sektor A abgelegt werden bis dieser keinen freien Speicherplatz mehr aufweist, also gefüllt ist. Die Erkennung des gefüllten Zustands von Sektor A erfolgt beispielsweise über dessen finale absolute Speicheradresse. Wenn der Sektor A gefüllt ist, erfolgt die Vorbereitung des nächsten Sektors B, auf welchen die Speicherzugriffe der Applikationssoftware nach dem Wechselsvorgang erfolgen.

[0031] Insbesondere um, wie in der Beschreibungseinleitung bereits beschrieben, eine möglichst homogene Abnutzung des Flash-Speichers zu ermöglichen, erfolgt anschließend ein Wechsel („Swap“) von Sektor A nach Sektor B, wobei die letzten Daten des gegenwärtigen Sektors A im Zustand Ready in den nächsten verfügbaren gelöschten Sektor B kopiert werden, welcher Zustand Swap zugewiesen bekommt 2.2. Dem gefüllten Sektor A wird entsprechend Zustand Full zugewiesen 2.3. Die Phase des Erfassens und Kopierens des Inhalts des Sektors im Zustand Full in den Sektor im Zustand Swap kann vergleichsweise zeitaufwändig sein, weshalb Zustand Swap über mehrere Schritte dargestellt ist. Nach dem Kopieren erhält Sektor B Zustand SwapBeforeEraseFull 2.4 und Sektor A im Zustand Full wird in Schritt 2.5 gelöscht und erhält somit Zustand Erase. Abschließend erhält Sektor B Zustand Ready, wodurch angezeigt ist, dass dieser für den Zugriff vorgesehen ist. Dadurch wird die Wechseloperation beendet und der Emulator akzeptiert neue Speicherzugriffe der Applikationssoftware. Während eines Wechsels der Sektoren wird seitens des Emulators kein Zugriff einer Applikationssoftware auf den Flash-Speicher umgesetzt. Einzelne Schritte des Wechselsvorgangs können dabei in abweichender Reihenfolge vorgenommen werden bzw. können weitere Zwischenschritte vorgesehen sein.

[0032] Fig. 3 zeigt in Form eines Ablaufdiagramms einen Teil der Initialisierungssequenz des Flash-Speichers, z.B. nach einem Trennen bzw. Unterbrechen der Energieversorgung, wobei die Emulationssoftware nach dem Start 3.1 die Zustandsinformationen der Header von wenigstens drei Sektoren durch Vergleich 3.2 mit Werten – z.B. Erased, Full, Swap, SwapBeforeEraseFull – eines oder mehrerer definierter Wertebereiche auf Plausibilität überprüft. Entspricht der Wert einem der vordefinierten Zustände – Erased, Ready, Swap oder SwapBeforeEraseFull – kann die Überprüfung dieses Sektors abgeschlossen werden. Stimmt der Inhalt nicht mit einem der vordefinierten Zustände überein, beispielsweise weil in Folge eines Energieversorgungsabbruchs ein undefinierter Zustand vorliegt, wird der entsprechende Sektor in 3.3 gelöscht und erhält den Zustand Erased. Tritt bei diesem Löschvorgang ein Fehler auf, was in 3.5

überprüft wird, erfolgt die Ausgabe eines Fehlers **3.5** und die Überprüfung dieses Sektors wird abgeschlossen **3.6**, andernfalls wird die Überprüfung dieses Sektors ohne Ausgabe eines Fehlers beendet. Ein Sektor, welcher einen Fehler beim Löschen verursacht, kann ein entsprechender Zustand zugewiesen werden, welcher einen Defekt anzeigt. Das Löschen des Sektors kann alternativ auch an die Bedingung geknüpft werden, dass der Sektor nicht bereits gelöscht ist, was mittels einer Speicherinhaltsprüfung eruiert werden kann. Die Überprüfung kann für alle oder eine Vielzahl von Sektoren zunächst eigenständig durchgeführt werden oder in Verbindung mit einer Suche nach einem Datensektor, dessen Header auf einen nicht gelöschten Zustand hinweist: Full, SwapBeforeEraseFull, Ready, Swap. Nachdem ein solcher Header gefunden ist, kann insbesondere in Abhängigkeit des jeweiligen Zustands festgelegt sein, welche angrenzenden Datensektoren für eine Betrachtung einer Konstellation drei aufeinanderfolgender Datensektoren herangezogen werden. Aufeinanderfolgend ist dabei nicht einschränkend im Sinne einer physikalischen adressierten Aneinanderreihung zu verstehen sondern kann sich beispielsweise auch auf deren virtuelle Speicheradressierung beziehen. Im Anschluss an die Sequenz gemäß der Beschreibung zur **Fig. 3** sind ausgehend von der Beschreibung zu **Fig. 2** die möglichen Szenarien der Konstellationen der Zustände von Sektoren in **Fig. 4** gezeigt.

[0033] Die einzelnen Szenarien sind dabei von dem Zeitpunkt der Unterbrechung der Energieversorgung während eines Wechselvorgangs abhängig. Für jedes dieser Szenarien werden jeweils zugeordnete Handlungen vorgenommen:

S1: Sektor A = Ready
Sektor B = Erased
Sektor C = Erased

[0034] Die Emulationssoftware wird in der Weise initialisiert, dass die Speicherzugriffe der Applikationssoftware auf den Sektor A im Zustand Ready gerichtet werden.

S2: Sektor A = Ready
Sektor B = Swap
Sektor C = Erased

[0035] Der Sektor B im Zustand Swap wird gelöscht und der Wechselvorgang erneut gestartet.

S3: Sektor A = Full
Sektor B = Swap
Sektor C = Erased

[0036] Der Sektor B im Zustand Swap wird gelöscht und der Wechselvorgang erneut gestartet.

S4: Sektor A = Full
Sektor B = SwapBeforeEraseFull
Sektor C = Erased

[0037] Der Sektor A im Zustand Full wird gelöscht und der Sektor im Zustand SwapBeforeEraseFull erhält den Zustand Ready, da in diesem Fall der Speicherinhalt des Sektors im Zustand Full bereits vor der Unterbrechung vollständig in den nächsten Sektor kopiert wurde.

S5: Sektor A = Erased
Sektor B = SwapBeforeEraseFull
Sektor C = Erased

[0038] Der Sektor B, welcher sich vor der Unterbrechung im Zustand SwapBeforeEraseFull befunden hat, wird gelöscht und erhält den Zustand Erase.

[0039] Das in S5 gezeigte Szenario entspricht dem besonders kritischen Fall, dass eine Unterbrechung der Energieversorgung im Verlauf einer Löschopeation während eines Wechselvorgangs der Sektoren, also beim Übergang von Schritt **2.4** in Schritt **2.5** von **Fig. 2**, eintritt. Vor dem Löschen des sich im Zustand Full befindlichen Sektors ist der Zustand SwapBeforeEraseFull des nächsten Sektors insbesondere deshalb vorgesehen, da im Falle einer Unterbrechung der Energieversorgung während eines Löschovorgangs eines Sektors im Zustand Full, Speicherzugriffe der Applikationssoftware nach Neustart des Flash-Speichers auf den Sektor nicht erfolgen konnten, obwohl der Speicherinhalt des Sektors als solches einen gelöschten Zustand aufwies. Dies tritt insbesondere auf, wenn die vom Löschoprozess nicht unabhängig löschobaren Headerinformationen eines Sektors gelöscht werden, bevor dieser Sektor vollständig gelöscht ist. Bei Auslösen eines Löschobefehls wird der gesamte Sektor gelöscht und löscht ebenfalls den Header des Sektors (z.B. 0xFF). Bei einem Neustart würden diese Headerinformationen fälschlicherweise einen gelöschten Zustand des gesamten Sektors anzeigen. Um sicherzustellen, dass der Sektor A keine Fehler bei Speicheroperationen in Folge eines unterbrochenen Löschovorgangs vor der Unterbrechung der Energieversorgung hervorruft, wird dieser gelöscht und erhält den Zustand Erased, auch wenn dessen Inhalt auf einen leeren Sektor hindeuten sollte. Dadurch kann ein fehlerhaftes Verhalten des Sektors, in Folge eines unterbrochenen Löschoprozesses vor der Unterbrechung der Energieversorgung, für anschließende Speicheroperationen ausgeschlossen werden.

[0040] Somit kann durch Vergleich von Headerinformationen von nur 3 aufeinanderfolgenden Sektoren, der Sektor identifiziert werden, auf den die schreibenden und lesenden Speicherzugriffe einer Applikationssoftware erfolgen können.

[0041] Bevorzugt ist für jeden Sektor separat ein Zählwert zur Erfassung der Anzahl der Löschovorgänge des jeweiligen Sektors vorgesehen, welcher in dem Sinne Berücksichtigung findet, dass eine gleichmäßige Abnutzung der Sektoren erfolgt.

ZITATE ENTHALTEN IN DER BESCHREIBUNG

Diese Liste der vom Anmelder aufgeführten Dokumente wurde automatisiert erzeugt und ist ausschließlich zur besseren Information des Lesers aufgenommen. Die Liste ist nicht Bestandteil der deutschen Patent- bzw. Gebrauchsmusteranmeldung. Das DPMA übernimmt keinerlei Haftung für etwaige Fehler oder Auslassungen.

Zitierte Patentliteratur

- DE 102008002494 A1 [0005]
- DE 10349595 B3 [0006]

Patentansprüche

1. Verfahren zur Behandlung von Unterbrechungen der Energieversorgung eines löschbaren nichtflüchtigen Datenspeichers, wobei der Datenspeicher einzeln löschbare Speichersektoren (A, B, C) aufweist, die eine Vielzahl von Speichereinheiten enthalten, welche individuell beschrieben werden können, **dadurch gekennzeichnet**, dass nach einem Einschalten der Energieversorgung des Datenspeichers von wenigstens drei aufeinanderfolgenden Speichersektoren (A, B, C) ein Vergleich von Zustandsindikatoren (Ready, Swap, Erased, SwapBeforeEraseFull, Full) mit definierten Zustandsindikatoren erfolgt und in Abhängigkeit einer Konstellation der Zustandsindikatoren dieser wenigstens drei Speichersektoren, dieser festgestellten Konstellation zugeordnete Handlungen vorgenommen werden.

2. Verfahren gemäß Anspruch 1, **dadurch gekennzeichnet**, dass vor dem Vergleich der Zustandsindikatoren mit den definierten Zustandsindikatoren eine Suche nach wenigstens einem Daten Sektor erfolgt, welcher einen Zustandsindikator aufweist, der auf einen nicht gelöschten Zustand (Ready, Swap, SwapBeforeEraseFull, Full) des Datensektors hindeutet.

3. Verfahren gemäß Anspruch 2, **dadurch gekennzeichnet**, dass die drei aufeinanderfolgenden Speichersektoren, deren Konstellation festgestellt wird, von dem Datensektor, welcher einen Zustandsindikator aufweist, der auf einen nicht gelöschten Zustand des Datensektors hindeutet, sowie zumindest zwei weiteren Datensektoren gebildet werden.

4. Verfahren gemäß Anspruch 1 bis 3, **dadurch gekennzeichnet**, dass die Zustandsindikatoren (Ready, Swap, Erased, SwapBeforeEraseFull, Full) dieser wenigstens drei Sektoren vor dem Vergleich der Zustandsindikatoren (Ready, Swap, Erased, SwapBeforeEraseFull, Full) mit den definierten Zustandsindikatoren auf Plausibilität überprüft werden.

5. Verfahren gemäß einem der Ansprüche 1 bis 4, **dadurch gekennzeichnet**, dass wenigstens ein Zustandsindikator (SwapBeforeEraseFull) vorgesehen ist, welcher einen Abschluss eines Kopiervorgangs eines Inhalts von einem Speichersektor (A, B, C) auf einen weiteren Speichersektor (A, B, C) anzeigt.

6. Verfahren gemäß Anspruch 5, **dadurch gekennzeichnet**, dass bei einer Konstellation (E-SB-WF-E) der Speichersektoren in der Reihenfolge aufeinanderfolgender Sektoren:
 Speichersektor in einem gelöschten Zustand (Erased),
 Speichersektor in dem einen Abschluss eines Kopiervorgangs anzeigenden Zustand (SwapBeforeEraseFull) und

Speichersektor in einem gelöschten Zustand (Erased),
 der Speichersektor gelöscht wird, der den einen Abschluss eines Kopiervorgangs anzeigenden Zustand (SwapBeforeEraseFull) aufweist.

7. Verfahren gemäß einem der Ansprüche 1 bis 6, **dadurch gekennzeichnet**, dass der Inhalt eines in einem undefinierten Zustand befindlichen Sektors gelöscht wird.

8. Elektronisches System umfassend einen löschbaren nichtflüchtigen Datenspeicher, wobei der Datenspeicher einzeln löschbare Speichersektoren (A, B, C) aufweist, die eine Vielzahl von Speichereinheiten enthalten, welche individuell beschrieben werden können, sowie Mittel zum Schreiben von Daten in den Datenspeicher aufweist, **dadurch gekennzeichnet**, dass das System ausgestaltet ist ein Verfahren gemäß einem der Ansprüche 1 bis 7 auszuführen.

9. Computerprogrammprodukt, welches eingerichtet ist, dass es, wenn es auf einem elektronischen System ausgeführt wird, ein Verfahren gemäß einem der Ansprüche 1 bis 8 ausführt.

Es folgen 2 Seiten Zeichnungen

Anhängende Zeichnungen

Fig. 1

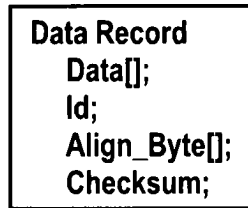


Fig. 2

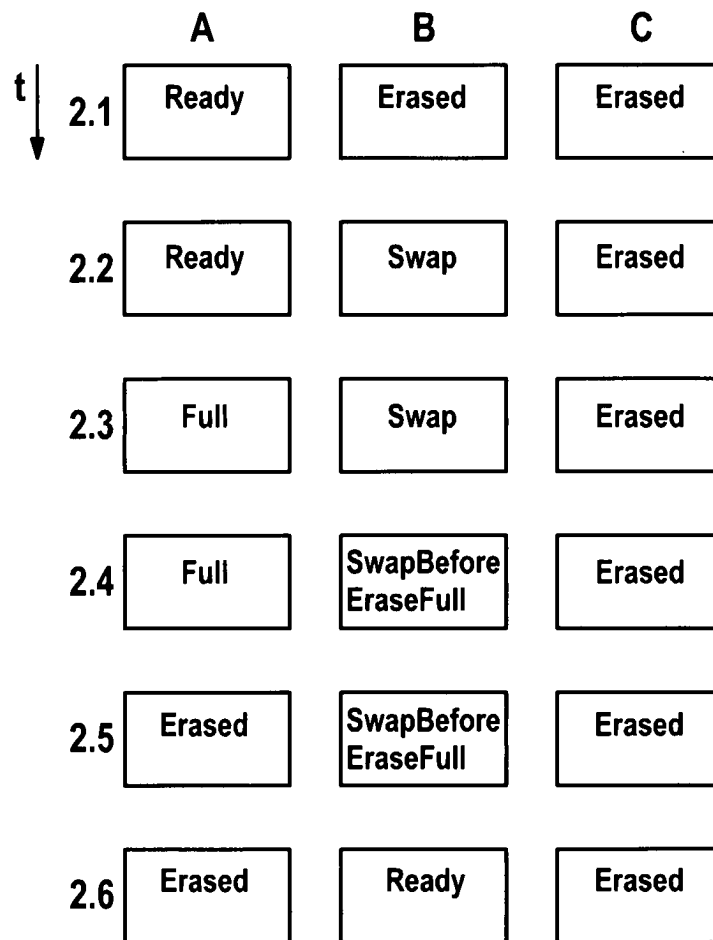


Fig. 3

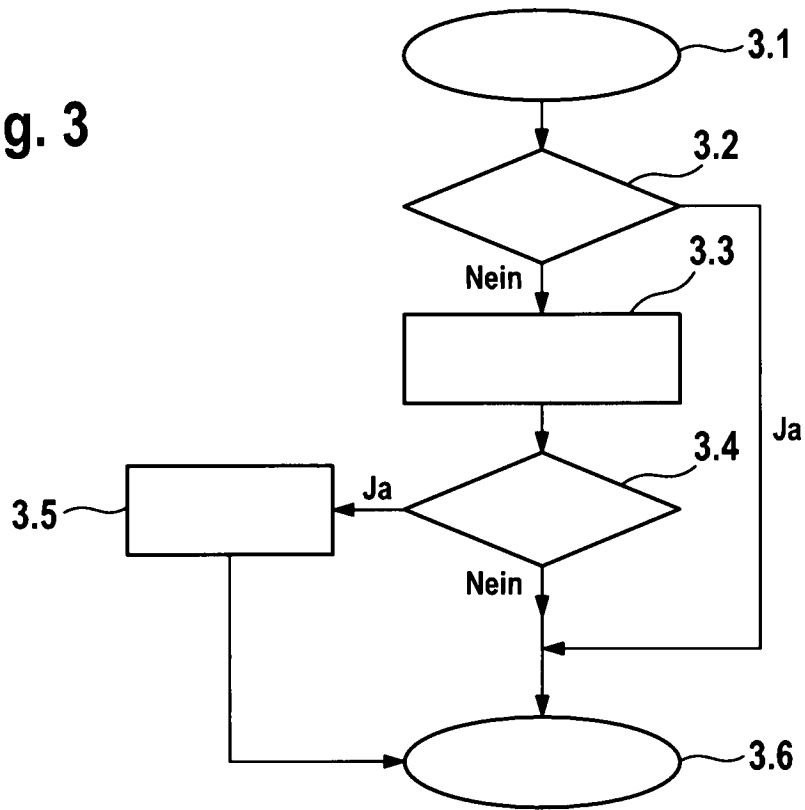


Fig. 4

	A	B	C
S1	Ready	Erased	Erased
S2	Ready	Swap	Erased
S3	Full	Swap	Erased
S4	Full	SwapBefore EraseFull	Erased
S5	Erased	SwapBefore EraseFull	Erased