



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) DE 699 00 971 T2 2004.03.18

(12)

## Übersetzung der europäischen Patentschrift

(97) EP 1 020 798 B1

(51) Int Cl.<sup>7</sup>: G06F 11/267

(21) Deutsches Aktenzeichen: 699 00 971.5

(96) Europäisches Aktenzeichen: 99 114 613.5

(96) Europäischer Anmeldetag: 26.07.1999

(97) Erstveröffentlichung durch das EPA: 19.07.2000

(97) Veröffentlichungstag  
der Patenterteilung beim EPA: 06.03.2002

(47) Veröffentlichungstag im Patentblatt: 18.03.2004

(73) Patentinhaber:

Agilent Technologies Inc., A Delaware Corp., Palo  
Alto, Calif., US

(84) Benannte Vertragsstaaten:

DE, FR, GB

(74) Vertreter:

derzeit kein Vertreter bestellt

(72) Erfinder:

Tiedemann, Herbert, 71134 Aidlingen, DE; Rivoir,  
Jochen, 71106 Magstadt, DE; Wendel, Tilmann,  
71083 Herrenberg, DE

(54) Bezeichnung: Unidirektionale Prüfung von bus-basiertem System

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 eingereicht, 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

### GRUNDLAGEN DER ERFINDUNG

[0001] Die vorliegende Erfindung betrifft ein Prüfsystem zum Prüfen eines busbasierten Systems.

[0002] Moderne Datenverarbeitungssysteme werden oftmals aus einer Vielzahl von einzelnen Komponenten aufgebaut, die mit Hilfe von einem oder mehreren internen oder externen Bussen miteinander verbunden werden. Die einzelnen Komponenten können beispielsweise Zentraleinheiten (CPUs), Speicher, RAID-Teilsysteme, ein Cachespeichersystem, NIC-Karten, Massenspeichereinheiten usw. sein. Verschiedene interne oder externe Busse (im Folgenden nur als Busse bezeichnet) werden im Allgemeinen unter Verwendung von einer oder mehreren Busschnittstellen (miteinander) verbunden. Jede Busschnittstelle (auch als Brücke bezeichnet) ist zwischen (mindestens) zwei Busse oder zwischen einen Bus und z. B. ein CPU-Teilsystem geschaltet, und nur solche Informationen, die für die Weiterleitung auf die 'andere' Seite der Busschnittstelle vorgesehen sind, werden zu dieser weitergeleitet. Zu diesem Zweck "kennt" jede Busschnittstelle die Adresse(n) von einer oder mehreren Komponenten, die mit dem jeweiligen Bus verbunden sind, und/oder die Typen von Übertragungen, die sie weiterleiten muss (im Falle eines PCI-Busses sind diese Informationen Konfigurationszyklen, spezielle Zyklen oder Unterbrechungsbestätigungszyklen).

[0003] **Fig. 1** zeigt ein Beispiel einer typischen Struktur eines busbasierten Systems. Das System umfasst drei Busse **10**, **20** und **30**. Eine Busschnittstelle **40** ist zwischen den Bus **10** und den Bus **20** geschaltet, und eine Busschnittstelle **50** ist zwischen den Bus **20** und den Bus **30** geschaltet. Beispiele für einzelne Komponenten im busbasierten System sind die jeweils mit dem Bus **10** verbundenen Komponenten **52**, **54** und **56**, die jeweils mit dem Bus **20** verbundenen Komponenten **60** und **62** und die jeweils mit dem Bus **30** verbundenen Komponenten **70** und **72**.

[0004] Wenn die Komponente **52** im Betrieb beispielsweise eine Information an die Komponente **72** schreiben möchte oder eine Anforderung diesbezüglich erhält, wird diese Information zuerst auf dem Bus **10** platziert. Die Busschnittstelle **40**, die "weiß", dass die Komponente **72** "irgendwie" mit dem Bus **20** verbunden ist, leitet diese Information zum Bus **20** weiter. Dementsprechend leitet die Busschnittstelle **50**, die "weiß", dass die Komponente **72** mit dem Bus **30** verbunden ist, diese Information weiter zum Bus **30**. Die mit dem Bus **30** verbundene Komponente **72** kann schließlich diese Information empfangen und annehmen. Wenn die Komponente **60** in einem anderen Beispiel beispielsweise aus der Komponente **70** lesen möchte oder eine Anforderung diesbezüglich erhält, initialisiert die Komponente **60** ein Lesesignal auf den Bus **20**. Die Busschnittstelle **40**, die weiß, dass die Komponente **70** nicht mit dem Bus **10** ver-

bunden ist, leitet dieses Lesesignal nicht weiter auf den Bus **10**. Die Busschnittstelle **50**, die weiß, dass die Komponente **70** mit dem Bus **30** verbunden ist, leitet das Lesesignal jedoch weiter zum Bus **30**, und die Komponente **70** empfängt folglich das Lesesignal vom Bus **30**.

[0005] Um die Struktur des busbasierten Systems und auch die Funktionalität und Richtigkeit von diesem zu prüfen, insbesondere von den Busschnittstellen, werden Prüfungen normalerweise im "nackten" System ausgeführt, im Allgemeinen ohne oder nur mit einer verringerten Anzahl von Komponenten, die möglicherweise im System angewandt werden können. Zu diesem Zweck wird das System normalerweise vollständig mit Prüfeinheiten 'bevölkert', z. B. werden diese in jeden freien Steckplatz zum Einfügen von Komponenten eingefügt, so dass jeder Steckplatz und jeder Bus an der Prüfung des Systems teilnehmen können.

[0006] Ein nach dem Stand der Technik bekanntes Prüfsystem wird von der Hewlett-Packard HP E2920 PCI-Reihe und dem HP E2976A Systemprüfpalet verwendet und verwendet mit einzelnen Bussen verbundene Prüfkarten zur Überprüfung des busbasierten Systems. Im Beispiel von **Fig. 1** könnte eine Prüfkarte **80** mit dem Bus **10**, eine Prüfkarte **85** mit dem Bus **20** und eine Prüfkarte **90** mit dem Bus **30** verbunden werden. Zum Überprüfen der Struktur des busbasierten Systems kann jede Prüfkarte Informationen an eine andere Prüfkarte schreiben und schließlich diese Informationen wieder aus dieser anderen Prüfkarte lesen. Falls die aus der anderen Prüfkarte gelesenen Informationen gleich den an diese geschriebenen Informationen sind, kann davon ausgegangen werden, dass die Übertragung der Informationen ohne Fehler ausgeführt wurde und folglich die Struktur insoweit fehlerfrei ist.

[0007] **Fig. 2** zeigt die Grundstruktur der Prüfkarte **80** als Beispiel für alle Prüfkarten **80**, **85** oder **90**. Die Prüfkarte **80** umfasst einen Speicher **100**, einen Sender (der auch als Datenquelle oder Initiator bezeichnet werden könnte) **110**, einen Empfänger (der auch als Datenziel oder Endeinrichtung bezeichnet werden könnte) **120** und einen Vergleicher **130**. Zum Schreiben von Informationen auf den Bus **10** liest der Sender **110** diese Informationen aus dem Speicher **100** und stellt sie auf den Bus **10**. Es ist klar, dass die externe Adresse und die interne Adresse keinen Bezug zueinander haben müssen.

[0008] Zum Vergleichen des Ergebnisses eines Schreibvorgangs auf eine andere Prüfkarte (Richtung des Pfeils I in **Fig. 2**), z. B. auf die Prüfkarte **85** (wie in **Fig. 2** als Verbindung der Prüfkarte **85** mit dem Bus **10** angezeigt wird), leitet der Sender **110** einen Lesevorgang (Richtung des Pfeils II in **Fig. 2**) auf die andere Prüfkarte **85** ein. Die aus der Prüfkarte **85** gelesenen Informationen sind schließlich auf dem Bus **10** verfügbar, und der Empfänger **120** stellt diese Informationen in den Vergleicher **130**, der diese Informationen mit den Informationen vergleicht, die dieser

im Speicher **100** gespeicherten Adresse AD entsprechen.

[0009] Obwohl die Prüfkarten, wie sie in **Fig. 2** dargestellt werden, bereits zu einer komplizierten Überprüfung von busbasierten Systemen führen, sind einige strukturelle Fehler nur schwer zu erkennen. Ein solcher Fehler könnte beispielsweise auftreten, wenn Datenbits während der Datenübertragung ausgetauscht werden, so dass die Daten z. B. zuerst in der einen Richtung und dann wieder zurück in der zweiten Richtung der Übertragung geändert werden. In diesem Fall empfängt der Empfänger Daten, die von der anfänglichen Information vom Sender verschieden sind. Die geänderten Daten werden jedoch erneut geändert (und könnten folglich korrigiert werden), wenn sie ausgelesen werden, so dass die schließlich vom Sender empfangenen Daten 'zufällig' wieder gleich den ursprünglichen Daten sein könnten. In einem solchen Fall würde der Fehler nicht erkannt. Ein anderer Fehler, der nicht gefunden werden kann, ist ein Adressierfehler. Falls sich z. B. die Adresse auf dem Bus **20** auf Grund von Rechenfehlern in der Brücke **40** leicht von der Adresse auf dem Bus **10** unterscheidet, könnte die Prüfeinheit noch immer in der Lage sein, die Daten zu empfangen und wieder zurückzugeben. In der Praxis würde dies in einem System, in dem es unabdingbar ist, dass die Daten korrekt und an der richtigen Stelle eingehen, einen Fehler verursachen.

[0010] US-A-4 622 669 beschreibt ein Prüfmodul zum Suchen und Erkennen von Hardwarefehlern in der Schnittstellenlogik zu einem asynchronen Mikroprozessorbus bei echter Betriebsgeschwindigkeit bis zum Auftreten eines Fehlers. Falls ein Fehler erkannt wird, stoppt die Schaltung einen geprüften Mikroprozessor, der mit dem asynchronen Mikroprozessorbus verbunden ist, und hält den Zustand der Bussignalleitungen fest. Der geprüfte Mikroprozessor weist Bitmustersätze auf. Ein Prüfmikroprozessor weist im internen Speicher gespeicherte Bitmustersätze auf. Prüfmusterzwischenspeicher sind mit dem Prüfmikroprozessor verbunden, um die Prüfmustersätze sequenziell zwischenzuspeichern. Adressbuszwischenspeicher und Datenbuszwischenspeicher sind mit dem asynchronen Bus verbunden, um den Zustand der Adressenleitungen als zu prüfendes Muster zwischenzuspeichern. Ein Vergleicher ist mit dem asynchronen Bus, dem Prüfmikroprozessor, dem ersten Mittel und dem zweiten Mittel verbunden, um eines der Bitmuster mit einem der Prüfmuster zu vergleichen, wobei ein Fortsetzungssignal zum asynchronen Bus und zum Prüfmikroprozessor gesendet wird, wenn das Prüfmuster und das Bitmuster gleich sind, so dass ein anderes der Prüfmuster und ein anderes der Bitmuster verglichen werden können. Ein Diagnosesignal wird zum Prüfmikroprozessor gesendet, wenn das Testmuster und das Bitmuster nicht gleich sind. Fehlerzwischenspeicher sind mit den Adressbuszwischenspeichern und den Datenbuszwischenspeichern, mit dem Prüfmikroprozessor und

mit dem asynchronen Bus verbunden, um das Bitmuster auszugeben, wenn das Diagnosesignal erzeugt wird.

[0011] DE-A-4 333 953 beschreibt ein Verfahren zum Prüfen eines unidirektionalen Datenkanals.

## ZUSAMMENFASSUNG DER ERFINDUNG

[0012] Eine Aufgabe der vorliegenden Erfindung ist eine weitere Verbesserung der Überprüfung von busbasierten Systemen. Die Aufgabe wird durch die unabhängigen Ansprüchen gelöst. Bevorzugte Ausführungsformen werden von den abhängigen Ansprüchen gezeigt.

[0013] Im Gegensatz zum herkömmlichen bidirektionalen Datenübertragungsschema über das Bussystem zu Prüfzwecken (wie durch die Pfeile I und II in **Fig. 2** dargestellt wird) beruht die Erfindung auf einem Konzept einer unidirektionalen Datenübertragung über das zu überprüfende busbasierte System.

[0014] Ein erfindungsgemäßes Prüfsystem stellt eine Sendeeinheit und eine Empfangseinheit bereit, die beide mit mindestens einem Bus im busbasierten System verbunden sind oder durch diesen verbunden werden. Zu Prüfzwecken sendet ein Datengenerator der Sendeeinheit festgelegte Daten zur Empfangseinheit. Die Empfangseinheit vergleicht die von der Sendeeinheit gesendeten Daten (direkt) mit von einem Datengenerator der Empfangseinheit erzeugten Daten, wobei die Datenerzeugung der Empfangseinheit in einem definierten Zusammenhang mit der Datenerzeugung der Sendeeinheit steht. Folglich müssen die (Prüf-) Daten lediglich einmal das busbasierte System durchlaufen und können in und von der Empfangseinheit erkannt und direkt überprüft werden.

[0015] Es gibt verschiedene deterministische Möglichkeiten, den definierten Zusammenhang zwischen der Daten- (Muster-) Erzeugung der Sendeeinheit und der Empfangseinheit bereitzustellen. Eine Möglichkeit könnte ein algorithmischer Zusammenhang sein, wobei die Empfangseinheit unter Verwendung von allgemeinen Informationen oder Kenntnissen, die beiden Einheiten zur Verfügung stehen, ein Datenmuster erzeugt. Folglich können beide Einheiten das Datenmuster der anderen Einheit auf dessen Richtigkeit überprüfen.

[0016] Vorzugsweise beruht der algorithmische Zusammenhang auf der Adresse der Empfangseinheit, da dies eine beiden Einheiten bekannte Information ist. Es sei darauf hingewiesen, dass in fast allen busbasierten Systemen ein eindeutiger Zusammenhang zwischen Adresse und Daten garantiert ist. Die Verwendung der Adresse (als 'Ausgangspunkt') zum Erzeugen von Datenmustern vermeidet außerdem Probleme, falls die Übertragung unterbrochen oder durch Systeme, beispielsweise eine Busschnittstelle, zwischen den beiden Einheiten geteilt wird. Bussysteme garantieren normalerweise nur, dass ein bestimmtes Datenmuster bei einer bestimmten Adresse

ankommt. Normalerweise garantieren sie jedoch nicht, dass die Folge von auszuführenden Übertragungen identisch mit der ursprünglich vom Initiator bereitgestellten Folge ist. Ein Beispiel für die adressenbasierte Mustererzeugung ist z. B., dass das Datenmuster bereits die Adresse der Empfangseinheit selbst oder irgendwie davon abgeleitet ist (z. B. negiert oder zyklisch verschoben).

[0017] Der algorithmische Zusammenhang ermöglicht die Berechnung eines Datenmusters ohne die Verwendung eines kostspieligen Speichers und ermöglicht das Umgehen von Größenbegrenzungen, die automatisch durch die Verwendung von internem Speicher gegeben sind. Ein anderer Vorteil ist die Fähigkeit zum Berechnen des geeigneten Datenmusters, selbst wenn die Übertragung unterbrochen oder die Folge von Übertragungen durch das verbindende System vermischt wurde. Dies kann mit Hilfe des folgenden Beispiels verdeutlicht werden: Eine Karte 1 (Sender) sendet ein Datenpaket A zu einer Adresse A1 und ein Paket B zu einer Adresse B1 aus, wobei sich beide Adressen im selben busbasierten System oder in derselben Einheit befinden. Die verbindende Busschnittstelle (oder Brücke) trennt die Übertragung A in Teilübertragungen Aa und Ab zu den Adressen A1 und A1 b (was A1 plus die Länge von Aa ist). Zwischen diese beiden Übertragungen wird die Übertragung B gesetzt. Bei Betrachtung zweier Busse ist die Folge von Übertragungen auf dem Bus 1A und B, auf dem Bus 2 wären in diesem Beispiel Aa, B und Ab. Das Ergebnis ist dasselbe, die Folge ist jedoch geändert. Die empfangende Karte 2 (Empfänger) muss nun eine Möglichkeit haben, das erwartete Datenmuster aus der Adresse zu ermitteln, in die dieses Muster geschrieben wird. Dies wäre nicht möglich, wenn eine feststehende oder auf andere Art erzeugte Folge von Datenmuster erwartet wird, ohne die Adressinformationen zu berücksichtigen. Der Hardwareaufwand einer solchen Vorgehensweise ist sehr gering ohne zusätzliche Verbindung (die zu einem komplexen Netz führen würde, falls mehrere Prüfkarten in verschiedenen Bussen verwendet werden).

[0018] Eine andere Lösung könnte eine (direkte) Verbindung zwischen den entsprechenden Datengeneratoren außerhalb des busbasierten Systems sein. Eine solche Verbindung könnte eine (Hardware-) Verbindung z. B. über entsprechende Verbindungsleitungen oder eine drahtlose Kommunikation zwischen den Datengeneratoren sein.

[0019] Eine weitere Lösung könnte bereitgestellt werden, indem eine Vorinformation von der Sendeeinheit zur Empfangseinheit gesendet wird, die die Empfangseinheit zum Erzeugen eines definierten Datenmusters anweist, das später mit einem von der Sendeeinheit aufeinanderfolgend gesendeten Datenmuster verglichen werden muss.

[0020] Eine weitere Möglichkeit zum Einrichten eines definierten Zusammenhangs zwischen den entsprechenden Datenerzeugungen könnte darin bestehen, nur vordefinierte Daten zu verwenden, so dass

die Empfangseinheit im Voraus weiß, welches Datenmuster von der Sendeeinheit gesendet wird. Dies kann z. B. erreicht werden, indem stets dasselbe Datenmuster wird oder verschiedene Datenmuster verwendet werden, jedoch mit einem definierten Zusammenhang zwischen ihnen (z. B. das Datenmuster 1 zum Zeitpunkt 1, das Datenmuster 2 zum Zeitpunkt 2 und so weiter). Vorzugsweise könnten die Zeitzusammenhänge unter Verwendung der Zeit ermittelt werden, die z. B. seit einem Urlade- oder Einschaltvorgang verstrichen ist, oder aus der Menge von übertragenen Daten.

[0021] Noch eine weitere Lösung könnte die Bereitstellung eines definierten zeitlichen Zusammenhangs zwischen der Datenerzeugung der jeweiligen Einheiten sein. Dies kann z. B. erreicht werden, indem ein feststehender Zusammenhang für die zeitliche Steuerung der Erzeugung der Datenmuster verwendet wird, so dass es für die Empfangseinheit zu einem bestimmten Zeitpunkt klar ist, welches Datenmuster von der Sendeeinheit erwartet wird.

[0022] Die Empfangseinheit vergleicht das von der Sendeeinheit gesendete empfangene Datenmuster mit dem von der Empfangseinheit empfangenen Datenmuster. Folglich kann die Empfangseinheit Rückschlüsse ziehen, ob die Weiterleitung der Informationen im busbasierten System ordnungsgemäß funktioniert oder, mit anderen Worten, ob die Daten zur richtigen Adresse gelangen. Außerdem können Datenfehler im busbasierten System ausgewertet werden, oder, mit anderen Worten, es kann überprüft werden, ob Daten in einem busbasierten System geändert werden.

[0023] Durch die Bereitstellung eines unidirektionalen Flusses der Prüfdaten verringert die Erfindung die Prüfzeit deutlich, die in busbasierten Systemen mit einer Vielzahl von verschiedenen Adressen und auszuführenden Prüfungen besonders lang ist. Die Erfindung ermöglicht außerdem eine Verringerung des für das gesamte Prüfsystem erforderlichen Aufwands. Die zum Prüfen benötigte Zeit ist nicht nur kürzer (auf Grund der Datenübertragung in nur eine Richtung), sondern es können auch zwei Prüfungen gleichzeitig ausgeführt werden, d. h. die Richtigkeit der Daten und die korrekte Adressierung.

[0024] In einem Beispiel soll ein System mit insgesamt sechzehn möglichen Steckplätzen zum Einfügen von Karten geprüft werden. Jede Prüfkarte zum Überprüfen des Systems muss in der Lage sein, jede andere der verbleibenden neunzehn Prüfkarten zu prüfen. In einer herkömmlichen Anwendung müsste jede Karte mit einem internen Speicher von mindestens 1 MB versehen werden, wobei 64 kB ausschließlich für jede Prüfkarte reserviert werden. Bei Hochgeschwindigkeitsspeichern würde dies bei gegenwärtig etwa DM 100,- je Prüfkarte eine Summe von DM 2000,- für das gesamte Prüfsystem ergeben. Dieser Speicherplatz wird in Prüfsystemen gemäß der Erfindung nicht benötigt (und die DM 2000,- könnten in diesem Beispiel eingespart werden), da das Prüf-

muster nicht gespeichert werden muss, sondern unabhängig auf der Grundlage von allgemeinen Kenntnissen zwischen Sender und Empfänger erzeugt werden kann.

[0025] Eine andere wesentliche Verbesserung, die von der Erfindung bereitgestellt wird, besteht darin, dass ein fast unbegrenzter Adressbereich geprüft werden kann, da ein Datengenerator problemlos z. B. einen 32- oder sogar 64-Bit-Adressbereich mit eindeutigen Datenmustern hinter jeder Adresse darstellen kann, was mit einem herkömmlichen Speicher entweder extrem kostspielig oder unmöglich auszuführen wäre. Folglich können Adressbereiche geprüft werden, die mit den nach dem Stand der Technik bekannten Prüfverfahren unter wirtschaftlichen Gesichtspunkten nicht angemessen geprüft werden könnten.

[0026] Es ist klar, dass die Begriffe Sende- oder Empfangseinheit sich nicht unbedingt auf einzelne, physisch getrennte Einheiten beziehen müssen, sondern auch auf Einheiten angewandt werden können, in denen die sendende und/oder empfangende Funktionalität gemäß der vorliegenden Erfindung als eine Teilfunktion oder ein Teilmodul ausgeführt werden. Die Erfindung ermöglicht es, dass eine Datenerzeugungs- und/oder Datenvergleichseinheit gemäß der Erfindung in Teilen oder Komponenten des Systems realisiert werden kann, z. B. in einer Busschnittstelle, einer Brücke, einem speziellen Chip oder einem Chipsatz, der mit einem Systemspeicher verbunden werden könnte. Folglich kann jede Schnittstelle zwischen diesem Teil oder dieser Komponente und einem anderen Prüfempfänger geprüft werden, selbst wenn eine außerhalb befindliche Karte normalerweise nicht auf diese zugreifen kann. Auf diese Weise ist die Prüfung nicht auf extern zugängliche Steckplätze z. B. für Prüfkarten begrenzt, sondern diese Einheiten mit (integrierter) Prüffähigkeit gemäß der Endung können aktiv auf andere Einheiten zugreifen, als wären sie z. B. spezifische Prüfkarten (da ihr Verhalten, das Erzeugen oder Annehmen von Datenmustern eines bestimmten Typs, bekannt ist).

[0027] Der hierin verwendete Begriff 'Bus' muss im weitest möglichen Sinne des Wortes verstanden werden und könnte jede Art von Verbindung (verdrahtet oder drahtlos; elektrisch, optisch, magnetisch, akustisch – in jedem Wellenlängenbereich) bedeuten, gleichgültig ob sie in ihrer einfachsten Form oder als eine Kombination aus einer Vielzahl von einzelnen Einheiten, Geräten oder Komponenten vorliegt. Typische Beispiele von Bussen könnten der AGP-Bus, der PCI-Bus, der PCI-X-Bus, der EISA-Bus, der ISA-Bus, Firewire usw. sein. In diesem Sinne könnte sogar eine serielle Verbindung zwischen zwei oder mehr Einheiten einen 'Bus' darstellen.

#### KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0028] Andere Aufgaben und viele der dazugehörigen Vorteile der vorliegenden Erfindung werden pro-

blemlos und besser mit Bezugnahme auf die folgende ausführliche Beschreibung verstanden, wenn sie in Verbindung mit den begleitenden Zeichnungen betrachtet wird. Merkmale, die im Wesentlichen oder funktionell gleich oder ähnlich sind, werden mit demselben (denselben) Bezugszeichen bezeichnet.

[0029] **Fig. 1** zeigt ein Beispiel einer typischen Struktur eines busbasierten Systems,

[0030] **Fig. 2** zeigt die Grundstruktur einer Prüfkarte

**80**, wie sie nach dem Stand der Technik bekannt ist,

[0031] **Fig. 3** zeigt die Grundstruktur des Überprüfungsschemas eines busbasierten Systems gemäß der vorliegenden Erfindung,

[0032] **Fig. 4** zeigt eine Ausführungsform einer Prüfeinheit **300** gemäß der vorliegenden Erfindung, und

[0033] **Fig. 5** zeigt eine Ausführungsform eines generischen Datengenerators **250**.

#### AUSFÜHRLICHE BESCHREIBUNG DER ERFINDUNG

[0034] **Fig. 3** zeigt die Grundstruktur des Überprüfungsschemas eines busbasierten Systems gemäß der vorliegenden Erfindung. Eine Sendeeinheit **200** und eine Empfangseinheit **210** sind beide mit einem Bus **220** verbunden. Der Bus **220** kann ein einzelner Bus oder ein Bussystem mit einer Vielzahl von einzelnen Bussen sein, wie in **Fig. 1** dargestellt wird. Die Sendeeinheit **220** umfasst einen Datengenerator **230** und einen Sender (Master oder Initiator) **240**. Die Empfangseinheit **210** umfasst einen Datengenerator **250**, einen Vergleicher **260** und einen Empfänger (Ziel oder Endeinrichtung) **270**.

[0035] Im Betrieb leitet der Initiator einer Übertragung (der die Sendeeinheit **200** oder die Empfangseinheit **210** sein kann) eine Transaktion ein, und der andere Agent (in diesem Fall die Empfangseinheit **210** oder die Sendeeinheit **200**) nimmt an dieser Übertragung teil. Die Richtung der Übertragung hängt vom Initiator ab, der einen Lese- oder einen Schreibbefehl ausgeben kann. Der Sender **240** sendet die Daten aus, während der Empfänger **270** diese Transaktion durch Empfangen der Daten erfüllt. In einer PCI-Umgebung muss der Initiator, der Daten aus einem Ziel lesen möchte, z. B. diese Daten stets anfordern, bis der Empfänger die angeforderten Daten schließlich bereitstellt (im Falle eines Lesevorgangs). In einer PCI-X-Umgebung sendet der Empfänger die Daten unabhängig und automatisch zum Sender zurück, um die Anforderung zu beenden.

[0036] Ausführlicher beschrieben, platziert der Sender **240** der Sendeeinheit **200** ein vom Datengenerator **230** erzeugtes und an die Empfangseinheit **210** adressiertes Datenmuster auf den Bus **220**. Der Empfänger **270** der Empfangseinheit **210** empfängt das (von der Sendeeinheit über den Bus **220** gesendete) Datenmuster und leitet diese empfangenen Daten zum Vergleicher **260** weiter. Der Vergleicher **260** empfängt außerdem ein vom Datengenerator **250** der Empfangseinheit **210** erzeugtes Datenmuster

und vergleicht diese beiden Datenmuster. Die Erzeugung des Datenmusters durch den Datengenerator **250** wird in einem definierten und deterministischen Zusammenhang mit der Erzeugung des Datenmusters durch den Datengenerator **230** der Sendeeinheit **200** ausgeführt. Der definierte Zusammenhang zwischen der Datenerzeugung des Datengenerators **230** und des Datengenerators **250** wird in **Fig. 3** durch eine gepunktete Linie **280** angedeutet.

[0037] Wenn die beiden Datenmuster im Vergleicher **260** übereinstimmen, hat die Datenübertragung von der Sendeeinheit **200** zur Empfangseinheit **210** ohne Fehler stattgefunden. Andernfalls ist es klar, dass bei der Datenübertragung zwischen der Sendeeinheit **200** und der Empfangseinheit **210** ein Fehler aufgetreten ist und die Quelle dieses Fehlers zurückverfolgt werden kann.

[0038] In einer bevorzugten Ausführungsform und nach der Ausführung einer Prüfung erhält der Empfänger **210** eine Anfrage (z. B. von einer Prüfsoftware) zur Übergabe von Daten (z. B. zur Prüfsoftware), ob der Empfänger **210** die vom Sender **200** gesendeten Daten empfangen hat, um welche empfangenen Daten es sich handelte oder was sie enthielten (z. B. wie viele Bytes) und ob vom Empfänger **210** ein Fehler festgestellt wurde. Falls auf diese Weise irgendwelche Fehler oder Unstimmigkeiten erkannt werden, wird ein Bericht erstellt, und es kann ein Alarm signal ausgegeben werden. Dies hängt von der tatsächlichen Prüfrealisierung ab.

[0039] In einer anderen bevorzugten Ausführungsform wird der definierte Zusammenhang zwischen den Datengeneratoren **230** und **250** durch eine Datenerzeugung auf der Grundlage der Adresse der Empfangseinheit **210** bereitgestellt. Die Adresse der Empfangseinheit **210** wird sowohl vom Datengenerator **230** der Sendeeinheit **200** als auch vom Datengenerator **250** der Empfangseinheit **210** für die Erzeugung eines injektiven (oder bijektiven) Datenmusters verwendet. Ausführliche Beispiele werden an späterer Stelle angegeben.

[0040] **Fig. 4** zeigt eine Ausführungsform einer Prüfeinheit **300** gemäß der vorliegenden Erfindung. Im Gegensatz zu der in **Fig. 3** dargestellten Empfangseinheit **210** umfasst die Prüfeinheit **300** von **Fig. 4** sowohl eine Sende- als auch eine Empfangsfunktionalität. Folglich kann die Prüfeinheit **300** die Funktionalität der Sendeeinheit **200** sowie der Empfangseinheit **210** bereitstellen. Die Prüfeinheit **300** umfasst den Empfänger **270** und den Datengenerator **250**, die beide mit dem Vergleichen **260** verbunden sind. Die Prüfeinheit **300** umfasst außerdem einen mit dem Datengenerator **250** verbundenen Sender **310**, um beispielsweise vom Datengenerator **250** erzeugte Daten zum Bus **220** zu senden. Folglich stellt jede Prüfeinheit **300** eine unabhängige Prüfeinheit bereit, so dass eine Überprüfung des busbasierten Systems in beide Richtungen ausgeführt werden kann.

[0041] Im Beispiel von **Fig. 4** kann die Prüfung eines busbasierten Systems von der Prüfeinheit **300** zu

(Richtung von Pfeil A) einer Prüfeinheit **350** (die im Wesentlichen entsprechend der Prüfeinheit **300** aufgebaut ist) und umgekehrt bereitgestellt werden (Richtung von Pfeil B).

[0042] Allgemein gesprochen werden die Datenmuster auf dem Bus **220** von einer Datenquelle (d. h. entsprechend der Sendeeinheit **200** oder der Prüfeinheit **300**) unter Verwendung eines Datengenerators erzeugt. Die Datenmuster werden auf Seiten von (mindestens) einem anderen Teilnehmer (z. B. entsprechend der Prüfeinheit **300** oder der Empfangseinheit **210**) der Übertragung in Bezug auf eine identische (insofern als der Empfänger das ursprünglich gesendete Datenmuster ermitteln kann) oder eine entsprechend zusammenhängende (insofern als der Empfänger nur ein dem ursprünglich gesendeten Datenmuster entsprechendes Datenmuster ermitteln kann) Datenquelle überprüft.

[0043] Es ist klar, dass es nicht erforderlich ist, dass der Empfänger genau dasselbe Datenmuster wie das vom Sender gesendete Datenmuster erzeugt. Es ist lediglich erforderlich, dass der Empfänger irgendwie das empfangene mit dem eigenen erzeugten Datenmuster vergleichen kann, um zu überprüfen, ob das ursprüngliche vom Sender gesendete Datenmuster während der Übertragung vom Sender zum Empfänger geändert wurde.

[0044] In einer Ausführungsform sind die Datengeneratoren des Senders und des Empfängers identisch aufgebaut. In einer anderen Ausführungsform verknüpft der Empfänger (z. B. EXKLUSIV-ODER) eine negierte Version des erwarteten Senderdatenmusters mit dem tatsächlich empfangenen Senderdatenmuster.

[0045] Es gibt verschiedene Möglichkeiten zum Erzeugen deterministischer Datenmuster. Vorzugsweise wird ein definierter Eingangspunkt in die Folge von Datenmustern angewandt. Dies ist insbesondere hilfreich, da die Übertragung zwischen den beiden Agenten zu jedem Zeitpunkt vom zugrundeliegenden System unterbrochen werden kann (z. B. durch die Brücken oder das Teilsystem zwischen den beiden Bussen, wo sich die beiden Agenten physisch befinden), und die Prüfkarte ein automatisches System zur Wiederherstellung aus diesen Unterbrechungen benötigt. Vorzugsweise wird die (Sende-) Adresse des Empfängers der Informationen als dieser Eingangspunkt verwendet, um ein Datenmuster zu erzeugen, dass wiederholbar ist und problemlos vom anderen Teilnehmer überprüft werden kann.

[0046] Im Folgenden werden Beispiele für verschiedene Vorgehensweisen zum Erzeugen von Datenmustern auf der Grundlage der Adresse angegeben:

1. Die Adresse selbst wird als Muster verwendet, oder mit anderen Worten, die Adresse stellt die Daten dar oder umgekehrt. In diesem Fall vergleicht der empfangende Teilnehmer einfach das empfangene Datenmuster mit seiner eigenen Adresse.
2. Ein Datenmuster wird mit einem algorithmi-

schen Zusammenhang mit der Adresse erzeugt. 3. Bits der Adresse werden ausgetauscht oder verwürfelt (auf wiederholbare Weise). In einem Beispiel: Bit 1 wird auf Platz 17 ausgetauscht, Bit 2 auf Platz 30, Bit 3 auf Platz 1, Bit 4 auf Platz 9, Bit 5 auf Platz ... und so weiter bis Bit 64 (oder Bit 32 in einem reinen 32-Bit-System).

4. Die Adresse wird negiert.

5. Ein interner Speicher wird verwendet und mit einem willkürlichen Datenmuster gefüllt. Jedes Datenmuster ist geeignet, die Größe kann jedoch Einschränkungen unterliegen.

6. Es können Zähler verwendet werden, die gemäß Teilen (z. B. bis hin zu allen Bits) der Adresse gesteuert oder vorgegeben werden. Zu möglichen Realisierungen gehören laufende Einsen/Nullen, Umschaltwerte (z. B. 0000/FFFF oder 5555/AAAA - im hexadezimalen Code - oder ähnliche), Graycode- oder ähnliche Coderealisierungen.

[0047] **Fig. 5** stellt eine Ausführungsform eines generischen Datengenerators **250** zum Erzeugen eines Datenmusters gemäß eines Adressmusters (Vektor A[0] bis A[63]) und eines vorprogrammierten Satzes von Registern (Vektor Prog[i,0] bis Prog[i,63] für ein Datenbit D[i]). Im Falle eines 64-Bit-Busses (z. B. PCI, PCI-X) wären 65 Register mit einer Breite von 64 Bit erforderlich. **Fig. 5** zeigt eine logische Schaltung zum Erzeugen des Datenbit D[i]. Jedes einzelne Datenbit D[i] wird von einer unabhängigen Schaltung gemäß **Fig. 5** bereitgestellt, und jedes Datenbit D[i] hängt von allen Adressenanschlüssen (Vektor A[0] bis A[63] als vollständige 64-Bit-Adresse) und von einem Programmierregister ab (Vektor Prog[i], ebenfalls als 64-Bit-Datenwort mit Prog[i,0] bis Prog[i,63]). Jene 64 Programmierregister zusammen mit dem Negierungsregister Neg[i] müssen auf jeder Prüfkarte im System im Wesentlichen identisch sein.

[0048] Die Abhängigkeit des Datenmusters der Adressbits ((Vektor A[0] bis A[63]) in **Fig. 5** ist programmierbar und folglich äußerst flexibel. Beispielsweise erfolgt die Erzeugung der negierten Adresse als Datenmuster auf eine Weise, dass Neg[i] auf 1 gesetzt wird, Prog[i,j] ebenfalls auf 1 gesetzt wird und alle anderen Prog[j,k], wobei j nicht gleich k ist, auf 0 gesetzt werden.

[0049] Es muss verstanden werden, dass jede Prüfeinheit **300** oder Empfangseinheit **210** eine Vielzahl von verschiedenen Adressen verwalten kann, so dass nur eine Prüfeinheit **300** oder Empfangseinheit **210** erforderlich wäre, um fast einen vollständigen Adressbereich zu prüfen. Folglich kann ein fast unbegrenzter Bereich von Adressen geprüft werden, ohne dass zusätzliche Hardware, beispielsweise kostspielige Speicher, benötigt wird.

[0050] In einer bevorzugten Ausführungsform der Erfindung werden eine Datenerzeugungseinheit (beispielsweise der Datengenerator **230** und/oder **250**) und/oder einen Datenvergleichseinheit (beispielsweise der Vergleicher **260**) gemäß der Erfindung als Teil-

le oder Komponenten eines busbasierten Systems realisiert. Im Beispiel von **Fig. 1** umfasst z. B. die Busschnittstelle **40** oder irgendeine andere der Komponenten **52** bis **56**, **60** bis **62** oder **70** bis **72** mindestens eine zusätzliche Funktionseinheit mit dem Datengenerator **230**, um die Datensendefunktionalität der Sendeeinheit **200** oder der Prüfeinheit **300** bereitzustellen, oder den Datengenerator **250** zusammen mit dem Vergleicher **260**, um die Datenempfangsfunktionalität der Sendeeinheit **210** oder der Prüfeinheit **300** bereitzustellen. Weitere Umschalteinheiten zum Umschalten von der ursprünglichen Funktionalität der Einheit zur zusätzlichen Prüffunktionalität könnten bereitgestellt werden, wenn sie von der jeweiligen Anwendung benötigt würden.

[0051] Das folgende spezifischere Beispiel verwendet als Datenmuster die negierte Adresse. Es wird von zwei Agenten in einem System ausgegangen, einer Prüfkarte TC1 (z. B. die Prüfeinheit **300**) und einer Prüfkarte TC2 (z. B. die Prüfeinheit **350**). Die Prüfkarte TC1 möchte  $2 \times 32$  Bit auf die Prüfkarte TC2 schreiben. Die Prüfkarte TC2 hat eine Startadresse 0x000080000 (hexadezimal), die übertragenen Daten (negiert) sind 0xffff7fff und 0xffff7ffb. Der Grund für die Verschiebung 'b' des zweiten Datenmusters ist, dass die Adresse auf Grund der Tatsache, dass sie in Bytes gezählt wird und vier Bytes automatisch mit einer Übertragung übertragen werden, um den Wert 4 springt. Eine Brücke (oder ein Teilsystem) zwischen diesen beiden Agenten nimmt die Übertragung in einem geschlossenen Übertragungsblock entgegen.

[0052] Um die Daten zur Prüfkarte TC2 zu senden, verwendet die Brücke zwei kleinere Datenpakete, eines mit der Adresse 0x000080000 und dem Datenelement 0xffff7fff und ein anderes mit der Adresse 0x00008004 und dem Datenelement 0xffff7ffb. Die Prüfkarte TC2 nimmt beide Pakete an (sogar auf die Folge dieser beiden Pakete kommt es nicht an) und kann nun die Regel anwenden, wie die Daten aus der Adresse erzeugt werden können (in diesem Fall: negieren), um die Richtigkeit des empfangenen Datenmusters zu überprüfen. Die in **Fig. 5** beschriebenen Register haben in diesem Fall die folgenden Werte: Neg wird mit 0xffffffff gefüllt, Prog[i,j] ist 0 für alle  $j < i$ . Mit anderen Worten, Prog[0] = 0x00000001, Prog[1] = 0x00000002, Prog[3] = 0x00000004, Prog[4] = 0x00000008 und so weiter bis Prog[31] = 0x80000000 (Prog[32] bis Prog[63] = 0x00000000). Diese Werte müssen entweder vor der Prüfung programmiert werden, oder es kann in der Prüfkarte voreingestellt werden, dass sie automatisch vom Einschaltzeitpunkt an verwendet oder programmiert werden. Dies ist realisierungsspezifisch.

## Patentansprüche

1. Prüfeinheit (**210**; **300**) zum Prüfen einer Datenübertragung oder eines Adressierschemas in einem busbasierten System, wobei die Prüfeinheit (**210**;

**300)** eine Empfangseinheit (210; 300) umfasst, die für das Empfangen eines Datenmusters auf ein von einer Sendeeinheit (200; 300) gesendetes Senderdatenmuster innerhalb des Systems hin geeignet ist, wobei die Empfangseinheit (210; 300) Folgendes umfasst:

einen Vergleicher (260) zum Vergleichen des erwarteten Datenmusters mit dem empfangenen Datenmuster,

**dadurch gekennzeichnet**, dass die Empfangseinheit außerdem Folgendes umfasst:

einen Datengenerator (250) zum Erzeugen eines erwarteten Datenmusters, und

wobei der Datengenerator (250) für das Erzeugen des erwarteten Datenmusters in einem definierten Zusammenhang mit der Datenerzeugung des Senderdatenmusters geeignet ist, um Rückschlüsse zu ziehen, ob die Weiterleitung des Datenmusters im System ordnungsgemäß funktioniert oder ob Daten im System geändert werden, und wobei die Erzeugung des erwarteten Datenmusters auf allgemeinen Kenntnissen zwischen der Prüfeinheit (210; 300) und der Sendereinheit (200; 300) beruht.

2. Prüfeinheit (210; 300) nach Anspruch 1, wobei der definierte Zusammenhang ein algorithmischer Zusammenhang ist.

3. Prüfeinheit (210; 300) nach Anspruch 1 oder 2, wobei der definierte oder algorithmische Zusammenhang auf der Adresse der Prüfeinheit (210; 300) beruht, vorzugsweise insofern, als das erwartete Datenmuster die Adresse der Empfangseinheit der Prüfeinheit (210; 300) oder davon abgeleitet ist.

4. Prüfeinheit (210; 300) nach Anspruch 1, wobei der definierte Zusammenhang von einer vom busbasierten System unabhängigen Verbindung zwischen der Prüfeinheit (210; 300) und der Sendereinheit (200; 300) bereitgestellt wird, die vorzugsweise verdrahtet oder drahtlos ist.

5. Prüfeinheit (210; 300) nach Anspruch 1, wobei der definierte Zusammenhang bereitgestellt wird, indem eine Vorinformation von der Sendeeinheit (200; 300) zur Prüfeinheit (210; 300) gesendet wird, die die Prüfeinheit (210; 300) anweist, das erwartete Datenmuster zu erzeugen, das mit dem später von der Sendeeinheit gesendeten Datenmuster verglichen werden muss.

6. Prüfeinheit (210; 300) nach Anspruch 1, wobei der definierte Zusammenhang dadurch bereitgestellt wird, dass nur vordefinierte Informationen zum Erzeugen des erwarteten Datenmusters verwendet werden oder ein definierter Zeitzusammenhang zwischen der Datenerzeugung der Empfangseinheit und der Sendeeinheit verwendet wird.

7. Prüfeinheit (210; 300) nach irgendeinem der

obigen Ansprüche, die außerdem Folgendes umfasst:

einen Empfänger (270), der mit einem Bus des busbasierten Systems verbunden ist, um eine von einem anderen Teilnehmer des busbasierten Systems, zum Beispiel der Sendeeinheit, angeforderte Transaktion zu erfüllen, wobei der Vergleicher (260) mit dem Empfänger (270) und mit dem Datengenerator (250) verbunden ist.

8. Prüfeinheit (210; 300) nach Anspruch 7, die außerdem Folgendes umfasst:

einen Sender (310) zum Senden von vom Datengenerator erzeugten Daten zum Bus (220).

9. System zum Prüfen einer Datenübertragung oder eines Adressierschemas in einem busbasierten System, das Folgendes umfasst: eine Prüfeinheit (210; 300) nach irgendeinem der obigen Ansprüche, und

eine Sendeeinheit (200; 300) zum Erzeugen eines Senderdatenmusters und zum Senden des Senderdatenmusters zu einer Prüfeinheit (210; 300).

10. Verfahren zum Prüfen einer Datenübertragung oder eines Adressierschemas zwischen einer Sendeeinheit (200; 300) und einer Empfangseinheit (210; 300) in einem busbasierten System, das die folgenden Schritte umfasst:

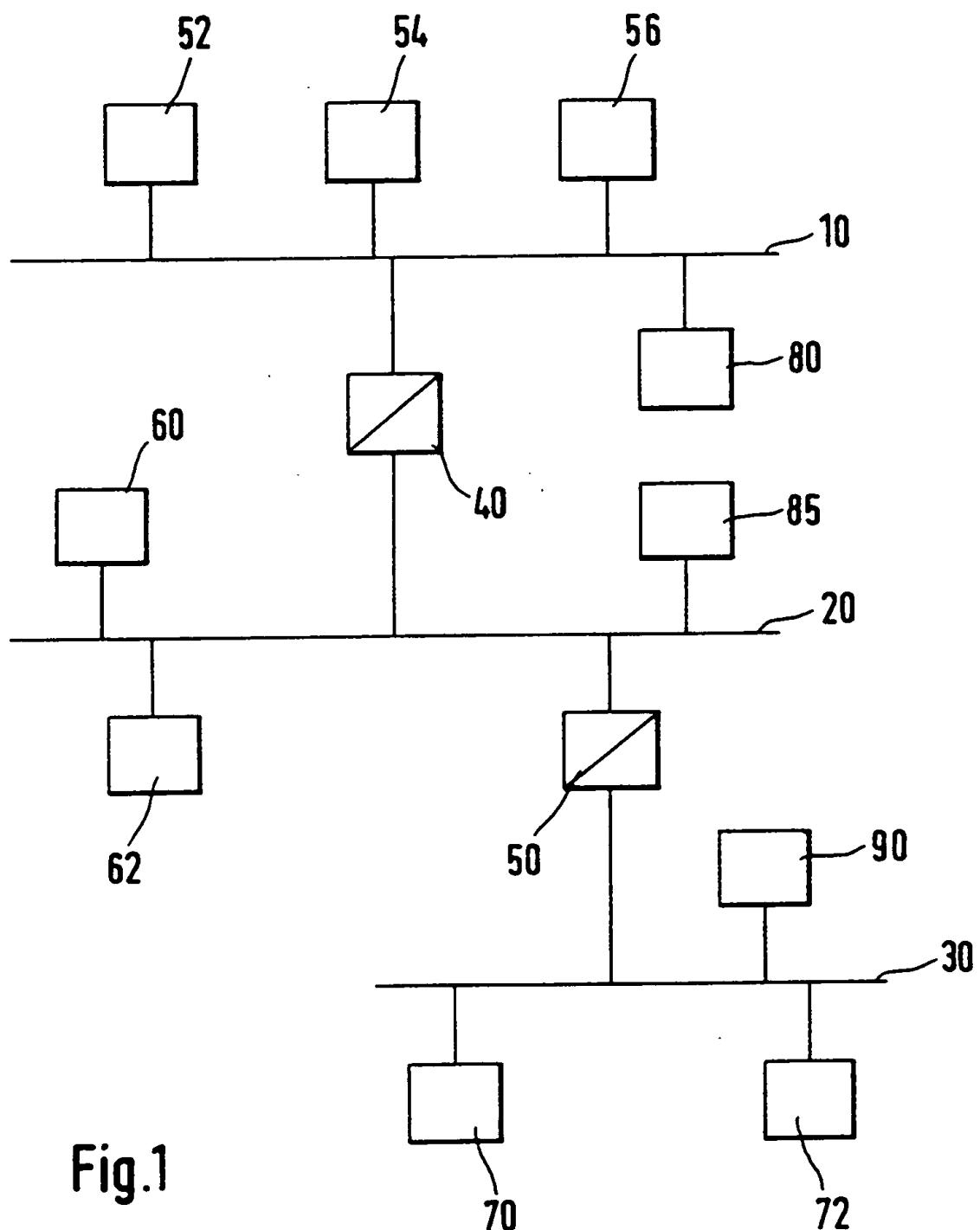
(a) Erzeugen eines Senderdatenmusters durch die Sendeeinheit und Senden des Senderdatenmusters zur Empfangseinheit,

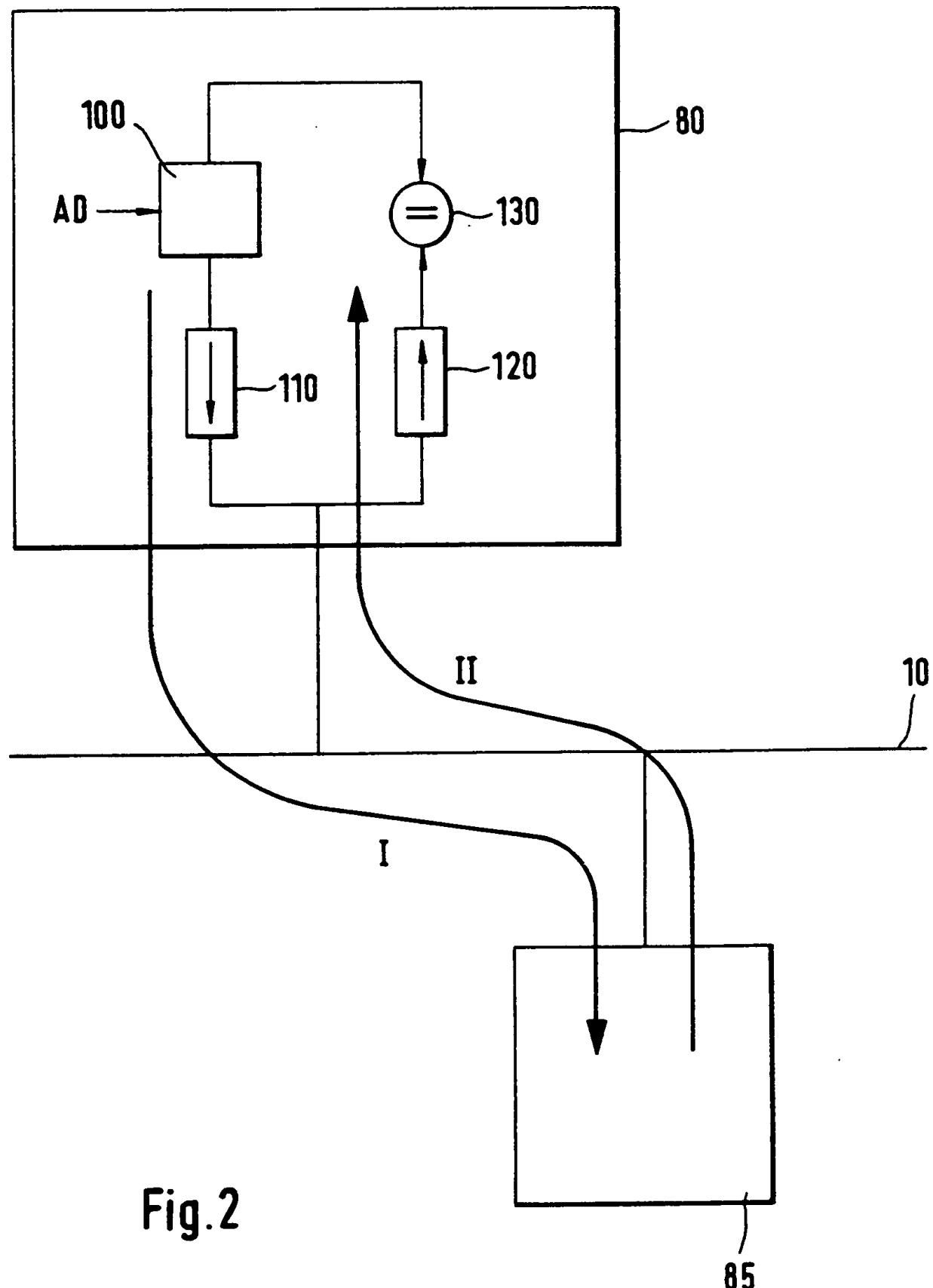
(b) Empfangen eines Datenmusters durch die Empfangseinheit auf das von der Sendeeinheit gesendete Senderdatenmuster hin,

(c) Erzeugen eines Datenmusters durch die Empfangseinheit in einem definierten Zusammenhang mit der Erzeugung des Senderdatenmusters, auf der Grundlage der allgemeinen Kenntnisse zwischen der Sendeeinheit und der Empfangseinheit, und

(d) Vergleichen des empfangenen Datenmusters mit dem von der Empfangseinheit erzeugten Datenmuster, um Rückschlüsse zu ziehen, ob die Weiterleitung des Datenmusters im System ordnungsgemäß funktioniert oder ob Daten im System geändert werden.

Es folgen 5 Blatt Zeichnungen





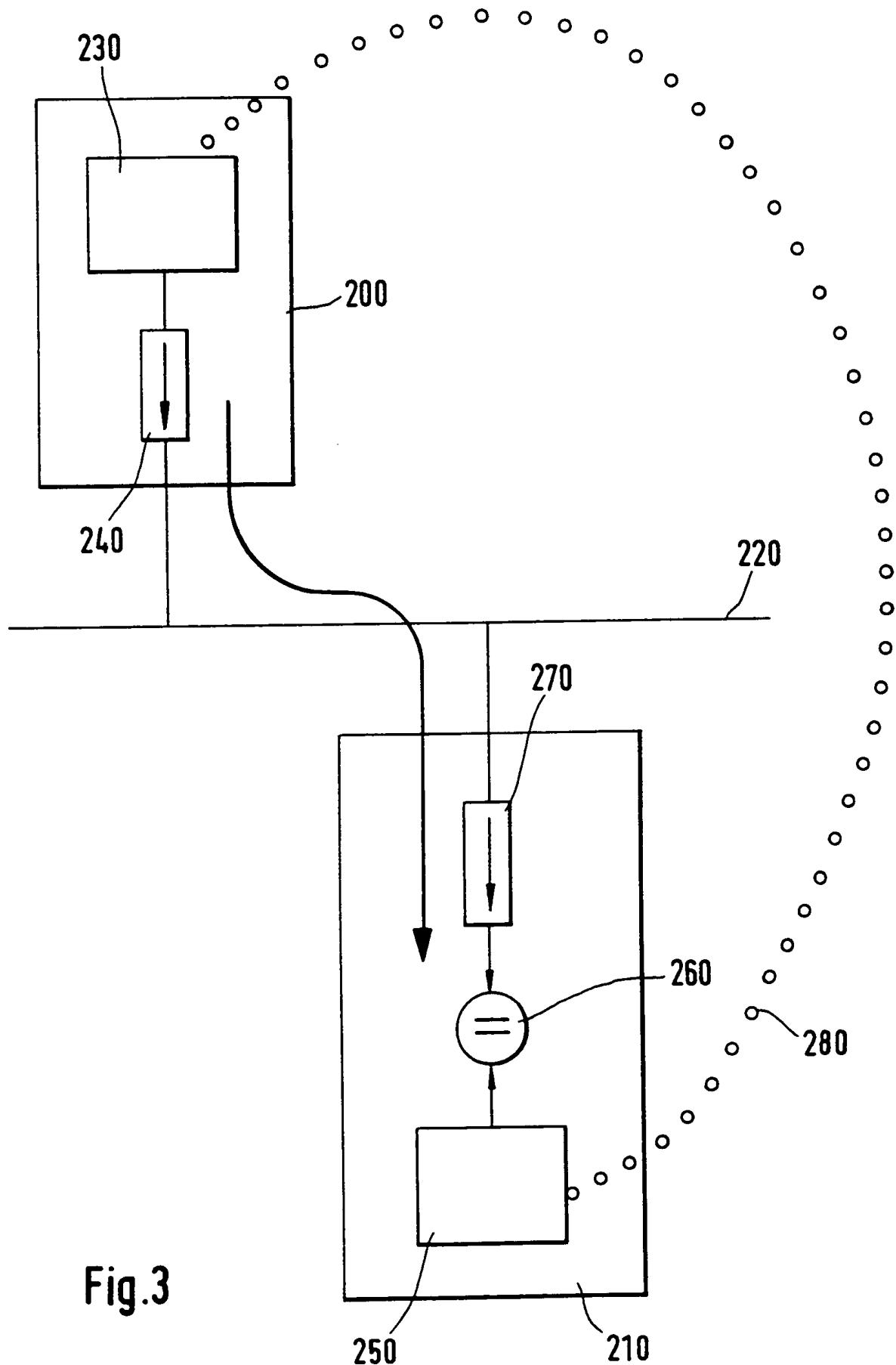


Fig.3

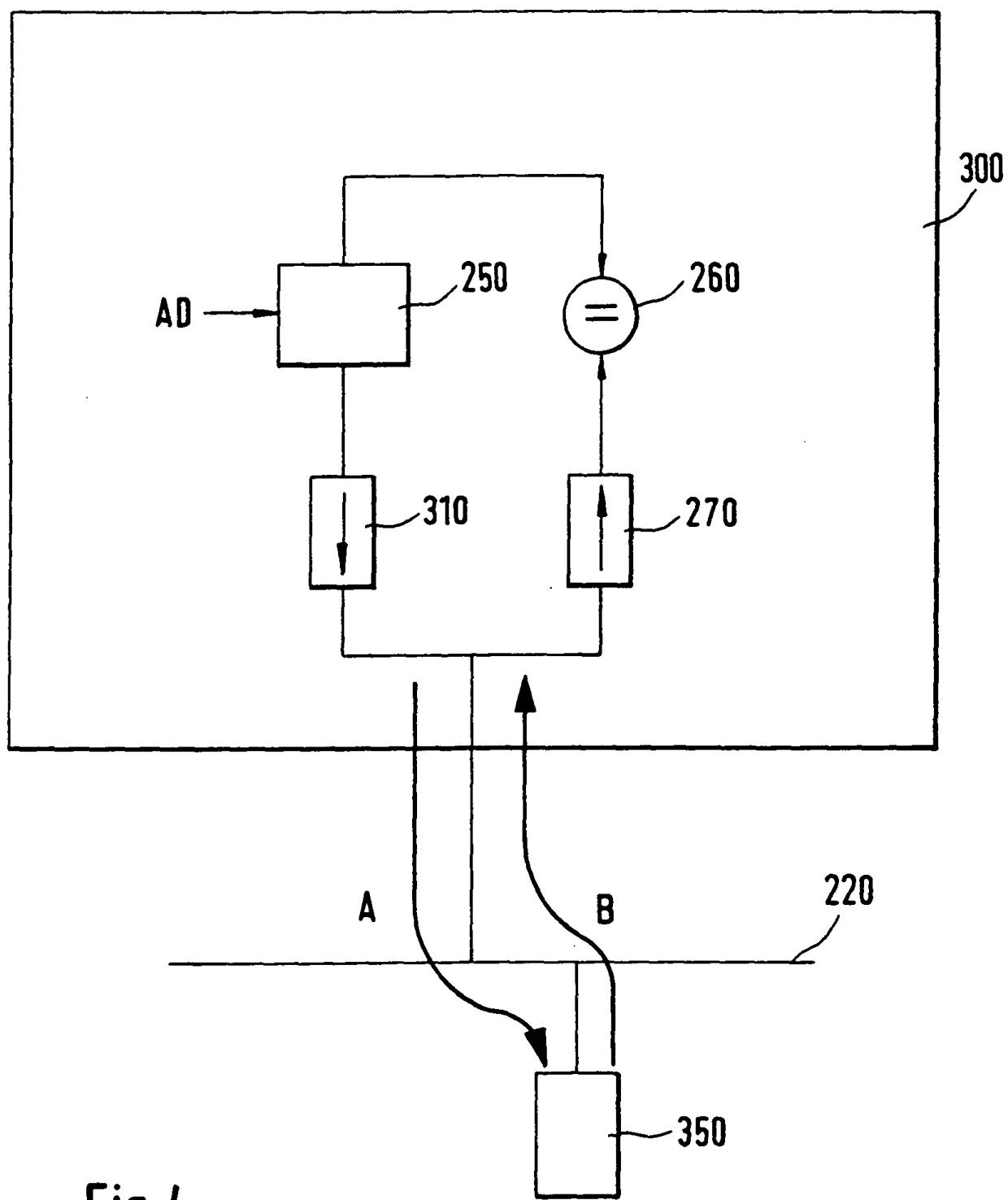


Fig. 4

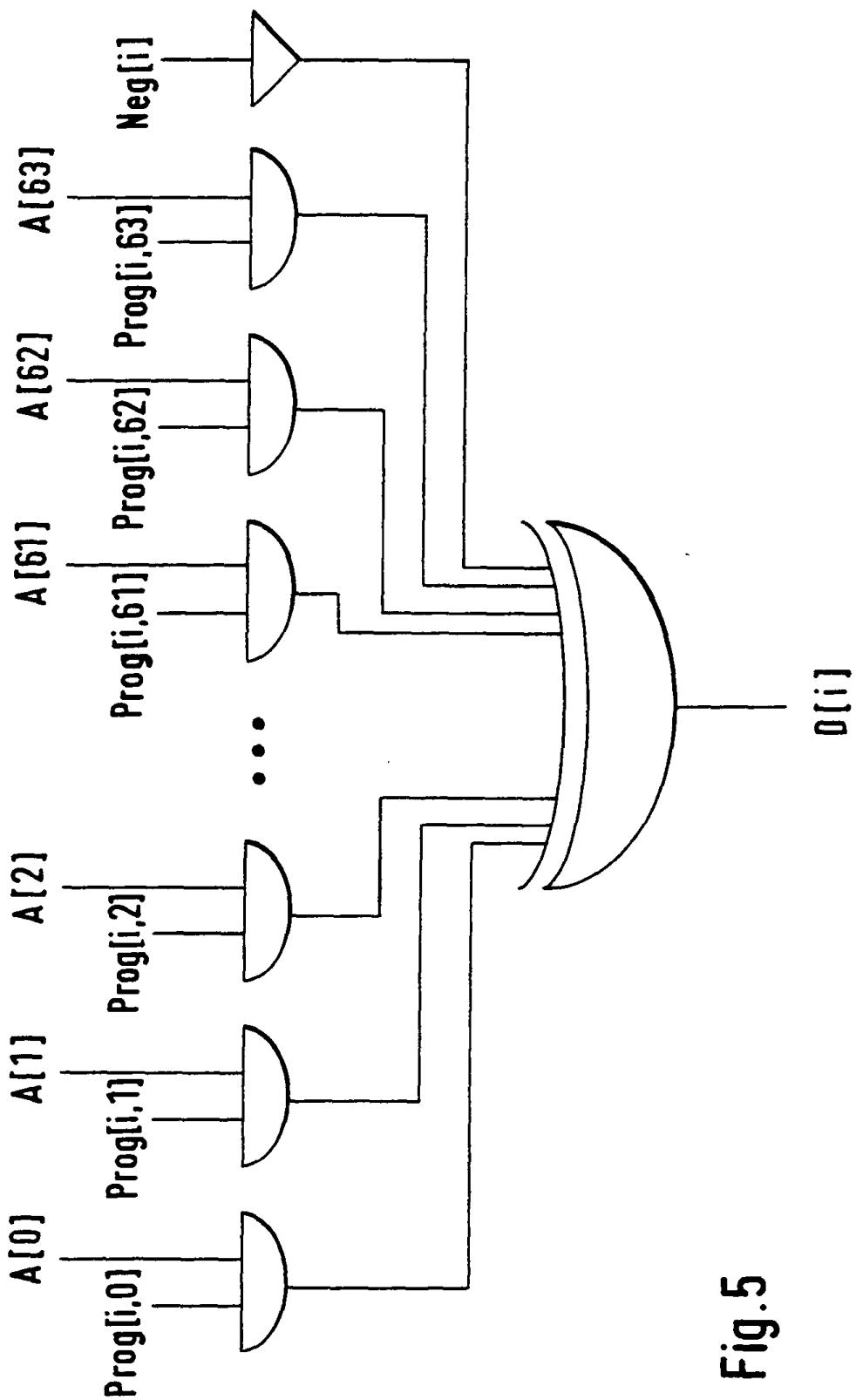


Fig. 5