(19) Deutsches Patent- und Markenamt

(10) **DE 100 85 013 B3** 2017.11.30

(12) # Patentschrift

(21) Deutsches Aktenzeichen: **100 85 013.8**
(86) PCT-Aktenzeichen: **PCT/US00/24848**
(87) PCT-Veröffentlichungs-Nr.: **WO 2001/022209**
(86) PCT-Anmeldetag: **11.09.2000**
(87) PCT-Veröffentlichungstag: **29.03.2001**
(45) Veröffentlichungstag der Patenterteilung: **30.11.2017**

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

Innerhalb von neun Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

(30) Unionspriorität:
09/400,570 **21.09.1999** **US**

(73) Patentinhaber:
Intel Corporation, Santa Clara, Calif., US

(74) Vertreter:
ZENZ Patentanwälte Partnerschaft mbB, 45128 Essen, DE

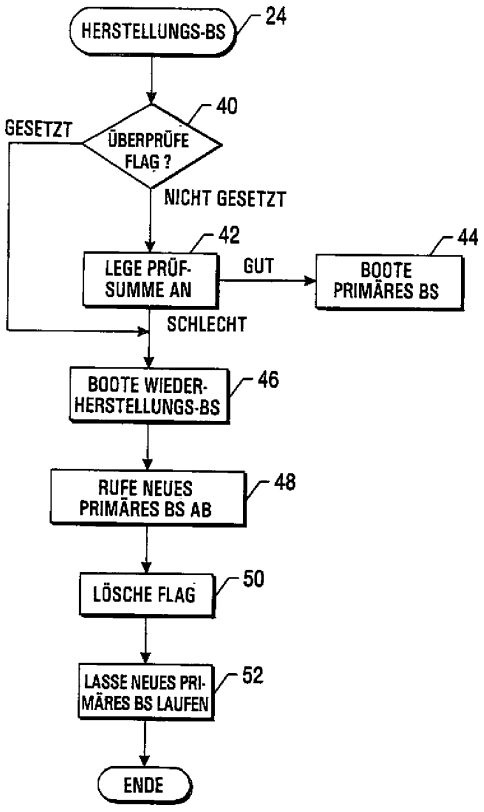
(72) Erfinder:
Rhoads, Edward R., Sherwood, Oreg., US;
Ketrenos, James P., Austin, Tex., US

(56) Ermittelter Stand der Technik:

US	**56 57 445**	**A**
US	**57 01 492**	**A**
US	**57 87 491**	**A**
US	**59 44 820**	**A**

(54) Bezeichnung: **Anordnen von in nicht-flüchtigen wiederprogrammierbaren Halbleiterspeichern gespeicherten Informationen**

(57) Hauptanspruch: Verfahren zum Organisieren gespeicherter Informationen in einem nicht-flüchtigen, wiederprogrammierbaren Halbleiterspeicher (14), wobei:
der Speicher (14) in eine Mehrzahl von Partitionen aufgeteilt wird, wobei jede Partition eine definierte Adresse hat;
ein erster Bootlader für ein Wiederherstellungsbetriebssystem an einer ersten Adresse und in einer ersten Partition (20) gespeichert wird, wobei das Wiederherstellungsbetriebssystem für das Aktualisieren eines primären Betriebssystems und/oder das Erlangen eines Ersatzes für das primäre Betriebssystem verantwortlich ist, wobei das Wiederherstellungsbetriebssystem ein Kernel (26) aufweist, das auf nur denjenigen Befehlscode reduziert ist, der zum Implementieren der Wiederherstellungs- und Aktualisierungsfunktionen erforderlich ist;
ein zweiter Bootlader (102) für ein primäres Betriebssystem an einer zweiten Adresse und in einer zweiten Partition gespeichert wird; und
die Adressen für die erste Bootlader-Partition und für die zweite Bootlader-Partition (102) in einer weiteren Partition (100) gespeichert werden.



Beschreibung

[0001] Diese Erfindung bezieht sich auf ein Verfahren zum Organisieren gespeicherter Informationen in einem nicht-flüchtigen, wiederprogrammierbaren Halbleiterspeicher, wobei der Speicher in eine Mehrzahl von Partitionen aufgeteilt wird, wobei jede Partition eine definierte Adresse hat.

[0002] Es besteht ein wachsendes Interesse an sogenannten eingebetteten prozessorbasierten Systeme. Diese Systeme arbeiten oftmals mit verringerten Funktionalitäten, um die gewünschte Leistung bei relativ geringen Kosten zur Verfügung zu stellen. In vielen Fällen können diese eingebetteten Systeme batteriebetrieben sein. Somit sind ihre Fähigkeiten eingeschränkt, um die Batterielebensdauer zu erhöhen.

[0003] Aus einer Vielzahl von Gründen, die das Bewahren der Batterielebensdauer, das Verringern der Kosten und das Bereitstellen eines kompakten Formfaktors einschließen, können prozessorbasierte Systeme zur Verfügung gestellt werden, welche als ihr nicht-flüchtiges Speichermedium kein Festplattenlaufwerk benutzen. Bei vielen prozessorbasierten Systemen stellt ein Festplattenlaufwerk ein bequemes nicht-flüchtiges Speichermedium zur Verfügung, das die meisten Informationen speichert, die der Benutzer permanent zu halten wünscht. Diese können neben anderen Dingen das Betriebssystem, Anwendungssoftware, Dateien und Daten als Beispiele einschließen. Die Informationen, die in dem Festplattenlaufwerk gespeichert sind, können zur Ausführung in den Systemspeicher übertragen werden, welcher herkömmlicherweise ein flüchtiger Speicher ist.

[0004] Bei vielen Systemen stellen Festplattenlaufwerke ein relativ schnelles Speichermedium sehr hoher Kapazität zur Verfügung. Jedoch benötigen Festplattenlaufwerke mehr Raum und verwenden mehr Energie als nicht-flüchtige Halbleiterspeicher. Bei vielen eingebetteten Systemen werden wiederprogrammierbare nicht-flüchtige Halbleiterspeicher als das primäre Speichersystem für prozessorbasierte Systeme benutzt. Diese Halbleiterspeicher speichern die vollständige Informationsausrüstung, die normalerweise in Festplattenlaufwerken gespeichert ist, einschließlich der Betriebssysteme.

[0005] In vielen Fällen verwenden die als primäres nicht-flüchtiges Speichermedium für prozessorbasierte Systeme benutzten Halbleiterspeicher FLASH-Speicher. Diese FLASH-Speicher können ohne Benutzereingriff unter Verwendung gut bekannter platineneigener Fähigkeiten neu programmiert werden. Auf diese Speicher wird im allgemeinen unter Verwendung von Zeilen- und Spaltenadressen zugegriffen. So sind die Speicher grundsätzlich monolithisch, wobei der Ort der Dateien und der anderen Daten in diesem Speicher grundsätzlich außerhalb des Speichers gespeichert wird.

[0006] Während diese Systeme bei relativ einfachen eingebetteten prozessorbasierten Systemen gut arbeiteten, kann in dem Maße, wie sich die Anforderungen an die prozessorbasierten Systeme erhöhen, dieses einfache Speichersystem unzureichend sein, um einige gewünschte Funktionen zu handhaben. Somit besteht ein Bedürfnis an einem verbesserten Weg der Verwendung nicht-flüchtiger wiederprogrammierbarer Halbleiterspeicher als primäres nicht-flüchtiges Dateisystem für prozessorbasierte Systeme.

[0007] Aus der Patentschrift US 5,701,492 ist eine Schnittstellenplatine zum Ankoppeln eines Druckers an ein LAN bekannt, die einen Mikroprozessor, einen flüchtigen Speicher (DRAM) und einen wiederprogrammierbaren Halbleiterspeicher (Flash-Speicher) aufweist. Der Flash-Speicher enthält einen Boot-Block-Bereich, einen Datei-Bereich und einen Directory-Bereich. Der Boot-Block-Bereich enthält zumindest zwei verschiedene Versionen eines Boot-Blocks, jeweils mit einem Startsektor. Jeder Startsektor enthält den Code, der zum Bestimmen der aktuellen Directory, zum Prüfen eines Boot-Block-Select-Feldes in der aktuellen Directory zum Bestimmen des aktuellen Boot-Blocks, zum Kopieren des aktuellen Boot-Blocks in den obersten Bereich des DRAMs (Shadow-DRAM) und zum Ausführen des Rests des Boot-Blocks aus dem DRAM heraus erforderlich ist. Der Directory-Bereich speichert mindestens zwei Directories mit jeweils einem Header und 84 Dateieinträgen. Der Header enthält ein Boot-Block-Select-Feld, das bestimmt, welcher der Boot-Blöcke in den Shadow-DRAM kopiert werden soll. Jeder Dateieintrag enthält einen 3-Buchstaben-Dateinamen, ein Flag und einen Zeiger zum Startort der Datei im Datei-Bereich. Der Dateiname MON bezeichnet ein Monitorprogramm. Zum Aktualisieren des Boot-Blocks und der Dateien des Monitorprogramms wird zunächst ein neuer Boot-Block in den Flash-EPROM in den ungenutzten Boot-Block-Bereich, der nicht der Bereich des aktuellen Boot-Blocks ist, gespeichert. Danach werden neue kritische Dateien in den Flash-EPROM gespeichert. Dann wird eine entsprechende neue Directory erstellt und flash-gespeichert. Nach dem Neu-Booten wird der neu flash-gespeicherte Boot-Block der aktuelle Boot-Block und werden die neuen kritischen Dateien das aktuelle Monitorprogramm.

[0008] Aus der Patentschrift US 5,944,820 ist ein Verfahren zum Bereitstellen eines modifizierbaren Partition-Boot-Records bekannt. Ein Speicher, auf den durch einen Prozessor zugegriffen werden kann, enthält zumindest zwei Partitionen, von denen die erste anfangs aktiv und die zweite anfangs inaktiv ist. Die inaktive Partition enthält Software, die nur dann

als aktiv gekennzeichnet wird, wenn ein Benutzer eine Software-Lizenz akzeptiert hat.

[0009] Aufgabe der Erfindung ist es ausgehend von dem eingangs genannten Stand der Technik die Wiederherstellung und Aktualisierung eines Betriebssystems an die Verwendung eines Flash-Speichers anzupassen und flexibel zu gestalten.

[0010] Diese Aufgabe wird erfindungsgemäß durch ein Verfahren zum Organisieren gespeicherter Informationen in einem nicht-flüchtigen, wiederprogrammierbaren Halbleiterspeicher mit den Merkmalen des Anspruchs 1 gelöst.

[0011] Vorteilhafte und/oder bevorzugte Ausführungsformen der Erfindung sind in den Unteransprüchen gekennzeichnet.

[0012] Weitere Aspekte werden in der nachfolgenden detaillierten Beschreibung angegeben.

[0013] **Fig.** 1 ist eine schematische Darstellung eines Client/Server-Systems gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

[0014] **Fig.** 2 ist ein Schema der Speicherarchitektur der in **Fig.** 1 gezeigten Speichereinrichtung;

[0015] **Fig.** 3 ist ein Schema einer Speicherarchitektur eines BIOS und Wiederherstellungsbetriebssystems, die bei dem in **Fig.** 2 gezeigten System verwendet wird;

[0016] **Fig.** 4 ist ein Ablaufdiagramm zum Implementieren von Software zum Wieder-Laden von Betriebssystemen;

[0017] **Fig.** 5 ist eine Darstellung einer Speicherarchitektur für das in **Fig.** 2 gezeigte primäre Betriebssystem;

[0018] **Fig.** 6 ist eine Hardwareimplementierung des in **Fig.** 1 gezeigten Clients; und

[0019] **Fig.** 7 ist ein Ablaufdiagramm, das den Betrieb der in **Fig.** 5 gezeigten FLAT veranschaulicht.

[0020] Ein Client/Server-Computersystem **10**, das in **Fig.** 1 gezeigt ist, kann einen oder mehrere Server **18** enthalten, die über ein Netzwerk **16** mit einem oder mehreren Clients **12** gekoppelt sein können. Jeder Client **12** kann eine Speichereinrichtung **14** aufweisen. Der Client **12** kann ein prozessor-basiertes System, wie beispielsweise ein Desktop-Computersystem, ein Handheld-Computersystem, ein prozessorbasiertes Fernsehsystem, eine Set-Top-Box, ein Gerät, ein Thin-Client, ein Funktelefon oder dergleichen sein. Das Netzwerk **16** kann irgendeines einer Vielzahl von Netzwerken sein, einschließlich eines lokalen Netzwerks (LAN), eines Stadtnetzwerks (MAN), eines Weitbereichsnetzwerks (WAN), eines drahtlosen Netzwerks, eines Heimnetzwerks oder eines internationalen Netzwerks, wie beispielsweise dem Internet.

[0021] Bei dem System **10** kann der Client **12** permanent sein Betriebssystem auf einer wiederprogrammierbaren Speichereinrichtung **14** speichern. Die Speichereinrichtung **14** kann in herkömmlicher Weise ein Festplattenlaufwerk oder ein Flash-Speicher sein. Wenn das Betriebssystem zerstört ist oder aktualisiert werden muss, kann der Client **10** auf das Netzwerk **16** und den Server **18** zugreifen, um ein ungestörtes oder aktualisiertes Betriebssystem zu erlangen und das neue Betriebssystem automatisch auf die Speichereinrichtung **14** neu laden.

[0022] Die Speichereinrichtung **14** kann elektrisch umprogrammierbar sein. Die Speichereinrichtung **14** kann darüber hinaus bei einem Ausführungsbeispiel der Erfindung als BIOS-Speicher für den Client **12** dienen. Während herkömmlicherweise der BIOS-Speicher ein Nur-Lese-Speicher (ROM) ist, kann durch Verwendung eines wiederprogrammierbaren Speichers das Betriebssystem ebenso wie der BIOS-Befehlscode aktualisiert oder ersetzt werden, wenn er zerstört ist, wie im folgenden erläutert wird. Bei anderen Ausführungsbeispielen kann ein herkömmlicher BIOS-ROM zusätzlich zu der Speichereinrichtung **14** verwendet werden.

[0023] Es sind eine Vielzahl von Flash-Speichern zum Implementieren der Speichereinrichtung **14** verfügbar, wie beispielsweise Intels Speicher der Marke StrataFlash™. Ein vorteilhafter Speicher ist das 28F64OJ5-Acht-Megabyte-FLASH-Array, das von der Intel Corporation erhältlich ist. Dieser Speicher enthält mehrere 128-Kilobyte-Blöcke. Jeder Block kann datengeschützt sein, so dass er nicht gelöscht oder überschrieben werden kann. Mit anderen Worten, der Datenschutz kann selektiv auf einen oder mehrere einer Mehrzahl von Blöcken in dem Speicher angewendet werden.

[0024] Das BIOS kann in einem oder mehreren datengeschützten Blöcken in dem Flash-Speicher gespeichert sein. In gleicher Weise kann das Wiederherstellungsbetriebssystem in einem oder in mehreren Blöcken so gespeichert sein, dass es ebenfalls datengeschützt ist. Bei einem Ausführungsbeispiel kann das BIOS in zwei 128 Kilobyte-Blöcken gespeichert sein, und das Wiederherstellungsbetriebssystem kann zwei 128-Kilobyte-Blöcke verwenden.

[0025] Es wird jetzt auf **Fig.** 2 Bezug genommen; die Speicherarchitektur der Speichereinrichtung **14** kann adressierbare Orte für ein BIOS und ein Wiederherstellungsbetriebssystem **20** und ein primäres Betriebssystem **22** enthalten. Das primäre Betriebs-

system kann ein Mehrzweckbetriebssystem, wie beispielsweise Microsoft Windows® 98 oder CE, LINUX oder das BE-Betriebssystem beispielsweise sein. Das primäre Betriebssystem kann auch ein Echtzeit-Betriebssystem (RTOS), wie beispielsweise das PalmOS sein. Das BIOS und das Wiederherstellungsbetriebssystem **20** arbeiten in Fällen, in denen das primäre Betriebssystem **22** zerstört ist oder aktualisiert werden muss. Das Wiederherstellungsbetriebssystem **20** kann ein Betriebssystem einer verringerten Größe sein, welches grundlegende wesentliche BIOS-Funktionen und die zum Gewinnen des neuen primären Betriebssystems erforderliche eingeschränkte Software enthält. So ist ein "Wiederherstellungsbetriebssystem" in diesem Sinne ein Betriebssystem, das zum Aktualisieren und/oder Gewinnen eines Ersatzes für ein primäres Betriebssystem verantwortlich ist.

[0026] Gemäß **Fig.** 3 enthält bei einem Ausführungsbeispiel der Erfindung das Wiederherstellungsbetriebssystem **20** ein Kernel **26**, Netzwerkschnittstellensteuereinrichtungs(NIC)-Treiber **30** und einen Netzwerkstapel **28**. Das Kernel **26** ist der Kern des Wiederherstellungsbetriebssystems **20**. Der Stapel **28** beispielsweise kann das User Datagram Protocol/Internet Protocol (UDP/IP), das Trivial File Transfer Protocol (TFTP), das Dynamic Host Control Protocol (DHCP), das Address Resolution Protocol (ARP) und das Anfangsladeprotokoll (BOOTP – Bootstrap Protocol) enthalten. (Diese Protokolle finden sich bei www.ietf.org/rfc.html.). Das Wiederherstellungsbetriebssystem **20** kann darüber hinaus die Betriebssystem-Wiederherstell- und -Aktualisierungs-Anwendungssoftware **24** enthalten. Ein FLASH-Treiber **34** und BIOS-Dienste **35** können ebenfalls enthalten sein. Der FLASH-Treiber wird verwendet, um ein neues primäres Betriebssystem in den FLASH-Speicher einzuschreiben, wenn ein FLASH-Speicher als Speichereinrichtung **14** verwendet wird. Die Hardwareschnittstelle **36** bildet eine Schnittstelle zwischen den Softwareschichten und einer Hardwaremutterplatine.

[0027] Idealerweise kann das Wiederherstellungsbetriebssystem **22** so viel wie möglich abgerüstet werden, um Speicher einzusparen. Wenn möglich, kann das Kernel **26** auf nur denjenigen Befehlscode reduziert sein, welcher erforderlich ist, um seine Wiederherstellungs- und Aktualisierungsfunktionen zu implementieren. Ein Kernel, welches besonders gut anwendbar ist, ist das LINUX-Kernel. Das LINUX-Kernel enthält eine X-basierte Kernelkonfigurationsutility, die make xconfig genannt wird. Diese Utility schafft eine graphische Benutzeroberfläche, um das Auswählen der Elemente des Kernels und des Betriebssystems zu erleichtern. Das heißt, das LINUX-Betriebssystem ermöglicht es dem Benutzer, auf eine Reihe von Fragen zu antworten, die über eine graphische Benutzerschnittstelle gestellt werden, um anzuzeigen, ob bestimmte Funktionalitäten erwünscht sind. Der Code für nicht ausgewählte Funktionalitäten könnte dann ausgeschlossen werden. Im Ergebnis kann ein relativ abgerüstetes Betriebssystem auf einfache Weise ohne Zugriff auf Objektcode entwickelt werden.

[0028] Im Falle irgendwelcher Softwarefehler oder Abstürze kann das System neu laden, wodurch der Fehler aufgelöst wird. Ein Überwachungszeitgeber in dem CMOS-Speicher verfolgt eine Zahl der erfolglos versuchten Neuladevorgänge. Wenn diese Zahl einen Schwellenwert überschreitet (beispielsweise 3), könnte das Wiederherstellungsbetriebssystem aufgerufen werden. Wenn das System versucht, neu zu laden, überprüft es die CMOS-Speicher-Neuanfangslade-Anzahl und lädt das Wiederherstellungsbetriebssystem automatisch, sofern der Wiederladezahlschwellenwert überschritten ist. Das Wiederherstellungsbetriebssystem **20** wird gestartet, so dass eine neue Version des Abbilds des primären Betriebssystems abgerufen werden kann.

[0029] Das Wiederherstellungsbetriebssystem **20** kann darüber hinaus Betriebssystem-Updates erlangen. Dies kann auf eine Reihe von Wegen geschehen. Bei einem Ausführungsbeispiel könnte der Benutzer eine Aktualisierung anfordern, wodurch ein separates Update-Bit in dem CMOS-Speicher gesetzt wird. Bei einem anderen Ausführungsbeispiel könnte ein Betriebssystemanbieter an seine Benutzer eine Nachricht ausstrahlen (broadcast), welche anzeigt, dass ein Update verfügbar ist. Die Benutzersysteme, die die Nachricht empfangen, können ihr eigenes Update-Bit haben, das automatisch in dem CMOS-Speicher gesetzt wird. Bei dem nächsten versuchten Anfangsladen (boot), wird das Wiederherstellungsbetriebssystem gebootet, um automatisch das Update zu erwerben.

[0030] Alternativ könnte die Wiederherstellungs- und Update-Anwendungssoftware **24** so konfiguriert werden, dass das Update automatisch zu einer Zeit einer vorhergesagten niedrigen Benutzung erworben wird. Wenn beispielsweise das System erfasst, dass das Update-Bit gesetzt ist, was anzeigt, dass ein Update erwünscht wird, könnte das System bis Mitternacht warten, um automatisch das Update herunterzuladen.

[0031] Das Wiederherstellungsbetriebssystem wiederum kann über die Netzwerkschnittstellensteuereinrichtung und das Netzwerk **16** kommunizieren, um eine neue Version des primären Betriebssystemabbilds abzurufen. Dies könnte ausgeführt werden, indem auf eine weitere Einrichtung in demselben Netzwerk zugegriffen wird oder bei einem anderen Beispiel, indem auf das gewünschte Betriebssystem über das Internet zugegriffen wird.

[0032] Nachdem das neue Betriebssystem im Systemspeicher überprüft worden ist und in den Speicher **14** geladen worden ist, wird das System neu gebootet. Wenn das System das primäre Betriebssystem lädt, setzt das primäre Betriebssystem das Update-Bit in dem CMOS-Speicher zurück.

[0033] In einigen Fällen, wenn ein Booten versucht wird, könnte eine Analyse des gespeicherten Betriebssystems feststellen, dass das Betriebssystem zerstört ist. Beispielsweise könnte während des Bootens eine Prüfsummenanalyse vorgenommen werden. Wenn das gespeicherte Betriebssystem gestört ist, könnte ein Wiederherstellungsbit in dem CMOS-Speicher gesetzt und das Anfangsladen (boot) abgebrochen werden. Das nächste Mal, wenn ein Boot versucht wird, wird das Wiederherstellungsbit identifiziert, und das System lädt das Wiederherstellungsbetriebssystem.

[0034] Es wird jetzt auf **Fig.** 4 Bezug genommen; die Wiederherstellungs- und -Aktualisierungsanwendungssoftware **24** beginnt mit dem Überprüfen der Speichereinrichtung **14**, wie es in dem Rhombus **40** gezeigt ist. Beim Einschalten, nach dem Durchlaufen des Einschaltselbsttests (POST), überprüft der Startcode das primäre Betriebssystemabbild in dem Speicher **14** nach Prüfsummenfehlern. Wenn es einen Fehler gibt, bootet das System das Wiederherstellungsbetriebssystem **20** und startet die Wiederherstellungsanwendung. Ein Fehlercode kann auftreten, weil das Betriebssystemabbild zerstört oder eines der Wiederherstellungs- oder Aktualisierungs-Flags gesetzt ist. Das Wiederherstellungs-Flag kann beispielsweise wegen eines Defekts in dem Betriebssystem gesetzt sein. Die Aktualisierungs-Flags können beispielsweise gesetzt sein, weil eine Zeitdauer für ein altes primäres Betriebssystem abgelaufen ist oder weil der Benutzer einen Wunsch zur Erlangung eines Upgrades angezeigt hat. So wird nach dem Anlegen der Prüfsumme, wie es im Block **42** angezeigt ist, das primäre Betriebssystem gebootet, wie es im Block **44** angezeigt ist, sofern die Prüfsumme ein gültiges Betriebssystem anzeigt. Anderenfalls wird das Wiederherstellungsbetriebssystem gebootet, wie es in Block **46** angezeigt ist.

[0035] Während der Boot-Routine setzt ggf. ein Startbefehlscode, welcher Teil des BIOS ist, das Wiederherstellungsbit in dem CMOS-Speicher. Der Startbefehlscode kann darüber hinaus den Befehlscode zum Zählen, wie oft ein Neu-Booten versucht worden ist, und zum Speichern von Informationen über die Anzahl der versuchten Neu-Boot-Vorgänge enthalten.

[0036] Bei einem Ausführungsbeispiel der vorliegenden Erfindung könnte die Anwendung **24** eine Anforderung über das Netzwerk an den Server **18** für ein Betriebssystemherunterladen (Block **48**) initiieren. Sobald das neue Abbild heruntergeladen ist, wird es in die Speichereinrichtung **14** geschrieben. Dann wird das Wiederherstellungsbit gelöscht, wie es im Block **50** angezeigt ist, und das System bootet erneut, wie es im Block **55** gezeigt ist. Beim nächsten Mal bootet das System in das primäre Betriebssystem und führt seine üblichen Funktionen aus.

[0037] Die Speicherarchitektur eines Abschnitts der Speichereinrichtung **14**, die das primäre Betriebssystem **22** speichert, das in **Fig.** 5 gezeigt ist, weist an der untersten Speicheradresse ein Prüfsummen- oder Zyklisches-Redundanzprüf(CRC)-Feld **96** auf. Über dem Prüfsummenfeld **96** befindet sich ein Feld **98**, welches die Anzahl der Einträge in einer Flash-Zuweisungstabelle (FLAT – FLASH Allocation Table) **100** anzeigt. Die FLASH-Zuweisungstabelle unterteilt (partitioniert) den FLASH-Speicherabschnitt **22** und gestattet, dass mehrere Befehlscode- und Datenabbilder in der Speichereinrichtung **14** gespeichert werden können. Dies wiederum ermöglicht, dass mehrere Anfangslader (Boot-Lader) in dem Flash-Speicher zum Booten verschiedener Betriebssystemabbilder vorhanden sind. Zur Anfangsladezeit wählt das BIOS auf der Grundlage des Status des Wiederherstellungsbits, wie es oben beschrieben wurde, aus, welcher Anfangslader zu laden und auszuführen ist.

[0038] Der Anfangslader **102** zum Laden des primären Betriebssystems ist über der Flash-Zuweisungstabelle **100** gespeichert. Über dem Anfangslader **102** befindet sich das Kernel **104** bzw. der Kern des primären Betriebssystems. Das Kernel des primären Betriebssystems kann dem durch das Wiederherstellungsbetriebssystem benutzten Kernel gleich sein oder von diesem abweichen. Während beispielsweise LINUX für das Wiederherstellungsbetriebssystem verwendet werden kann, könnte bei einem Ausführungsbeispiel Windows® CE als primäres Betriebssystem benutzt werden.

[0039] Über dem Kernel **104** befindet sich ein Dateisystem **106**. Die FLASH-Zuweisungstabelle **100** enthält jeweils einen Eintrag für jedes in dem FLASH-Speicherabschnitt **22** gespeicherte Item, einschließlich der in dem Dateisystem **106** gespeicherten Items (Einzelheiten). Das Dateisystem **106** enthält Dateien, Verzeichnisse und zum Lokalisieren und zum Zugreifen auf Betriebssystemdateien und Verzeichnisse verwendete Informationen.

[0040] Jedes in der FLASH-Zuweisungstabelle enthaltene Item enthält Informationen über die Softwareversion, die Flags, die Daten-Offsets, die Länge der Daten und ihre Ladeadresse. Die Versionsnummer verfolgt lediglich, welche Version der Software in einem bestimmten Speicher **14** geladen wurde. Das Datenoffset legt fest, wo in dem FLASH-Speicher ein Eintrag angeordnet ist.

[0041] Das Flag-Feld hat Informationen über die Art der zugehörigen Einträge. Das am geringsten bewertete Bit des Flag-Felds enthält Informationen über den Status der zyklischen Redundanzüberprüfung (CRC). Dies teilt im Endeffekt dem BIOS mit, ob eine CRC berechnet werden muss. Das nächsthöher bewertete Bit enthält den Blocktyp. Der Blocktyp schließt "Boot" ein, was einen Anfangslader anzeigt, "Kernel" oder "Dateisystem". Sofern der Blocktyp Anfangslader ist, sagt dieses Flag-Feld, wohin in den Speicher mit wahlfreiem Zugriff der Anfangslader aus dem FLASH-Speicher zu laden ist. Ein zusätzlicher Bereich in dem FLASH-Feld kann für weitere Informationen reserviert sein. Ein Boot-Lader oder Anfangslader lädt weitere Laderprogramme, welche ein Betriebssystem laden und übergibt die Kontrolle an diese.

[0042] Während die vorliegende Erfindung in Verbindung mit einer Vielzahl von prozessorbasierten Systemen verwendet werden kann, ist eine Anwendung, welche ein Set-Top-Computersystem verwendet, in **Fig.** 6 veranschaulicht. Ein Set-Top-Computersystem arbeitet mit einem Fernsehempfänger zusammen. Der Client **12** kann einen Prozessor **65** enthalten, der mit einem Chipsatz **66** eines beschleunigten Graphikports (AGP) gekoppelt ist. Die Accelerated Graphics Port Spezifikation, Rev. 2.0 ist von der Intel Corporation aus Santa Clara, Kalifornien, erhältlich. Der Chipsatz **66** kann mit dem Systemspeicher **68** und dem beschleunigten Graphikport-Bus **70** gekoppelt sein. Der Bus **70** wiederum kann mit einem Graphikbeschleuniger **72**, der außerdem mit einem Video- oder Fernsehempfänger **73** gekoppelt ist, gekoppelt sein.

[0043] Ein Abschnitt **75** des Systemspeichers **68**, der CMOS-Speicher genannt wird, kann durch eine integrierte Speicherschaltung implementiert sein, welche an das Sichern von Systemdaten angepasst ist. Herkömmlicherweise enthält der CMOS die Echtzeituhr (RTC), welche die Tageszeit verfolgt. Die Wiederherstellungs- und Aktualisierungsbits sind in dem CMOS-Speicher an vorgegebenen Stellen gespeichert.

[0044] Der Chipsatz **66** kann darüber hinaus mit einem Bus **74** gekoppelt sein, der eine Fernsehtuner/ Aufnahme-Karte **76** aufnimmt. Die Karte **76** kann mit einer Fernsehantenne **78** gekoppelt sein, welche auch eine Satellitenantenne oder Kabelverbindung sein kann, um zusätzliche Beispiele anzugeben. Eine Schnittstelle zu einem Netzwerk **16**, wie beispielsweise eine Modemschnittstellenverbindung zu dem Internet oder eine Netzwerkschnittstellensteuereinrichtungsverbindung zu einem Computernetzwerk, kann ebenfalls vorgesehen sein.

[0045] Eine Brücke **80** kann wiederum mit einem weiteren Bus **84** gekoppelt sein, welcher eine serielle Eingabe/Ausgabe-Schnittstelle **86** und eine Speicherschnittstelle **94** unterstützt. Die Schnittstelle **86** kann mit einem Modem **88** oder einer Tastatur **92** gekoppelt sein. Die Schnittstelle **94** kann den FLASH-Speicher **14**, der das Wiederherstellungsbetriebssystem und das BIOS **20** und das primäre Betriebssystem **22** speichert, ankoppeln. Die Brücke **80** kann der 82371AB-PCI-ISA-IDE-Xcelerator(PIIX4)-Chipsatz sein, der von der Intel Corporation erhältlich ist. So kann er Mehrzweck-Eingabe/ Ausgabe-Pins (GP[I,O]) enthalten.

[0046] Bei der Anzahl der zum Implementieren von Computersystemen verwendeten Chipsätze kann der Chipsatz derart eingerichtet sein, dass er jeweils nur eine bestimmte Anzahl von Zeilen des BIOS sieht. Bei Ausführungsbeispielen, bei denen das primäre Betriebssystem und das Wiederherstellungsbetriebssystem in dem FLASH-Speicher gespeichert sind, kann auf diese auf dieselbe Weise zugegriffen werden, wie auf den BIOS-Speicher zugegriffen wird. Da der FLASH-Speicher, auf den zugegriffen wird, beträchtlich größer ist als ein BIOS-Speicher, kann es somit wünschenswert sein, andere Techniken zu verwenden, um auf sämtliche Speicher in dem FLASH zuzugreifen. Eine Technik, um dies bei Prozessoren der Intel Corporation auszuführen, besteht darin, die GP[I,O]-Pins zu verwenden, beispielsweise an der PIIX4-Einrichtung. Diese Pins können mit den zum Entwickeln der das BIOS lesenden Signale verantwortlichen Pins gekoppelt sein. Indem geeignete GP [I,O]-Signale zur Verfügung gestellt werden, kann das FLASH-Speicher-Lesen bankgeschaltet sein, um sequentiell den gesamten Speicher zu lesen.

[0047] Wenden wir uns jetzt **Fig.** 7 zu; in Übereinstimmung mit einem Ausführungsbeispiel beginnt die Software, die die FLAT verwendet, um zu ermöglichen, dass mehrere Code- und Datenabbilder in dem FLASH-Speicher gespeichert werden, beim Einschalten oder Systemrücksetzen mit der BIOS-Ausführung und der Durchführung der Systeminitialisierung und der Einschaltselbsttestaktivitäten (Block **110**). Die Inhalte des FLASH-Speichers können gültig gemacht werden, indem der im Feld **96** in dem FLASH-Speicher gespeicherte CRC überprüft wird, wie es im Block **112** angezeigt ist. An diesem Punkt wählt das BIOS den Anfangslader zum Ausführen aus (Block **114**), indem die FLAT durchsucht wird und der als Anfangslader markierte Eintrag ausgewählt wird. Der Anfangslader verwendet dann die FLAT, um herauszufinden, wo in dem FLASH-Speicher das primäre Betriebssystem angeordnet ist (Block **116**), lädt das Betriebssystem an der richtigen Adresse in dem Systemspeicher (Block **118**) und startet seine Ausführung (Block **120**).

[0048] Bei einigen Ausführungsbeispiel könnte das BIOS weiterhin vom Betriebssystem unabhängig sein. Die Betriebssystemabhängigkeiten können sich

in dem Anfangslader aufhalten. Der Anfangslader ermöglicht es einem herkömmlichen Computerbetriebssystem, sich in dem FLASH-Speicher aufzuhalten.

[0049] Während die vorliegende Erfindung in Verbindung mit einem Ausführungsbeispiel, bei dem das primäre Betriebssystem und das Wiederherstellungsbetriebssystem in einer Speichereinrichtung, wie beispielsweise einem FLASH-Speicher, gespeichert sind, veranschaulicht worden ist, können andere wiederprogrammierbare Speichereinrichtungen ebensogut benutzt werden. Im Falle des FLASH-Speichers ist unter den gegebenen aktuellen ökonomischen Bedingungen der Speicher relativ teuer und eine Spiegelung wird grundsätzlich nicht verwendet. Somit ist die Verwendung des Wiederherstellungsbetriebssystems in Verbindung mit FLASH-Speichern besonders vorteilhaft. Jedoch könnte die vorliegende Erfindung in Verbindung mit anderen Konfigurationen benutzt werden. Bei Systemen beispielsweise, die das primäre Betriebssystem in einem Festplattenlaufwerk speichern, könnte sich das Wiederherstellungsbetriebssystem ebenfalls auf dem Festplattenlaufwerk befinden. Das BIOS könnte in solchen Fällen, sofern es gewünscht wird, weiterhin in einem BIOS-ROM gespeichert sein.

[0050] Alternativ könnte das Wiederherstellungsbetriebssystem aktuell auf einem externen oder entnehmbaren Speicher, wie beispielsweise einer CompactDisc-ROM (CD-ROM) zur Verfügung gestellt werden. Wenn es erforderlich ist, könnte der Benutzer einfach die CD-ROM in einen CD-Player laden. Ein Prozessor führt das Wiederherstellungsbetriebssystem von der CD-ROM herunter aus und verwendet dann die Wiederherstellungs- und Aktualisierungs-Anwendungssoftware, um das primäre Betriebssystem zu aktualisieren und zu ersetzen. Diese Lösung bietet Vorteile gegenüber dem Bereitstellen des vollständigen Betriebssystems in Plattenform, da die Verwendung eines kompakten Wiederherstellungsbetriebssystems die Aktualisierungen erleichtert. Das heißt, das kompakte Wiederherstellungssystem könnte schnell geladen werden und verwendet werden, um Updates zu erwerben. Anderenfalls müsste das vollständige Betriebssystem in Plattenform für jeden Benutzer für jede Aktualisierung (Update) zur Verfügung gestellt werden, so dass der Benutzer dann die Updates erwerben kann.

[0051] Während die vorliegende Erfindung unter Bezugnahme auf eine Client/Server-Umgebung beschrieben worden ist, ist die vorliegende Erfindung zusätzlich für eine Vielzahl weiterer Umgebungen verfügbar. Beispielsweise könnte die vorliegende Erfindung auf einem Server in einer Client/Server-Umgebung implementiert sein. Zusätzlich ist sie auf Stand-Alone-Computersysteme einschließlich prozessorbasierter Systeme, die batteriegestützt sind, anwendbar. In Verbindung mit Hand-held-Computersystemen beispielsweise könnte die vorliegende Erfindung eine Update- oder Ersetzungsfunktionalität unter Verwendung verfügbarer verdrahteter oder drahtloser Kommunikationsverbindungen zur Verfügung stellen. Bei einem System, welches vorübergehend mit einem Desktop-Computer über eine feste Verdrahtung verknüpft sein kann, wie beispielsweise ein persönlicher digitaler Assistent Palm-Pilot, könnte das Wiederherstellungsbetriebssystem mit dem Desktop kommunizieren, um ein neues Betriebssystem zu erlangen. In ähnlicher Weise könnten Upgrades unter Verwendung einer Vielzahl drahtloser Kommunikationsverbindungen einschließlich Radio- und Funktelefonverbindungen, gewonnen werden. Bei Systemen, welche über Kabel- oder Satellitenrundfunksysteme miteinander verknüpft sind, könnten neue Betriebssysteme darüber hinaus unter Verwendung dieser Kommunikationsverbindungen ebensogut gewonnen werden.

[0052] In Verbindung mit zugeschnittenen Betriebssystemen könnte es erforderlich sein, zu einem speziellen fernen Ort zu gehen, um das Betriebssystem zu aktualisieren oder zu ersetzen. In Verbindung mit nicht auf den Benutzer zugeschnittenen Betriebssystemen jedoch können eine Vielzahl von Sites innerhalb des erweiterten Computersystems des Benutzers, das über das Internet oder über eine Vielzahl von Kommunikationsverbindungen zugreifbar ist, benutzt werden, um solche Ersetzungen zu erwerben. Zusätzlich könnten eine Vielzahl solcher Sites in die Wiederherstellungsbetriebssystemanwendungssoftware vorprogrammiert sein, so dass dann, wenn das System beim Erwerben der erforderlichen Ersetzung bei einem Ort nicht erfolgreich ist, es eine Vielzahl weiterer Orte abfragen kann.

[0053] In einigen Fällen kann die Wiederherstellungsanwendungssoftware nicht mit Informationen über zusätzliche Orte, welche zukünftige Aktualisierungen enthalten, programmiert werden. Wenn jedoch ein Betriebssystemanbieter Informationen über Updates ausstrahlt, könnten diese Ausstrahlungen (broadcasts) auch Informationen darüber enthalten, wie die gewünschten Updates automatisch erwerbbar sind. Diese Informationen können dann von der Wiederherstellungsanwendungssoftware verwendet werden.

[0054] Bei einigen Ausführungsbeispielen beachtet der Systembenutzer die Operation des Wiederherstellungsbetriebssystems überhaupt nicht. Das Wiederherstellungsbetriebssystem arbeitet im Hintergrund, wobei es das primäre Betriebssystem dem Benutzer robuster erscheinen lässt.

Patentansprüche

1. Verfahren zum Organisieren gespeicherter Informationen in einem nicht-flüchtigen, wiederprogrammierbaren Halbleiterspeicher (**14**), wobei:
der Speicher (**14**) in eine Mehrzahl von Partitionen aufgeteilt wird, wobei jede Partition eine definierte Adresse hat;
ein erster Bootlader für ein Wiederherstellungsbetriebssystem an einer ersten Adresse und in einer ersten Partition (**20**) gespeichert wird, wobei das Wiederherstellungsbetriebssystem für das Aktualisieren eines primären Betriebssystems und/oder das Erlangen eines Ersatzes für das primäre Betriebssystem verantwortlich ist, wobei das Wiederherstellungsbetriebssystem ein Kernel (**26**) aufweist, das auf nur denjenigen Befehlscode reduziert ist, der zum Implementieren der Wiederherstellungs- und Aktualisierungsfunktionen erforderlich ist;
ein zweiter Bootlader (**102**) für ein primäres Betriebssystem an einer zweiten Adresse und in einer zweiten Partition gespeichert wird; und
die Adressen für die erste Bootlader-Partition und für die zweite Bootlader-Partition (**102**) in einer weiteren Partition (**100**) gespeichert werden.

2. Verfahren nach Anspruch 1, wobei ferner Informationen über die Anzahl der Partitionen (in **98**) gespeichert werden.

3. Verfahren nach Anspruch 1, wobei ferner ein Dateisystem (**106**) in einer der Partitionen gespeichert wird.

4. Verfahren nach Anspruch 1, wobei ferner ein Kernel für das primäre Betriebssystem (**22**) in einer der Partitionen (**104**) gespeichert wird.

5. Verfahren nach Anspruch 1, wobei ferner in Zuordnung zu den Adressen Informationen darüber, ob eine Integritätsüberprüfung an den an der zugeordneten Adresse gespeicherten Daten ausgeführt werden muss oder nicht, gespeichert werden.

6. Verfahren nach Anspruch 1, wobei ferner in Zuordnung zu der Adresse einer Partition Informationen über die Art der in der Partition gespeicherten Informationen gespeichert werden.

7. Verfahren nach Anspruch 6, wobei ferner Informationen darüber gespeichert werden, ob die bei einer gegebenen Partition gespeicherten Informationen ein Bootlader, ein Kernel oder ein Dateisystem sind.

8. Das Verfahren nach Anspruch 6, wobei ferner Informationen über die Ladeadresse für die Informationen in Zuordnung zu der Adresse gespeichert werden.

9. Artikel, umfassend ein Medium, das Befehle speichert, die bei ihrer Ausführung ein prozessorbasiertes System veranlassen, ein Verfahren nach einem der Ansprüche 1–8 auszuführen.

10. Prozessorbasiertes System (**12**), aufweisend:
einen Prozessor (**65**);
einen mit dem Prozessor (**65**) gekoppelten flüchtigen Speicher (**68**); und
ein mit dem Prozessor (**65**) gekoppelten nichtflüchtigen wiederprogrammierbaren Halbleiterspeicher (**14**),
wobei der Halbleiterspeicher (**14**) in eine Mehrzahl von Partitionen (**20**, **96**–**106**) aufgeteilt ist, von denen jede eine definierte Adresse hat,
wobei ein erster Bootlader für ein Wiederherstellungsbetriebssystem (**20**) an einer ersten Adresse und in einer ersten Partition (**20**) gespeichert ist, wobei das Wiederherstellungsbetriebssystem für das Aktualisieren eines primären Betriebssystems und/oder das Erlangen eines Ersatzes für das primäre Betriebssystem verantwortlich ist, wobei das Wiederherstellungsbetriebssystem ein Kernel (**26**) aufweist, das auf nur denjenigen Befehlscode reduziert ist, der zum Implementieren der Wiederherstellungs- und Aktualisierungsfunktionen erforderlich ist,
wobei ein zweiter Bootlader (**102**) für ein primäres Betriebssystem an einer zweiten Adresse und in einer zweiten Partition (**102**) gespeichert ist; und
wobei die Adressen für die erste und für die zweite Bootlader-Partition in einer weiteren Partition (**100**) gespeichert sind.

11. System nach Anspruch 10, wobei der Halbleiterspeicher (**14**) ein FLASH-Speicher ist.

12. System nach Anspruch 10, wobei eine der Partitionen (**20**) ein Basis-Eingabe/Ausgabe-System (**32**) speichert.

13. System nach Anspruch 10, wobei eine der Partitionen ein Dateisystem (**106**) speichert.

Es folgen 4 Seiten Zeichnungen

Anhängende Zeichnungen

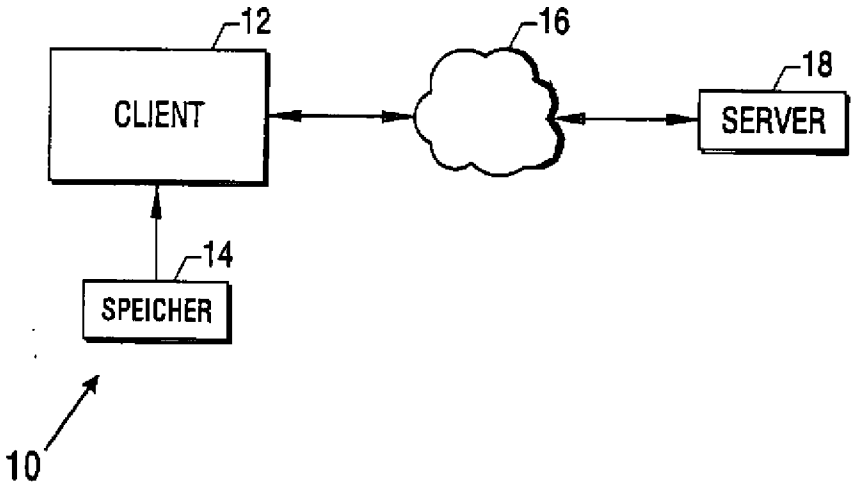


FIG. 1

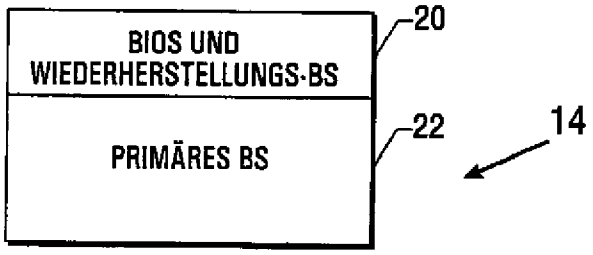


FIG. 2

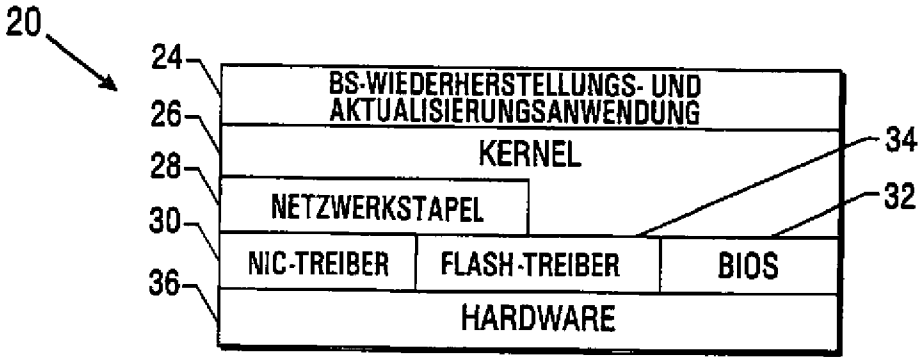


FIG. 3

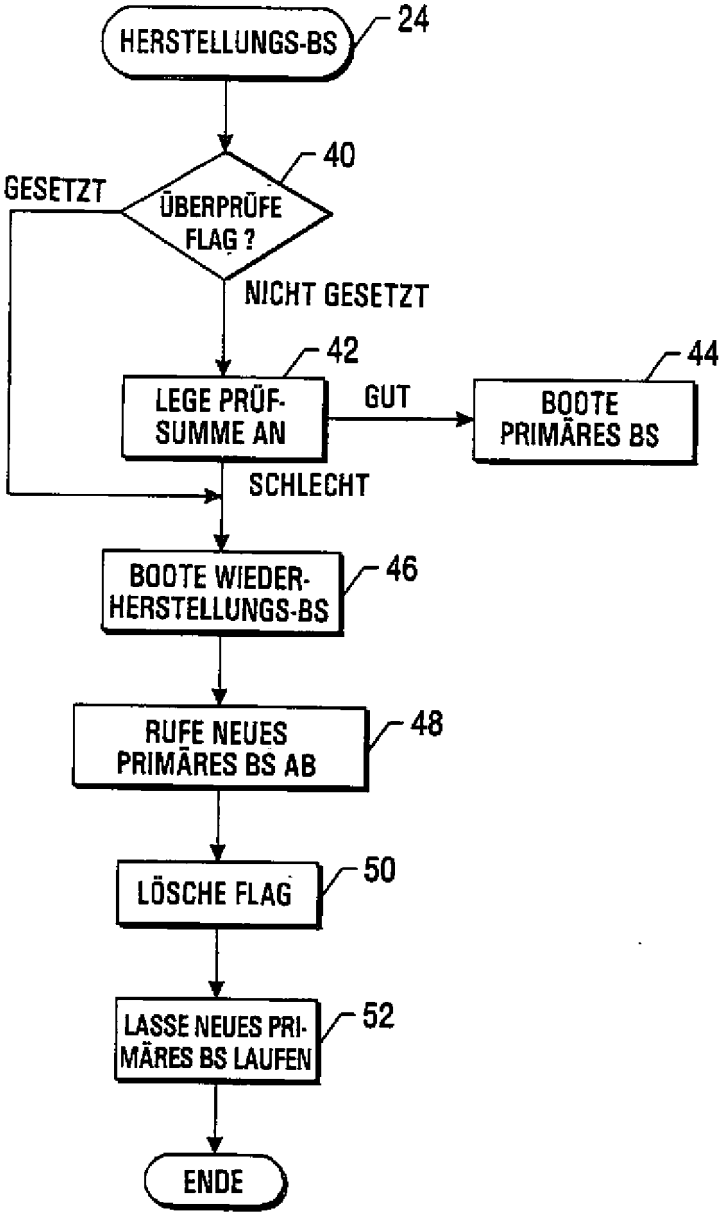

HERSTELLUNGS-BS
24
ÜBERPRÜFE FLAG ?
40
GESETZT
NICHT GESETZT
LEGE PRÜF-SUMME AN
42
GUT
BOOTE PRIMÄRES BS
44
SCHLECHT
BOOTE WIEDER-HERSTELLUNGS-BS
46
RUFE NEUES PRIMÄRES BS AB
48
LÖSCHE FLAG
50
LASSE NEUES PRIMÄRES BS LAUFEN
52
ENDE

FIG. 4

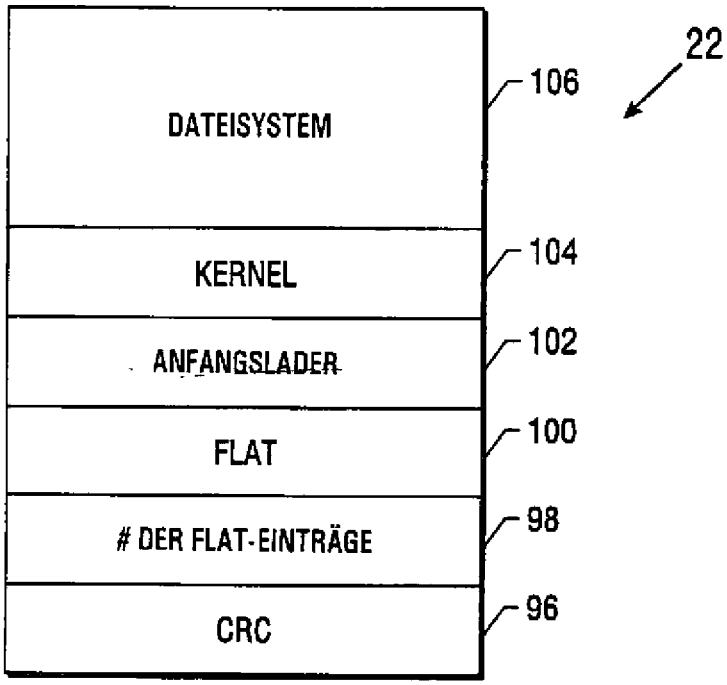

22
DATEISYSTEM
106
KERNEL
104
ANFANGSLADER
102
FLAT
100
DER FLAT-EINTRÄGE
98
CRC
96

FIG. 5

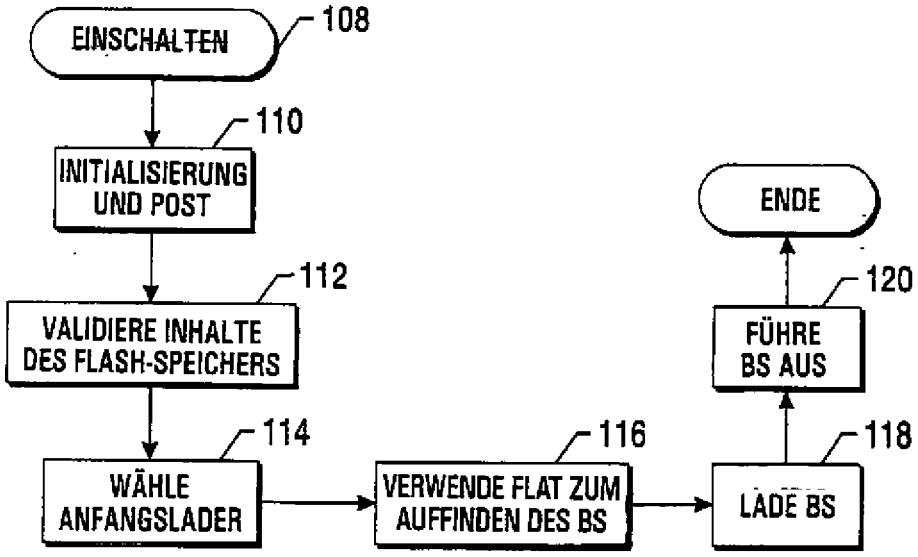

EINSCHALTEN
108
INITIALISIERUNG UND POST
110
VALIDIERE INHALTE DES FLASH-SPEICHERS
112
WÄHLE ANFANGSLADER
114
VERWENDE FLAT ZUM AUFFINDEN DES BS
116
LADE BS
118
FÜHRE BS AUS
120
ENDE

FIG. 7

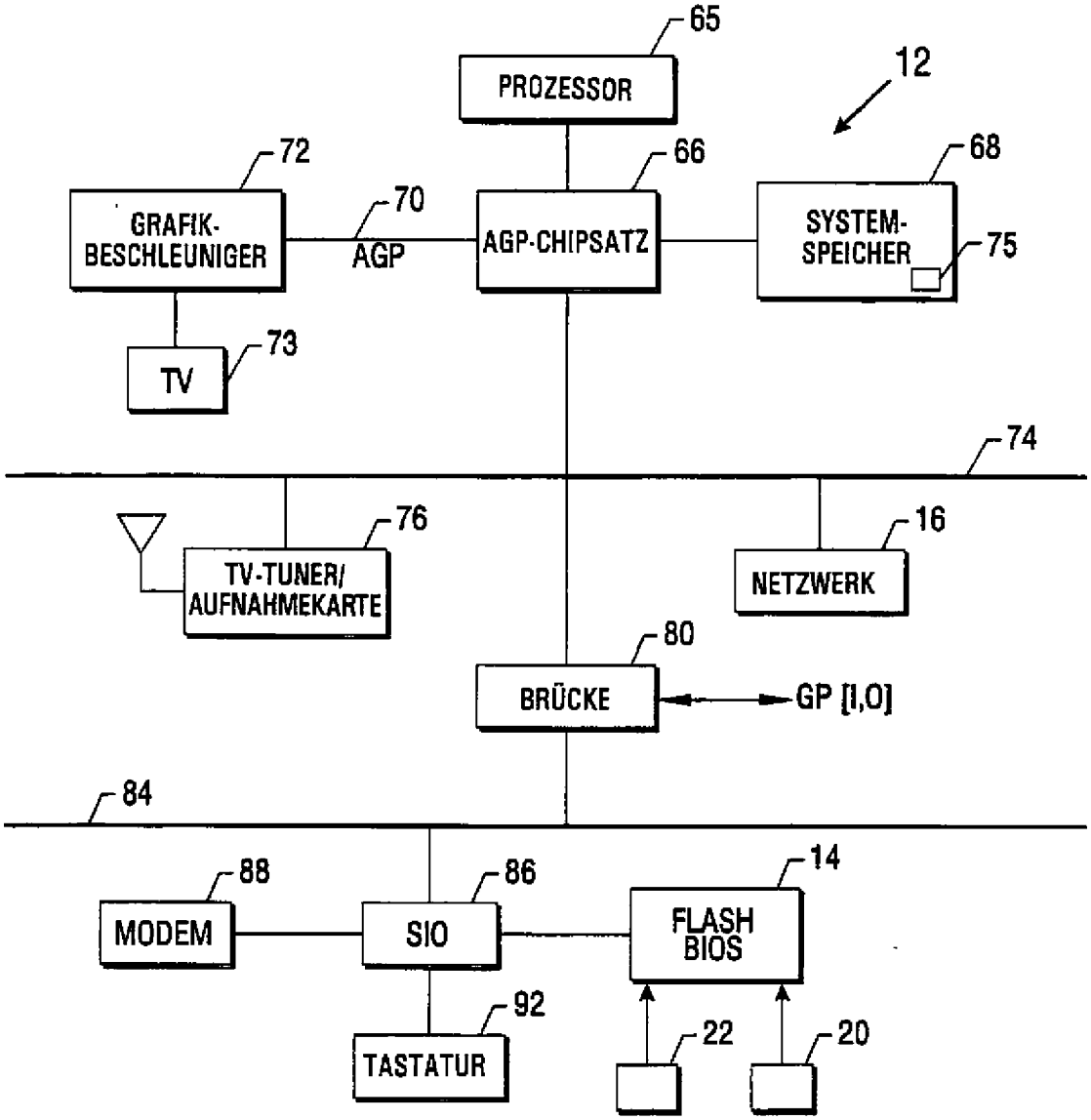

65
PROZESSOR
12
72
66
68
GRAFIK-
BESCHLEUNIGER
70
AGP
AGP-CHIPSATZ
SYSTEM-
SPEICHER
75
73
TV
74
76
16
TV-TUNER/
AUFNAHMEKARTE
NETZWERK
80
BRÜCKE
GP [I,O]
84
88
86
14
MODEM
SIO
FLASH
BIOS
92
22
20
TASTATUR

FIG. 6