



BUNDESPATENTGERICHT

17 W (pat) 29/09

(Aktenzeichen)

BESCHLUSS

In der Beschwerdesache

betreffend die Patentanmeldung 10 2005 019 842.2 - 53

...

hat der 17. Senat (Technischer Beschwerdesenat) des Bundespatentgerichts am 15. Oktober 2013 unter Mitwirkung des Vorsitzenden Richters Dipl.-Phys. Dr. Morawek, der Richterin Eder, des Richters Dipl.-Ing. Baumgardt und des Richters Dipl.-Ing. Hoffmann

beschlossen:

Auf die Beschwerde der Anmelderin wird der Beschluss der Prüfungsstelle für Klasse G06F des Deutschen Patent- und Markenamts vom 14. Januar 2009 aufgehoben und das Patent mit folgenden Unterlagen erteilt:

Patentansprüche 1 bis 20 vom 6. September 2013, eingegangen am 9. September 2013,

Beschreibung Seiten 1, 2, 3, 3a und 4 bis 26 vom 6. September 2013, eingegangen am 9. September 2013

und 14 Blatt Zeichnungen mit Figuren 1 bis 14 vom 21. Juli 2005.

Gründe

I.

Die vorliegende Patentanmeldung wurde am 28. April 2005 beim Deutschen Patent- und Markenamt eingereicht. Sie trägt die Bezeichnung:

„System und Verfahren zum sequentiellen Schreiben
von Daten in einen Flash-Speicher“.

Die Anmeldung wurde durch Beschluss der Prüfungsstelle für Klasse G 06 F des Deutschen Patent- und Markenamts mit der Begründung zurückgewiesen, dass der Gegenstand des Hauptanspruchs des Hauptantrags und auch die Gegenstände des jeweiligen Hauptanspruchs des ersten und zweiten Hilfsantrags mangels erfinderischer Tätigkeit nicht gewährbar seien, da sie sich unter Zuhilfenahme

des fachmännischen Könnens in naheliegender Weise aus der Zusammenschau zweier entgegengehaltener Druckschriften ergäben.

Gegen diesen Beschluss ist die Beschwerde der Anmelderin gerichtet. Zur Begründung führt sie im Wesentlichen aus, der Zurückweisung lägen aus ihrer Sicht eine rückschauende Betrachtungsweise und eine Fehlinterpretation der Druckschriften zugrunde.

Auf einen Hinweis des Senats hin hat die Anmelderin ihr Patentbegehren klargestellt und die Beschreibung angepasst.

Die Anmelderin beantragt nunmehr sinngemäß,

den angegriffenen Beschluss aufzuheben und das nachgesuchte Patent mit folgenden Unterlagen zu erteilen:

Patentansprüche 1 bis 20 vom 6. September 2013, eingegangen am 9. September 2013;

Beschreibung Seiten 1, 2, 3, 3a und 4 bis 26 vom 6. September 2013, eingegangen am 9. September 2013;

sowie 14 Blatt Zeichnungen mit Figuren 1 bis 14 vom 21. Juli 2005.

Das geltende Patentbegehren, hier bezüglich des Hauptanspruchs mit einer möglichen Gliederung versehen, lautet:

1. System (200), das imstande ist, Daten sequentiell in einen Flash-Speicher (30) zu schreiben, wobei das System (200) aufweist:
 - 1.1. mindestens einen virtuellen Block, wobei jeder virtuelle Block eine Mehrzahl von virtuellen Seiten und eine virtuelle Block-

adresse aufweist, wobei jede virtuelle Seite einen virtuellen Seiten-Offsetwert besitzt;

- 1.2. mindestens einen physikalischen Block im Flash-Speicher (30), wobei jeder physikalische Block eine Mehrzahl von physikalischen Seiten und eine physikalische Blockadresse aufweist, wobei jede physikalische Seite im physikalischen Block einen physikalischen Seiten-Offsetwert besitzt;
- 1.3. wobei jeder virtuelle Block auf einen physikalischen Block abgebildet ist und in den virtuellen Blocks jede virtuelle Seite auf eine physikalische Seite in dem entsprechenden physikalischen Block abgebildet ist und der virtuelle Seiten-Offsetwert gleich dem physikalischen Seiten-Offsetwert ist;
- 1.4. wobei das System ferner aufweist:
eine erste Abbildungstabelle (40), die eine Mehrzahl von ersten Abbildungsdaten aufweist, wobei jede der ersten Abbildungsdaten eine virtuelle Blockadresse und eine physikalische Blockadresse speichert, um einen spezifischen virtuellen Block auf einen spezifischen physikalischen Block abzubilden; und

gekennzeichnet ist durch:

- 1.5. mindestens einen aktiven Block im Flash-Speicher (30), wobei jeder aktive Block eine Mehrzahl von physikalischen Seiten und eine aktive Blockadresse aufweist, jede physikalische Seite im aktiven Block einen physikalischen Seiten-Offsetwert besitzt, wobei jeder aktive Block auf einen physikalischen Block abgebildet ist;
- 1.6. wobei das System ferner aufweist:
eine zweite Abbildungstabelle (50), die eine Mehrzahl von zweiten Abbildungsdaten aufweist, wobei jede der zweiten Abbildungsdaten eine physikalische Blockadresse und eine aktive

Blockadresse aufweist, um einen spezifischen aktiven Block auf einen spezifischen physikalischen Block abzubilden; und

1.7. eine dritte Abbildungstabelle (60), die eine Mehrzahl von dritten Abbildungsdaten aufweist, wobei jede der dritten Abbildungsdaten einen virtuellen Seiten-Offsetwert und einen physikalischen Seiten-Offsetwert aufweist, um eine spezifische virtuelle Seite auf eine spezifische physikalische Seite eines spezifischen aktiven Blocks abzubilden.

2. System (200) gemäß Anspruch 1, ferner gekennzeichnet durch:
- ein Empfangsmodul (202), um eine Zugriffsanfrage zu empfangen, wobei die Zugriffsanfrage eine virtuelle Seitennummer aufweist, die eine virtuelle Seite repräsentiert;
 - ein Suchmodul für einen physikalischen Block (204), um einen spezifischen physikalischen Block entsprechend der virtuellen Seitennummer aufzufinden; und
 - ein Entscheidungsmodul (206), um zu bestimmen, ob der spezifische physikalische Block auf einen aktiven Block abgebildet ist.
3. System (200) gemäß Anspruch 1, ferner gekennzeichnet durch:
- ein Empfangsmodul (202) zum Empfang einer Schreibanfrage, wobei die Schreibanfrage eine virtuelle Seitennummer aufweist, die eine virtuelle Seite repräsentiert;
 - ein Suchmodul (204) für einen physikalischen Block, zum Ermitteln einer virtuellen Blockadresse entsprechend der virtuellen Seitennummer, um einen spezifischen physikalischen Block aufzufinden;
 - ein Entscheidungsmodul (206) zum Ermitteln, ob der spezifische physikalische Block auf einen aktiven Block abgebildet wurde;
 - ein Suchmodul (208) zum Suchen des abgebildeten aktiven Blocks, um zu ermitteln, ob der abgebildete aktive Block eine leere physikalische Seite besitzt;

ein Offsetwert-Suchmodul (210) zum Ermitteln eines virtuellen Seiten-Offsetwerts entsprechend der virtuellen Seitennummer, um zu erfassen, ob eine physikalische Seite, die den gleichen Offsetwert wie der virtuelle Seiten-Offsetwert aufweist, eine leere physikalische Seite im spezifischen physikalischen Block ist; und

ein Schreibmodul (222) zum Schreiben von Daten auf die leere physikalische Seite des abgebildeten aktiven Blocks oder der physikalischen leeren Seite des spezifischen physikalischen Blocks.

4. System (200) gemäß Anspruch 3, ferner gekennzeichnet durch ein Mischmodul (216), um einen spezifischen aktiven Block und einen entsprechend abgebildeten physikalischen Block zu mischen, um einen neuen physikalischen Block zu erzeugen.
5. System (200) gemäß Anspruch 4, ferner gekennzeichnet durch ein Erzeugungsmodul (218) für einen aktiven Block, um einen neuen aktiven Block zu erzeugen, wobei der neue aktive Block auf den spezifischen physikalischen Block abgebildet ist und ein leerer aktiver Block ist.
6. System (200) gemäß Anspruch 4, ferner gekennzeichnet durch ein Wahlmodul (214), um einen spezifischen aktiven Block zum Blockmischen auszuwählen.
7. System (200) gemäß Anspruch 6, dadurch gekennzeichnet, dass das Wahlmodul (214) den spezifischen aktiven Block zufällig auswählt.
8. System (200) gemäß Anspruch 6, dadurch gekennzeichnet, dass das Wahlmodul (214) den spezifischen aktiven Block entsprechend der im aktiven Block gespeicherten Datenmenge auswählt.

9. System (200) gemäß Anspruch 4, ferner gekennzeichnet durch ein Abfragemodul (212), um abzufragen, ob es einen freien aktiven Block im Flash-Speicher (30) gibt.
10. System (200) gemäß Anspruch 9, ferner gekennzeichnet durch ein Abbildungsmodul (215), um den freien aktiven Block auf den spezifischen physikalischen Block abzubilden.
11. System (200) gemäß Anspruch 1, ferner gekennzeichnet durch:
 - ein Empfangsmodul (202) zum Empfangen einer Leseanfrage, wobei die Leseanfrage eine virtuelle Seitennummer aufweist;
 - ein Suchmodul (204) für einen physikalischen Block, zum Ermitteln einer virtuellen Blockadresse entsprechend der virtuellen Seitennummer, um einen spezifischen physikalischen Block aufzufinden;
 - ein Entscheidungsmodul (206) zur Ermittlung, ob der spezifische physikalische Block auf einen aktiven Block abgebildet wurde;
 - ein Suchmodul (208) zum Suchen des abgebildeten aktiven Blocks, um eine benötigte physikalische Seite zu ermitteln;
 - ein Offsetwert-Suchmodul (210) zum Ermitteln eines virtuellen Seiten-Offsetwerts entsprechend der virtuellen Seitennummer, um die physikalische Seite aufzufinden, die den gleichen Offsetwert wie der virtuelle Seiten-Offsetwert besitzt; und
 - ein Lesemodul (220) zum Lesen der Daten der physikalischen Seite.
12. Verfahren zum Datenzugriff in einem Flash-Speicher (30), das in einer elektronischen Vorrichtung verwendet wird, wobei die elektronische Vorrichtung aufweist:
 - mindestens einen virtuellen Block,
 - wobei jeder virtuelle Block eine Mehrzahl von virtuellen Seiten und eine virtuelle Blockadresse aufweist,

wobei jede virtuelle Seite einen virtuellen Seiten-Offsetwert besitzt; und mindestens einen physikalischen Block im Flash-Speicher (30), wobei jeder physikalische Block eine Mehrzahl von physikalischen Seiten und eine physikalische Blockadresse aufweist, wobei jede physikalische Seite einen physikalischen Seiten-Offsetwert besitzt;

gekennzeichnet durch:

mindestens einen aktiven Block im Flash-Speicher (30),

wobei jeder aktive Block eine Mehrzahl von physikalischen Seiten und eine aktive Blockadresse aufweist,

wobei jede physikalische Seite einen physikalischen Seiten-Offsetwert besitzt, jeder virtuelle Block auf einen physikalischen Block abgebildet wird, jeder aktive Block auf einen physikalischen Block abgebildet wird, jede virtuelle Seite im entsprechenden virtuellen und physikalischen Block auf eine physikalische Seite abgebildet wird, der virtuelle Seiten-Offsetwert in den entsprechenden virtuellen und physikalischen Seiten gleich dem physikalischen Seiten-Offsetwert ist;

und das Verfahren aufweist:

Empfangen einer Zugriffsanfrage, die eine virtuelle Seitennummer aufweist, die eine virtuelle Seite repräsentiert; Auffinden eines spezifischen physikalischen Blocks gemäß der virtuellen Seitennummer; und Ermitteln, ob der spezifische physikalische Block auf einen aktiven Block abgebildet wurde, wobei, wenn das Verfahren verwendet wird, um Daten in den Flash-Speicher (30) zu schreiben, das Verfahren ferner aufweist:

(a) Empfangen einer Schreibanfrage, wobei die

Schreibanfrage eine virtuelle Seitennummer aufweist, die eine spezifische virtuelle Seite repräsentiert;

(b) Auffinden eines spezifischen physikalischen Blocks entsprechend der virtuellen Seitennummer;

- (c) Ermitteln, ob der spezifische physikalische Block auf einen aktiven Block abgebildet wurde; falls ja, dann
Ausführen von Schritt (d) und falls nein, Ausführen von Schritt (e);
- (d) Ermitteln, ob der abgebildete aktive Block eine leere physikalische Seite besitzt; falls ja, Ausführen von Schritt (f) und falls nein, Ausführen von Schritt (g);
- (e) Verwenden der virtuellen Seitennummer, um einen virtuellen Seiten-Offsetwert zu ermitteln und zu ermitteln, ob eine physikalische Seite, die einen gleichen Offsetwert wie der virtuellen Seiten-Offsetwert besitzt, eine leere physikalische Seite im spezifischen physikalischen Block ist; falls ja, Ausführen von Schritt (h) und falls nein, Ausführen von Schritt (i);
- (f) Schreiben von Daten auf die leere physikalische Seite;
- (g) Mischen des spezifischen physikalischen Blocks und seinem entsprechenden abgebildeten aktiven Block;
- (h) Schreiben von Daten auf die physikalische Seite; und
- (i) Mischen eines physikalischen Blocks und seinem entsprechenden abgebildeten aktiven Block.

13. Verfahren gemäß Anspruch 12, ferner gekennzeichnet durch:

Auswählen eines physikalischen Blocks, der auf einen aktiven Block vor Ausführung von Schritt (i) abgebildet wurde.

14. Verfahren gemäß Anspruch 12, dadurch gekennzeichnet, dass Schritt (g) ferner aufweist:

nach Mischen des spezifischen physikalischen Blocks und seinem entsprechenden abgebildeten aktiven Block, Erzeugen eines neuen physikalischen Blocks und Abbilden des neuen physikalischen Blocks auf einen virtuellen Block, auf den der spezifische physikalische Block abgebildet wurde.

- 15.** Verfahren gemäß Anspruch 14, dadurch gekennzeichnet, dass das Verfahren nach Schritt (g) ferner die Erzeugung eines neuen aktiven Blocks aufweist, der auf den neuen physikalischen Block abgebildet wurde und ein leerer aktiver Block ist.
- 16.** Verfahren gemäß Anspruch 12, dadurch gekennzeichnet, dass Schritt (i) ferner aufweist: nach Mischen eines spezifischen physikalischen Blocks und seinem entsprechenden abgebildeten aktiven Block, Erzeugen eines neuen physikalischen Blocks und Abbilden des neuen physikalischen Blocks auf einen virtuellen Block, auf den der spezifische physikalische Block abgebildet wurde.
- 17.** Verfahren gemäß Anspruch 16, dadurch gekennzeichnet, dass nach Schritt (i) das Verfahren ferner das Erzeugen eines neuen aktiven Blocks aufweist, der auf den spezifischen physikalischen Block abgebildet wurde und ein leerer aktiver Block ist.
- 18.** Verfahren gemäß Anspruch 12, dadurch gekennzeichnet, dass vor Schritt (i) das Verfahren ferner eine Abfrage aufweist, ob es einen leeren aktiven Block gibt.
- 19.** Verfahren gemäß Anspruch 18, dadurch gekennzeichnet, dass der leere aktive Block auf den spezifischen physikalischen Block abgebildet wird, wenn der leere aktive Block existiert.
- 20.** Verfahren gemäß Anspruch 12, dadurch gekennzeichnet, dass das Verfahren verwendet wird, um Daten vom Flash-Speicher zu lesen und ferner aufweist:

 - (a) Empfangen einer Leseanfrage, wobei die Leseanfrage eine virtuelle Seitennummer aufweist, die eine virtuelle Seite repräsentiert;

- (b) Ermitteln einer virtuellen Blockadresse gemäß der virtuellen Seitennummer, um einen spezifischen physikalischen Block aufzufinden;
- (c) Ermitteln, ob der spezifische physikalische Block auf einen aktiven Block abgebildet wurde;
und falls ja, Ausführen von Schritt (d) und falls nein, Ausführen von Schritt (e);
- (d) Suchen nach dem abgebildeten aktiven Block, um eine benötigte physikalische Seite aufzufinden; und
- (e) Ermitteln eines virtuellen Seiten-Offsetwerts gemäß der virtuellen Seitennummer, um den spezifischen physikalischen Block für eine physikalische Seite, die einen physikalischen Seiten-Offsetwert besitzt, der identisch zum virtuellen Seiten-Offsetwert ist, zu suchen.

Dem Patentbegehren liegt die **Aufgabe** zugrunde, ein System und Verfahren vorzusehen, das imstande ist, Daten sequentiell in einen Flash-Speicher zu schreiben, um die Update-Effizienz zu begünstigen (siehe geltende Beschreibung Seite 3a).

II.

Die Beschwerde wurde rechtzeitig eingelegt und ist auch sonst zulässig. Sie hat Erfolg, da das geltende Patentbegehren durch den bekannt gewordenen Stand der Technik nicht vorweggenommen oder nahegelegt ist und auch die übrigen Kriterien für eine Patenterteilung erfüllt sind (PatG §§ 1 bis 5, § 34).

1. Die vorliegende Patentanmeldung betrifft ein System und ein Verfahren, mit dem Daten sequentiell in einen Flash-Speicher geschrieben werden können.

Flash-Speicher sind im Vergleich zu magnetischen Speichern relativ günstig und haben einen geringen Energieverbrauch. Sie haben aber den Nachteil, dass

Daten nicht einfach in einen Block neu geschrieben werden können, sondern der Block zuerst gelöscht werden muss, und dass die Anzahl der Lösch- bzw. Schreibzyklen eines Blocks und somit dessen Lebensdauer begrenzt ist.

Ein Flash-Speicher ist in eine Vielzahl von physikalischen (Speicher-) Blöcken unterteilt. Jeder dieser physikalischen Blöcke ist über eine physikalische Blockadresse ansprechbar und enthält eine Mehrzahl von physikalischen Seiten, die einen physikalischen Seiten-Offsetwert besitzen.

Das vorliegende Patentbegehren führt verschiedene Ausgestaltungen an, die aus dem Stand der Technik zu entnehmen sind.

Eine erste Ausführung verwendet ein FFS (Flash-File-System), das mit einer Abbildungstabelle die virtuellen Blöcke auf die physikalischen Blöcke abbildet (vgl. OS [0025] - [0027], Fig.1 und Fig.2). Da in dieser Ausführung die Abbildung blockweise geschieht, werden die Seiten innerhalb eines Blocks hintereinander abgebildet, wobei eine virtuelle Seite mit einem virtuellen Seiten-Offsetwert auf eine physikalische Seite abgebildet wird, die den gleichen Seiten-Offsetwert besitzt. Die Abbildungstabelle müsste jedoch ständig angepasst werden, damit schlechte Blöcke nicht beschrieben werden und ein zu häufiges Beschreiben der gleichen Blöcke verhindert wird.

In der nächsten Ausführung (vgl. [0028] - [0032], Fig.3 und Fig.4) wird beschrieben, wie über die Beziehung in einer Abbildungstabelle zuerst eine Blockadresse und anschließend eine Seite innerhalb des Blocks mit einem spezifischen physikalischen Seiten-Offsetwert ermittelt wird, der dem virtuellen Seiten-Offsetwert entspricht. Dieses Vorgehen benötigt für die Abbildungstabelle nur einen geringen Speicherplatz. Für ein sequentielles Schreiben von Daten, wie es bei den neuen Flash-Speichern nötig ist, wird hier lediglich eine niedrige Aktualisierungseffizienz erreicht. Denn für ein sequentielles Schreiben von Daten ist es nötig, dass in einen physikalischen Block eine Seite mit einem kleinen physikalischen Seiten-Offset-

wert vor eine Seite mit einem größeren Seiten-Offsetwert geschrieben werden muss. Wird beim Schreiben der Daten einer Seite mit einem bestimmten physikalischen Seiten-Offsetwert festgestellt, dass in dem Block bereits eine Seite mit einem größeren physikalischen Seiten-Offsetwert vorhanden ist, so müssen alle Daten des Blocks mit den Seiten, die einen kleineren Seiten-Offsetwert aufweisen zuerst in einen neuen Block kopiert werden, anschließend wird die Seite mit dem bestimmten Seiten-Offsetwert eingefügt und schließlich alle Seiten des Block mit einem höheren Seiten-Offsetwert angefügt.

Ein weiterer, aus dem Stand der Technik bekannter, Ansatz zum sequentiellen Schreiben von Daten wird ebenso beschrieben (vgl. [0033] - [0035] und Fig.5). Dabei soll eine Seite mit einem bestimmten physikalischen Seiten-Offsetwert in einem Block aktualisiert werden. Dazu durchsucht das System den gesamten Block nach einer leeren Seite und schreibt die zu aktualisierenden Daten in die erste gefundene leere Seite. Findet das System keine leere Seite in dem Block, so werden die weiteren Blöcke durchsucht, bis eine leere Seite gefunden wird, in die die zu aktualisierenden Daten geschrieben werden. Damit ist aber die Beziehung zwischen den virtuellen Seiten-Offsetwerten und den physikalischen Seiten-Offsetwerten nicht mehr gegeben. Dies würde bedeuten, dass bei der Suche nach einer Seite mit einem spezifischen Seiten-Offsetwert alle Seiten durchsucht werden müssen, was die Leseeffizienz erheblich verschlechtert. Zur Vermeidung dieses Problems wird in der Abbildungstabelle ein weiterer Datenbereich eingeführt, in dem diese Beziehungen hinterlegt sind. Für die Aktualisierung ist bei dieser Ausführung die Verschiebung einer großen Datenmenge nötig, was zu einer schlechten Aktualisierungseffizienz und zu einem vermehrten Auftreten von schlechten Blocks führt.

Davon ausgehend wird in der vorliegenden Anmeldung (insbes. Anspruch 1) ein Flash-Speicher mit einer Mehrzahl von physikalischen Blöcken und einer spezifischen Anzahl von aktiven Blöcken vorgestellt. Dabei besitzt jede physikalische Seite der Mehrzahl von Blöcken und die zugehörige virtuelle Seite den gleichen

Offsetwert, jedoch kann die physikalische Seite eines aktiven Blocks einen anderen Offsetwert als die zugeordnete virtuelle Seite aufweisen. In dieser Zuordnung wird sichergestellt, dass die Vorschrift für das sequentielle Schreiben, kleinerer Seiten-Offsetwert vor größerem Seiten-Offsetwert, in den physikalischen und in den aktiven Blöcken eingehalten wird.

Zum Erreichen dieses Ziels wird eine erste Abbildungstabelle verwendet, die eine eindeutige Beziehung zwischen den physikalischen und den virtuellen Blocks herstellt. Mit einer zweiten Abbildungstabelle wird eine Beziehung zwischen einem virtuellen Block und einem aktiven Block angegeben. In einer dritten Abbildungstabelle ist die Beziehung zwischen den virtuellen Seiten eines virtuellen Blocks und den physikalischen Seiten des aktiven Blocks hinterlegt, um somit auch Seiten mit unterschiedlichen Offsetwerten adressieren zu können.

Im Gegensatz zu den aus dem Stand der Technik bekannten Lösungen, bei denen maximal zwei Tabellen benutzt werden, wird durch die Verwendung der drei Tabellen das Schreiben und Lesen von Daten, sowie das Aktualisieren des Speichers, wesentlich beschleunigt.

Als **Fachmann**, der mit der Aufgabe betraut wird, das sequentielle Schreiben von Daten in einen Flash-Speicher zu verbessern, ist ein Diplom-Ingenieur der Elektrotechnik oder ein Physiker mit Hochschulausbildung und mehrjähriger Berufserfahrung im Bereich der Entwicklung von Flash-Speichern anzusehen.

2. Der Erteilungsantrag verlässt nicht den Rahmen der ursprünglichen Offenbarung.

Der neue Anspruch 1 entspricht den ursprünglichen Ansprüchen 1 bis 4, ergänzt auf Basis der ursprünglichen Beschreibung S.7 und S.13. Die Ansprüche 2 bis 4 und 5 bis 11 entsprechen den ursprünglichen Ansprüchen 5 bis 7 und 9 bis 15. Der neue Anspruch 12 entspricht den ursprünglichen Ansprüchen 16 und 17 und

die Ansprüche 13 bis 20 entsprechen den ursprünglichen Ansprüchen 18 bis 25. Zusätzlich wurden in den Ansprüchen noch redaktionelle Änderungen vorgenommen.

Die Beschreibung wurde redaktionell überarbeitet und der Stand der Technik wurde aufgenommen.

3. Die Patentansprüche geben nunmehr klar und deutlich an, was durch sie unter Schutz gestellt werden soll.

4. Der Gegenstand des geltenden Patentanspruchs 1 ist durch den bekannt gewordenen Stand der Technik weder vorweggenommen noch nahegelegt.

4.1 Das System zum sequentiellen Schreiben von Daten in einen Flash-Speicher ist neu.

Im Laufe des Verfahrens wurden folgende Druckschriften berücksichtigt:

- D1:** DE 198 45 295 A1
- D2:** US 2003 / 0101327 A1
- D3:** EP 1 244 019 A2
- D4:** US 6 678 785 B2
- D5:** US 2003 / 0070035 A1
- D6:** DE 197 82 041 T1
- D7:** EP 0 712 067 A2
- D8:** US 5 473 569 A

D1 zeigt ein Zugriffsverfahren für Flash-Speicher, mit dem die Suchzeiten verkürzt werden. Hierzu wird der Speicherbereich in Blöcke unterteilt. Der Speicher wird mit einer logischen Adresse angesprochen, die auf einen Operationsblock verweist, dessen Verweistabelle geladen wird. Anschließend wird überprüft, ob in der

Verweistabelle eine entsprechende logische Adresse vorhanden ist und falls dies zutrifft, kann der physikalische Speicherbereich angesprochen werden und die Daten können sequentiell in den Speicher geschrieben werden. Dabei fehlt in der **D1** die Verwendung von drei Abbildungstabellen, da die Verweistabelle eine Unterteilung des Operationsblocks darstellt und somit keine weitere Abbildungstabelle gemäß der vorliegenden Anmeldung darstellt. Auch ist keine Verwendung eines aktiven Blocks und auch keine Verwendung eines Seiten-Offsetwertes angegeben.

D2 beschreibt eine Zugriffssteuerung für einen Flash-Speicher, bei der die physikalischen Adressen mit einer Zuordnungstabelle aus den logischen Adressen ermittelt werden. Zusätzlich wird eine Correction-Table verwendet, die, bspw. bei einer Verschiebung der Daten in andere physikalische Speicherbereiche, die Änderung der Zuordnung speichert. Die in der Correction-Table gespeicherten Zuordnungen werden anschließend auf die Zuordnungstabelle übertragen. Dies entspricht einem Aktualisieren der Zuordnungstabelle. Besonderes Augenmerk wird dabei in der **D2** auf die Wiederherstellung nach einem Stromausfall bzw. nach einem Systemabsturz gelegt. Hier ist es mit Hilfe der Correction-Table möglich auf die gespeicherten Daten auch nach einem unvorhergesehen Ereignis zugreifen zu können, selbst wenn die Zuordnungstabelle noch nicht aktualisiert ist. In **D2** ist keine zweite und dritte Abbildungstabelle im Sinne der vorliegenden Anmeldung und auch keine Verwendung einer Tabelle mit aktiven Blocks gezeigt. Ebenso beschreibt die **D2** keine Verwendung von Seiten-Offsetwerten.

Aus **D3** ist ein Verfahren zum Zugriff auf einen Flash-Speicher zu entnehmen, bei dem eine erste Zuordnungstabelle für die Adressumsetzung und eine weitere Tabelle zur Anzeige von sog. Write-Units verwendet wird. Ein Hinweis diese zweite Tabelle im Sinne einer Zuordnungstabelle für aktive Blöcke zu benutzen ist nicht gegeben. Auch macht die **D3** keine Aussagen zu einer dritten Tabelle und zu den Seiten-Offsetwerten.

Auch in **D4** ist die Adressierung eines Flash-Speichers mit einer Übersetzungstabelle gezeigt. Dazu erfolgt nach einem Schreibbefehl eine Überprüfung, ob ein freier Bereich vorliegt und abhängig von dieser Abfrage das Schreiben in den ausgewählten Bereich oder in einen neuen (freien) Bereich. Weiterhin ist in **D4** eine Übertragung von vorhandenen Daten in einen freien Bereich und das anschließende Löschen des vorigen Bereichs erläutert. Aus **D4** sind keine drei Abbildungstabellen und auch keine aktiven Blöcke zu entnehmen.

D5 beschreibt einen Zugriff auf einen Flash-Speicher, wobei in jedem physikalischen Block des Speichers ein Overhead-Bereich bzw. ein Offset-Bereich vorhanden ist. Dieser Bereich des Speichers ist reserviert für ein BIOS Programm und kann somit nicht als aktiver Block oder als Seiten-Offsetwert benutzt werden. Zusätzlich ist auch keine zweite und dritte Zuordnungstabelle gezeigt.

D6 behandelt die Verwendung eines Flash-Speichers analog zu einer Festplatte. Die dazu notwendige Steuerung ist speziell für die Umsetzung der Zugriffsbefehle einer Festplatte auf die Speicherbereiche eines Flash-Speichers ausgebildet. Auch in **D6** sind keine Hinweise auf drei Abbildungstabellen, auf aktive Blöcke und auf Seiten-Offsetwerte vorhanden.

Aus **D7** ist ein Verfahren zu entnehmen, bei dem unter Angabe von Zylinder, Header und Sektor eine Adresse in eine physikalische Speicheradresse eines Flash-Speichers umgesetzt wird. Die **D7** zeigt dabei auch nicht die Verwendung von aktiven Blöcken, Speicher-Offsetwerten und drei Abbildungstabellen.

Die Lehre der **D8** betrifft das Verwalten der Speicherbereiche eines Flash-Speichers, wenn Daten in einen Speicherbereich geschrieben werden sollen. Hierzu wird zuerst überprüft, ob genügend Speicherplatz in einem bereits benutzten Bereich zur Verfügung steht und falls dies nicht der Fall ist, werden die vorhandenen Daten in einen neuen Bereich kopiert und die neuen Daten angehängt. Anschließend kann der vorherige Bereich gelöscht werden. Aus **D8** sind ebenso

keine drei Abbildungstabellen, Seiten-Offsetwerte oder aktive Blöcke gemäß der vorliegenden Anmeldung zu entnehmen.

Somit lehrt keine der bekannt gewordenen Druckschriften das Konzept eines „aktiven Blocks“ und speziell die Verwendung einer dritten Abbildungstabelle, deren Abbildungsdaten einen virtuellen Seiten-Offsetwert und einen physikalischen Seiten-Offsetwert aufweisen, um damit eine spezifische virtuelle Seite auf eine spezifische physikalische Seite eines spezifischen aktiven Blocks abzubilden (Merkmal (1.7)).

4.2 Das System gemäß Anspruch 1 beruht auf einer erfinderischen Tätigkeit.

Als nächstkommenden Stand der Technik sieht der Senat die Druckschrift **D4** an. Dieser Druckschrift liegt ebenso das Problem zugrunde die Vorschriften bei einem sequentiellen Schreiben einzuhalten. Wie oben erläutert, ist in **D4** die Adressierung eines Flash-Speichers mit einer Übersetzungstabelle gezeigt, wobei bei einem Schreibbefehl eine Überprüfung, ob ein freier Bereich vorliegt, stattfindet. Anschließend erfolgt, abhängig vom Ergebnis der Abfrage, das Schreiben in den ausgewählten Bereich oder in einen neuen (freien) Bereich. Darüber hinaus ist in **D4** die Aktualisierung beschrieben, indem vorhandene Daten in einen freien Bereich übertragen werden und anschließend der vorher belegte Bereich gelöscht wird.

Für den Fachmann wären, ausgehend von **D4**, mehrere gedankliche Schritte notwendig, um zum Gegenstand der Erfindung zu gelangen. Denn zuerst müssten in die Abbildungsvorschrift die Bedingungen hinsichtlich der Seiten-Offsetwerte und der aktiven Blöcke gemäß den Merkmalen (1.3 und 1.5) aufgenommen werden. Anschließend wäre eine dritte Abbildungstabelle zu implementieren und es müssten aktive Blöcke festgelegt werden (entsprechend den Merkmalen (1.5 und 1.7)).

Dass diese Abfolge von Schritten nahegelegen hätte, kann nicht festgestellt werden. Hinzu kommt, dass es für die in Anspruch 1 angegebene Lösung, welche sich erst bei rückschauender Betrachtungsweise erschließt, auch alternative Lösungen gibt, die zu ganz anderen Ergebnissen geführt hätten (vgl. BGH GRUR 2006, 930 „Mikrotom“, Leitsatz b). Darüber hinaus ist auch aus dem Stand der Technik kein Anlass für den Fachmann ersichtlich, gerade in der beanspruchten Richtung vorzugehen (vgl. BGH GRUR 2010, 407 „Einteilige Öse“).

Das beanspruchte System gilt somit als auf einer erfinderischen Tätigkeit beruhend (PatG § 4 Satz 1).

5. Ebenso konnte der Stand der Technik keine Anregungen liefern, ein Verfahren zum Datenzugriff in einem Flash-Speicher gemäß dem geltenden nebengeordneten Anspruch 12 auszubilden.

6. Die Unteransprüche 2 bis 11 und 13 bis 20 betreffen nicht selbstverständliche Ausgestaltungen des beanspruchten Systems bzw. des beanspruchten Verfahrens und sind in Verbindung mit den Ansprüchen 1 und 12 ebenfalls gewährbar.

Dr. Morawek

Eder

Baumgardt

Hoffmann

Fa