



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) **DE 698 14 182 T2** 2004.04.01

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 119 814 B1**

(21) Deutsches Aktenzeichen: **698 14 182.2**

(86) PCT-Aktenzeichen: **PCT/US98/27784**

(96) Europäisches Aktenzeichen: **98 965 007.2**

(87) PCT-Veröffentlichungs-Nr.: **WO 00/10094**

(86) PCT-Anmeldetag: **29.12.1998**

(87) Veröffentlichungstag  
der PCT-Anmeldung: **24.02.2000**

(97) Erstveröffentlichung durch das EPA: **01.08.2001**

(97) Veröffentlichungstag  
der Patenterteilung beim EPA: **02.05.2003**

(47) Veröffentlichungstag im Patentblatt: **01.04.2004**

(51) Int Cl.7: **G06F 13/10**

**G06F 13/12, G06F 9/46, G06F 13/24**

(30) Unionspriorität:

|               |                   |           |
|---------------|-------------------|-----------|
| <b>131446</b> | <b>10.08.1998</b> | <b>US</b> |
| <b>131447</b> | <b>10.08.1998</b> | <b>US</b> |
| <b>131497</b> | <b>10.08.1998</b> | <b>US</b> |
| <b>131922</b> | <b>10.08.1998</b> | <b>US</b> |

(73) Patentinhaber:

**Micron Technology, Inc., Boise, Id., US**

(74) Vertreter:

**FROHWITTER Patent- und Rechtsanwälte, 81679 München**

(84) Benannte Vertragsstaaten:

**AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LI, LU, MC, NL, PT, SE**

(72) Erfinder:

**KLEIN, A., Dean, Eagle, US**

(54) Bezeichnung: **PROZESSOR ODER ZENTRALEINHEIT MIT INTERNEM REGISTER FÜR PERIPHERIEZUSTAND**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

**Beschreibung**

Hintergrund

Gebiet der Erfindung

[0001] Die vorliegende Erfindung bezieht sich Peripheriegeräte in Computersystemen und insbesondere auf ein System mit einem zentralisierten Register, das in einer Kernlogikeinheit oder einem entsprechenden Prozess angeordnet ist, zum Aufrechterhalten von Statusinformation für Peripheriegeräte in einem Computersystem.

Stand der Technik

[0002] Computersysteme weisen typischerweise eine Zentraleinheit auf, die mit mehreren Peripheriegeräten verbunden ist und mit ihnen kommuniziert, was typischerweise über einen Computerebus geschieht. Diese Peripheriegeräte können zum Beispiel folgende sein: Datenspeichergeräte, wie zum Beispiel Plattenlaufwerke und Bandlaufwerke; Dateneingabegeräte, wie zum Beispiel eine Tastatur oder eine Maus; Datenausgabegeräte, wie zum Beispiel ein Bildschirm oder ein Lautsprecher; und Kommunikationsgeräte, wie zum Beispiel ein Netzwerkschnittstellencontroller. Ein Peripheriegerät erfordert oft die Aufmerksamkeit der Zentraleinheit, um Daten zwischen der Zentraleinheit und dem Peripheriegerät zu übertragen oder um sonst das Peripheriegerät zu steuern und zu manipulieren. Diese Aufmerksamkeit wird typischerweise durch eine Unterbrechung ausgelöst, welche das Peripheriegerät an die Zentraleinheit sendet, um die normale Verarbeitung durch die Zentraleinheit zu "unterbrechen". Während einer Unterbrechung hebt die Zentraleinheit die normale Verarbeitung zeitweise auf und führt einen Code aus, der als "Unterbrechungs-Service-routine" bekannt ist, um den erforderlichen Service für das Peripheriegerät auszuführen. Nachdem die Unterbrechungs-Service-routine abgeschlossen ist, fährt die Zentraleinheit mit der normalen Verarbeitung fort.

[0003] Viele Computersysteme verwenden eine gemeinsam genutzte Unterbrechungsarchitektur, bei der mehrere Peripheriegeräte das gleiche Unterbrechungssignal aktivieren können. Eine gebräuchliche gemeinsam genutzte Unterbrechungsarchitektur ist eine verkettete ("daisy-chained") Struktur, bei der Peripheriegeräte über eine oder mehrere Unterbrechungsleitungen miteinander "verkettet" sind. Ein Peripheriegerät in der Kette kann ein Unterbrechungssignal erzeugen, und dieses Unterbrechungssignal wird durch die Kette weitergeleitet, bis es schließlich die Zentraleinheit erreicht. Bei einer weiteren häufig verwendeten gemeinsam genutzten Unterbrechungsarchitektur haben die Peripheriegeräte eine gemeinsame Unterbrechungsbusleitung; Peripheriegeräte können eine Unterbrechung an den Prozessor signalisieren, indem sie sich mit ihrem Signal auf die

ser Unterbrechungsbusleitung durchsetzen.

[0004] Eine gemeinsam genutzte Unterbrechungsarchitektur hat bestimmte Vorteile. Sie ist sehr einfach; typischerweise erfordert sie nur eine kleine Anzahl von Signalleitungen zum Tragen von Unterbrechungssignalen. Sie ist außerdem erweiterbar, wobei es typischerweise möglich ist, dass zusätzliche Peripheriegeräte in ein Computersystem integriert werden, ohne dass zusätzliche Leitungen für Unterbrechungssignale nötig sind.

[0005] Eine gemeinsam genutzte Unterbrechungsarchitektur leidet jedoch unter einem größeren Nachteil. Sie erfordert, dass die Zentraleinheit feststellt, welches Peripheriegerät eine Verarbeitung erfordert. Dies deshalb, weil alle Peripheriegeräte das gleiche Unterbrechungssignal erzeugen, und die Zentraleinheit aufgrund des Unterbrechungssignals nicht feststellen kann, welches Peripheriegerät einer Verarbeitung bedarf. Daher muss die Zentraleinheit typischerweise die Peripheriegeräte abfragen (poll), um festzustellen, welche Peripheriegeräte einer Verarbeitung bedürfen.

[0006] Dieser Abfragevorgang kann sehr zeitaufwendig sein. Es kann sein, dass die Zentraleinheit jedes Peripheriegerät im Computersystem abfragen muss, auch wenn typischerweise nur ein Peripheriegerät zu einer bestimmten Zeit einer Verarbeitung bedarf. Das Abfragen verringert den Wirkungsgrad der Zentraleinheit (CPU), weil die CPU eine Vielzahl von Bustransaktionen durchführen muss, um die Peripheriegeräte abzufragen, und jede Bustransaktion kann in einem Hochleistungscomputersystem eine große Anzahl von GPU-Zyklen erfordern. Das Abfragen belastet auch den Peripheriebus mit einer großen Anzahl von Abfragezugriffen. Außerdem erhöht das Abfragen die Zeit, die für die Verarbeitung einer Unterbrechung erforderlich ist. Dies kann Probleme für Peripheriegeräte hervorrufen, die zu einer bestimmten Zeit einer Verarbeitung bedürfen. Zum Beispiel kann es sein, dass ein Netzwerk-Schnittstellen-Controller sofort einer Verarbeitung bedarf, um zu verhindern, dass ein Puffer mit eintreffenden Daten überfließt. Diese sofortige Verarbeitung kann durch die Abfrage verzögert werden.

[0007] Es besteht daher ein Bedarf nach einem System zum Abfragen von Statusinformation von Peripheriegeräten in einer gemeinsam genutzten Unterbrechungsarchitektur, das die Zeit und Busaktivität verringert, die zum Feststellen des Status der Peripheriegeräte benötigt wird.

[0008] In der WO-A-91/18342 (Star Semiconductor Corp.) ist eine programmierbare integrierte Digital-Signalprozessor-Architektur (SPROC™) beschrieben, welche allgemeine Signalprozessoren analog zu einer Zentraleinheit, an einen Bus angeschlossene Statusregister (Warteflagregister), an den Bus angeschlossene serielle Ports analog zu Peripheriegeräten, einen durch einen Flag erzeugenden Decoder überwachten Datenflussmanager aufweist. Die Bits des Warte-Flag-Registers können durch über ei-

nen internen Bus gesendete Daten gesetzt, d. h. geschrieben, werden. Außerdem sind die Warte-Flag-Register an einen Flagbus angeschlossen, der jedes Mal beschrieben wird, wenn vorbestimmte Speicherplätze im Speicher adressiert werden. Auf diese Weise kann jedes Bit in den Registern selektiv gelöscht werden. Wenn alle Bits aufgrund des Auftretens bestimmter Ereignisse gelöscht wurden, wird ein ODER-Gatter zum Vorsehen eines Statusflags verwendet. Der Prozessor kann dann feststellen, dass die zum Durchführen einer Operation erforderlichen Daten dem Prozessor zur Verfügung stehen, worauf dieser aus einer Warteschleife ausklinkt. Adressen werden dadurch erzeugt, dass in verschiedenen anderen Registern Werte hinzugefügt werden, so zum Beispiel Offset, Dekrement, Loop, Basisadresse und Rahmenadressenregister.

### Zusammenfassung

[0009] Die vorliegende Erfindung sieht ein Verfahren zum Verwalten von Statusinformation für mehrere Peripheriegeräte in einem Computersystem und eine entsprechende Vorrichtung gemäß der folgenden Ansprüche vor.

[0010] In einer Ausführungsform der vorliegenden Erfindung ist ein Computersystem vorgesehen, das Statusinformation für mehrere Peripheriegeräte in einem Statusregister unterhält, das in einem Prozessor oder in einer Kernlogikeinheit im Computersystem angeordnet ist. In dieser Ausführungsform aktualisiert ein Peripheriegerät das Statusregister, wenn sich sein Status ändert, indem eine Busmasteroperation zum Übertragen der Statusinformation an das Statusregister ausgeführt wird. Dann erzeugt es eine Unterbrechung, um einem Prozessor anzuzeigen, dass es einer Verarbeitung bedarf. Wenn der Prozessor die Unterbrechung abarbeitet, muss der Prozessor lediglich das Statusregister lesen, um festzustellen, welches Peripheriegerät einer Verarbeitung bedarf. Dies ist ein sehr schneller Vorgang, weil das Statusregister innerhalb des Prozessors oder der Kernlogik ist. Es ist also keine zeitaufwendige Abfrage der Peripheriegeräte erforderlich, um den Status der Peripheriegeräte festzustellen. Demnach sieht eine Ausführungsform der vorliegenden Erfindung eine Vorrichtung vor, die Statusinformation für Peripheriegeräte in einem Statusregister unterhält. Diese Vorrichtung enthält einen Peripheriekommunikationskanal, der an eine Anzahl von Peripheriegeräte angeschlossen ist. Eine Aktualisierungsschaltung ist zwischen den Peripherie-Kommunikationskanal und das Statusregister geschaltet. Diese Aktualisierungsschaltung enthält einen Mechanismus zum Aktualisieren des Statusregisters in Reaktion auf Statusinformation enthaltende Signale, die sie von den Peripheriegeräten über den Peripherie-Kommunikationskanal erhalten haben.

[0011] In einer Ausführungsform der vorliegenden Erfindung weist der Kommunikationskanal einen

CPU-Bus auf. In einer Variation dieser Ausführungsform weist der Kommunikationskanal einen Prozessor-Speicher-Bus auf. In einer anderen Ausführungsform weist der Peripherie-Kommunikationskanal einen PC-Bus auf.

[0012] Bei einer Ausführungsform der vorliegenden Erfindung liest ein Spezial-Prozessorbefehl das Statusregister und zweigt auf der Grundlage des Inhalts des Statusregisters automatisch zur entsprechenden Unterbrechungs-Serviceroutine ab.

[0013] In einer Ausführungsform der vorliegenden Erfindung enthält der Kommunikationskanal einen Computersystembus. In einer weiteren Variation dieser Ausführungsform trägt der Kommunikationskanal Signale zum Aufrechterhalten einer Kohärenz zwischen mehreren Caches im Computersystem.

[0014] In einer Ausführungsform der vorliegenden Erfindung weist das Computersystem mehrere Zentraleinheiten und mehrere Statusregister auf, die zwischen den Zentraleinheiten und dem Kommunikationskanal eingeschaltet sind.

[0015] In einer Ausführungsform der vorliegenden Erfindung weist das Computersystem mehrere Zentraleinheiten und ein einziges Statusregister auf, das zwischen die Zentraleinheiten und den Peripherie-Kommunikationskanal geschaltet ist.

[0016] In einer weiteren Ausführungsform der vorliegenden Erfindung beinhaltet das Computersystem eine durch ODER-Funktion vergattete Unterbrechungsstruktur, welche die Peripheriegeräte mit der Zentraleinheit über eine Kernlogikeinheit verbindet.

[0017] Eine weitere Ausführungsform der vorliegenden Erfindung kann als eine Vorrichtung innerhalb einer Kernlogikeinheit eines Computersystems zum Aktualisieren eines Statusregisters zum Anzeigen von Veränderungen im Status von Peripheriegeräten in einem Computersystem charakterisiert werden. Diese Vorrichtung enthält mehrere Adresseingänge, die mit den Adressleitungen eines Busses verbunden sind, wobei der Bus an die Peripheriegeräte angeschlossen ist. Diese Vorrichtung enthält auch eine Adresserfassungsschaltung, die in den mehreren Adresseingängen an einen Satz werthöherer Bits angeschlossen ist. Diese Adresserfassungsschaltung ist so konfiguriert, dass sie eine Adresse in einem reservierten Bereich von Adressen erfasst, die durch den Satz werthöherer Bits angegeben sind. Die Vorrichtung enthält zusätzlich eine Decoderschaltung, die an einen Satz wertniedrigerer Bits in den mehreren Adresseingängen angeschlossen ist. Diese Decoderschaltung erfasst Referenzen auf bestimmte Adressen im reservierten Bereich von Adressen. Ein Statusregister ist an einen Satz von Ausgängen aus der Decoderschaltung angeschlossen, so dass eine durch ein Peripheriegerät vorgenommene Referenzierung auf eine bestimmte Adresse im reservierten Bereich von Adressen durch den Coder geleitet wird und Statusinformation für das Peripheriegerät im Statusregister aktualisiert. Das Statusregister weist auch Ausgänge auf, die an eine Zentraleinheit angeschlos-

sen sind, so dass das Statusregister von der Zentraleinheit gelesen werden kann.

[0018] In einer Variation dieser Ausführungsform enthält die Adresserfassungsschaltung einen Decoder. In einer anderen Variation dieser Ausführungsform enthält die Adresserfassungsschaltung einen Komparator.

[0019] In einer Variation dieser Ausführungsform residiert die Kernlogikeinheit innerhalb eines einzelnen Halbleiterchips. In einer weiteren Variation dieser Ausführungsform residiert die Kernlogikeinheit innerhalb eines Halbleiterchipsatzes.

[0020] Eine weitere Ausführungsform der vorliegenden Erfindung kann als ein Verfahren zum Verwalten von Statusinformation für mehrere Peripheriegeräte in einem Computersystem charakterisiert werden. Dieses Verfahren enthält das Empfangen an einem Peripheriegerät von Statusinformation über das Peripheriegerät und das Kommunizieren der Statusinformation durch einen Kommunikationskanal an ein Statusregister, das an eine Zentraleinheit im Computersystem angeschlossen ist. Das Verfahren enthält auch das Senden einer Unterbrechung vom Peripheriegerät an die Zentraleinheit.

#### Beschreibung der Figuren

[0021] **Fig. 1** veranschaulicht ein bekanntes Computersystem, bei dem ein Prozessor 100 Statusregister 112, 122 und 132 liest, die an entsprechenden Peripheriegeräten 110, 120 und 130 angeordnet sind.

[0022] **Fig. 2** veranschaulicht ein Computersystem, das einen Prozessor 100 mit einem internen Statusregister 102 oder einer Kernlogikeinheit 103 mit einem internen Statusregister 107 zum Speichern des Status von Peripheriegeräten gemäß einer Ausführungsform der vorliegenden Erfindung aufweist.

[0023] **Fig. 3** veranschaulicht ein Computersystem, das mehrere Prozessoren aufweist, die Statusregister 102, 302 und 312 oder ein einzelnes Statusregister 107 innerhalb einer Kernlogikeinheit 103 zum Speichern des Status von Peripheriegeräten gemäß einer Ausführungsform der vorliegenden Erfindung aufweisen.

[0024] **Fig. 4** veranschaulicht die Struktur eines Verarbeitungssystems gemäß einer Ausführungsform der vorliegenden Erfindung.

[0025] **Fig. 5** veranschaulicht die Struktur eines Statusregisters 102 oder Statusregisters 107 gemäß einer Ausführungsform der vorliegenden Erfindung.

[0026] **Fig. 6** veranschaulicht, wie eine Statusregisteraktualisierung über eine Speicherabbildung gemäß einer Ausführungsform der vorliegenden Erfindung durchgeführt wird.

[0027] **Fig. 7** veranschaulicht einige der Funktionseinheiten innerhalb des Prozessors 100 gemäß einer Ausführungsform der vorliegenden Erfindung.

[0028] **Fig. 8** veranschaulicht einen Teil der internen Struktur der Northbridge 408 gemäß einer Ausführungsform der vorliegenden Erfindung.

[0029] **Fig. 9** veranschaulicht einen Teil einer Innenstruktur einer Statusregistereinheit 712 gemäß einer Ausführungsform der vorliegenden Erfindung.

[0030] **Fig. 10** ist ein Fließdiagramm, das veranschaulicht, wie ein Peripheriegerät das Statusregister 102 oder 107 gemäß einer Ausführungsform der vorliegenden Erfindung aktualisiert.

[0031] **Fig. 11** ist ein Fließdiagramm, das veranschaulicht, wie der Prozessor 100 Information aus dem Statusregister 102 oder 107 zum Auslösen einer entsprechenden Unterbrechungs-Serviceroutine gemäß einer Ausführungsform der vorliegenden Erfindung verwendet.

#### Detaillierte Beschreibung der Erfindung

[0032] Die folgende Beschreibung dient dazu, es einem Fachmann zu ermöglichen, die Erfindung herzustellen und zu verwenden und wird im Zusammenhang einer bestimmten Anwendung und ihrer Anforderungen geliefert. Verschiedene Modifikationen der offenbarten Ausführungsformen werden dem Fachmann problemlos ersichtlich sein, und die hier definierten allgemeinen Prinzipien können auch auf andere Ausführungsformen und Anwendungsbereiche angewendet werden, ohne dass dadurch vom Geist und Umfang der vorliegenden Erfindung abgewichen wird. Daher soll die vorliegende Erfindung nicht auf die gezeigten Ausführungsformen eingeschränkt sein, sondern soll den weitesten Umfang genießen, der mit den hier offenbarten Prinzipien und Merkmalen vereinbar ist.

#### Überblick über eine erste Ausführungsform der Erfindung

[0033] **Fig. 1** veranschaulicht ein bekanntes Computersystem, bei dem ein Prozessor 100 Statusregister 112, 122 und 132 liest, die in entsprechenden Peripheriegeräten 110, 120 und 130 angeordnet sind. Der Prozessor 100 ist über die Kernlogikeinheit 103 mit einem Speicher 101 und einem Bus 105 verbunden. Der Prozessor 100 kann über den Bus 105 auf Peripheriegeräte 110, 120 und 130 zugreifen. In Reaktion auf eine Unterbrechung fragt der Prozessor 100 die Statusregister 112, 122 und 132 ab, um festzustellen, welche Peripheriegeräte 110, 120 und 130 einer Verarbeitung bedürfen. Dieses Abfragen benötigt vielfältige Operationen über den Bus 105.

[0034] **Fig. 2** veranschaulicht ein Computersystem mit einem Prozessor 100 mit einem internen Statusregister 102 oder einer Kernlogikeinheit 103 mit einem internen Statusregister 107 zum Speichern des Status von Peripheriegeräten gemäß einer Ausführungsform der vorliegenden Erfindung. Wie beim in **Fig. 1** gezeigten System ist der Prozessor 100 über die Kernlogikeinheit 103 mit dem Speicher 101 und dem Bus 105 verbunden. Der Prozessor 100 kann über den Bus 105 auf die Peripheriegeräte 110, 120 und 130 zugreifen.

[0035] Die in **Fig. 2** gezeigte Ausführungsform unterscheidet sich jedoch in einer Anzahl von Punkten vom in **Fig. 1** gezeigten System. In **Fig. 2** referenziert der Prozessor 100 das Statusregister 102 oder 107 während Unterbrechungen, um den Status von Peripheriegeräten 110, 120 und 130 festzustellen. Dies ist ein sehr schneller Vorgang, weil die Statusregister 102 und 107 sich im Inneren des Prozessors 100 bzw. der Kernlogikeinheit 103 befinden. Eine Referenz auf das Statusregister 102 oder 107 durch den Prozessor 100 erfordert keine Zugriffe über den Bus 105 zum Abfragen der Peripheriegeräte 110, 120 und 130.

[0036] Stattdessen sind die Peripheriegeräte 110, 120 und 130 für das Aktualisieren der Statusinformation im Statusregister 102 oder 107 verantwortlich. Diese Aktualisierung braucht nur dann stattzufinden, wenn sich der Status eines Peripheriegerätes ändert. Um ein Statusregister zu aktualisieren, schreibt ein Peripheriegerät, wie zum Beispiel das Peripheriegerät 110, an einen reservierten Speicherplatz im Adressraum des Busses 105. In diesem reservierten Adressraum residiert tatsächlich kein Speicher. Stattdessen fängt eine an das Statusregister angeschlossene Logik Referenzen auf diese reservierten Plätze ab und verwendet die Referenzen zum entsprechenden Aktualisieren der Statusregister zum Widerspiegeln der angegebenen Veränderung des Status eines Peripheriegeräts.

[0037] Hier ist festzustellen, dass das Statusregister 107 innerhalb der Kernlogikeinheit 103 eine alternative Ausführungsform zum Statusregister 102 innerhalb des Prozessors 100 darstellt. Das Referenzieren des Statusregisters 107 innerhalb der Kernlogikeinheit 103 ist nicht so schnell wie das Referenzieren des Statusregisters 102 innerhalb des Prozessors 100. Das Statusregister 107 erfordert jedoch keine speziellen Modifikationen am Prozessor 100, um es zu implementieren.

[0038] Allgemein kann der Prozessor 100 ein beliebiger Typ einer Computermaschine für ein Computersystem sein. Dies können zum Beispiel Mainframeprozessoren, Mikroprozessoren und Mikrocontroller sein, jedoch hierauf nicht eingeschränkt. Der Bus 105 kann ein beliebiger Typ eines Kommunikationskanals zur Verbindung eines Prozessors mit anderen Geräten in einem Computersystem, wie zum Beispiel Peripheriegeräten, Speichergeräten und anderen Prozessoren sein. Diese sind zum Beispiel der PCI-Bus und Busse, welche Signale zum Erhalten der Kohärenz zwischen mehrfachen Caches in einem Multiprozessorsystem mit gemeinsam genutztem Speicher enthalten, sind hierauf jedoch nicht eingeschränkt. Peripheriegeräte 110, 120 und 130 können ein beliebiger Typ von Peripheriegeräten sein, die an ein Computersystem angeschlossen werden können. Diese sind zum Beispiel Datenspeichergeräte, wie zum Beispiel Plattenlaufwerke und Bandlaufwerke; Dateneingabegeräte, wie zum Beispiel eine Tastatur oder eine Maus; Datenausgabegeräte, wie zum Bei-

spiel ein Bildschirm oder ein Lautsprecher; und Kommunikationsgeräte, wie zum Beispiel ein Netzwerk-Schnittstellen-Controller, sind jedoch hierauf nicht eingeschränkt.

[0039] **Fig. 3** veranschaulicht ein Computersystem mit mehreren Prozessoren, das gemäß einer Ausführungsform der vorliegenden Erfindung innerhalb der Kernlogikeinheit 103 Statusregister 102, 302 und 312 oder ein einziges Statusregister 107 zum Speichern des Status von Peripheriegeräten aufweist. In dieser Ausführungsform enthalten die Prozessoren 100, 300 und 310 Caches 113, 303 bzw. 313. Die Caches 113, 303 und 313 speichern Kopien von Code und Daten aus dem Speicher 320 zur Verwendung durch die Prozessoren 100, 300 bzw. 310. Die Prozessoren 100, 300 und 310 sind durch den Bus 320 miteinander verbunden, und die Kernlogikeinheit 103 ist das auch. Die Kernlogikeinheit 103 verbindet den Bus 320 mit dem Speicher 101 und dem Bus 105. Peripheriegeräte 110, 120 und 130 sind an den Bus 105 angeschlossen.

[0040] In dieser Ausführungsform enthält der Bus 320 Signale zum Aufrechterhalten einer Kohärenz zwischen im Speicher 320 gespeicherten Daten sowie von Kopien der Daten, die in den Caches 113, 303 und 313 gespeichert sind. Die Kohärenz wird typischerweise dadurch aufrechterhalten, dass ein Eintrag in einem Cache ungültig gemacht wird, wenn eine Kopie der Daten, die in dem Eintrag enthalten ist, durch einen anderen Cache oder im Speicher 320 modifiziert wird. Die Prozessoren 100, 300 und 310 verwenden eine Snoop-Logik zum Mithören eines Satzes von Signalen auf dem Bus 320, um festzustellen, ob ein Eintrag in einem lokalen Prozessor-Cache ungültig zu machen ist.

[0041] In dieser Ausführungsform enthalten die Prozessoren 100, 300 und 310 zusätzlich Statusregister 102, 302 bzw. 312, welche Statusinformation über Peripheriegeräte 110, 120 und 130 speichern. In dieser Ausführungsform arbeiten die Statusregister 102, 302 und 312 durch Überwachen der von der Snoop-Logik überwachten Signale, um Cache-Einträge ungültig zu machen. Um ein Statusregister zu aktualisieren, schreibt ein Peripheriegerät, wie zum Beispiel das Peripheriegerät 110, an einen reservierten Speicherplatz im Adressraum des Busses 105.

[0042] Allgemein verursacht das Aktualisieren eines Platzes im Speicher 101, dass ein Entwertungssignal auf dem Bus 105 alle Kopien der Daten, die in dem Speicherplatz gespeichert sind, ungültig werden, die in den Caches 113, 303 und 313 gegebenenfalls existieren. Da in diesem reservierten Adressraum tatsächlich kein Speicher residiert, verursachen die Entwertungssignale immer einen Cache-Fehlertreffer. Eine an das Statusregister angeschlossene Logik fängt Referenzen auf diese reservierten Speicherplätze ab und verwendet diese Referenzen zum entsprechenden Aktualisieren der Statusregister zum Widerspiegeln der angezeigten Veränderung des Status eines Peripheriegeräts. Dieser Vorgang ist im

Einzelnen mit Bezug auf die **Fig. 4-11** unten beschrieben.

#### Beschreibung des Computersystems

[0043] **Fig. 4** veranschaulicht die Struktur eines Verarbeitungssystems gemäß einer Ausführungsform der vorliegenden Erfindung. In der veranschaulichten Ausführungsform ist die CPU 404 über die Northbridge 408 an den Speicher 405 und an den Bus 430 angeschlossen. Der Speicher 405 kann ein beliebiger Typ eines Halbleiterspeichers sein, der in einem Computersystem verwendet werden kann. Der Bus 430 kann ein beliebiger Typ eines Computersystembusses sein. In einer Ausführungsform enthält der Bus 430 einen PCI-Bus. Der Bus 430 ist an ein Grafikmodul 414 angeschlossen, das grafische Bilder zur Ausgabe an die Anzeige 416 verarbeitet. Der Bus 430 ist zusätzlich an eine Soundkarte 415 angeschlossen, die Audiosignale erzeugt. Die Soundkarte 415 ist an einen Lautsprecher 417 angeschlossen, so dass die von der Soundkarte 415 erzeugten Audiosignale über den Lautsprecher 417 ausgegeben werden.

[0044] In der veranschaulichten Ausführungsform ist die CPU 404 über die Northbridge 408 zusätzlich mit einer Southbridge 410 verbunden. Die Northbridge 408 und Southbridge 410 bilden ein Teil der "Kernlogik" für das Computersystem. Diese Kernlogik bindet Operationen und Komponenten im Computersystem zusammen und koordiniert sie. Die Southbridge 410 ist mit einer Platte 406 verbunden, die ein beliebiger Typ einer nichtflüchtigen Speichervorrichtung sein kann. Diese können magnetische, optische, magneto-optische und Flashspeichergeräte sein. Die Southbridge 410 ist auch mit dem Bus 432 verbunden, der ein beliebiger Typ eines Computersystembusses sein kann. In einer Ausführungsform weist der Bus 432 einen ISA-Bus auf. Der Bus 432 erlaubt es der CPU 404, mit dem BIOS-ROM 412 und dem Modem 422 zu kommunizieren, die an den Bus 432 angeschlossen sind. Das Modem 422 kann ein beliebiger Typ eines Modems sein, durch das ein Computersystem über eine Telefonleitung kommunizieren kann.

[0045] **Fig. 4** veranschaulicht das Statusregister 102 innerhalb der CPU 404 sowie ein Statusregister 107 innerhalb der Northbridge 408. In einer anderen Ausführungsform kann das Statusregister 102 ein eigenständiges Register im Computersystem sein, das nicht innerhalb der CPU 404 oder der Northbridge 408 ist.

#### Beschreibung des Statusregisters

[0046] **Fig. 5** veranschaulicht die Struktur des Statusregisters 107 (oder des Statusregisters 102) gemäß einer Ausführungsform der vorliegenden Erfindung. In der veranschaulichten Ausführungsform enthält das Statusregister 107 (oder das Statusregister

102) mehrere Bits, die Statusinformation für Peripheriegeräte im Computersystem enthalten. Diese Bits enthalten das Statusgerät 1 502, Statusgerät 2 504, Statusgerät 3 506 und Statusgerät N 508. Wenn ein Statusbit gesetzt ist, zeigt dies an, dass das entsprechende Gerät einer Verarbeitung bedarf. In anderen Ausführungsformen der vorliegenden Erfindung enthält das Statusregister 107 (oder Statusregister 102) mehr als ein Bit Statusinformation für jedes Gerät. Diese Bits enthalten zusätzliche Statusinformationen für jedes Gerät, über das hinaus, dass ein bestimmtes Gerät einer Verarbeitung bedarf. Zum Beispiel kann die Statusinformation die Art der Verarbeitung angeben, die das Gerät benötigt.

[0047] **Fig. 6** veranschaulicht, wie die Statusregisteraktualisierung über die Speicherabbildung gemäß einer Ausführungsform der vorliegenden Erfindung bewerkstelligt wird. In dieser Ausführungsform ist der Adressraum 600 ein Adressraum für Adressleitungen auf einem Bus, wie zum Beispiel dem Bus 105 aus **Fig. 1**. Der Adressraum 600 enthält ein BIOS-Image 610 am unteren Ende des Adressraums 600. Das BIOS-Image 610 enthält Code zum Implementieren niedriger Betriebssystemfunktionen. Der Adressraum 600 enthält zusätzlich einen physischen Speicher 630 am oberen Ende des Adressraums 600. Der physische Speicher 630 enthält Code und Daten, die von einem Prozessor zum Ausführen von Programmen verwendet werden. Ein Abschnitt des Adressraums 600 zwischen dem BIOS-Image 610 und dem physischen Speicher 630 ist zur Aktualisierung des Statusregisters 107 (oder des Statusregisters 102) reserviert. Es gibt keinen tatsächlichen Speicher in diesen Speicherplätzen. Zugriffe auf diese Speicherplätze aktualisieren jedoch den Inhalt des Statusregisters 107 (oder des Statusregisters 102). Zum Beispiel setzt ein Zugriff auf die Adresse 640 das Statusbit 502 für das Gerät 1 so, dass angezeigt wird, dass das Gerät 1 einer Verarbeitung bedarf, und ein Zugriff auf die Adresse 642 setzt das Bit 502 zurück, um anzuzeigen, dass das Gerät 1 keiner Verarbeitung bedarf. In ähnlicher Weise setzen Zugriffe auf Adressen 644, 648 und 652 Statusbits 504, 506 bzw. 508 und Zugriffe auf Adressen 646, 650 und 654 setzen die gleichen Statusbits zurück.

#### Beschreibung der Statusregisterplätze

[0048] **Fig. 7** veranschaulicht die Funktionseinheiten im Prozessor 100 gemäß einer Ausführungsform der vorliegenden Erfindung. In der veranschaulichten Ausführungsform enthält der Prozessor 100 eine Integer-ALU (Arithmetic Logic Unit) 702 und eine Floating-Point-Einheit 704, die Rechneroperationen durchführen. Der Prozessor enthält auch einen Controller 706, der Aktionen von Funktionseinheiten im Prozessor 100 koordinieren kann. Eine Anzahl von Einheiten im Prozessor 100 sind an den Bus 105 angeschlossen. Diese enthalten einen L1-Cache 708, der Befehle und Daten speichert, die vom Prozessor

100 während Rechneroperationen verwendet werden. In manchen Ausführungsformen enthält der L1-Cache 708 getrennte Befehls- und Daten-Caches. Die Snoop-Logik 710 ist auch an den Bus 105 angeschlossen. Die Snoop-Logik 710 hört Signale auf dem Bus 105 mit, die "Snoop"-Informationen enthalten. Die Snoop-Logik 710 verwendet diese Snoop-Information zum Entwerten von Einträgen innerhalb des L1-Cache 708. Der Prozessor 105 enthält zusätzlich Register 712, die zeitweise Datenwerte für Rechneroperationen im Prozessor 100 speichern. Die Register 712 enthalten zusätzlich Statusregister 102 aus **Fig. 2**.

[0049] **Fig. 8** veranschaulicht einen Teil der internen Struktur der Northbridge 408 gemäß einer Ausführungsform der vorliegenden Erfindung. In dieser Ausführungsform residiert das Statusregister 107 in der Statusregistereinheit 712 innerhalb der Northbridge 408. Die Northbridge 408 enthält zusätzlich einen Switch 805, der Daten zwischen der GPU 404, dem Speicher 405 und dem Bus 430 vermittelt. In dieser Ausführungsform hört die Statusregistereinheit 712 bei Zugriffen auf den Bus 430 mit, um Zugriffe auf den reservierten Bereich der Adressen 620 zu erfassen.

#### Beschreibung einer Ausführungsform des Statusregisters

[0050] **Fig. 9** veranschaulicht einen Teil einer internen Struktur der Statusregistereinheit 712 gemäß einer Ausführungsform der vorliegenden Erfindung. In dieser Ausführungsform werden Adressleitungen vom Bus 105 durch eine Logik innerhalb der Statusregistereinheit 712 überwacht, um Referenzen auf reservierte Adressen 620 zu erfassen. Insbesondere werden werthöhere Adressbits aus dem Bus 105 in Eingänge des Decoders 900 eingespeist. Wenn die werthöheren Adressbits 902 eine Adresse in den reservierten Adressen 620 angeben, erzeugt der Decoder 900 ein Registertreffersignal 906, das in einen Freigabeeingang des Decoders 910 eingespeist wird. In einer anderen Ausführungsform werden werthöhere Adressbits 902 in eine Komparatorschaltung eingespeist, welche die gleiche Adresserfassungsfunktion durchführt. Allgemein kann eine beliebige allgemein bekannte Adresserfassungsschaltung zum Erfassen von Adressen im reservierten Bereich 620 verwendet werden. Wertniedrige Adressbits werden zum Auswählen von verschiedenen Ausgangssignalen des Decoders 910 verwendet. Diese Ausgangssignale setzen entweder Bits im Statusregister 107 (oder Statusregister 102) oder setzen diese zurück. In der veranschaulichten Ausführungsform wird jedes Bit des Statusregisters 107 (oder Statusregisters 102) in einer bistabilen Schaltung gespeichert, welche zwei NAND-Gatter enthält, die zirkulär verschaltet sind, wie das in **Fig. 9** gezeigt ist. Jedes NAND-Gattenpaar erhält zwei Eingänge vom Decoder 910. Wenn der obere Eingang auf LOW geht, wird das Bit gesetzt und wenn der untere Eingang auf

LOW geht, wird das Bit rückgesetzt. Zum Beispiel verursacht ein Zugriff auf Adresse 640, dass der obere Ausgang des Decoders 910 auf LOW geht, was verursacht, dass das entsprechende Bit einen Wert eins bekommt. Im Gegensatz dazu verursacht ein Zugriff auf Adresse 642, dass der nächste untere Ausgang des Decoders 910 auf LOW geht, was das gleiche Bit rücksetzt. Wenn schließlich das Status-Lesesignal 912 ausgegeben wird, werden die damit verbundenen Treiber aktiviert, um die Bits aus dem Statusregister 102 oder dem Statusregister 107 (oder dem Statusregister 102) auszulesen. Diese Ausführungsform veranschaulicht eine von vielen möglichen Strukturen für die Statusregister 107 (oder Statusregister 102). Allgemein kann eine beliebige andere üblicherweise bekannte Struktur für ein Register verwendet werden.

#### Beschreibung des Betriebs des Statusregisters

[0051] **Fig. 10** ist ein Fließdiagramm, das illustriert, wie ein Peripheriegerät das Statusregister 102 (oder 107) gemäß einer Ausführungsform der vorliegenden Erfindung aktualisiert. In dieser Ausführungsform beginnt das Peripheriegerät im Zustand 1000 und geht zum Zustand 1002 weiter. Im Zustand 1002 erfasst das Peripheriegerät eine Veränderung seines Status. Diese Veränderung des Status kann anzeigen, dass eine Verarbeitung benötigt wird. Zum Beispiel kann es sein, dass Daten zur Übertragung an den Prozessor 100 bereitstehen. Dann geht das Peripheriegerät weiter zum Zustand 1004. Im Zustand 1004 führt das Peripheriegerät eine Busmasteroperation auf dem Bus 105 durch, um das Statusregister 102 (oder 107) zu aktualisieren, um anzuzeigen, dass das Gerät einer Verarbeitung bedarf. Das Peripheriegerät geht dann zum Zustand 1006 weiter. Im Zustand 1006 erzeugt das Peripheriegerät eine Unterbrechung, um den Prozessor 100 anzuzeigen, dass ein Peripheriegerät einer Verarbeitung bedarf. Das Peripheriegerät geht dann zum Zustand 1008 weiter, der ein Endzustand ist.

[0052] **Fig. 11** ist ein Fließdiagramm, das illustriert, wie der Prozessor 100 gemäß einer Ausführungsform der vorliegenden Erfindung Information vom Statusregister 102 (oder 107) zum Auslösen einer Unterbrechungs-Serviceroutine verwendet. Der Prozessor 100 startet im Zustand 1100 und geht zum Zustand 1102 weiter. Im Zustand 1102 empfängt der Prozessor 100 eine Unterbrechung von einem der an den Bus 105 angeschlossenen Peripheriegeräte. Der Prozessor 100 geht dann zum Zustand 1104 weiter, in dem der Prozessor 100 den Zustand speichert, um die Unterbrechung zu verarbeiten. Dann geht der Prozessor 100 zum Zustand 1106 weiter. Im Zustand 1106 holt der Prozessor 100 Unterbrechungsbefehle von demjenigen Ort ein, auf den ein Unterbrechungsvektor zeigt. Der Prozessor 100 geht dann zum Zustand 1108 weiter. Im Zustand 1108 kopiert der Prozessor 100 das Statusregister 102 (oder 107) in ein

Prozessorregister, um den Inhalt des Statusregisters 102 (oder 107) zu untersuchen. Dann geht das System weiter zum Zustand 1110.

[0053] Im Zustand 1110 zweigt der Prozessor 100 auf der Grundlage des Inhalts des Statusregisters 102 (oder 107) zu verschiedenen Unterbrechungs-Serviceroutinen 1112, 1114 und 1116 ab. Diese Abzweigung kann tatsächlich eine Anzahl von Befehlen erfordern, um das Statusregister 107 zu testen und um eine entsprechende bedingte Abzweigung durchzuführen. Wenn das Statusregister 102 (oder 107) anzeigt, dass das Gerät 1 einer Verarbeitung bedarf, zweigt der Prozessor 100 zum Zustand 1112 ab, welches der Anfang einer Unterbrechungs-Serviceroutine für einen Service des Geräts 1 ist. Diese Unterbrechungs-Serviceroutine enthält allgemein eine große Anzahl von Unterbrechungs-Servicebefehlen, welche nicht gezeigt sind. Nachdem die Unterbrechungs-Serviceroutine abgeschlossen ist, geht der Prozessor 100 zum Zustand 1118 weiter, der ein Endzustand ist. Wenn das Statusregister 102 (oder 107) anzeigt, dass das Gerät 2 einer Verarbeitung bedarf, zweigt der Prozessor 100 zum Zustand 1114 ab, welches der Anfang einer Unterbrechungs-Serviceroutine für einen Service von Gerät 2 ist. Nachdem diese Unterbrechungs-Serviceroutine abgeschlossen ist, geht der Prozessor 100 zum Zustand 1118 weiter, der ein Endzustand ist. Wenn das Statusregister 102 (oder 107) anzeigt, dass das Gerät N einer Verarbeitung bedarf, zweigt der Prozessor 100 zum Zustand 1116 ab, welches der Start einer Unterbrechungs-Serviceroutine für einen Service von Gerät N ist. Nachdem diese Unterbrechungs-Serviceroutine abgeschlossen ist, geht der Prozessor 100 zum Zustand 1118 weiter, der ein Endzustand ist.

[0054] In einer Ausführungsform der vorliegenden Erfindung werden der Vorgang des Abbildens von Peripheriegeräten auf bestimmte Bits eines Statusregisters 102 (oder 107) und der Vorgang des Zuweisens bestimmter Unterbrechungs-Serviceroutinen zu bestimmten Peripheriegeräten als Initialisierungsroutinen während der Systeminitialisierung durchgeführt. In einer Ausführungsform der vorliegenden Erfindung residiert der Code zum Durchführen dieser Funktionen in einem Festwertspeicher, der während des Hochfahrens des Systems gelesen wird.

#### Definitionen

[0055] Kernlogik – Schaltung innerhalb eines Computersystems, die einen Prozessor mit einem Speicher und einem Peripheriebus verbindet und andere Funktionen durchführt.

[0056] Snoop-Bus – Ein Bus, der Signale zur Aufrechterhaltung einer Konsistenz oder Kohärenz zwischen einer Vielzahl von Caches in einem mehrere Prozessoren enthaltenden Computersystem trägt.

[0057] Die vorhergehenden Beschreibungen von Ausführungsformen der Erfindung wurden ausschließlich zu Zwecken der Veranschaulichung und

Beschreibung gegeben. Sie sollen nicht erschöpfend sein oder die Erfindung auf die offenbarten Ausführungsformen einschränken. Selbstverständlich werden dem Fachmann viele Modifikationen und Variationen ersichtlich sein.

#### Patentansprüche

1. Verfahren zum Verwalten von Statusinformationen für mehrere Peripheriegeräte (110, 120, 130) in einem Computersystem, mit den folgenden Schritten:

- Empfangen (1002), über einen Kommunikationskanal (105), von Statusinformationen von einem Peripheriegerät;
- Aktualisieren (1004) eines Statusregisters (102, 107) in Kommunikation mit einer Zentraleinheit (100) in Reaktion auf die Statusinformationen vom Peripheriegerät;
- Empfangen (1102), an der Zentraleinheit, einer Unterbrechung vom Peripheriegerät;
- Testen (1110) des Statusregisters in Reaktion auf die Unterbrechung zum Bestimmen, welche Peripheriegeräte aus den mehreren Peripheriegeräten bedient werden müssen; und
- Bedienen (1112, 1114, 1116) der Peripheriegeräte, die bedient werden müssen,

**dadurch gekennzeichnet**, dass das Empfangen von Statusinformationen vom Peripheriegerät das Empfangen eines Zugriffs auf eine bestimmte Adresse (640, 642, 644, 648, 650, 652, 654) in einem Satz reservierter Adressen (620) beinhaltet, wobei der Zugriff auf die bestimmte Adresse einen bestimmten Status für ein bestimmtes Peripheriegerät bedeutet.

2. Verfahren nach Anspruch 1, bei dem das Aktualisieren des Statusregisters zum Angeben des Status eines Peripheriegeräts das Modifizieren eines Bits (502, 504, 506, 508) in einem Statusregister (102, 107) beinhaltet.

3. Verfahren nach Anspruch 2, weiter mit den folgenden Schritten:

- Überwachen von Eingaben von Adressleitungen eines Busses (105);
- Erfassen einer Adresse im reservierten Bereich (620) von Adressen, die von einem Satz werthöherer Bits (902) angegeben werden;
- Erfassen von Hinweisen auf bestimmte Adressen (640, 642, 644, 648, 650, 652, 654) im reservierten Bereich von Adressen, die durch einen Satz wertniedrigerer Bits (904) angegeben werden; und
- Verwenden der wertniedrigeren Bits (904) zum Setzen oder Rücksetzen von Bits (502, 504, 506, 508) im Statusregister (102, 107).

4. Verfahren nach Anspruch 3, bei dem das Empfangen der Statusinformationen umfasst:

- Empfangen von Statusinformationen von einem Peripheriegerät über einen Bus (320), der auch Sig-

nale zur Aufrechterhaltung der Kohärenz von Daten und Code zwischen mehreren Caches (103, 303, 313) und einem Speicher (320) im Computersystem trägt; und

- Verwenden einer Snoop-Logik (710) zum Mithören der Signale zum Bestimmen, ob ein Eintrag in einem Cache zur Aufrechterhaltung der Kohärenz zu löschen ist.

5. Verfahren nach Anspruch 1, das bei der Systeminitialisierung oder beim Hochfahren die folgenden Schritte aufweist:

- Abbilden von Peripheriegeräten auf bestimmte Bits des Statusregisters (102, 107); und
- Zuweisen bestimmter Unterbrechungsdienstroutinen zu bestimmten Peripheriegeräten.

6. Vorrichtung zum Verwalten von Statusinformationen für mehrere Peripheriegeräte (110, 120, 130) in einem Computersystem, mit:

- einer Einrichtung zum Empfangen (1002), über einen Kommunikationskanal (105), von Statusinformationen von einem Peripheriegerät;
- einer Einrichtung zum Aktualisieren (1004) eines Statusregisters (102, 107) in Kommunikation mit einer Zentraleinheit (100) in Reaktion auf die Statusinformationen vom Peripheriegerät;
- einer Einrichtung zum Empfangen (1102), an der Zentraleinheit, einer Unterbrechung vom Peripheriegerät;
- einer Einrichtung zum Testen (1110) des Statusregisters in Reaktion auf die Unterbrechung zum Bestimmen, welche Peripheriegeräte aus den mehreren Peripheriegeräten bedient werden müssen; und
- einer Einrichtung zum Bedienen (1112, 1114, 1116) der Peripheriegeräte, die bedient werden müssen, dadurch gekennzeichnet, dass die Einrichtung zum Empfangen von Statusinformationen vom Peripheriegerät eine Einrichtung zum Empfangen eines Zugriffs auf eine bestimmte Adresse (640, 642, 644, 648, 650, 652, 654) in einem Satz reservierter Adressen (620) beinhaltet, wobei der Zugriff auf die bestimmte Adresse einen bestimmten Status für ein bestimmtes Peripheriegerät bedeutet.

7. Vorrichtung nach Anspruch 6, bei der die Einrichtung zum Aktualisieren des Statusregisters zum Angeben des Status eines Peripheriegeräts eine Einrichtung zum Modifizieren eines Bits (502, 504, 506, 508) in einem Statusregister (102, 107) aufweist.

8. Vorrichtung nach Anspruch 7, weiter mit den folgenden Elementen:

- einer Einrichtung zum Überwachen von Eingaben von Adressleitungen eines Busses (105);
- einer Adresserfassungsschaltung (900) zum Erfassen einer Adresse im reservierten Bereich (620) von Adressen, die von einem Satz werthöherer Bits (902) angegeben werden;
- einer Decoderschaltung (910) zum Erfassen von

Verweisen auf bestimmte Adressen (640, 642, 644, 648, 650, 652, 654) im reservierten Bereich von Adressen, die durch einen Satz wertniedrigerer Bits (904) angegeben werden; und

- einer Einrichtung zum Verwenden der wertniedrigeren Bits (904) zum Setzen oder Rücksetzen von Bits (502, 504, 506, 508) im Statusregister (102, 107).

9. Vorrichtung nach Anspruch 8, bei der die Einrichtung zum Empfangen von Statusinformationen aufweist:

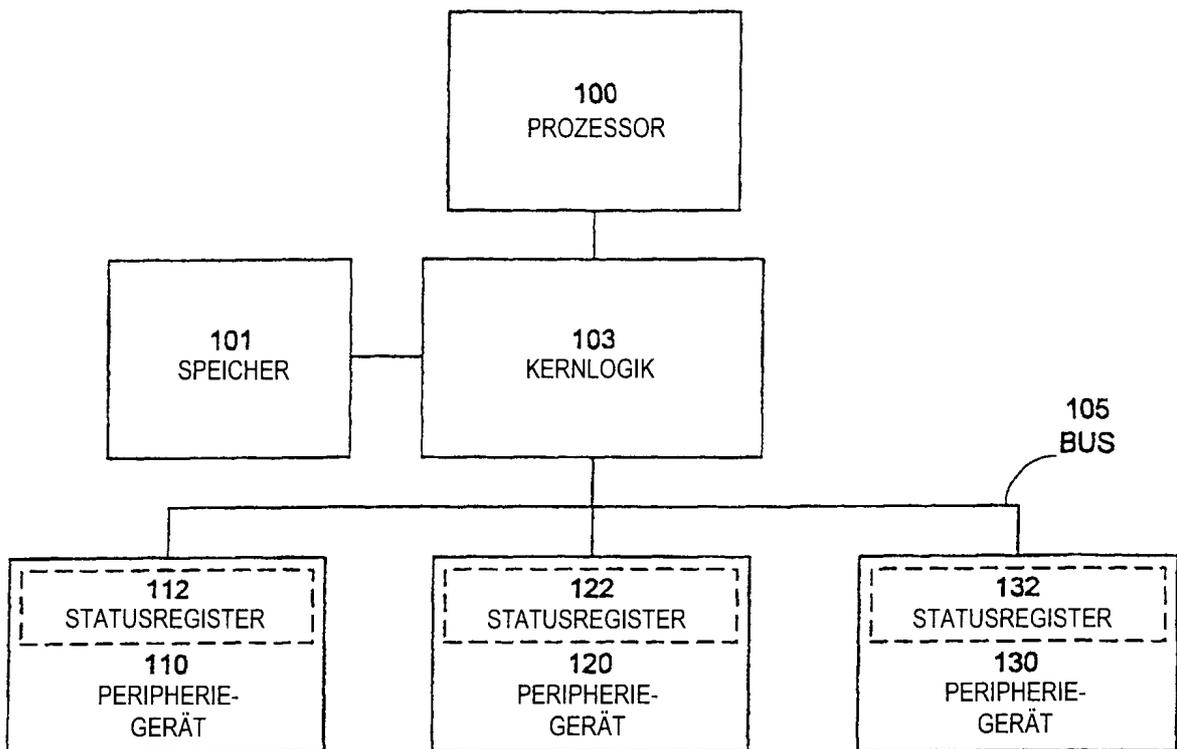
- eine Einrichtung zum Empfangen von Statusinformationen von einem Peripheriegerät über einen Bus (320), der auch Signale zur Aufrechterhaltung der Kohärenz von Daten und Code zwischen mehreren Caches (103, 303, 313) und einem Speicher (320) im Computersystem trägt; und
- eine Einrichtung zum Verwenden einer Snoop-Logik (710) zum Mithören der Signale zum Bestimmen, ob ein Eintrag in einem Cache zur Aufrechterhaltung der Kohärenz zu löschen ist.

10. Vorrichtung nach Anspruch 6, die bei der Systeminitialisierung oder zum Hochfahren weiter die folgenden Elemente aufweist:

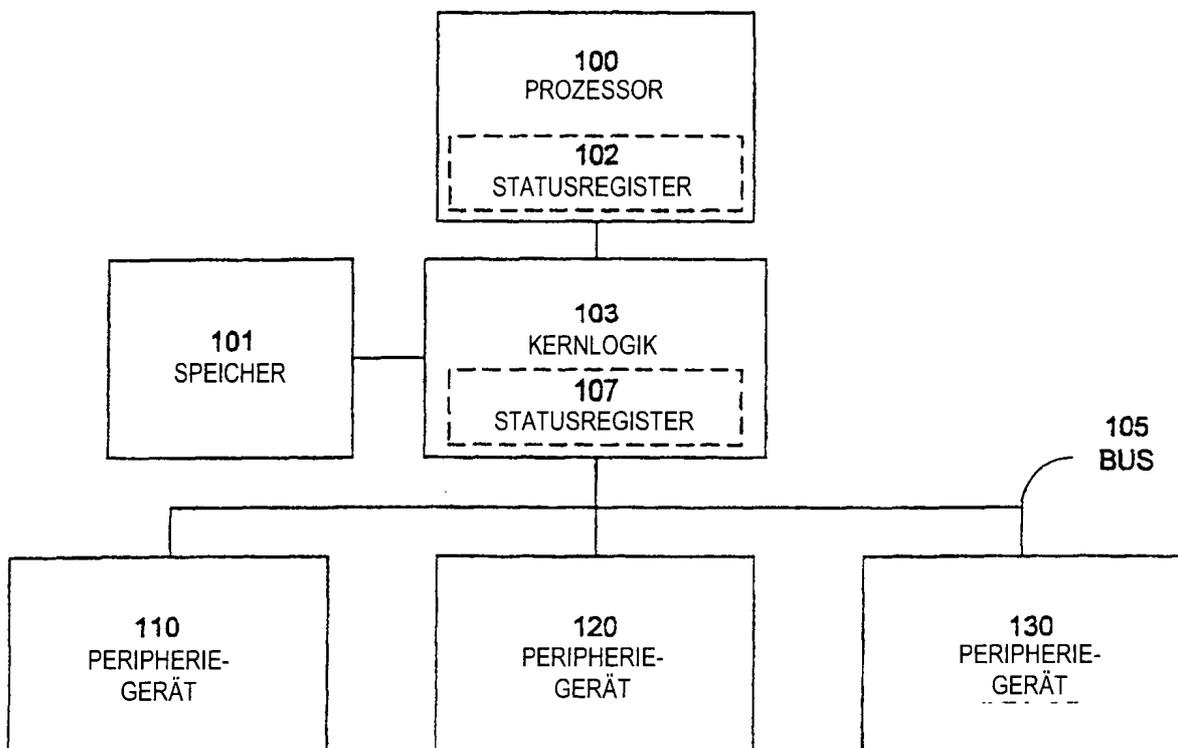
- eine Einrichtung zum Abbilden von Peripheriegeräten auf bestimmte Bits des Statusregisters (102, 107); und
- eine Einrichtung zum Zuweisen bestimmter Unterbrechungsdienstroutinen zu bestimmten Peripheriegeräten.

Es folgen 8 Blatt Zeichnungen

Anhängende Zeichnungen



**FIG. 1**  
(STAND DER TECHNIK)



**FIG. 2**

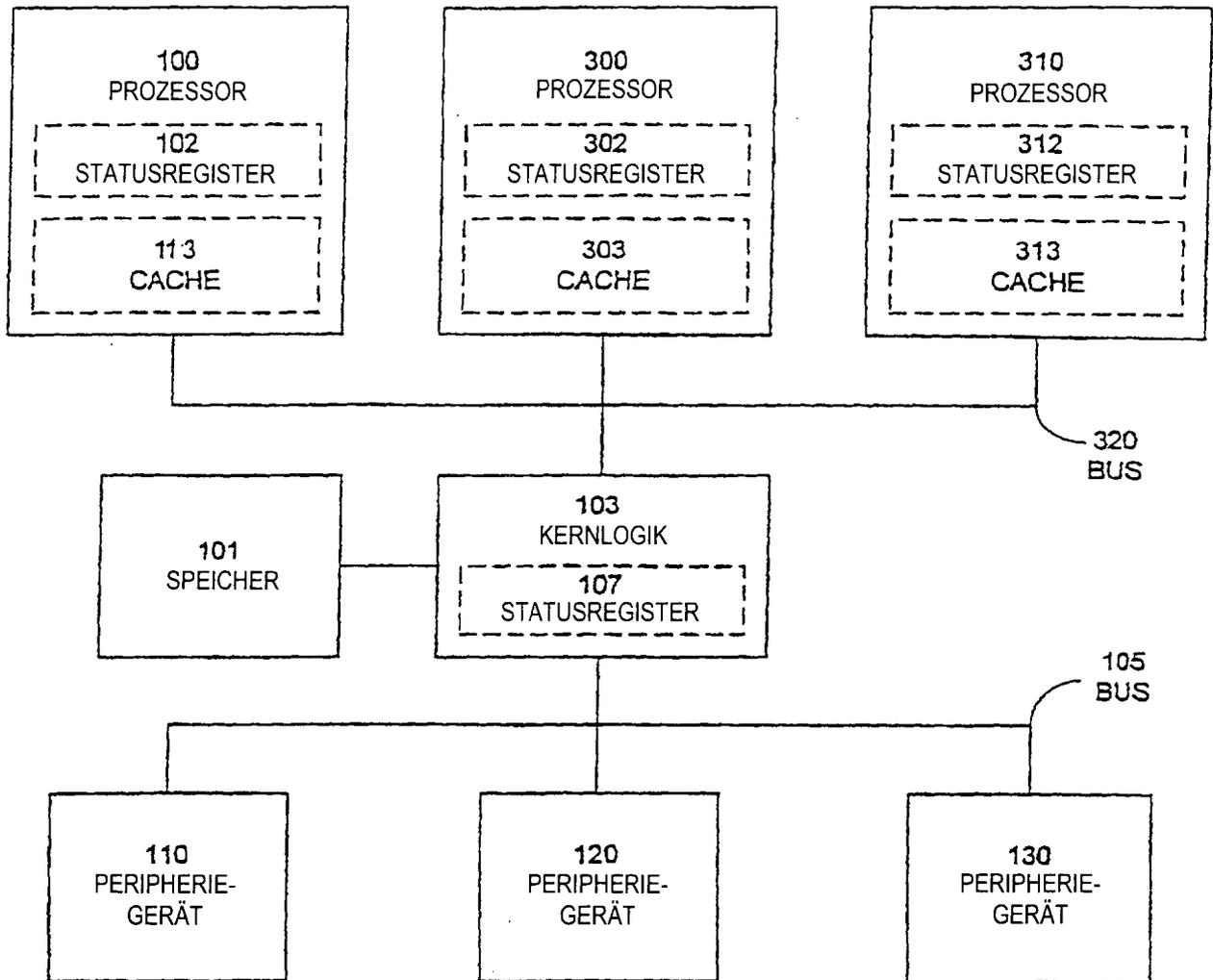


FIG. 3

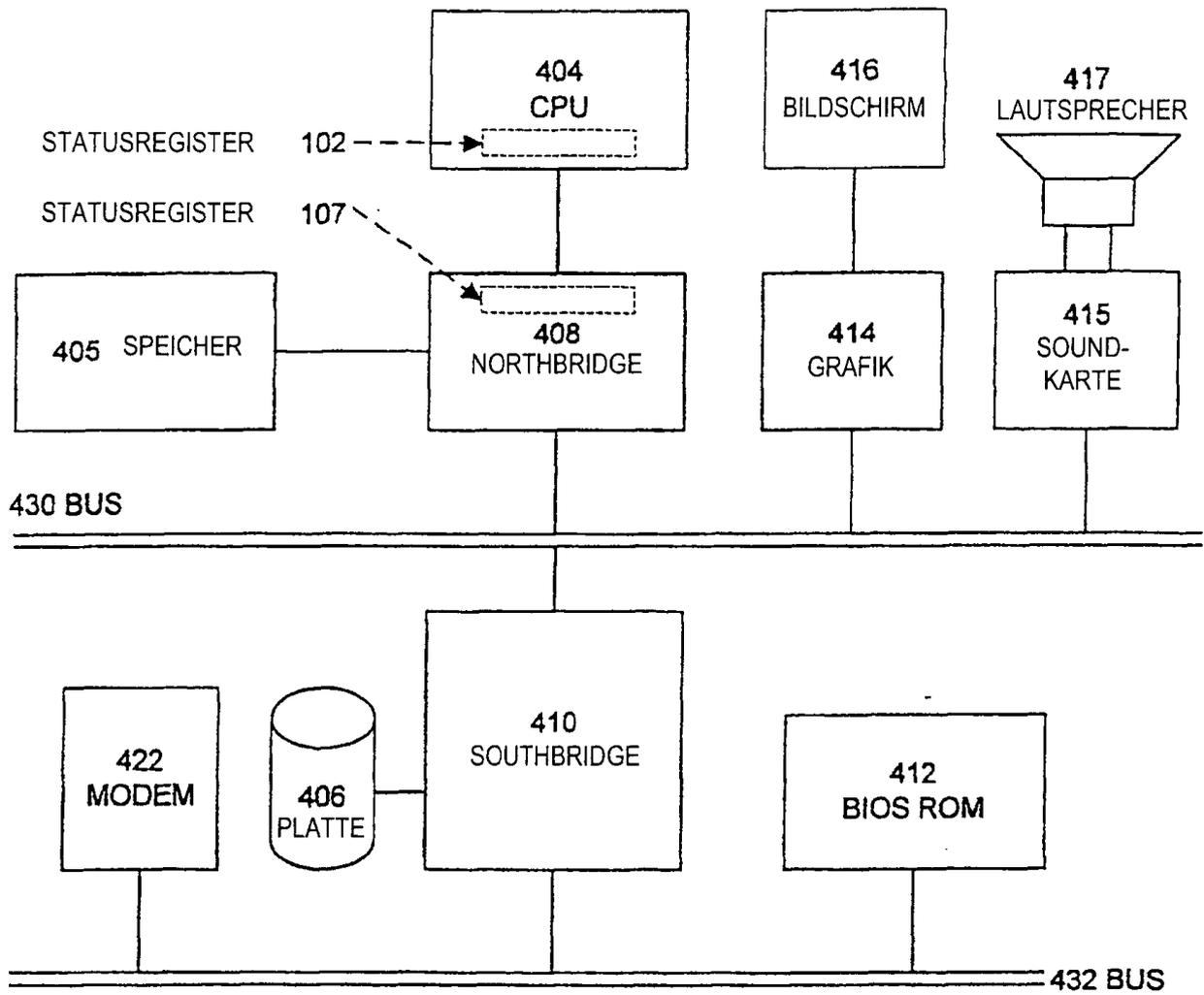


FIG. 4

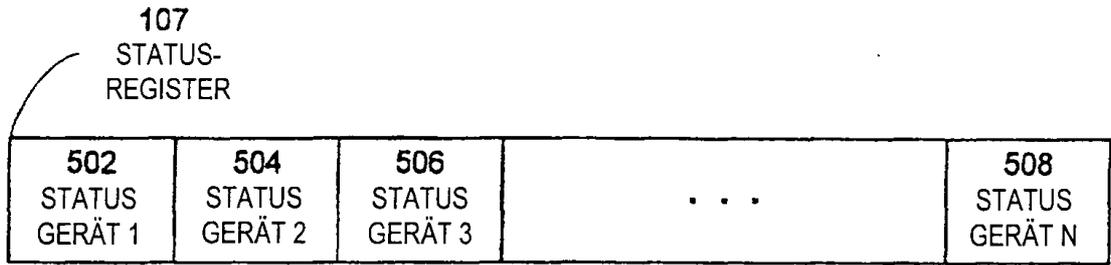


FIG. 5

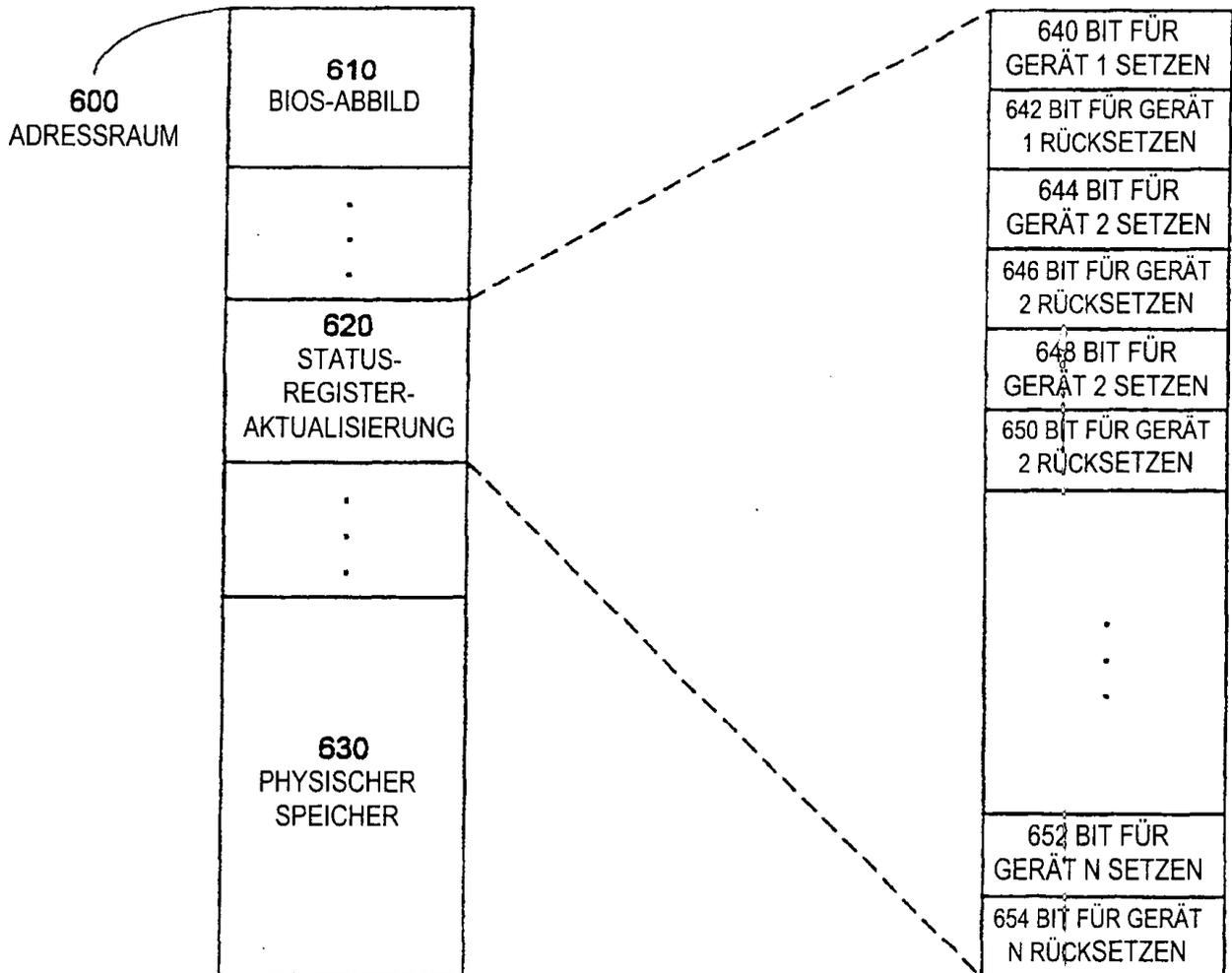


FIG. 6

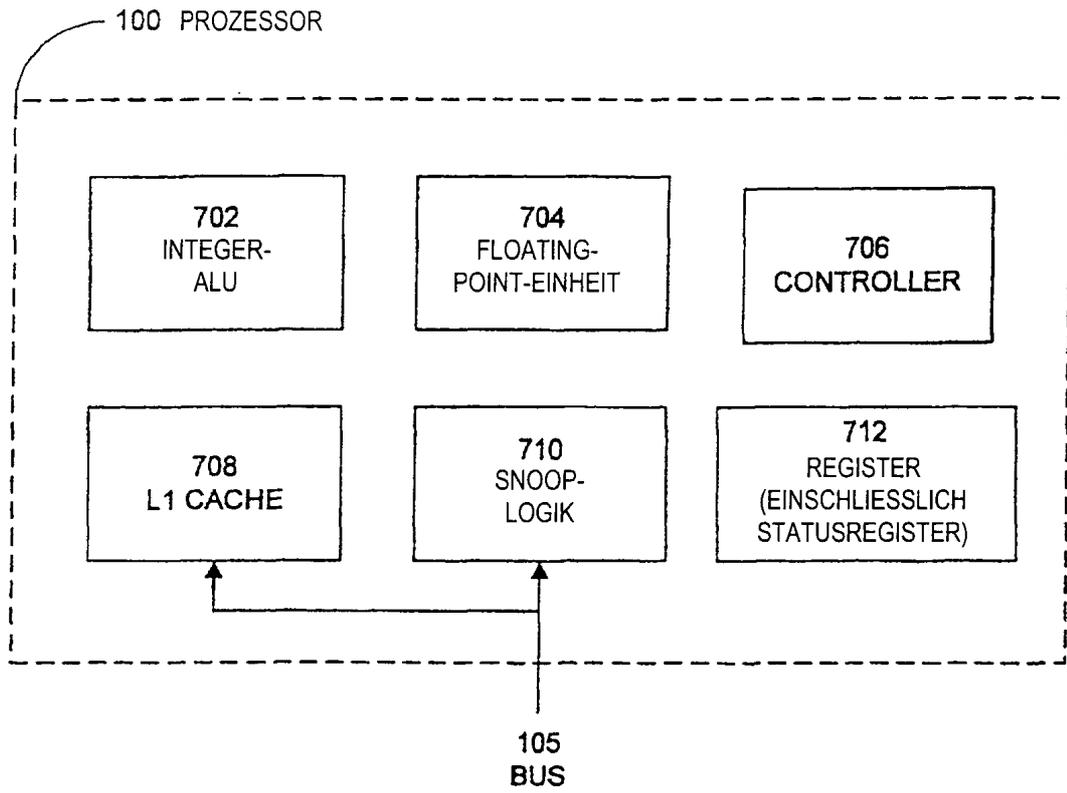


FIG. 7

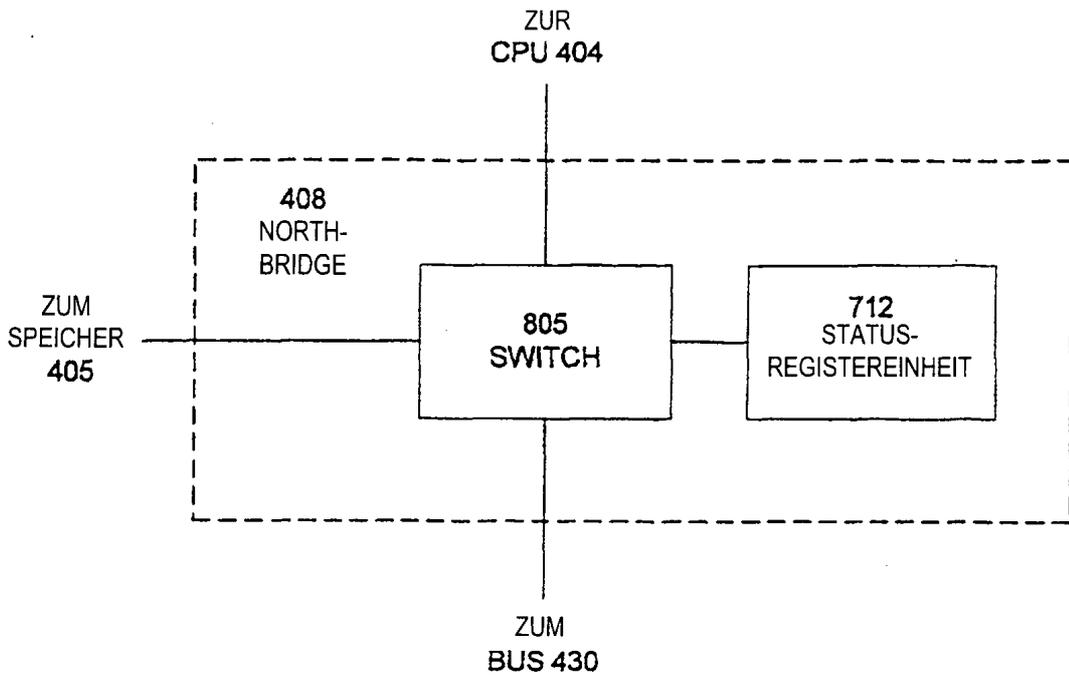


FIG. 8

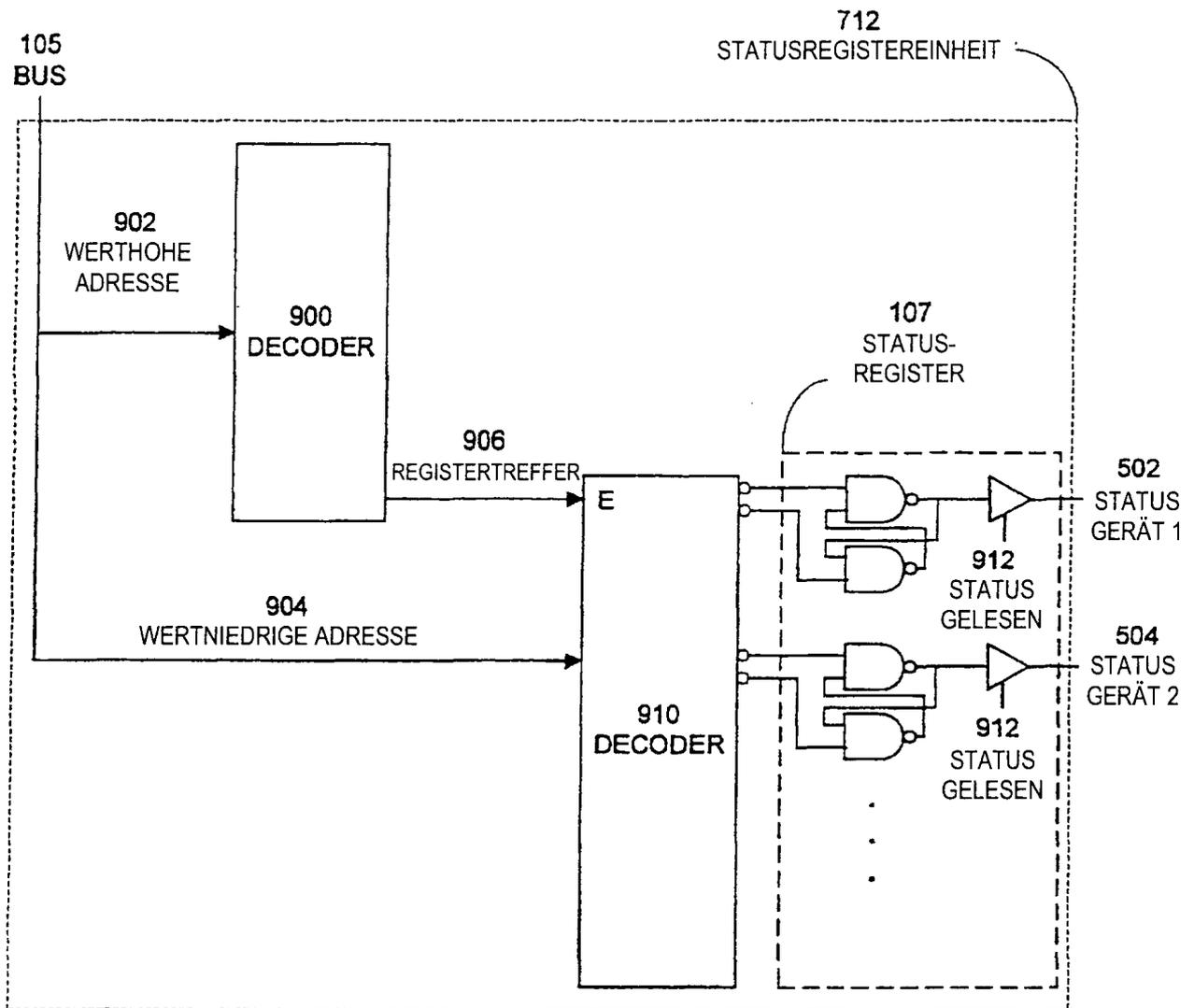


FIG. 9

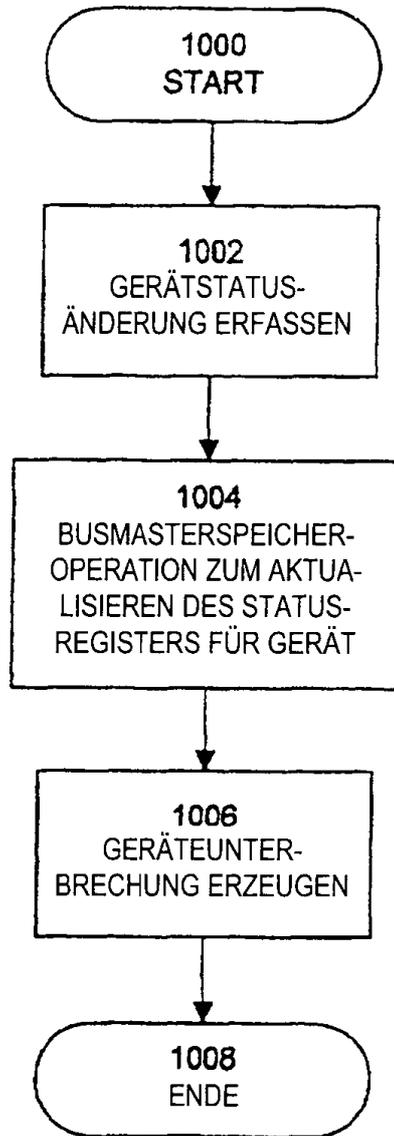


FIG. 10

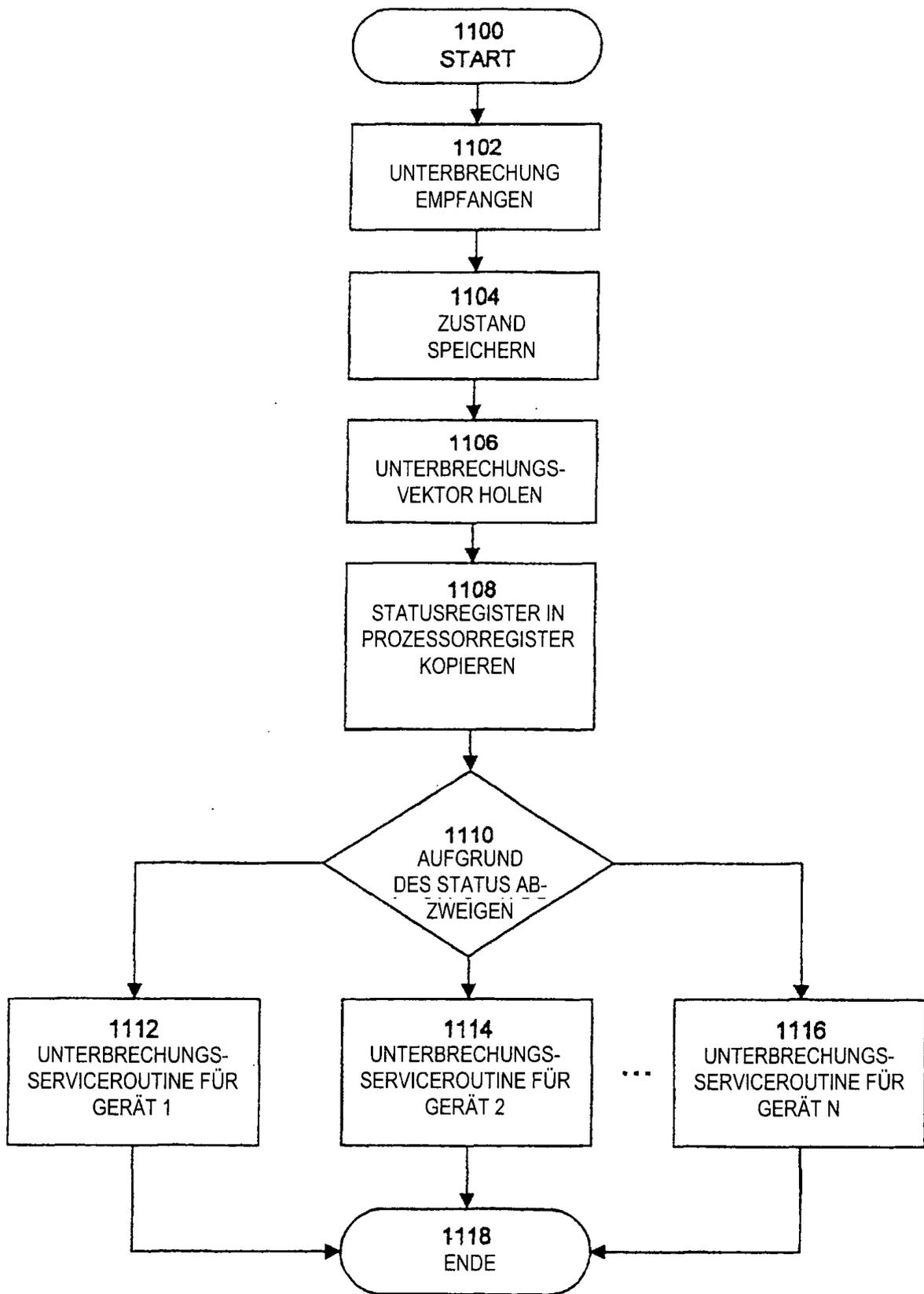


FIG. 11