



(10) **DE 10 2010 052 726 A1** 2012.05.31

(12) **Offenlegungsschrift**

(21) Aktenzeichen: **10 2010 052 726.2**

(22) Anmeldetag: **26.11.2010**

(43) Offenlegungstag: **31.05.2012**

(51) Int Cl.: **H04L 12/24 (2006.01)**

(71) Anmelder:
**Fujitsu Technology Solutions Intellectual
Property GmbH, 80807, München, DE**

(74) Vertreter:
**Epping Hermann Fischer,
Patentanwalts-gesellschaft mbH, 80339, München,
DE**

(72) Erfinder:
**Linne, Johannes, 86447, Todtenweis, DE;
Schieferdecker, Markus, 86415, Mering, DE**

(56) Für die Beurteilung der Patentfähigkeit in Betracht
gezogene Druckschriften:

US	2004 / 0 186 688	A1
US	2005 / 0 060 526	A1
US	2005 / 0 289 333	A1
US	2007 / 0 011 536	A1
US	6 145 098	A

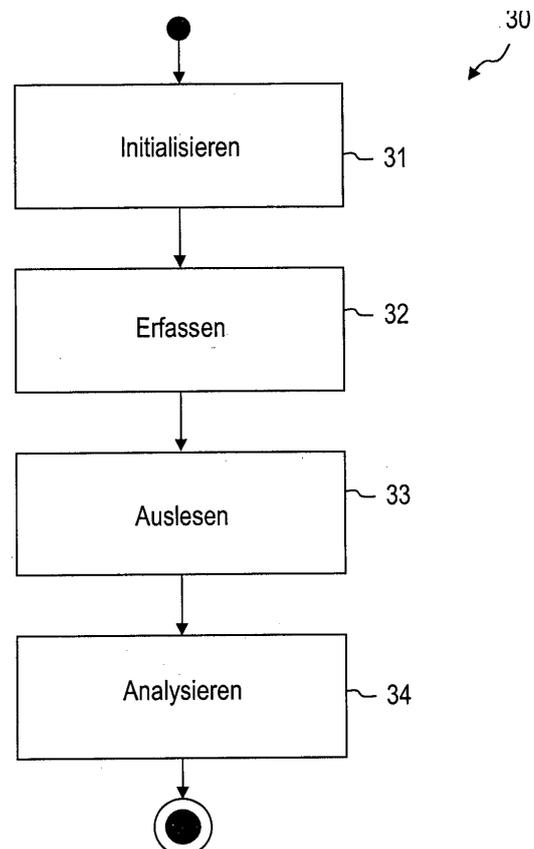
Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Verfahren zur Bestimmung einer Systemstabilität, Computersystem und
Computerprogrammprodukt**

(57) Zusammenfassung: Die Erfindung betrifft ein Verfahren (30) zur Bestimmung einer Systemstabilität einer Systemkonfiguration eines Computersystems (1) mit den Schritten Initialisieren wenigstens eines Bussystems (8, 9, 10, 11, 12) durch eine Bussteuerung unter Verwendung wenigstens einer vorgegebenen, hardware-spezifischen Testroutine, Erfassen wenigstens eines Konfigurationsparameters zum Betrieb des Bussystems (8, 9, 10, 11, 12) in einer aktuellen Konfiguration auf Grundlage der wenigstens einen Testroutine durch eine Firmwarekomponente (21), Auslesen des wenigstens einen erfassten Konfigurationsparameters über eine Schnittstelle (25) der Firmwarekomponente (21) und Bestimmen einer Systemstabilität auf Grundlage des ausgelesenen wenigstens einen Konfigurationsparameters durch eine Analysekomponente (26).

Die Erfindung betrifft außerdem ein Computersystem (1) und ein Computerprogrammprodukt, die zur Implementierung des Verfahrens (30) geeignet sind.



Beschreibung

[0001] Die Erfindung betrifft ein Verfahren zur Bestimmung einer Systemstabilität einer Systemkonfiguration eines Computersystems. Darüber hinaus betrifft die Erfindung ein Computersystem mit wenigstens einem Bussystem mit einer Bussteuerung und wenigstens einer Buskomponente sowie ein Computerprogrammprodukt umfassend Programmcode zur Bestimmung einer Systemstabilität.

[0002] Zur Qualität und Stabilität eines Computersystems tragen die Systemkomponenten des Computersystems und ihre Verbindung miteinander bei. Dabei tragen sowohl so genannten On-Board-Komponenten, die fest auf einer Systemplatine des Computersystems angeordnet sind, als auch Erweiterungskomponenten, die beispielsweise in Bussteckplätzen der Systemplatine eingesteckt sind, zur Gesamtstabilität bei. Aufgrund der Vielzahl unterschiedlicher Systemkomponenten, wie beispielsweise Prozessoren, Speichermodule, Schnittstellenkarten und so weiter, ist in der Regel eine Überprüfung der Systemstabilität des zusammengestellten Gesamtsystems neben der Prüfung der Funktionalität der einzelnen Komponenten erforderlich.

[0003] Computersystemhersteller, die einzelne Komponenten zu einem Gesamtsystem zusammensetzen, führen dazu in der Regel einen so genannten Fertigungsausgangstest durch. Bekannte Fertigungsausgangstests werden als so genannter Go- oder No-Go-Test durchgeführt, bei dem in möglichst kurzer Zeit festgestellt werden soll, ob ein Computersystem in einer konkreten Systemkonfiguration ausgeliefert werden kann oder nicht.

[0004] Dazu wird beispielsweise ein auf dem Computer installiertes Betriebssystem gebootet und innerhalb des Betriebssystems eine Testsoftware gestartet. Die Testsoftware führt dabei funktionale Tests aus, die beispielsweise überprüfen, ob Dateien auf eine Festplatte geschrieben und wieder zurückgelesen werden können.

[0005] Mit derartigen Tests werden in der Regel nur so genannte "harte" Fehler gefunden. Eine Aussage über die Qualität der Systemstabilität oder einzelner Systemkomponenten der Systemkonfiguration kann hierdurch nicht getroffen werden. Zusätzlich ist es aus Zeitgründen im Rahmen von Fertigungsausgangstests in der Regel nicht möglich, weitere Randbedingungen, der Testumgebung, wie beispielsweise eine Betriebstemperatur oder eine Betriebsspannung des Computersystems zu variieren.

[0006] Eine Aufgabe der vorliegenden Erfindung ist es, Verfahren zur Bestimmung einer Systemstabilität einer Systemkonfiguration anzugeben, die eine verbesserte, schnellere und/oder einfachere Aussage

über die Qualität der Systemstabilität als bekannte Fertigungsausgangstests ermöglichen. Darüber hinaus soll ein Computersystem sowie ein Computerprogrammprodukt beschrieben werden, die sich zur Durchführung des Verfahrens eignen.

[0007] Gemäß einer ersten Ausgestaltung wird ein Verfahren zur Bestimmung einer Systemstabilität einer Systemkonfiguration mit den folgenden Schritten vorgeschlagen:

- Initialisieren wenigstens eines Bussystems durch eine Bussteuerung unter Verwendung wenigstens einer vorgegebenen, hardwarespezifischen Testroutine
- Erfassen wenigstens eines Konfigurationsparameters zum Betrieb des Bussystems in einer aktuellen Konfiguration auf Grundlage der wenigstens einen Testroutine durch eine Firmwarekomponente
- Auslesen des wenigstens einen erfassten Konfigurationsparameters über eine Schnittstelle der Firmwarekomponente und
- Bestimmen einer Systemstabilität auf Grundlage des ausgelesenen wenigstens einen Konfigurationsparameter durch eine Analysekomponente:

[0008] Das oben genannte Verfahren macht sich Ergebnisse einer hardwarespezifischen Testroutine zunutze, die während der Initialisierung eines Bussystems durchgeführt wird. Dabei werden Konfigurationsparameter zum Betrieb des Bussystems in einer aktuellen Konfiguration durch eine Firmwarekomponente erfasst und durch eine Analysekomponente ausgewertet. Hierzu greift die Analysekomponente über eine Schnittstelle der Firmwarekomponente auf die gespeicherten Konfigurationsparameter zurück.

[0009] Das oben genannte Verfahren weist unter anderem den Vorteil auf, dass Testroutinen, die zum korrekten Betrieb und zu einer notwendigen Initialisierung eines Bussystems ohnehin ausgeführt werden, auch zur Bestimmung einer Systemstabilität verwendet werden. Dadurch dass die Testroutinen auf einer sehr niedrigen, hardwarespezifischen Ebene des Computersystems ausgeführt werden, können auch solche Fehler oder Mängel einer Systemkonfiguration erkannt werden, die bei einem rein funktionalen Go- oder No-Go-Test bekannter Fertigungsausgangstests nicht erkannt werden können.

[0010] Gemäß einer vorteilhaften Ausgestaltung wird die Systemstabilität auf Grundlage einer Abweichung des ausgelesenen wenigstens einen Konfigurationsparameters von einem vorgegebenen Referenzwert bestimmt. Die Bestimmung einer Abweichung von einem Referenzwert erlaubt die Erkennung so genannter "weicher" Fehler, wie beispielsweise die Erkennung besonders langer Buszugriffszeiten oder niedriger Datenübertragungsraten, die im

Rahmen eines funktionalen Tests nicht erkannt werden.

[0011] Gemäß einer weiteren vorteilhaften Ausgestaltung werden die Schritte des Initialisierens, Erfassens und Auslesens wiederholt durchgeführt und die Systemstabilität wird auf Grundlage einer statistischen Auswertung des wenigstens einen wiederholt ausgelesenen Konfigurationsparameters bestimmt. Die wiederholte Ausführung der Verfahrensschritte zusammen mit einer statistischen Auswertung des ausgelesenen Konfigurationsparameters erlaubt eine Erkennung von Änderungen und insbesondere eine Erfassung einer graduellen Verschlechterung der Systemstabilität. Auf diese Weise können zum Beispiel alterungsbedingte Fehlerquellen des Computersystems rechtzeitig vor einem Totalausfall erkannt werden.

[0012] Gemäß einer weiteren vorteilhaften Ausgestaltung umfasst die vorgegebene, hardware-spezifische Testroutine eine Trainingssequenz zum Trainieren einer Datenübertragung zwischen der Bussteuerung und wenigstens einer mit dem wenigstens einen Bussystem verbundenen Buskomponente. Solche Trainingssequenzen, die insbesondere dazu verwendet werden, ein Bustiming festzulegen, gestatten einen Aufschluss über die korrekte Funktion, Leistungsfähigkeit und Anordnung einzelner Buskomponenten.

[0013] Gemäß einer zweiten Ausgestaltung der Erfindung wird ein Computersystem beschrieben, das wenigstens ein Bussystem mit einer Bussteuerung und wenigstens eine mit dem Bussystem verbundene Buskomponente, wenigstens eine Firmwarekomponente mit Programmcode sowie eine Analysekomponente mit Programmcode umfasst. Die Firmwarekomponente weist Programmcodes zum Initialisieren des Bussystems durch die Bussteuerung unter Verwendung wenigstens einer vorgegebenen, hardware-spezifischen Testroutine umfassend eine Trainingssequenz zum Trainieren einer Datenübertragung zwischen der Bussteuerung und der Buskomponente und zum Erfassen wenigstens eines Konfigurationsparameters zum Betrieb des Bussystems in einer aktuellen Konfiguration des Computersystems auf Grundlage der wenigstens einen Testroutine auf. Die wenigstens eine Analysekomponente weist Programmcode zum Auslesen des wenigstens einen erfassten Konfigurationsparameters über eine Schnittstelle der Firmwarekomponente und zum Bestimmen einer Systemstabilität auf Grundlage des ausgelesenen wenigstens einen Konfigurationsparameters auf.

[0014] Ein derartiges Computersystem eignet sich zur Ausführung des oben genannten Verfahrens. Des Weiteren eignet sich ein derartiges Computersystem insbesondere zur Bestimmung einer Systemstabilität einer aktuellen Systemkonfiguration unabhängig

von einer speziellen Testumgebung. Somit kann die Systemstabilität jederzeit bestimmt werden. Ein derartiges Computersystem eignet sich insbesondere zur fortwährenden Überwachung einer aktuellen Systemkonfiguration, auch noch nach Auslieferung des Computersystems durch dessen Hersteller beim Einsatz des Computersystems durch einen Nutzer.

[0015] Gemäß einer vorteilhaften Ausgestaltung sind die Firmwarekomponente und/oder die Analysekomponente in einer BIOS-Komponente des Computersystems integriert. Dadurch kann das Verfahren zur Bestimmung einer Systemstabilität unabhängig von einem Betriebssystem und weiterer Testsoftware des Computersystems durchgeführt werden.

[0016] Gemäß einer weiteren vorteilhaften Ausgestaltung ist die Analysekomponente in einer Systemmanagementkomponente des Computersystems integriert, die zur Kopplung mit einer Wartungsvorrichtung eingerichtet ist. Ein derartiges Computersystem kann entweder fortwährend oder im Bedarfsfall mit einer Wartungsvorrichtung analysiert werden. Dabei können Rückschlüsse auf harte und weiche Fehler sowie auf Fehlkonfigurationen auf Hardwareebene gezogen werden.

[0017] Gemäß einer dritten Ausgestaltung der vorliegenden Erfindung wird ein Computerprogrammprodukt umfassend Programmcode zur Bestimmung einer Systemstabilität einer Systemkonfiguration beschrieben. Das Computerprogrammprodukt ist dadurch gekennzeichnet, dass bei Ausführung des Programmcodes durch einen Prozessor eines Computersystems, die folgenden Schritte ausgeführt werden:

- Auslesen wenigstens eines während einer Initialisierung wenigstens eines Bussystems durch eine Bussteuerung unter Verwendung wenigstens einer vorgegebenen, hardware-spezifischen Testroutine erfassten Konfigurationsparameters über eine Schnittstelle einer Firmwarekomponente und
- Bestimmen einer Systemstabilität auf Grundlage des ausgelesenen wenigstens einen Konfigurationsparameters.

[0018] Weitere vorteilhafte Ausgestaltungen der Erfindung sind in den abhängigen Patentansprüchen sowie der nachfolgenden ausführlichen Beschreibung von Ausführungsbeispielen offenbart.

[0019] Die Erfindung wird nachfolgend unter Verwendung unterschiedlicher Ausführungsbeispiele unter Bezugnahme auf die Figuren näher erläutert.

[0020] In den Figuren zeigen:

[0021] [Fig. 1](#) eine beispielhafte Systemkonfiguration eines Computersystem,

[0022] [Fig. 2](#) eine Softwarearchitektur zur Implementierung eines Verfahrens zur Bestimmung einer Systemstabilität und

[0023] [Fig. 3](#) ein Ablaufdiagramm eines Verfahrens zur Bestimmung einer Systemstabilität.

[0024] [Fig. 1](#) zeigt schematisch eine beispielhafte Systemkonfiguration eines Computersystems **1**. Das Computersystem **1** umfasst im Ausführungsbeispiel einen Prozessor **2**, einen Hub-Baustein **3** sowie zwei Speichermodule **4a** und **4b**. Des Weiteren umfasst das Computersystem **1** einen BIOS-Baustein **5** und eine so genannte PCI-Bridge **6**. Schließlich umfasst das Computersystem **1** Erweiterungskomponenten **7a** bis **7c**, die in korrespondierenden Bussteckplätzen des Computersystems **1** eingesteckt sind.

[0025] Im Ausführungsbeispiel ist der Prozessor **2** mit dem Hub-Baustein **3** über einen Prozessorbus **8** gekoppelt. Des Weiteren ist der Hub-Baustein **3** über einen Speicherbus **9** mit den Speichermodulen **4a** und **4b** gekoppelt. Der Hub-Baustein **3** ist mit dem BIOS-Baustein **5** über einen Systemmanagementbus **10** und mit der PCI-Bridge **6** über einen Chipsatzbus **11** gekoppelt. Die Erweiterungskomponenten **7a** bis **7c** sind mit der PCI-Bridge **6** über einen PCI-Express-Bus **12** gekoppelt.

[0026] Selbstverständlich ist die oben beschriebene Konfiguration nur beispielhafter Natur. In der Praxis variiert die Art und Anzahl der verwendeten Systemkomponenten abhängig vom Einsatzzweck des Computersystems **1**. Auch die Architektur und Topografie der Bussysteme des Computersystems **1** können in vielfältiger Weise variiert werden. Beispielsweise sind andere Systemarchitekturen bekannt, bei der eine Vielzahl von unterschiedlichen Systemkomponenten an einem gemeinsamen Systembus angeschlossen ist. Die konkrete Anordnung und Ausgestaltung der einzelnen, zum Teil oben beschriebenen Bussysteme hängt von deren jeweiligen Einsatzzweck ab.

[0027] Insbesondere bei modernen, besonders leistungsfähigen Computersystemen müssen die eingesetzten Bussysteme oft sehr hohe Datenübertragungsraten aufweisen. Um gleichzeitig eine Flexibilität bezüglich der einsetzbarer Systemkomponenten zu erreichen, wird eine Datenübertragung zwischen zwei Buskomponenten, in der Regel einer Bussteuerung als Busmaster und einer weiteren Buskomponente als so genannter Slave, bei der ersten Initialisierung des jeweiligen Bussystems trainiert. Auf diese Weise können optimale Konfigurationsparameter zum Betrieb des Bussystems gefunden werden.

[0028] Einzelne oder alle solcher Konfigurationsparameter können auch durch die einzelnen Systemkomponenten vorgegeben werden. Im Ausführungsbeispiel umfassen beispielsweise die Speichermodu-

le **4a** und **4b** jeweils einen nicht-flüchtigen Speicher mit so genannten Serial Presence Detect (SPD) Daten **13a** und **13b**.

[0029] In den SPD-Daten sind Konfigurationsparameter zur automatischen Konfiguration der Speichermodule **4a** und **4b**, insbesondere deren Speicherkapazität, Versorgungsspannung und Timingparameter, wie beispielsweise erforderliche Refresh- und Latenzzeiten, gespeichert. Diese Daten werden von einer Firmwarekomponente des Computersystems **1** während der Initialisierung des Speicherbusses **9** im Rahmen des so genannten Power-on-Self-Tests (POST) abgerufen und zum Konfigurieren eines in dem Hub-Baustein **3** integrierten Speicherbuscontrollers verwendet. Die dabei bestimmten Werte zum Betrieb des Speicherbuscontrollers werden in geeigneten Registern, beispielsweise des Hub-Bausteins **3** oder des BIOS-Bausteins **5** abgelegt. Im Ausführungsbeispiel gemäß [Fig. 1](#) sind beispielhaft die nicht flüchtigen CMOS-Register **17** des BIOS-Bausteins **5** dargestellt.

[0030] Ähnliche Parameter zum Konfigurieren und Betrieb einzelner Komponenten von unterschiedlichen Bussystemen beziehungsweise zugehöriger Bussteuerungskomponenten eines Bussystems, wie beispielsweise der PCI-Bridge **6**, sind auch in den Erweiterungskarten **7a** bis **7c** gespeichert. In der [Fig. 1](#) sind sie als Konfigurationsdaten **14a** bis **14c** dargestellt.

[0031] Darüber hinaus kann das Computersystem **1** Sensoren zum Erfassen von Umgebungsbedingungen umfassen. Im Ausführungsbeispiel umfasst das Computersystem **1** einen Temperatursensor **15** sowie einen Spannungssensor **16** der eine Betriebstemperatur beziehungsweise eine Betriebsspannung eines eingesetzten Systemboards erfasst und über den Systemmanagementbus **10** zur Verfügung stellt.

[0032] In der [Fig. 2](#) ist eine beispielhafte Softwarearchitektur **20** zur Bestimmung einer Systemstabilität des Computersystems gemäß [Fig. 1](#) dargestellt. Im Ausführungsbeispiel umfasst die Softwarearchitektur **20** eine Firmwarekomponente **21** sowie ein Betriebssystem **22**.

[0033] Bei der Firmwarekomponente **21** handelt es sich beispielsweise um eine konventionelle BIOS-Komponente mit Programmcode, der rückwärtskompatibel zum BIOS des ursprünglichen IBM-PCs ist. Alternativ kann es sich bei der Firmwarekomponente **21** auch um eine modular erweiterbare Firmware gemäß dem Extensible Firmware Interface (EFI) handeln.

[0034] Die Firmwarekomponente **21** umfasst im Ausführungsbeispiel eine Bootkomponente **23**, eine Bussteuerungskomponente **24** und eine Analyseschnittstelle **25**. Die Bootkomponente **23** wird als

erstes nach Anlegen einer Betriebsspannung an den Prozessor **2** und den BIOS-Baustein **5** ausgeführt. Die Bootkomponente **23** lädt weitere Komponenten der Firmwarekomponente **21** und gegebenenfalls weitere Firmwarekomponenten anderer Systemkomponenten des Computersystems **1** nach, um das Computersystem **1** und die darin enthaltenen Systemkomponenten zu initialisieren.

[0035] Im Ausführungsbeispiel lädt die Bootkomponente **23** die Bussteuerungskomponente **24** nach, die beispielsweise den Speicherbus **9** initialisiert. Hierzu werden im Ausführungsbeispiel zunächst die SPD-Daten **13a** und **13b** aus den Speichermodulen **4a** und **4b** abgerufen. Nachfolgend werden im Laufe einer so genannten Trainingsphase unterschiedliche Datenübertragungsraten und/oder Speicherzugriffarten zwischen der Speicherbussteuerung und den Speichermodulen **4a** und **4b** getestet. Dabei erfasste Konfigurationsparameter, wie beispielsweise zur Verfügung stehende Datenübertragungsmodi oder deren Latenzzeiten werden in Registern, beispielsweise dem CMOS-Register **17** des BIOS-Bausteins **5** für den weiteren Betrieb des Computersystems **1** hinterlegt. Diese Daten können nachfolgend über die Analyseschchnittstelle **25** abgerufen werden.

[0036] Im Ausführungsbeispiel wird innerhalb des Betriebssystems **22** eine Analysekomponente **26** sowie eine Fernwartungskomponente **27** ausgeführt. Selbstverständlich können die Komponenten **26** und **27** alternativ auch Teil der Firmwarekomponente **21** oder einer davon unabhängigen Überwachungskomponente, wie einem Systemmanagementbaustein, auch bekannt als Baseboard-Management-Controller (BMC), bilden.

[0037] Die Analysekomponente **26** ruft über die Analyseschchnittstelle **25** die von der Bussteuerungskomponente **24** ermittelten Konfigurationsparameter für ein Bussystem ab. Ausgehend von vorgegebenen Referenzwerten, die beispielsweise für unterschiedliche Systemkonfigurationen fest vorgegeben sein können oder im Rahmen einer früheren Bestimmung von Konfigurationsparametern in einer ursprünglichen Systemkonfiguration bestimmt wurden, kann somit eine gegebenenfalls vorhandene Abweichung von gewünschten Sollwerten bestimmt werden. Beispielsweise kann ein bevorstehender Fehler oder eine Fehlkonfiguration des Computersystems **1** erkannt werden, wenn die Latenzzeit zum Ansprechen bestimmter Buskomponenten ansteigt oder bestimmte Datenübertragungsmodi nicht länger zur Verfügung stehen. Selbstverständlich kann auch überprüft werden, ob erkannte Buskomponenten zur Verwendung mit anderen Komponenten des Computersystems **1** grundsätzlich geeignet sind.

[0038] Die dabei erfassten Daten können entweder lokal ausgewertet und zur Anzeige von Warnmeldun-

gen in dem Computersystem **1** eingesetzt werden oder über die Fernwartungskomponente **27** an einen zentralen Wartungsrechner weitergegeben werden. Sofern das Computersystem **1** in ein Netzwerk, beispielsweise in ein Firmennetzwerk, integriert ist, kann diese Weitergabe vollautomatisch im Hintergrund erfolgen. Andernfalls können die Daten durch die Fernwartungskomponente **27** lokal auf einem nichtflüchtigen Speichermedium gespeichert und bei Bedarf an eine entsprechende Wartungskomponente abgegeben werden.

[0039] In der [Fig. 3](#) ist ein Ablaufdiagramm eines Verfahrens **30** zum Bestimmen einer Systemstabilität dargestellt.

[0040] In einem ersten Schritt **31** wird wenigstens ein Bussystem eines Computersystems **1** initialisiert. Beispielsweise können der Prozessorbus **8**, der Speicherbus **9**, der Systemmanagementbus **10**, der Chipsatzbus **11** und/oder der PCI-Express-Bus **12** des Computersystems **1** gemäß [Fig. 1](#) initialisiert werden.

[0041] Zumindest einige Bussysteme, wie beispielsweise ein so genannte Frontsidebus (FSB) des Prozessors **2**, ein Interprozessorbuss wie beispielsweise Intels Quickpath (QPI) oder ein PCI-Express-Bus werden dabei über ein so genanntes Training optimiert. Das heißt, dass derartige Bussysteme bei jedem Neustart beziehungsweise bei jeder Firmware-Initialisierung neu vermessen und initialisiert werden. Dieses Training ist Bestandteil des Programmcodes der Firmwarekomponente **21**, beispielsweise der Bussteuerungskomponente **24**. Zur Durchführung des Trainings werden oftmals von den jeweiligen Herstellern der beteiligten Mikrochips bereits auf Chipebene verwirklichte Testmechanismen verwendet. In diesen Testalgorithmen sind oftmals auch als so genannte Margin-Tests bezeichnete Testalgorithmen enthalten, bei denen Tests unter Variation bestimmter Randbedingungen, wie beispielsweise der Betriebsspannung, oder Variation von Timingwerten durchgeführt werden.

[0042] Als Ergebnis dieser Testtrainings- und Optimierungsphase werden im Schritt **32** Konfigurationsparameter zum Betrieb des jeweiligen Bussystems in einer aktuellen Konfiguration des Computersystems **1** durch eine Firmwarekomponente **21** erfasst.

[0043] Beispielsweise können solche Parameterwerte, die das Timing bestimmter Bussignale oder das Vorhandensein und die Position von in Einsteckplätzen angeordneten Erweiterungskomponenten **7** beschreiben, in Form von geeigneten Registerwerten oder Variablen im Speicher des Computersystems **1** abgelegt werden. Diese Konfigurationsparameter stehen dann nachfolgend für den Betrieb der jeweiligen Buskomponente beziehungsweise des jeweiligen Bussystems zur Verfügung und können dann bei-

spielsweise von einer Buskomponente beziehungsweise einer Bussteuerung ausgelesen werden.

[0044] In einem Schritt **33** werden die erfassten Konfigurationsparameter über eine Schnittstelle einer Firmwarekomponente **21** ausgelesen. Auf diese Weise werden die hardwarespezifischen Ergebnisse der Businitialisierung im Schritt **31** und den darauf basierenden im Schritt **22** erfassten Konfigurationsparameter auch für andere Softwarekomponenten zur Verfügung gestellt. Beispielsweise können weitere so genannte EFI-Erweiterungsmodule der Firmwarekomponente **21** auf die Trainingsergebnisse eines Hardwaredreibers zugreifen. Alternativ können die Ergebnisse auch wie in der [Fig. 2](#) dargestellt einer höheren Softwareebene, beispielsweise einer Betriebssystemebene, bereitgestellt werden.

[0045] Der Vorteil der Verwendung von Ergebnissen hardwarespezifischer Testroutinen liegt unter anderem darin, dass sie aufgrund der hardwarenahen Realisierung besonders schnell durchgeführt werden können, was zu einer Verminderung der erforderlichen Testzeit führt. Des Weiteren stehen damit Testergebnisse zur Verfügung, die auf anderem Wege in der Regel nicht oder nur mit erheblichem Aufwand abgerufen oder reproduziert werden können, da Chiphersteller vielfach die Details der einzelnen Testroutinen nicht vollständig offen legen.

[0046] In einem Schritt **34** wird dann auf Grundlage wenigstens eines ausgelesenen Konfigurationsparameters eine Systemstabilität durch eine Analysekomponente bestimmt. Dabei können die Ergebnisse der Initialisierungs- beziehungsweise Trainingsphase auf verschiedene Weise ausgewertet werden.

[0047] Beispielsweise ist es möglich, für jede gegebene Systemkonfiguration unter einer eindeutigen Kennung die erfassten Konfigurationsparameter abzulegen. Die gespeicherten Konfigurationsparameter können dann insbesondere für Vergleiche mit anderen Systemkonfigurationen, statistische Überwachungen einer Systemstabilität und eine Fehlernachverfolgung verwendet werden. Beispielsweise kann über eine statistische Betrachtung der Lebenszyklus des Computersystems **1** nachverfolgt werden. Ebenso ist es möglich, über eine Korrelation mit Konfigurationsparametern bekannter Fehlersituationen auf eine Fehlkonfiguration des Computersystems **1** zu schließen. Zudem kann aus bestimmten Parametern direkt auf die Qualität der jeweiligen Systemkonfiguration geschlossen werden, beispielsweise über die Auswertung von zur Datenübertragung über ein Bussystem zur Verfügung stehenden Datentransferraten. Auf diese Weise können Systemkonfigurationen eines Computersystems gegebenenfalls schon vor Auslieferung optimiert werden. Dabei werden Konfigurationen vermieden, die zwar noch zu einem funktionsfähigen Computersystem **1** führen, jedoch nicht

die optimale Leistungsfähigkeit der beteiligten Komponenten nutzen.

[0048] Die beschriebenen Verfahrensschritte können entweder einmalig, beispielsweise vor einer Auslieferung eines fertig zusammen gestellten Computersystems **1**, oder wiederholt, beispielsweise im Rahmen des Power-on-Self-Test (POST), ausgeführt werden.

[0049] Bei einem Ausgangstest können mit derartigen Untersuchungen auffällige Systeme aussortiert werden, die zwar einen funktionalen Ausgangstest bestanden haben, jedoch von bekannten Sollwerten bezüglich der erfassten Konfigurationsparameter deutlich abweichen. Auf diese Weise kann einem späteren Ausfall des Computersystems **1** mit nachfolgenden Wartungs- und Gewährleistungsansprüchen vorgebeugt werden. Selbstverständlich kann ein derartiger Ausgangstest bekannte, teilweise zeitaufwändige Ausgangstest auch ganz oder vollständig ersetzen.

[0050] Die beschriebenen Verfahrensabschnitte eignen sich auch zu einer Überwachung der Systemqualität bereits ausgelieferter Computersysteme **1**. Dabei kann die Überwachung sowohl präventiv, beispielsweise zu einem vorgegebenen Zeitpunkt, etwa bei jedem Booten oder einmal im Monat, oder konkret im Problemfall ausgelöst werden. Sollen im Schritt **34** verhältnismäßig aufwändige Analysen durchgeführt werden, kann das Verfahren in größeren zeitlichen Abständen durchgeführt werden.

[0051] Es ist auch möglich das Verfahren beim Auftreten vorbestimmter Systemmanagementinterrupts (SMI) durchzuführen oder auszulösen, wie sie beispielsweise regelmäßig durch einen Zeitgeber oder beim Auftreten schwerer Fehler verursacht werden. Selbstverständlich können auch nur einzelne Teile des Verfahrens, wie beispielsweise der Schritt **33** zum Abrufen der erfassten Konfigurationsparameter, bei jedem Auftreten eines bestimmten Interrupts durchgeführt werden, während der nachfolgende Schritt **34** nur in vorher vorgegebenen Zeitabständen, beispielsweise nach 100 Betriebsstunden oder einmal im Jahr, durchgeführt wird.

[0052] Um die Wartungskosten gering zu halten, empfiehlt es sich, das Verfahren auch mit Mitteln zur Fernwartung auslösen zu können. Hierfür eignet sich neben der bereits bezüglich der [Fig. 2](#) beschriebenen Fernwartungskomponente **27** auf Softwareebene auch ein so genannter Baseboard-Management-Controller (BMC), der einen Fernzugriff unabhängig von dem Prozessor **2** auf Komponenten einer Systemplatine erlaubt. Mittels einer wiederholten oder fortlaufenden Überwachung von Konfigurationsparametern können damit beispielsweise die Alterung von Komponenten, zum Beispiel ausgetrocknete Elektro-

lytkondensatoren oder korrodierte Steckverbindungen, erkannt werden und daraus auf eine mögliche Verschlechterung der Systemstabilität geschlossen werden.

[0053] Des Weiteren erlaubt eine derartige Überwachung auch die aktive Erkennung von Systemkonfigurationsänderungen, beispielsweise durch die nachträgliche Installation unbekannter oder ungeeigneter Komponenten durch einen Kunden. Sofern weitere Betriebsparameter, wie beispielsweise die Temperatur, in die Überwachung einbezogen werden, kann auch erkannt werden, ob oder gegebenenfalls welche Komponenten besonders anfällig auf eine erhöhte oder erniedrigte Betriebstemperatur reagieren und somit zu einer Verminderung der Systemstabilität führen. Schließlich kann auch erkannt werden, ob eine in dem Computersystem **1** eingesetzte Firmwarekomponente, insbesondere eine BIOS-Version, geeignet ist, die in dem Computersystem **1** enthaltenen Komponenten korrekt anzusteuern oder ob gegebenenfalls ein Update auszuführen ist.

Bezugszeichenliste

1	Computersystem
2	Prozessor
3	Hub-Baustein
4	Speichermodul
5	BIOS-Baustein
6	PCI-Bridge
7	Erweiterungskomponente
8	Prozessorbus
9	Speicherbus
10	Systemmanagementbus
11	Chipsatzbus
12	PCI-Express-Bus
13	SPD-Daten
14	Konfigurationsdaten
15	Temperatursensor
16	Spannungssensor
17	CMOS-Register
20	Softwarearchitektur
21	Firmwarekomponente
22	Betriebssystem
23	Bootkomponente
24	Bussteuerungskomponente
25	Analyseschnittstelle
26	Analysekomponente
27	Fernwartungskomponente
30	Verfahren zur Bestimmung einer Systemstabilität

Patentansprüche

1. Verfahren **(30)** zur Bestimmung einer Systemstabilität einer Systemkonfiguration eines Computersystems **(1)** mit den Schritten:
– Initialisieren wenigstens eines Bussystems **(8, 9, 10, 11, 12)** durch eine Bussteuerung unter Verwen-

dung wenigstens einer vorgegebenen, hardware-spezifischen Testroutine;

– Erfassen wenigstens eines Konfigurationsparameters zum Betrieb des Bussystems **(8, 9, 10, 11, 12)** in einer aktuellen Konfiguration auf Grundlage der wenigstens einen Testroutine durch eine Firmwarekomponente **(21)**;

– Auslesen des wenigstens einen erfassten Konfigurationsparameters über eine Schnittstelle **(25)** der Firmwarekomponente **(21)**; und

– Bestimmen einer Systemstabilität auf Grundlage des ausgelesenen wenigstens einen Konfigurationsparameters durch eine Analysekomponente **(26)**.

2. Verfahren **(30)** nach Anspruch 1, dadurch gekennzeichnet, dass die Systemstabilität auf Grundlage einer Abweichung des ausgelesenen wenigstens einen Konfigurationsparameters von einem vorgegebenen Referenzwert bestimmt wird.

3. Verfahren **(30)** nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass die Schritte des Initialisierens, Erfassens und Auslesens wiederholt durchgeführt werden und die Systemstabilität auf Grundlage einer statistischen Auswertung des wenigstens einen wiederholt ausgelesenen Konfigurationsparameters bestimmt wird.

4. Verfahren **(30)** nach Anspruch 3, dadurch gekennzeichnet, dass die Schritte des Initialisierens, Erfassens und Auslesens bei einer Aktivierung des wenigstens einen Bussystems **(8, 9, 10, 11, 12)**, bei einer Veränderung der Systemkonfiguration, bei Auslösen einer Unterbrechungsanforderung oder nach Ablauf eines vorbestimmten Zeitintervalls wiederholt werden.

5. Verfahren **(30)** nach einem der Ansprüche 1 bis 4, dadurch gekennzeichnet, dass die vorgegebene, hardware-spezifische Testroutine eine Trainingssequenz zum Trainieren einer Datenübertragung zwischen der Bussteuerung und wenigstens einer mit dem wenigstens einen Bussystem **(8, 9, 10, 11, 12)** verbundenen Buskomponente umfasst.

6. Verfahren **(30)** nach einem der Ansprüche 1 bis 5, dadurch gekennzeichnet, dass der wenigstens eine Konfigurationsparameter wenigstens einen der folgenden Parameter umfasst: Anzahl von Bussteckplätzen, Art von Bussteckplätzen, Anzahl von Buskomponenten, Art von Buskomponenten, Betriebsart von Buskomponenten, Position von Buskomponenten, Zugriffszeiten auf Buskomponenten, Temperatur von Buskomponenten, Höhe einer Versorgungsspannung des Bussystems **(8, 9, 10, 11, 12)**.

7. Computersystem **(1)**, umfassend:
– wenigstens ein Bussystem **(8, 9, 10, 11, 12)** mit einer Bussteuerung und wenigstens eine mit dem Bussystem verbundene Buskomponente;

– wenigstens eine Firmwarekomponente (**21**) mit Programmcode zum Initialisieren des Bussystems (**8, 9, 10, 11, 12**) durch die Bussteuerung unter Verwendung wenigstens einer vorgegebenen, hardware-spezifischen Testroutine umfassend eine Trainingssequenz zum Trainieren einer Datenübertragung zwischen der Bussteuerung und der Buskomponente und zum Erfassen wenigstens eines Konfigurationsparameters zum Betrieb des Bussystems in einer aktuellen Konfiguration des Computersystems (**1**) auf Grundlage der wenigstens einen Testroutine; und

– wenigstens eine Analysekomponente (**26**) mit Programmcode zum Auslesen des wenigstens einen erfassten Konfigurationsparameters über eine Schnittstelle (**25**) der Firmwarekomponente (**21**) und zum Bestimmen einer Systemstabilität auf Grundlage des ausgelesenen wenigstens einen Konfigurationsparameters.

8. Computersystem (**1**) nach Anspruch 7, dadurch gekennzeichnet, dass die Firmwarekomponente (**21**) und/oder die Analysekomponente (**26**) in einer BIOS-Komponente des Computersystems (**1**) integriert sind.

9. Computersystem (**1**) nach Anspruch 7 oder 8, dadurch gekennzeichnet, dass die Analysekomponente (**26**) in einer Systemmanagementkomponente des Computersystems (**1**) integriert ist, die zur Kopplung mit einer Wartungsvorrichtung eingerichtet ist.

10. Computerprogrammprodukt umfassend Programmcode zur Bestimmung einer Systemstabilität einer Systemkonfiguration, dadurch gekennzeichnet, dass bei Ausführung des Programmcodes durch einen Prozessor (**2**) eines Computersystems (**1**) die folgenden Schritte ausgeführt werden:

– Auslesen wenigstens eines während einer Initialisierung wenigstens eines Bussystems (**8, 9, 10, 11, 12**) durch eine Bussteuerung unter Verwendung wenigstens einer vorgegebenen, hardware-spezifischen Testroutine erfassten Konfigurationsparameters über eine Schnittstelle (**25**) einer Firmwarekomponente (**21**); und

– Bestimmen einer Systemstabilität auf Grundlage des ausgelesenen wenigstens einen Konfigurationsparameters.

Es folgen 3 Blatt Zeichnungen

Anhängende Zeichnungen

FIG 1

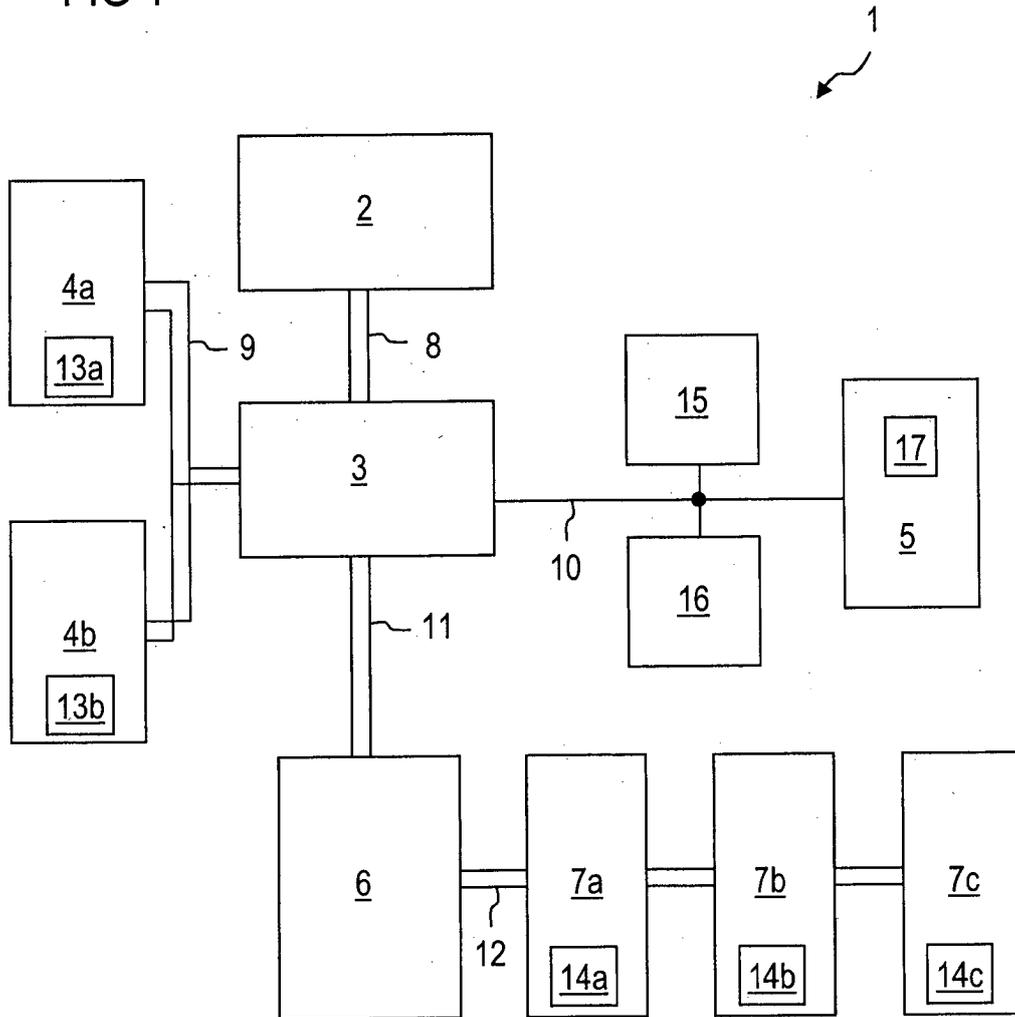


FIG 2

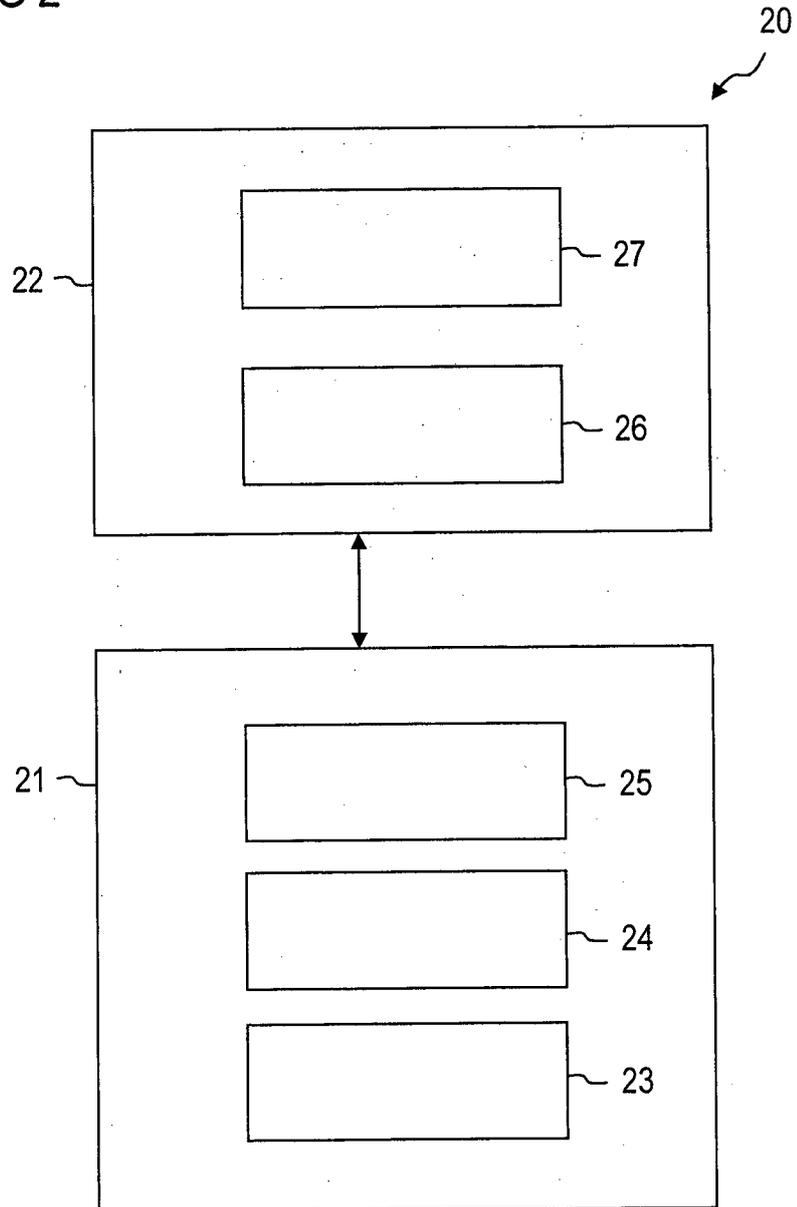


FIG 3

