(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 690 34 133 T2** 2004.09.16

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 611 014 B1**
(21) Deutsches Aktenzeichen: **690 34 133.4**
(96) Europäisches Aktenzeichen: **94 200 820.2**
(96) Europäischer Anmeldetag: **28.06.1990**
(97) Erstveröffentlichung durch das EPA: **17.08.1994**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **24.03.2004**
(47) Veröffentlichungstag im Patentblatt: **16.09.2004**

(51) Int Cl.[7]: **H04L 12/56**
H04L 12/44

(30) Unionspriorität:
8915135 **30.06.1989** **GB**

(84) Benannte Vertragsstaaten:
DE, FR, GB, IT

(73) Patentinhaber:
STMicroelectronics Ltd., Marlow, Buckinghamshire, GB

(74) Vertreter:
Samson & Partner, Patentanwälte, 80538 München

(72) Erfinder:
May, Michael David, Clifton, GB; Parsons, Brian Jeremy, Clifton, GB; Thompson, Peter William, Horfield, Bristol BS7 0DE, GB; Walker, Christopher Paul Hulme, Portishead, Avon BS20 9LD, GB

(54) Bezeichnung: **Kommunikationsgerät**

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

[0001] Diese Erfindung bezieht sich auf eine Kommunikationsvorrichtung für die Verwendung bei dem Weiterleiten von Nachrichten und insbesondere dem Weiterleiten von Nachrichten in einem Computernetz sowie auf eine Router/Switch-Vorrichtung für eine derartige Kommunikationsvorrichtung.

[0002] Es ist häufig wünschenswert, Verarbeitungsvorrichtungen wie Mikrocomputer in einem Netz zu verbinden, bei dem ein Mikrocomputer mit anderen Mikrocomputern in dem Netz kommunizieren kann. Bei solchen Netzen wäre es eine unnötige Einschränkung, wenn ein Mikrocomputer nur mit seinen unmittelbaren Nachbarn kommunizieren könnte, d. h. mit den Vorrichtungen, die mit ihm fest verdrahtet sind. Es ist gegenwärtig bekannt, dass Mikrocomputer in einem Netz Prozesse durchführen können, deren Funktion es ist, Nachrichten durch das Netz zu leiten. Das heißt, bei diesem Prozess wird eine Nachricht empfangen und ein Befehl ausgeführt, der zur Folge hat, dass die Nachricht an den entsprechenden benachbarten Mikrocomputer weitergeleitet wird, wo diese Prozedur möglicherweise wiederholt werden muss. Dies ist aufwendig sowohl bezüglich der Bearbeitungszeit für den Prozess in dem Mikrocomputer als auch in Bezug auf den Aufwand, um die Prozesse in einem Programm bei einer bestimmten Netzwerkarchitektur verschiedenen Mikrocomputern zuzuweisen.

[0003] Es sind Vorrichtungen für das Weiterleiten von Nachrichten verschiedener Art bekannt. Manche sind speziell für eine bestimmte Netzwerkarchitektur ausgelegt, während andere so funktionieren, dass Switch-Kanäle zwischen den verarbeitenden Vorrichtungen vorgesehen sind, die mit der Router-Vorrichtung verbunden sind. Bei vielen Netzen speisen die Router-Vorrichtungen jeweils ein Paket ein, legen seinen Zielort fest und schicken das Paket an die nächste Vorrichtung. Dies ist insofern nachteilig, als in jeder Vorrichtung Speicherplatz für die übermittelten Pakete benötigt wird und möglicherweise große Verzögerungen zwischen Ausgabe eines Pakets und seinem Empfang entstehen. Bislang gibt es keine flexible, schnelle und effiziente Router-Vorrichtung für das Weiterleiten von Nachrichten in einem Netz.

[0004] In diesem Zusammenhang wird auf den Artikel "Hop-by-Hop Flow Control on Unreliable Connections" in IEEE International Conference of Communications, 1987, Communications-Sound to Light, Band 3, 7.–10. Juli 1987, Seattle, Seite 1683–1687 verwiesen. Hier wird die Ablaufsteuerung in Segmenten eines virtuellen Kreises durch ein Computernetz diskutiert.

[0005] In US 4 740 958 wird ein Datenübertragungssystem mit mehreren Verarbeitungsknoten beschrieben. Das Ziel besteht darin sicherzustellen, dass Puffer innerhalb der Verarbeitungsknoten nicht überlaufen, und dazu wird ein System von Bestätigungssignalen eingesetzt. Die Knoten werden durch ein Bündel von sechs optischen Fasern miteinander verbunden, wobei jede Faser ein Manchester-Code-Signal überträgt. Fünf der Fasern übertragen digitale Daten, die aus vier Datenbits und einem dazugehörigen Paritätsbit bestehen. Die sechste Faser überträgt ein Bestätigungssignal in Gegenrichtung. Das Bestätigungssignal wird verwendet, um Zähler zu steuern, die die Anzahl von Nachrichten speichern, die in dem Empfangspuffer zu einem bestimmten Kanal warten.

[0006] Dieses System benötigt eine eigene Faser für das Bestätigungssignal.

[0007] In US 4 380 063 wird ein Ablaufsteuerungsmechanismus für Block-Switch-Knoten beschrieben. Hier wird ein Pufferüberlauf vermieden, indem bei jedem Knoten ein Zähler für das Speichern der Angabe der verfügbaren Pufferkapazität am anderen Ende jeder Ausgangsverbindung vorgesehen wird. Wann immer eine Nachricht aus einem Puffer ausgelesen wird, wird ein Token oder ein Bestätigungssignal über die Verbindung zurückgeschickt, und dies wird verwendet, um den Zähler zu aktualisieren und anzuzeigen, dass die verfügbare Pufferkapazität zugenommen hat.

[0008] Token werden über eigene Token-Leitungen übertragen, so dass hierfür eigene Verbindungen zwischen den Knoten benötigt werden.

[0009] Es ist ein Ziel der vorliegenden Erfindung, ein Ablaufsteuerungssystem zu schaffen, das auf den Verbindungen zwischen den Knoten beruht, die für die normale Nachrichtenübertragung zur Verfügung stehen, so dass die Konnektivität in einem Netz von Routern vereinfacht wird.

[0010] Die vorliegende Erfindung bezieht sich insbesondere, jedoch nicht ausschließlich auf Netze mit untereinander kommunizierenden Mikrocomputern und anderen Vorrichtungen. Sie ist insbesondere einsetzbar bei der Kommunikation zwischen Vorrichtungen, die über Paare von unidirektionalen Kommunikationsleitungen verbunden sind, über die serielle Bitpakete in einem oder mehreren Formaten übertragen werden.

[0011] Gemäß der vorliegenden Erfindung wird eine Kommunikationsvorrichtung zum Übertragen und Empfangen von Nachrichten mit einer Eingangsschaltung geschaffen, die eine Speichervorrichtung zum Abspeichern von Daten und einen Empfangseingang zum Empfangen von Eingangsdaten umfasst, einem Zähler für das Abzählen des Umfangs von Daten, die in der Speichervorrichtung gespeichert sind, und eine Ausgangsschaltung zum Übertragen eines Ablaufsteuerungscodes vom Ausgang aus, wenn genügend Speicherkapazität in der Speichervorrichtung für einen vorgegebenen Umfang von Daten vorhanden ist, wobei die Ausgangsschaltung mit einem Dateneingang zum Empfangen einer ersten Folge von Daten und einem Ablaufsteuerungsanforderungseingang für das Empfangen von Ablaufsteuerungsinformation ausgestattet ist, wobei die

Eingangsschaltung mit der Ausgangsschaltung verbunden ist, um die Ablaufsteuerungsinformation über den Ablaufsteuerungsanforderungseingang bereit zu stellen, die dadurch gekennzeichnet ist, dass die Ausgangsschaltung dazu dient, die Daten der ersten Folge und die Ablaufsteuerungsinformation in mehrere Ausgangscodes zu wandeln, wobei jeweils eine vorgegebene Bitlänge festgelegt ist, und um die Ausgangscodes als eine Ausgangsfolge über den Ausgang zu übertragen, wobei die Codes aus einer Gruppe von Codes ausgewählt werden, wobei eine Menge von diesen als Datencodes bezeichnet wird und einer, der sich nicht in dieser Menge befindet, als Ablaufsteuerungscode bezeichnet wird, und dass die Eingangsdaten in Form einer Folge von ankommenden Codes vorliegen und die Eingangsschaltung dazu dient, die ankommenden Codes der Eingangsfolge in Dateneinheiten, die durch die Speicherschaltung abgespeichert werden, und in die Ablaufsteuerungsinformation zu wandeln, wobei die Ausgangsschaltung veranlasst wird, den Ablaufsteuerungscode in der Ausgangsfolge zu übertragen, wenn genügend Speicherkapazität in der Speichervorrichtung vorhanden ist.

[0012] Die Erfindung betrifft außerdem eine Kommunikationsvorrichtung mit der oben beschrieben Kombination aus Eingangs- und Ausgangsschaltung. Die Codes sind Bitsequenzen, die seriell oder parallel übertragen werden können. Hier wird hauptsächlich die serielle Übertragung erläutert.

[0013] Damit wird bei der vorliegenden Erfindung ein bestimmtes serielles Bitpaket (im Folgenden als Ablaufsteuerungs-Token bezeichnet) auf einer Kommunikationsleitung übertragen, um die Menge der anderen seriellen Bitpakete zu steuern, die über die andere Kommunikationsleitung übertragen werden. Dies ist insbesondere, aber nicht ausschließlich anwendbar auf die Kommunikation zwischen Vorrichtungen, bei denen Sequenzen von Bytes von Daten als Sequenzen von seriellen Bitpaketen übertragen werden, gefolgt von einem bestimmten seriellen Bitpaket, das als Ende einer Sequenzmarkierung dient. Insbesondere ist dies anwendbar auf die Kommunikation zwischen mehr als zwei Vorrichtungen, bei denen Sequenzen von Bytes von Daten als Sequenzen von seriellen Bitpaketen übertragen werden, wobei das erste oder mehrere erste Bytes jeder Sequenz den Header-Abschnitt bilden. Derartige Kommunikationssysteme werden in unserer britischen Patentanmeldung 8915136.9 (Page, White und Farrer, Aktenzeichen 64199) beschrieben.

[0014] Es wird auf die europäische Anmeldung 90307110.8 verwiesen, aus der diese Anmeldung als Teilanmeldung hervorgegangen ist. Bei der Anmeldung wird ein Verfahren zum Weiterleiten von Nachrichten beschrieben, bei dem das Lesen eines Header-Abschnittes ausgeführt wird, bevor das vollständige Paket empfangen wird mit, wonach die Übertragung des Nachrichtenabschnitts durch eine Router/Switch-Vorrichtung folgt Dies wird als "wormhole routing" bezeichnet. Es hat den erheblichen Vorteil gegenüber bekannten Weiterleitungstechniken, dass bei der bevorzugten Ausführungsform Pufferplatz nur für den Header-Abschnitt und nicht für das vollständige Paket benötigt wird.

[0015] Diese Anmeldung beschreibt außerdem die Auswahl eines Ausgangs einer Router/Switch-Vorrichtung für einen Bereich von Zielknotenidentifizierungen, hier als "interval labelling" bezeichnet.

[0016] Die Kommunikationsvorrichtung gemäß der vorliegenden Erfindung kann bei jeder Router/Switch-Vorrichtung in einem Netz mit Routern vorgesehen werden, das dazu dient, Nachrichten zwischen einer ersten Computervorrichtung am Ursprungsknoten und einer zweiten Computervorrichtung am Zielknoten weiterzuleiten, wobei jeder Paketausgang von dem Ursprungsknoten durch den Zielknoten bestätigt wird, wenn das Paket dort empfangen wurde. Hierzu ist die Schaltung zur Erzeugung von Paketen der Computervorrichtungen in der Lage, ein Bestätigungspaket bei Empfang eines Datenpakets zu erzeugen. In diesem Zusammenhang wird auf unsere britische Anmeldung 8915136.9 (Page, White und Farrer, Aktenzeichen 64199) verwiesen, in der Computervorrichtungen für den Aufbau von Computernetzen beschrieben werden, für die diese Erfindung besonders eingesetzt werden kann. Die Datenpakete und Bestätigungspakete bilden die Basis der synchronisierten Prozess-Prozess-Kommunikation zwischen Prozessen, die von Prozessoren entfernter Computervorrichtungen bearbeitet werden.

[0017] Unter synchronisierter Prozess-Prozess-Kommunikation ist zu verstehen, dass die kommunizierenden Prozesse sich auf korrespondierenden Programmstufen befinden, wenn die Nachrichtenübertragung durchgeführt wird. Eine Nachricht kann unter Umständen mehrere der oben genannten Datenpakete enthalten.

[0018] Erfindungsgemäß wird außerdem ein Verfahren zum Übertragen von mehreren Bitsequenzen von einer ersten Kommunikationsvorrichtung an eine zweite Kommunikationsvorrichtung angegeben, wobei die Bitsequenzen jeweils die gleiche Anzahl von Bits aufweisen und eine Gruppe bilden, wobei das Verfahren umfasst: Bezeichnen einer Bitsequenz aus der Gruppe als Ablaufsteuerungscode und anderer Bitsequenzen als Dateneinheiten und auf Seiten der zweiten Kommunikationsvorrichtung Abzählen der Anzahl von Dateneinheiten, die von der ersten Kommunikationsvorrichtung übertragen werden und in der zweiten Kommunikationsvorrichtung gespeichert werden, und für den Fall, dass die Kapazität bei der zweiten Kommunikationsvorrichtung für eine vorgegebene Anzahl von Dateneinheiten vorhanden ist, Übertragen des Ablaufsteuerungscodes von der zweiten Kommunikationsvorrichtung zu der ersten Kommunikationsvorrichtung, um die Übertragung der vorgegebenen Anzahl von Dateneinheiten an die zweite Kommunikationsvorrichtung freizugeben.

[0019] Bei einem bevorzugten Codierungsformat ist jede Bitsequenz sechs Bits lang, wobei jede Sequenz drei Einsen und drei Nullen enthält, von denen sechzehn Kombinationen verwendet werden, um verschiedene

Werte mit vier Daten-Bits darzustellen, wobei eine Kombination verwendet wird, um das Ende der Nachrichtenpakete anzuzeigen, und eine Kombination verwendet wird, um den Ablauf der Datencodes und das Ende der Paketcodes in entgegengesetzter Richtung zu steuern.

[0020] Gemäß einem weiteren Aspekt der Erfindung wird ein Verfahren für das Codieren von Daten zur Übertragung von Nachrichten durch mehrere Router-Vorrichtungen als Bitsequenzen angegeben, wobei bei dem Verfahren die Bitsequenzen aus einer endlichen Menge von zwanzig Bitsequenzen ausgewählt werden, wobei jede Bitsequenz sechs Bits aus drei Einsen und drei Nullen aufweist, wobei eine erste Teilmenge der endlichen Menge, die aus sechzehn Bitsequenzen besteht, für die Verwendung als Datencodes gekennzeichnet wird, und eine zweite Teilmenge der endlichen Menge für die Verwendung als Steuercodes zum Steuern der Übertragung der Datencodes gekennzeichnet wird, wobei die zweite Teilmenge aus vier Bitsequenzen besteht, die nicht Teil der ersten Teilmenge sind, wobei ein erster der Steuercodes als Ablaufsteuerungscode zum Steuern des Umfangs von Datencodes bei der Übertragung zwischen einer Ursprungsvorrichtung und einer Zielvorrichtung gekennzeichnet wird, ein zweiter der Steuercodes als Nullcode zum Aufrechterhalten der fortlaufenden Übertragung, wenn es keinen Datencode zum Übertragen gibt, gekennzeichnet wird, und ein dritter der Steuercodes als ein Paketende-Code gekennzeichnet wird, durch den das Ende eines Paketes identifiziert wird, das zusammen mit diesem Paketende-Code wenigstens einen Code enthält.

[0021] Zum besseren Verständnis der vorliegenden Erfindung und zum Nachweis, wie diese umgesetzt werden kann, bezieht sich das Folgende auf das Beispiel in den beigefügten Zeichnungen.

[0022] **Fig.** 1 ist ein Blockdiagramm einer Router/Switch-Vorrichtung.

[0023] **Fig.** 2 ist ein Blockdiagramm einer Kommunikationsvorrichtung der Router/Switch-Vorrichtung.

[0024] **Fig.** 3 ist ein Blockdiagramm des Header-Abschnitt-Puffers der Router/Switch-Vorrichtung.

[0025] **Fig.** 4 ist ein Blockdiagramm der Header-Leseschaltung der Router/Switch-Vorrichtung.

[0026] **Fig.** 5 ist ein Blockdiagramm des Random-Header-Generators.

[0027] **Fig.** 6 ist ein Blockdiagramm der Header-Löschschaltung.

[0028] **Fig.** 7 zeigt einen Teil der Switch-Schaltung der Router/Switch-Vorrichtung.

[0029] **Fig.** 8 zeigt den Fluss der Daten durch einen Teil der Switch-Schaltung.

[0030] **Fig.** 9 zeigt die externen Verbindungen zu einem der Switch-Elemente der Switch-Schaltung.

[0031] **Fig.** 10 ist ein Blockdiagramm der Struktur eines Switch-Elements.

[0032] **Fig.** 11 ist ein Beispiel eines Computergitternetzes.

[0033] **Fig.** 1 zeigt die grundsätzlichen Aufbaumerkmale einer Router/Switch-Vorrichtung **1** mit einer Kommunikationsvorrichtung gemäß einer bevorzugten Ausführungsform der Erfindung. In **Fig.** 1 stellen dick gezeichnete Pfeile Datenpfade dar, während dünn gezeichnete Pfeile Befehlspfade sind. Die Router/Switch-Vorrichtung beinhaltet eine Switch-Schaltung in Form eines Kreuzschienenschalters **10**, über den irgendeiner von mehreren Nachrichteneingängen **12a**, ..., **12n** mit irgendeinem von mehreren Nachrichtenausgängen **14a**, ..., **14n** verbunden werden kann. Bei einer bevorzugten Ausführungsform gibt es **32** Eingänge und **32** Ausgänge, die paarweise **32** Nachrichtenverbindungen bilden. Der Kreuzschienenschalter umfasst eine Switch-Schaltung **32** für das Verbinden der Ausgänge, die verhindert, dass mehr als ein Eingang mit jedem Ausgang zu irgendeiner Zeit verbunden wird. Aus Gründen der Klarheit sind nur zwei Verbindungen dargestellt, die mit La, Ln bezeichnet sind und durch eine unterbrochene Linie um die Komponenten der Verbindung herum angedeutet sind. Die folgende Beschreibung bezieht sich auf die Verbindung La, aber alle anderen Verbindungen sind genauso geeignet für die mögliche gemeinsame Nutzung einiger ihrer Komponenten, was später erläutert wird.

[0034] Die Verbindung La beinhaltet eine Kommunikationsvorrichtung, die hier als Verbindungsmodul **18** bezeichnet wird, das dazu dient, die Router/Switch-Vorrichtung mit einer weiteren Router/Switch-Vorrichtung oder irgendeiner anderen Vorrichtung zu verbinden, und die zum Empfangen und Senden von Nachrichtenpaketen an und von der Router/Switch-Vorrichtung ausgelegt ist. Diese Verbindung kann durch ein bidirektionales Paar von unidirektionalen Kanälen **4a**, **6a** hergestellt werden. Jedes Nachrichtenpaket hat ein serielles Datenformat und beinhaltet einen Header zum Identifizieren des Zielknotens des Paketes. Ein Ziel- oder Ursprungsknoten in einem Computernetz ist im Allgemeinen, aber nicht immer eine Computervorrichtung, mit der Prozesse durchgeführt werden können.

Serielles Datenformat

[0035] Ein besonders geeignetes serielles Datenformat für die Verbindungsmodule ist die getaktete Daten-Codierung (clock-with-data encoding). Indem drei Null-Bits und drei Eins-Bits jeweils als Abschnitt mit sechs Bits gesendet werden, bleibt die Strombilanz ausgeglichen (DC-balance), und das Taktsignal kann unter Einsatz eines Phasenkreises (phase-locked loop) gewonnen werden. Diese Technik wird als 3-aus-6-Codierung bezeichnet. Es gibt 20 unterschiedliche 3-aus-6-Codes. Um Daten-Bytes effizient zu übertragen, werden 16 dieser Codes ausgewählt, um die verschiedenen möglichen Werte eines Halbbytes (4 Bit nibble) darzustellen. Damit kann ein Byte als zwei Codes gesendet werden. Wo die Übertragung kontinuierlich erfolgen muss (beispielsweise in einem Phasenkreissystem), ist es notwendig, einen einzelnen Nullcode vorzusehen, der

übertragen wird, wenn kein anderer Code übertragen wird.

[0036] Wenn ein Nachrichtenpaket direkt von einer Ursprungsvorrichtung an eine Zielvorrichtung geschickt wird, ist es nicht notwendig, dass die Länge des Paketes innerhalb des Paketes angezeigt wird; es ist lediglich notwendig, sicherzustellen, dass die Ursprungsvorrichtung und die Zielvorrichtung bezüglich der Länge übereinstimmen. Wo jedoch Pakete Router/Switch-Vorrichtungen durchlaufen, muss es möglich sein, dass diese Router/Switch-Vorrichtungen die Länge des durchlaufenden Paketes bestimmen, so dass die (temporär verbundene) Route durch die Switch-Schaltung **10** abgekoppelt werden kann, wenn das Ende des Paketes ausgegeben worden ist. Außer für den Fall, dass jedes Paket die gleiche Länge hat, ist es deswegen erforderlich, dass in dem Protokoll die Anzeige der Paketlänge vorgesehen ist, entweder durch eine Abschlussmarkierung oder eine Längenangabe am Anfang. Diese Einführung einer Längenangabe am Anfang erhöht die Verzögerung des Pakets durch die Router/Switch-Vorrichtung und macht außerdem eine Logikeinrichtung erforderlich, um das Paket abzuzählen. Eine bessere Technik besteht daher darin, ein Steuerungs-Token als Paketendmarkierung zu reservieren.

[0037] Das vollständige Protokoll eines 3-aus-6-Codierungssystems wird daher in der folgenden Tabelle wiedergegeben. Welches Bitmuster exakt welchem Token entspricht, ist vollständig beliebig und wird daher nicht gezeigt.

16	Datenwerte	
1	flow-control token	(FCT)
1	end-of-packet token	(EOP)
1	null token	(NULL)
1	ohne Zuordnung	

[0038] Es versteht sich, dass das 3-aus-6-Codierungsschema nur eines von vielen ist, die verwendet werden können, vorausgesetzt, dass die obigen Anforderungen bezüglich Paketende und (die weiter unten erläuterten) "Ablaufsteuerungs-Token" erfüllt werden. Insbesondere versteht sich, dass keine Notwendigkeit besteht, dass die Steuercodierungen das gleiche Format oder die gleiche Länge wie die codierten Datenwerte haben, noch dass die Daten als Folge von 4-Bit-Werten übertragen werden, vorausgesetzt, dass eine Folge von Bytes effizient übertragen werden kann.

[0039] Das Verbindungsmodul ist mit einem Header-Puffer **20** über einen Eingangsdatenpfad **36** und einen Ausgangsbefehlspfad **34** verbunden. Das Verbindungsmodul ist außerdem mit einer Header-Abstreifvorrichtung **28** über einen Ausgangsdatenpfad **40** und einen Eingangsbefehlspfad **38** verbunden. Der Header-Puffer **20** kann mit der Header-Leseschaltung in der Form eines Intervallwahlschalters **22** über Discard- und Select-Pfade **21**, **61** kommunizieren und ist permanent mit dem Intervallwahlschalter **22** über eine Header-Verbindung **23** verbunden. Der Intervallwahlschalter **22** liest den Header-Abschnitt eines ankommenden Pakets, bestimmt daraufhin den Ausgang des Switch-Schaltkreises **10**, an den die Nachricht gesendet werden soll, und setzt den Switch-Schaltkreis **10** entsprechend. Hierzu ermöglicht ein Befehls-Adresspfad **26** dem Intervallwahlschalter **22**, mit der Switch-Schaltung **10** zu kommunizieren. Die Switch-Schaltung kann Befehle an den Header-Puffer über den Pfad **19** schicken. Der Header-Puffer **20** kann außerdem mit einem Random-Header-Generator **24** über Request- und Header-Pfad **2**, **27** kommunizieren, wobei deren Sinn und Zweck später erläutert wird. Ein Nachrichtenpaket wird von der Switch-Vorrichtung über die Header-Abstreifvorrichtung **28** ausgegeben, wobei deren Sinn und Zweck ebenfalls später erläutert wird. Die Header-Abstreifvorrichtung kann Befehle an die und von der Switch-Schaltung **10** über die Pfade **29**, **31**, **7** senden bzw. empfangen. Die Bezugsziffer **30** bezeichnet die Initialisierungsschaltung, die verwendet wird, um die Router/Switch-Vorrichtung für den Betrieb einzurichten. Im Folgenden wird der Betrieb jeder Komponente der Router/Switch-Vorrichtung beschrieben.

[0040] Es versteht sich, dass die Notwendigkeit, einen Paket-Header vor jedem Datenblock zu übertragen, die Zeit reduziert, die effektiv für die Übertragung von Daten verfügbar ist, und damit die effektive Datenbandbreite reduziert. Damit ist es wünschenswert, die Paket-Header so kurz wie möglich zu halten. Es ist jedoch auch wünschenswert, dass man irgendeine einer großen Anzahl von Anschlussverbindungen in einem Netz spezifizieren kann, wenn es sich um ein großes Netz handelt. Zu diesem Zweck ist es mit der vorliegenden Erfindung möglich, die Größe der Paket-Header zu variieren, so dass sie ausreichend groß gemacht werden können, damit alle Anschlussverbindungen in einem großen Netz spezifiziert werden können, oder dass sie im Fall eines kleinen Netzes kleiner gemacht werden können, um die Verringerung der effektiven Datenbandbreite zu reduzieren. Bei der bevorzugten Ausführungsform umfasst jede Router-Vorrichtung ein Flag, das festlegt, ob Paket-Header eine Länge von einem oder zwei Bytes haben. Ein Byte ist die kleinste Größe des Headers, bei der die Pakete noch nicht als aus einer Sequenz von Bytes bestehend angesehen werden, und zwei Bytes reichen aus, um 65.536 Anschlussverbindungen unterscheiden zu können.

[0041] Jedes Verbindungsmodul **18** lässt Anfragen nach Daten von dem Header-Puffer **20** über den Pfad **34** zu und gibt anschließend Daten über den Pfad **36** aus. Es werden auch Anfragen nach Daten von der Header-Abstreifvorrichtung **28** über den Pfad **38** ausgegeben, und anschließend werden Daten über den Pfad **40** empfangen. Es ist wichtig, dass die Ablaufsteuerung zwischen der Router/Switch-Vorrichtung und einer damit über das Verbindungsmodul verbundenen Vorrichtung aufrechterhalten bleibt. Um dies zu gewährleisten, multiplext das Verbindungsmodul **18** die Ablaufsteuerungsinformation in Datenfolgen zwischen der Router/Switch-Vorrichtung und einer damit verbundenen Vorrichtung. Um dies zu gewährleisten, ohne dass zu viel Bandbreite verbraucht wird, und um damit die Übertragungsrate von Daten so groß wie möglich zu lassen, fassen die Verbindungsmodule die Daten zu Gruppen (batches) zusammen, anstatt sie einzeln zu übertragen. Hierzu beinhaltet jedes Verbindungsmodul einen Puffer, der groß genug ist, um eine vollständige Gruppe von Daten (batch) aufzunehmen, und der in der Lage ist, zu zählen, wie viele Daten sowohl noch gesendet als auch noch empfangen werden müssen. Eine einfache Art, den Fluss der Gruppen von Daten (batches) zu regulieren, besteht darin, ein Ablaufsteuerungs-Token in Form eines bestimmten Codes aus der Menge von Codes zu reservieren, die durch das Verbindungsmodul **18** gesendet und empfangen werden können. Ein Ablaufsteuerungs-Token wird übertragen, wann immer das Verbindungsmodul **18** der Router/Switch-Vorrichtung ausreichend Pufferraum für eine vollständige Gruppe (batch) von Daten aufweist.

[0042] Die Verbindungsmodule regulieren den Datenfluss unabhängig von den Paketen, die darin enthalten sein können. Zu jedem Zeitpunkt können die Daten, die durch ein Verbindungsmodul gepuffert werden, einen Teil oder das Ganze eines oder mehrerer aufeinander folgender Pakete bilden.

[0043] In **Fig.** 2 umfasst jedes Verbindungsmodul: einen Ausgangspuffer **42**, der Daten und Ablaufsteuerungsinformation in ein serielles Format für die Übertragung wandelt und dies mit einer Frequenz übermittelt, die durch ein Taktsignal 0 vorgegeben wird, und einen Eingangspuffer **44**, der das serielle Format in Daten, die gespeichert werden, sowie Ablaufsteuerungsinformation, die von ihm an den Ausgangspuffer **42** übertragen werden, decodiert.

[0044] Die Einheit der Ablaufsteuerung ist eine Ablaufsteuerungsgruppe (batch). Jede Datengruppe besteht aus mehreren Codes, wobei jeder Code entweder Daten mit vier Bits oder einen Paketende-Code darstellt. Beispielsweise kann die Ablaufsteuerungsgruppe aus sechzehn Codes bestehen.

[0045] Jedes Verbindungsmodul, das mit einer separaten Vorrichtung verbunden ist, die ihr eigenes Verbindungsmodul umfasst, oder einer ähnlichen Kommunikationsvorrichtung, hält eine Ziffer für die Anzahl der Daten fest, die an das andere Verbindungsmodul geschickt werden können, ohne dass weitere Ablaufsteuerungsinformation empfangen wird. Diese Ziffer wird Kredit genannt. Jede Verbindung eines verbundenen Paars von Verbindungsmodulen hält eine Ziffer für die Anzahl von Daten fest, die durch das andere Verbindungsmodul gesendet werden können, ohne dass weitere Ablaufsteuerungsinformation gesendet wird. Diese Ziffer wird Debit genannt.

[0046] Jede Verbindung eines verbundenen Paars von Verbindungsmodulen zeigt an, dass sie bereit ist, eine weitere Datengruppe zu empfangen, indem ein Ablaufsteuerungs-Token gesendet wird. Wenn dies der Fall ist, erhöht sie ihren Debit um die Anzahl von Daten in der Gruppe. Wenn die andere Verbindung des Paars den Ablaufsteuerungs-Token empfängt, so erhöht sie ihren Kredit um die Zahl der Daten in der Gruppe.

[0047] Der Ablaufsteuerungsmechanismus der Verbindungsmodule wird im Folgenden mit Bezug auf **Fig.** 2 erläutert, wobei angenommen wird, dass ein serielles Datenformat ähnlich dem oben beschriebenen vorliegt. Es versteht sich jedoch, dass der Ablaufsteuerungsmechanismus im Wesentlichen auf die gleiche Art arbeitet, wenn ein anderes Datenformat verwendet wird, oder sogar wenn Daten als parallele Signale anstatt in einem seriellen Format gesendet werden.

[0048] Wann immer der Eingangspuffer **44** auf Kanal 4 ein Ablaufsteuerungs-Token empfängt und decodiert, signalisiert er dies dem Ausgangspuffer auf dem Pfad **41** "fct.seen". Wann immer der Ausgangspuffer ein Signal auf dem Pfad **41** "fct.seen" empfängt, erhöht er seine Ziffer für die Anzahl von Codes, die noch übertragen werden sollen (Kredit), um die Größe der Ablaufsteuerungsgruppe (batch).

[0049] Der Eingangpuffer **44** hält eine Ziffer für die Zahl der Codes fest, die empfangen werden sollen, bevor der Kredit des angeschlossenen Verbindungsmoduls der separaten Vorrichtung aufgebraucht ist. Diese Ziffer sollte die gleiche wie die entsprechende Kreditziffer in dem Ausgangspuffer des verbundenem Verbindungsmoduls der separaten Vorrichtung sein, die mit der Router/Switch-Vorrichtung verbunden ist, abgesehen von Abweichungen auf Grund von Codes, die zwischen den beiden Verbindungsmodulen gerade übertragen werden.

[0050] Der Eingangspuffer **44** enthält einen First in-First out-Puffer (FIFO), der wenigstens so viele Codes puffern kann, wie in einer Ablaufsteuerungsgruppe (batch) enthalten sind. Die Codes, außer den Nullcodes und den Ablaufsteuerungs-Token, werden in dem Puffer gespeichert, und in dem Puffer wird eine Ziffer bezüglich der Anzahl der Codes festgehalten. Wenn ein Signal auf dem Pfad **34** "request.in" empfangen wird, so beseitigt der Eingangspuffer **44** das erste Element aus dem FIFO, sendet es auf dem Pfad **36** "data.out" und reduziert

die Ziffer der Anzahl von Codes, die in dem FIFO gespeichert sind, es sei denn, es gibt keine Codes in dem FIFO, wobei in diesem Fall der erste Code, der außer den Nullcodes und den Ablaufsteuerungs-Token, empfangen wird, direkt auf dem Pfad "data.out" gesendet wird.

[0051] Wann immer die Summe der Anzahl von Codes in dem Eingangspuffer **44** und der Anzahl von Codes, die empfangen werden sollen, bevor die aktuelle Ablaufsteuerung aufgebraucht ist (Debit), kleiner ist als die Differenz zwischen der Größe des Puffers und der Größe einer AblaufsteuerungsGruppe, signalisiert der Eingangspuffer **44** dies dem Aufgangspuffer **42** auf dem Pfad "send.fct". Der Ausgangspuffer sendet dann einen Ablaufsteuerungs-Token bei der nächsten Gelegenheit.

[0052] Es versteht sich, dass für den Fall, dass die Größe des Eingangspuffers FIFO exakt gleich der Größe der Ablaufsteuerungsgruppe (batch) ist, der Eingangspuffer dem Ausgangspuffer signalisieren wird, ein Ablaufsteuerungs-Token nur zu senden, wenn der Eingangspuffer FIFO vollständig leer ist und die aktuelle Ablaufsteuerungsgruppe vollständig erledigt ist. Aus diesem Grund ist es von Vorteil, den Eingangspuffer FIFO in Bezug auf die Kapazität etwas größer als die Größe der Ablaufsteuerungsgruppe zu machen, damit der Datenfluss nicht unterbrochen wird.

[0053] Im Folgenden werden der Eingangs- und der Ausgangspuffer in der OCCAM 2-Sprache erläutert. Diese Beschreibungscodes werden als Bytes dargestellt, und Signale werden durch Kanäle dargestellt, die nur Boolsche-Werte enthalten. Die OCCAM-Sprache wird in "The OCCAM 2 Reference Manual" beschrieben, auf das hiermit verwiesen wird.

[0054] In OCCAM 2 wird das Verbindungsmodul **18** durch eine parallele Zusammensetzung der Eingangs- und Ausgangspuffer **44**, **42** dargestellt. Die Pfade aus dem und in das Modul seien in allgemeinerer Art und Weise definiert. Dann wird das Verbindungsmodul dargestellt durch:

```
CHAN OF BOOL fct.seen, send.fct, fct.sent:
PAR
  OutputBuffer (clock.in, codes.out, data.in, request.out, fct.seen, send.fct, fct.sent)
  InputBuffer (codes.in, data.out, request.in, fct.seen, send.fct, fct.sent)
```

[0055] Der Eingangspuffer wird in dem Teil von OCCAM beschrieben, der in Anhang 1 und 1A wiedergegeben ist. Der FIFO-Speicher wird hier als Ringpuffer mit einer Kapazität von "input.buffer.size" mit Zeigern darauf ("first" und "last") implementiert.

[0056] Dies entspricht der Verwendung eines kleinen Teils eines Schreib-/Lesespeichers in dem Eingangspuffer selbst. Es versteht sich, dass andere Techniken verwendet werden könnten, um den FIFO-Speicher zu implementieren, ohne dass dadurch der Ablaufsteuerungsmechanismus der Verbindungsmodule betroffen wird. Die Eingangspuffergröße muss wenigstens so groß wie die Größe der Ablaufsteuerungsgruppe (batch) sein. Wenn die Ablaufsteuerungsgruppe (batch) eine Größe von 16 hat, so müsste die Eingangspuffergröße wenigstens 16 betragen, und für einen möglichst störungsfreien Datenfluss könnte sie noch etwas größer sein, beispielsweise 20.

[0057] Der Ausgangspuffer enthält außerdem einen kurzen FIFO-Speicher, um den Datenfluss störungsfrei zu bewältigen. Es gibt keine Beschränkung in Bezug auf die Größe dieses FIFO-Speichers auf Grund des Ablaufsteuerungsmechanismus. Der Ausgangspuffer fordert dazu auf, Daten zu senden, indem er dies über den Pfad **38** "request.out" signalisiert, wann immer Platz zum Puffern eines Codes vorhanden ist. Zu jedem Zeitpunkt danach kann er einen Code empfangen, der auf den Pfad "data.in" übertragen wird.

[0058] Der Ausgangspuffer hält eine Ziffer bezüglich der Zahl von Codes fest, die noch übertragen werden müssen, bevor die gegenwärtige Ablaufsteuerungsgruppe (batch) erledigt ist. Zu jedem Zeitpunkt, zu dem ein Code außer einem Nullcode oder einem Ablaufsteuerungs-Token übertragen wird, wird diese Ziffer um eins reduziert.

[0059] Der Ausgangspuffer wird durch den Teil von OCCAM in Anhang 2 dargestellt. Er enthält einen FIFO-Speicher mit der Kapazität "output.buffer.size", die zum Beispiel 3 sein kann. Dieser Teil von OCCAM (Anhang 2) zeigt eine alternative Implementierung des FIFO-Speichers als lineare Matrix, deren Inhalt sukzessive reduziert wird, wann immer ein Code unten weggenommen wird. Es versteht sich, dass eine ähnliche Implementierung für den Eingangspuffer verwendet werden kann bzw. dass die Ringpufferimplementierung in dem Eingangspuffer in dem Ausgangspuffer ebenfalls hätte verwendet werden können.

[0060] Es versteht sich, dass ein äquivalenter Grad an Ablaufsteuerung in einem alternativen System erreicht werden kann, bei dem der Ausgangspuffer seinen "Kredit" ursprünglich auf die Größe einer Ablaufsteuerungsgruppe setzt und der Eingangspuffer seine "ausstehenden Token" anfangs auf den gleichen Wert setzt, anstatt dass beide zu Anfang auf null gesetzt werden, wie es hier gezeigt ist. In diesem Fall würde ein Ablaufsteuerungs-Token nur gesendet werden, nachdem die erste Ablaufsteuerungsgruppe übertragen wurde, anstatt vorher, wie es in der obigen Beschreibung der Fall war.

[0061] Es folgt aus Gründen der Vollständigkeit eine Beschreibung der weiteren Komponenten der Rou-

ter/Switch-Vorrichtung nach **Fig.** 1.

Header-Puffer

[0062] Die Struktur des Header-Puffers **20** bei jeder Verbindung La, ..., Ln ist in **Fig.** 3 dargestellt. Der Header-Puffer wird durch eine Steuerlogik gesteuert, die allgemein durch Block **51** dargestellt ist. Es gibt einen Token-Puffer **50**, in dem ein Token abgelegt werden kann, der entweder ein Daten-Halbbyte (nibble) oder ein Paketende-Token sein kann. Dieser Token-Puffer **50** signalisiert, wenn er einen Token über den Pfad **36** erhalten hat, der ein Paketende-Token ist, und zwar auf der Leitung "Got.EOP", und signalisiert auf der Leitung "Got.token.b", sobald er einen anderen Token erhält. Er übermittelt seinen Inhalt an eine Switch-Vorrichtung **52**, wenn er ein Signal von der Steuerlogik **51** erhält, und zwar auf "Send.t." Die Switch-Vorrichtung wird durch die Steuerlogik **51** gesteuert, so dass für den Fall, dass das Signal "Set.s1" hoch ist, die Switch-Vorrichtung **52** ihr Eingangssignal an einen Multiplexer **54** sendet. Wenn das Signal "Set.s1" niedrig ist, übermittelt die Switch-Vorrichtung **52** ihr Eingangssignal an den Ausgangsmultiplexer **56**. Die Multiplexer **54**, **56** werden durch die Signale "Set.m2" und "Set.m1" von der Steuerlogik **51** gesteuert. Der Header-Puffer **20** beinhaltet außerdem einen First in-First out-Puffer (FIFO) **58** mit ausreichend Kapazität für einen 2-Byte-Header (4 Token). Er ist permanent mit dem Intervallwahlschalter **22** über die Header-Verbindung **23** verbunden, die in **Fig.** 3 nicht dargestellt ist. Wenn das Signal "Set.m1" hoch ist, sendet der Multiplexer **56** das Ausgangssignal des FIFO-Speichers an den Kanal "DataOut" **12a**, wenn das Signal "Set.m1" niedrig ist, sendet der Ausgangsmultiplexer **56** das Ausgangssignal der Switch-Vorrichtung **52** an den Pfad **12a** "DataOut". Wenn das Signal "Set.m2" hoch ist, sendet der Multiplexer **54** das Ausgangssignal des Random-Header-Generators **24** (**Fig.** 1), das über den Pfad **27** empfangen wurde, an den FIFO-Speicher. Wenn das Signal "Set.m2" niedrig ist, sendet der Multiplexer **54** das Ausgangssignal der Switch-Vorrichtung **52** an den FIFO-Speicher. Sobald der FIFO-Speicher **58** ein Token empfängt, signalisiert er dies auf dem Pfad "Got.token.f", es sei denn, er wird durch den Token aufgefüllt, wobei er dies in diesem Fall auf dem Pfad "Full" signalisiert. Er gibt einen Token aus, wann immer er ein Signal auf dem Pfad "Send.h" empfängt. Wenn er leer wird, signalisiert er dies auf dem Pfad "Empty". Wenn er einen Eingang auf dem Pfad "Reset" empfängt, so lässt er seinen gesamten Inhalt unbeachtet, signalisiert dies allerdings nicht auf dem Pfad "Empty". Die effektive Größe des FIFO-Speichers wird durch ein dazugehöriges Flag **60** gesteuert, das ein Paket-Header-Längen-Flag ist. Wenn dieses Flag gesetzt ist, so sind die Paket-Header zwei Bytes lang, und der FIFO-Speicher signalisiert dies auf dem Pfad "Full" nur, wenn er vier Token enthält. Wenn diese Flag nicht gesetzt ist, sind die Paket-Header ein Byte lang, und der FIFO-Speicher signalisiert dies auf dem Pfad "Full" nur, wenn er zwei Token enthält. In diesem Fall werden die beiden oberen Token des FIFO-Speichers auf null gesetzt. Die Paket-Header-Längen-Flags **60** werden bei allen Verbindungen La bis Ln identisch gesetzt, und es versteht sich, dass sie mit einem einzigen Latch implementiert werden können, dessen Ausgang fächerförmig an alle Verbindungen ausgegeben wird. Es gibt außerdem ein Flag **62** zu der Steuerlogik **51**, das das Randomisierungs-Flag für die Verbindung ist. Wenn dieses Flag gesetzt wird, erhält man einen zufalls-Header von dem Random-Header-Generator **24** für jedes Paket, das über die Verbindung ankommt. Um dieses Flag für jede Verbindung anders setzen zu können, gibt es bei jeder Verbindung einen separaten Latch.

[0063] Das Verhalten des Header-Puffers **20** hängt von den Vorgaben dieses Randomisierungs-Flags der Verbindung La ab:

wenn das Flag **62** gesetzt wird, signalisiert der Header-Puffer **20** dies dem Random-Header-Generator **24** und speichert den Zufalls-Header, der zurückgeschickt wird. Sobald irgendwelche Daten von dem Verbindungsmodul **18** ankommen, signalisiert er dies auf dem Pfad **61** "Select" dem angeschlossenen Intervallwahlschalter **22**;

wenn das Flag **62** nicht gesetzt ist, wird der Header-Puffer mit den ersten ankommenden Daten gefüllt, und wenn er einen Header enthält (1 oder 2 Bytes, je nach Vorgabe der Paket-Header-Länge), so signalisiert er dies dem angeschlossenen Intervallwahlschalter **22**.

[0064] Wenn der Wahlschalter **22** dem Puffer **20** signalisiert, dass der Header unbeachtet bleiben soll, so erfolgt dies durch Zurücksetzen des FIFO-Speichers **58**, und der Header-Puffer wird durch den Datenstrom wieder aufgefüllt, und wenn er einen Header enthält (1 oder 2 Bytes, je nach Vorgabe der Paket-Header-Länge), so signalisiert er dies dem angeschlossenen Intervallwahlschalter. Der Header-Puffer überträgt seinen Inhalt als Reaktion auf Anfragen, die von dem Kreuzschienenschalter **10** empfangen wurden. Wenn das Randomisierungs-Flag oder die Verbindung gesetzt worden ist, so wird der Header-Puffer von selbst durch den Random-Header-Generator gefüllt, andernfalls wird er von selbst durch die Datenfolge gefüllt, und der Zyklus wird nach Weiterleitung eines Paketende-Tokens wiederholt.

Intervallwahlschalter

[0065] Der Betrieb des Intervallwahlschalters **22** wird im Folgenden mit Bezug auf **Fig.** 4 erläutert. Der Intervallwahlschalter **22** fällt die Entscheidung über die Weiterleitung bei jedem Paket mit Hilfe eines Intervallwahl-

algorithmus. Der Inhalt des FIFO-Speichers **58** des Header-Puffers **20** wird kontinuierlich einer "Leiter" von 33 Basis- und Grenzwertkomparatoren **66a**, ..., **66n** zugeführt, von denen vier rechts in der Figur dargestellt sind. Es gibt 33 Komparatoren, da es in der beschriebenen Ausführungsform 32 Verbindungen gibt. Jeder Komparator ist mit einem Paar von Registern **68a**, ..., **68n** verbunden, außer dem untersten Komparator **66a**, dessen Basiswert fest auf null liegt. Jedes Register **68i-1** wird mit dem Basiswert eines Komparators **68i** sowie dem Grenzwert eines anderen, **68i-1**, verbunden, außer dem obersten Register **68n**, das nur mit dem Grenzwert des obersten Komparators **66n** verbunden ist. Im Normalbetrieb werden die Register mit einer nicht abnehmenden Länge von 16-Bit-Werten bei jeder Verbindung programmiert, so dass die Spitze der "Leiter" nicht null ist. Der Ausgang jedes Komparators wird mit einem jeweiligen 5 Bit-Register **70a**, ..., 70n verbunden, dessen Inhalt an ein Adress-Gate **72** geschickt wird, wenn der Paket-Header größer als oder gleich dem Basiswert und kleiner als der Grenzwert ist, die mit dem entsprechenden Komparator verbunden sind. Unten in **Fig.** 4 ist ein "Portal"-Komparator **74** für Einzelwerte gezeigt, dessen Ausgänge durch ein Select-Gate **73** gegatet werden, das das "Select"-Signal von dem Header-Puffer **20** erhält. Wenn das Select-Signal von dem Header-Puffer gesendet wird, so ist der Inhalt des FIFO Header-Puffers **58** der Paket-Header, der über den Header-Pfad **23** mit dem Eingang des Komparators **74** verbunden ist. Wenn der Paket-Header nicht gleich dem Inhalt eines 16-Bit-Registers **76** ist, das mit dem Komparator **74** verbunden ist, wird ein "Nein"-Signal an das Select-Gate **73** geschickt, das es dann möglich macht, dass die Adresse, die von der "Leiter" aus Basis- und Grenzwertkomparatoren **60a**, ..., **66n** erzeugt wird, über den Pfad **26** an den Kreuzschienenschalter **10** geschickt wird. Wenn der Paket-Header gleich dem Inhalt des 16-Bit-Registers **76** ist, das mit dem Komparator **74** verbunden ist, wird ein "Ja"-Signal an den Header-Puffer **20** gesendet, das ihm mitteilt, den Header auf dem Pfad **21** unbeachtet zu lassen. In diesem Fall wird der Ausgang der Komparatoren-Leiter nicht an den Kreuzschienenschalter **10** gesendet.

Random-Header-Generator

[0066] Der Betrieb des Random-Header-Generators wird im Folgenden mit Bezug auf **Fig.** 5 erläutert. Ein Zufallszahlengenerator **80**, der oben in **Fig.** 5 gezeigt ist, erzeugt 16-Bit-Pseudozufallszahlen mit Hilfe eines Schieberegisters mit Rückkopplung oder auf eine andere Art. Wenn er ein Signal von einer Übertragseinheit **82** empfängt, die mit ihm verbunden ist, erzeugt er eine derartige Zahl. Die Übertragseinheit **82** berechnet den Übertrag dieser Zahl mit dem Wert, der in einem dazugehörigen 16-Bit-Register **83** gespeichert ist. Wenn die Übertragseinheit **82** ein Signal von einer Token-Einheit **86** erhält, sendet sie diesen 16-Bit-Rechnungsübertrag weiter und fordert eine weitere Pseudozufallszahl von dem Zufallszahlengenerator **80** an. Die Token-Einheit speichert den 16-Bit-Übertrag, bis sie eine Anfrage von dem Header-Puffer **20** über den Pfad **25** erhält. Sie sendet dann die niedrigsten vier Bits des Übertrags und verschiebt die restlichen Bits um vier Plätze nach unten. Sie wiederholt diese Aktion, bis sie die Anzahl von Halbbytes (nibbles) gesendet hat, die der Länge des Headers entsprechen (nämlich 2 oder 4, je nach Vorgabe des dazugehörigen Flags **60**, wie es in **Fig.** 3 dargestellt ist), und wiederholt dann den Zyklus. Es versteht sich, dass der Bereich der Random-Header, die erzeugt wurden, beliebig verallgemeinert werden kann, indem ein zusätzliches 16-Bit-Register sowie ein Addierer vorgesehen wird, um seinen Inhalt zu dem Ausgang der Übertragseinheit hinzuzuzählen, bevor das Ergebnis an die Token-Einheit weitergeleitet wird.

Header-Abstreifvorrichtung

[0067] Der Betrieb der Header-Abstreifvorrichtung wird im Folgenden mit Bezug auf **Fig.** 6 beschrieben. Wie ersichtlich, gibt es einen einzigen Token-Puffer **85**, der in der Tat identisch mit dem Token-Puffer **50** in dem Header-Puffer **20** (**Fig.** 3) ist. Der Token-Puffer **85** signalisiert dies der Steuerlogik **84**, wenn er einen Token über den Pfad **14a"** DataIn" empfangen hat, der ein Paketende-Token ist, auf der Leitung "Got.EOP", und er signalisiert es der Steuerlogik **84** über die Leitung "Got.token.b", wann immer er einen anderen Token empfängt. Er überträgt den Inhalt auf dem "DataOut"-Pfad **40**, wenn er von der Steuerlogik **84** ein Signal auf "Send.t" erhält.

Kreuzschienenschalter

[0068] Der Kreuzschienenschalter **10** ist eine 32-mal-32-Matrix aus Switch-Elementen **88**. **Fig.** 7 zeigt einen Abschnitt des Kreuzschienenschalters **10** mit 20 Switch-Elementen mit dicken Pfeilen, die den Datenfluss andeuten. Anfragen bezüglich Datenfluss in entgegengesetzter Richtung sind in **Fig.** 7 nicht dargestellt. Ursprünglich sind alle Switch-Elemente "aus", d. h. im Zustand A in **Fig.** 7. Wenn der Eingang i mit dem Ausgang j verbunden werden soll, so wird das Switch-Element **88** (i, j) "ein"-geschaltet, und die Daten fließen von dem Eingang zu dem ausgewählten Ausgang, wie es in dem Zustand B in **Fig.** 7 gezeigt ist.

[0069] **Fig.** 8 zeigt den Datenfluss in einem 4-mal-5-Abschnitt des Kreuzschienenschalters **10**, wenn zwei der

Switch-Elemente Y und Z "ein"-geschaltet und alle anderen "aus"-geschaltet sind. Man beachte, dass bei dieser Konfiguration der Eingang **12b** "in.2" mit dem Ausgang **14c** "out.3" verbunden ist und dass der Eingang **12c** "in.3" mit dem Ausgang **14b** "out.2" verbunden ist. Man beachte, dass das Switch-Element U **88** (2, 2), dass sich im Zustand "aus" befindet, Daten für beide Verbindungen liefert. Man beachte außerdem, dass durch den "Ein"-Zustand der Schalter Y **88** (3, 2) und Z **88** (2, 3) der Schalter V **88** (3, 3) abgekoppelt wurde, aber dies hat keine Konsequenzen, da der Eingang "in.3" mit "out.2" über den Schalter Y verbunden ist, so dass die Switch-Vorrichtung V nicht unbedingt Daten von dem Eingang weiterleiten muss; und außerdem wurde der Ausgang "out.3" mit dem Eingang "in.2" verbunden. Die Entscheidungsschaltung **32** hat zur Folge, dass die Switch-Vorrichtung V auch keine Daten zum Ausgang "out.3" zu schicken braucht.

[0070] Die externen Verbindungen jedes Switch-Elements **88** (i, j) sind in **Fig.** 9 dargestellt. Die Leitung ADDR, die den Adresspfad **26** von dem Intervallwahlschalter **22** darstellt, ist mit all den Switch-Elementen in jeder Zeile verbunden, d. h. mit denen mit dem gleichen Wert für i. Diese Verbindungen werden im Folgenden genauer erläutert. Jedes Switch-Element **88** (i, j) des Kreuzschienenschalters ist identisch, und die Blockstruktur eines dieser Elemente ist in **Fig.** 10 gezeigt. Es gibt drei Hauptkomponenten: eine Switch-Vorrichtung **90**, einen Komparator **92** und einen Arbiter-Server (ARB) **94**, die in der europäischen Patentanmeldung 90307110.8 erläutert werden, aus der die vorliegende Anmeldung als Teilanmeldung hervorgegangen ist und auf die hiermit verwiesen wird.

Initialisierung von Flags und Registern

[0071] Die folgenden Parameter müssen vorgegeben werden, bevor die Router/Switch-Vorrichtung arbeiten kann:
die Header-Paketlänge (1 oder 2 Bytes),
die Komparatorvorgaben für die Intervallwahlschalter **22** und die dazugehörigen Ausgangsverbindungsnummern für jeden Bereich von Zielknotenidentifizierungen,
der Portal-Komparatorwert für die Intervallwahlschalter **22**,
bei jeder Verbindung: es müssen die Flags "randomise on input" und "strip header on output" gesetzt werden,
der Bereich der Zufalls-Header (wenn eines der Flags "randomise on input" gesetzt ist).

[0072] Diese müssen durch die Router/Switch-Vorrichtung eingegeben werden und in die entsprechenden Latch-Vorrichtungen und Register geladen werden. Dies kann auf viel-fache Art erfolgen, beispielsweise durch Senden vordefinierter Befehlscodes über eine extra vorgesehene Verbindung, die Kanäle zu allen Latch-Vorrichtungen und Registern enthält. Man beachte, dass alle Intervallwahlschalter und Random-Header-Generatoren identisch programmiert werden müssen. Diese Initialisierung ist grafisch in **Fig.** 1 durch die Initialisierungslogik **30** dargestellt.

[0073] Die Vorrichtung wurde mit einem Intervallwahlschalter und einem Random-Header-Generator pro Verbindung dargestellt und beschrieben. Es versteht sich, dass beide dieser Einheiten nur einmal oder zweimal pro Paketeingabe bei der Verbindung verwendet werden und damit im Prinzip von verschiedenen Verbindungen gemeinsam genutzt werden können, die durch einen Bus verbunden sind, wobei ein Entscheidungsschema (arbitration scheme) verwendet wird, um mehrfache gleichzeitige Anfragen nach Ressourcen aufzulösen. Beispielsweise könnte das Entscheidungsverfahren (arbitration method), das in dem Kreuzschienenschalter **10** verwendet wird, eingesetzt werden, um einen Intervallwahlschalter oder einen Header-Generator für mehrere Verbindungen zu nutzen.

[0074] **Fig.** 11 zeigt das Beispiel eines Netzes, bei dem mehrere Router/Switch-Vorrichtungen RS eingesetzt werden, um Nachrichten zwischen Ursprungsknoten und Zielknoten weiterzuleiten, die in **Fig.** 11 durch die Kreise N dargestellt sind. Die Router/Switch-Vorrichtungen sind durch ihre Verbindungen L miteinander verbunden, wobei jede Verbindung ein fest verdrahtetes bidirektionales Paar von unidirektionalen Kanälen aufweist, wie es mit den Bezugsziffern **4**, **6** in **Fig.** 1 dargestellt ist. Die Ziel- und Ursprungsknoten N des Netzes können eine Computervorrichtung beliebiger Art sein, jedoch ist die Router/Switch-Vorrichtung, die hier beschrieben wurde, insbesondere anwendbar bei Netzen mit Mikrocomputern, wie es in GB 111399 und unserer ebenfalls anhängigen britischen Patentanmeldung Nr. ... (Page, White und Farrer, Aktenzeichen 64199) beschrieben wird. Verbindungen mit Ziel- und Ursprungsknoten N werden Anschlussverbindungen in **Fig.** 11 genannt. In **Fig.** 11 ist jede Router/Switch-Vorrichtung mit vier Nachrichtenverbindungen dargestellt – tatsächlich sind es in der oben beschriebenen Ausführungsform 32.

Patentansprüche

1. Kommunikationsvorrichtung zum Übertragen und Empfangen von Nachrichten mit einer Eingangsschaltung, die eine Speichervorrichtung zum Abspeichern von Daten und einen Empfangseingang (**4**) zum Empfangen von Eingangsdaten umfasst, einem Zähler für das Abzählen des Umfangs von Daten, die in der Speichervorrichtung gespeichert sind, und einer Ausgangsschaltung zum Übertragen eines Ablaufsteuerungscodes

vom Ausgang aus, wenn genügend Speicherkapazität in der Speichervorrichtung für einen vorgegebenen Umfang von Daten vorhanden ist,
wobei die Ausgangsschaltung (**42**) mit einem Dateneingang (**40**) zum Empfangen einer ersten Folge von Daten und einem Ablaufsteuerungsanforderungseingang (**43**) für das Empfangen von Ablaufsteuerungsinformation ausgestattet ist,
wobei die Eingangsschaltung mit der Ausgangsschaltung verbunden ist, um die Ablaufsteuerungsinformation über den Ablaufsteuerungsanforderungseingang (**43**) bereitzustellen,
dadurch gekennzeichnet, dass die Ausgangsschaltung dazu dient, die Daten der ersten Folge und die Ablaufsteuerungsinformation in mehrere Ausgangscodes (**6**) zu wandeln, wobei jeweils eine vorgegebene Bitlänge festgelegt ist, und um die Ausgangscodes als eine Ausgangsfolge über den Ausgang zu übertragen, wobei die Codes aus einer Gruppe von Codes ausgewählt werden, wobei eine Menge von diesen als Datencodes bezeichnet wird und einer, der sich nicht in dieser Menge befindet, als Ablaufsteuerungscode bezeichnet wird, und dass die Eingangsdaten in Form einer Folge von ankommenden Codes vorliegen und die Eingangsschaltung dazu dient, die ankommenden Codes der Eingangsfolge in Dateneinheiten, die durch die Speicherschaltung abgespeichert werden, und in die Ablaufsteuerungsinformation zu wandeln, wobei die Ausgangsschaltung veranlasst wird, den Ablaufsteuerungscode in der Ausgangsfolge zu übertragen, wenn genügend Speicherkapazität in der Speichervorrichtung vorhanden ist.

2. Kommunikationsvorrichtung nach Anspruch 1, bei der die Speichervorrichtung eine Speicherkapazität hat, die ausreicht, um exakt die vorgegebene Anzahl von Dateneinheiten abzulegen.

3. Kommunikationsvorrichtung nach Anspruch 1, bei der die Speichervorrichtung genügend Speicherkapazität für mehr als die vorgegebene Anzahl von Dateneinheiten hat.

4. Eine erste Kommunikationsvorrichtung zum Übertragen von Nachrichten an und Empfangen von Nachrichten von einer zweiten Kommunikationsvorrichtung, die mit der ersten Kommunikationsvorrichtung verbunden ist, wobei die erste Kommunikationsvorrichtung eine Kommunikationsvorrichtung gemäß einem der vorangehenden Ansprüche umfasst, wobei die Ausgangscodes im Betrieb als eine Ausgangsfolge an die zweite Kommunikationsvorrichtung übertragen werden und die Eingangsfolge von ankommenden Codes im Betrieb von der zweiten Kommunikationsvorrichtung empfangen wird, wobei die Ablaufsteuerungsinformation über den Ablaufsteuerungsanforderungseingang bereitgestellt wird und die Ausgangsschaltung der ersten Kommunikationsvorrichtung veranlasst, den Ablaufsteuerungscode an die zweite Kommunikationsvorrichtung zu übertragen, wenn genügend Speicherkapazität in der Speichervorrichtung der ersten Kommunikationsvorrichtung vorhanden ist.

5. Kommunikationsvorrichtung nach einem der vorangehenden Ansprüche, bei der die Ausgangsschaltung einen Zähler für das Abzählen der Anzahl von Dateneinheiten umfasst, die von der Ausgangsschaltung übertragen werden dürfen, wobei der Zähler in Abhängigkeit vom Empfang eines Ablaufsteuerungscodes bei der Eingangsschaltung um die vorgegebene Anzahl hochgesetzt wird und so arbeitet, dass er bei jeder Dateneinheit, die dann von der Ausgangsschaltung übertragen wird, um eins herabgesetzt wird.

6. Kommunikationsvorrichtung nach Anspruch 5, bei der die Ausgangsschaltung keine weiteren Dateneinheiten absendet, wenn der Zähler null erreicht, bis sie in Abhängigkeit vom Empfang eines Ablaufsteuerungscodes inkrementiert wird.

7. Verfahren zum Übertragen von mehreren Bitsequenzen von einer ersten Kommunikationsvorrichtung an eine zweite Kommunikationsvorrichtung, wobei die Bitsequenzen jeweils die gleiche Anzahl von Bits aufweisen und eine Gruppe bilden, wobei das Verfahren umfasst:
Bezeichnen einer Bitsequenz aus der Gruppe als Ablaufsteuerungscode und anderer Bitsequenzen der Gruppe als Dateneinheiten und
auf Seiten der zweiten Kommunikationsvorrichtung Abzählen der Anzahl von Dateneinheiten, die von der ersten Kommunikationsvorrichtung übertragen werden und in der zweiten Kommunikationsvorrichtung gespeichert werden,
und für den Fall, dass die Kapazität bei der zweiten Kommunikationsvorrichtung für eine vorgegebene Anzahl von Dateneinheiten vorhanden ist, Übertragen des Ablaufsteuerungscodes von der zweiten Kommunikationsvorrichtung zu der ersten Kommunikationsvorrichtung, um die Übertragung der vorgegebenen Anzahl von Dateneinheiten an die zweite Kommunikationsvorrichtung freizugeben.

8. Verfahren nach Anspruch 7, das außerdem umfasst: Hochsetzen eines Zählers bei der ersten Kommunikationsvorrichtung um eine vorgegebene Anzahl bei Empfang eines Ablaufsteuerungscodes und Herabsetz-

ten des Zählers jedes Mal, wenn eine Dateneinheit übertragen worden ist, wobei der Zähler einen Zählwert bezüglich der Anzahl von Dateneinheiten beibehält, die von der ersten Kommunikationsvorrichtung übertragen werden dürfen.

9. Verfahren nach Anspruch 8, bei dem die erste Kommunikationsvorrichtung keine weiteren Dateneinheiten absendet, wenn der Zähler null erreicht, bis er in Abhängigkeit vom Empfang eines Ablaufsteuerungscodes hochgesetzt worden ist.

10. Router/Switch für das Weiterleiten von Nachrichten zwischen Rechnervorrichtungen in einem Netz mit mehreren Knoten, von denen wenigstens einige entsprechende Identifizierungen aufweisen, wobei jede Rechnervorrichtung eine Paketerzeugungsschaltung zum Erzeugen von Nachrichtenpaketen jeweils mit einem Vorspannabschnitt für die Anzeige einer Zielknotenidentifizierung und einem Nachrichtenabschnitt nach dem Vorspannabschnitt umfasst, wobei der Router/Switch wenigstens einen Eingang für das Empfangen von Nachrichtenpaketen von einem Ursprungsknoten in der Form einer Eingangsfolge von Codes, die jeweils eine vorgegebene Bitlänge haben, und eine Kommunikationsvorrichtung nach Anspruch 1 umfasst,
wobei mehrere Ausgänge vorgesehen sind, wobei bei jedem von diesen die Ausgangsschaltung vorgesehen ist,
wobei eine Vorspannleseschaltung zum Lesen des Vorspannabschnitts eines an dem Eingang empfangenen Pakets vorgesehen ist und
wobei eine Switch-Schaltung zum selektiven Verbinden des Eingangs mit einem ausgewählten Ausgang in Abhängigkeit vom Einlesen des Vorspannabschnittes vorgesehen ist.

11. Router/Switch nach Anspruch 10, bei dem der wenigstens eine Eingang mehrere Eingänge aufweist, die jeweils selektiv mit irgendeinem der mehreren Ausgänge verbunden werden können.

12. Router/Switch nach Anspruch 10 oder 11, wobei der Router/Switch außerdem eine Vorspannlöschschaltung zum Löschen des Vorspannabschnittes eines empfangenen Paketes umfasst, so dass bei Empfang des Paketes durch einen nachfolgenden Router/Switch, der damit verbunden ist, ein nachfolgender Abschnitt des Paketes den Vorspannabschnitt bildet.

13. Router/Switch nach Anspruch 10 oder 11, wobei der Router/Switch außerdem eine freie Vorspannerzeugungsschaltung umfasst, um freie Adressen zu erzeugen und neue Vorspannabschnitte für ankommende Pakete zu bilden.

14. Verfahren zum Codieren von Daten für das Übertragen von Nachrichten durch mehrere Router-Vorrichtungen in Bitsequenzen, wobei bei dem Verfahren die Bitsequenzen aus einer endlichen Menge von zwanzig Bitsequenzen ausgewählt werden, wobei jede Bitsequenz sechs Bits aus drei Einsen und drei Nullen aufweist, wobei eine erste Teilmenge der endlichen Menge, die aus sechzehn Bitsequenzen besteht, für die Verwendung als Datencodes gekennzeichnet wird und eine zweite Teilmenge der endlichen Menge für die Verwendung als Steuercodes zum Steuern der Übertragung der Datencodes gekennzeichnet wird, wobei die zweite Teilmenge aus vier Bitsequenzen besteht, die nicht Teil der ersten Teilmenge sind, wobei
ein erster der Steuercodes als Ablaufsteuerungscode zum Steuern des Umfangs von Datencodes bei der Übertragung zwischen einer Ursprungsvorrichtung und einer Zielvorrichtung gekennzeichnet wird,
ein zweiter der Steuercodes als Null-Code zum Aufrechterhalten der fortlaufenden Übertragung, wenn es keinen Datencode zum Übertragen gibt, gekennzeichnet wird und
ein dritter der Steuercodes als ein Paketende-Code gekennzeichnet wird, durch den das Ende eines Paketes identifiziert wird, das zusammen mit diesem Paketende-Code wenigstens einen Code enthält.

Es folgen 8 Blatt Zeichnungen

Anhängende Zeichnungen

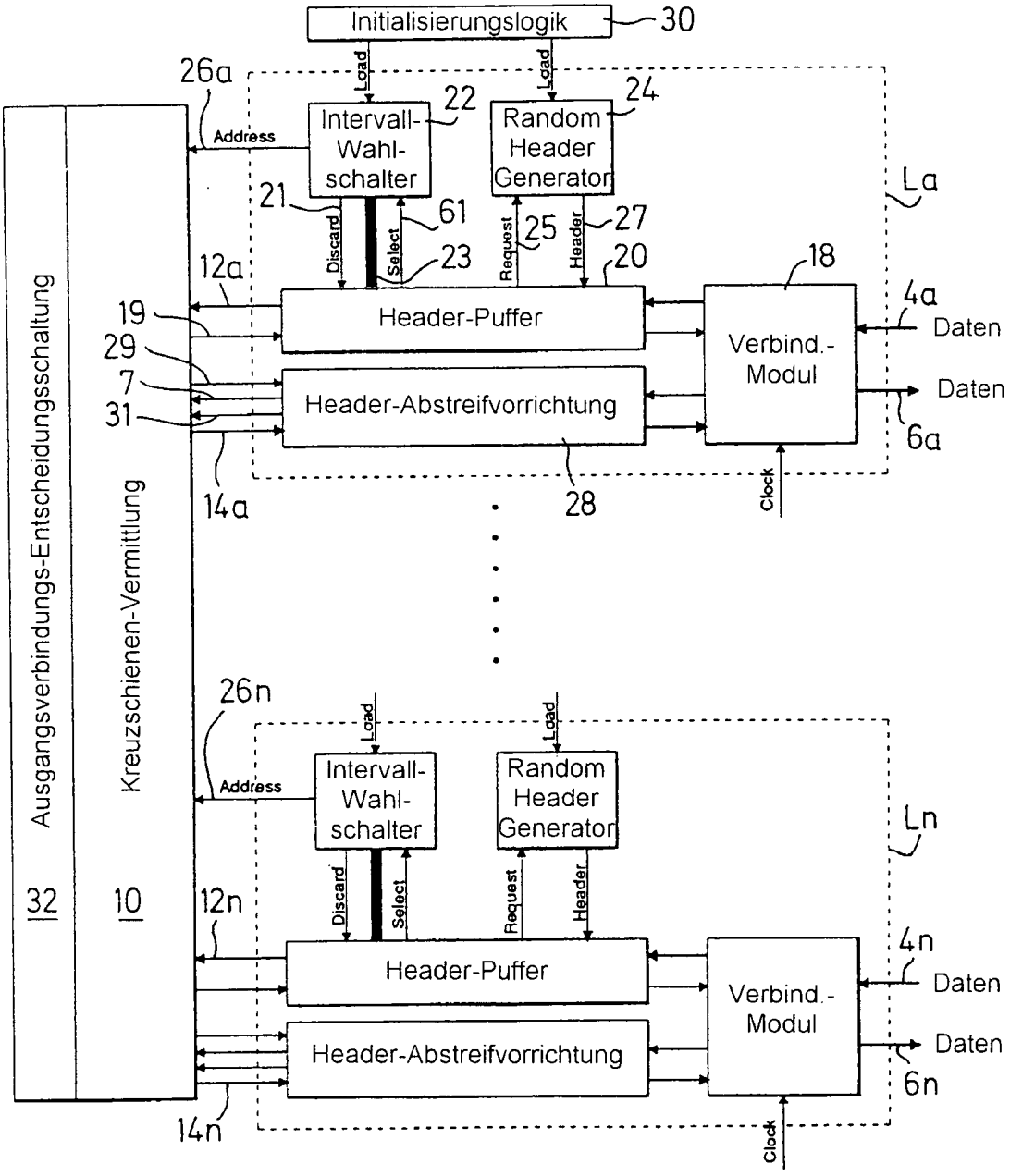


FIG.1.

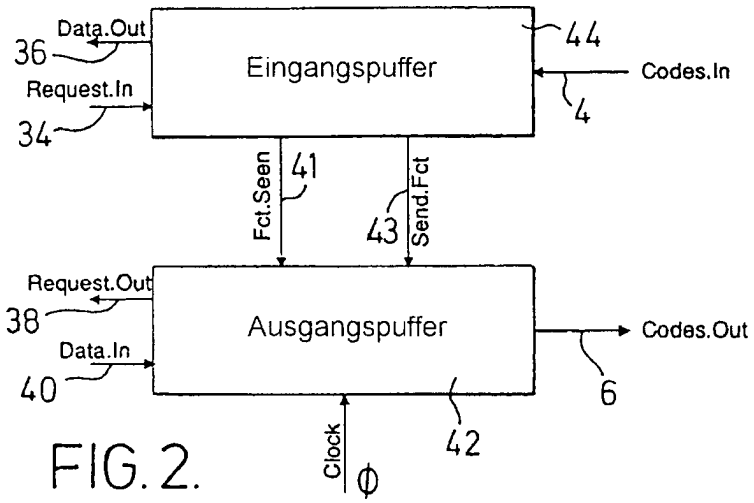

Data.Out
36
Request.In
34
Eingangspuffer
44
Codes.In
4
Fct.Seen
41
43
Send.Fct
Request.Out
38
Data.In
40
Ausgangspuffer
Codes.Out
6
42
Clock
ϕ

FIG. 2.

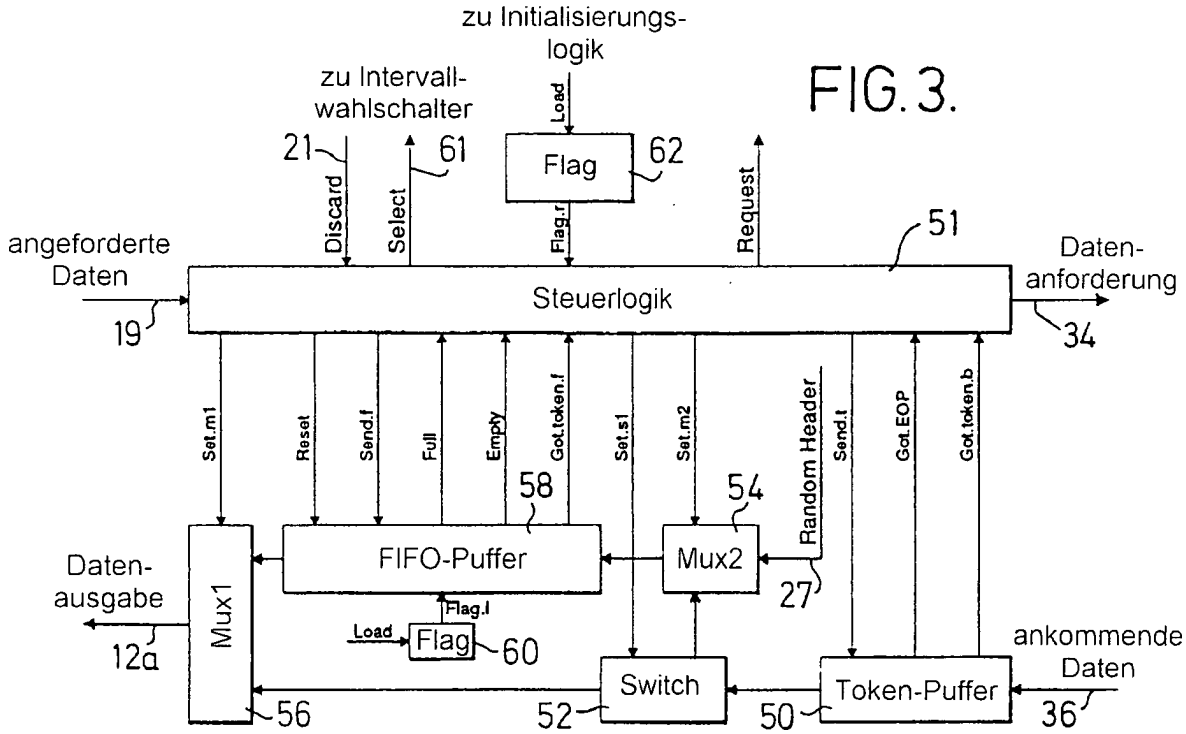

zu Initialisierungs-logik
zu Intervall-wahlschalter
Load
21
61
Flag
62
Discard
Select
Flag.r
Request
51
angeforderte Daten
19
Steuerlogik
Daten-anforderung
34
Set.m1
Reset
Send.f
Full
Empty
Got.token.f
Set.s1
Set.m2
Random Header
Send.t
Got.EOP
Got.token.b
58
54
Daten-ausgabe
12a
Mux1
FIFO-Puffer
Mux2
27
Flag.l
Load
Flag
60
ankommende Daten
Switch
Token-Puffer
56
52
50
36

FIG. 3.

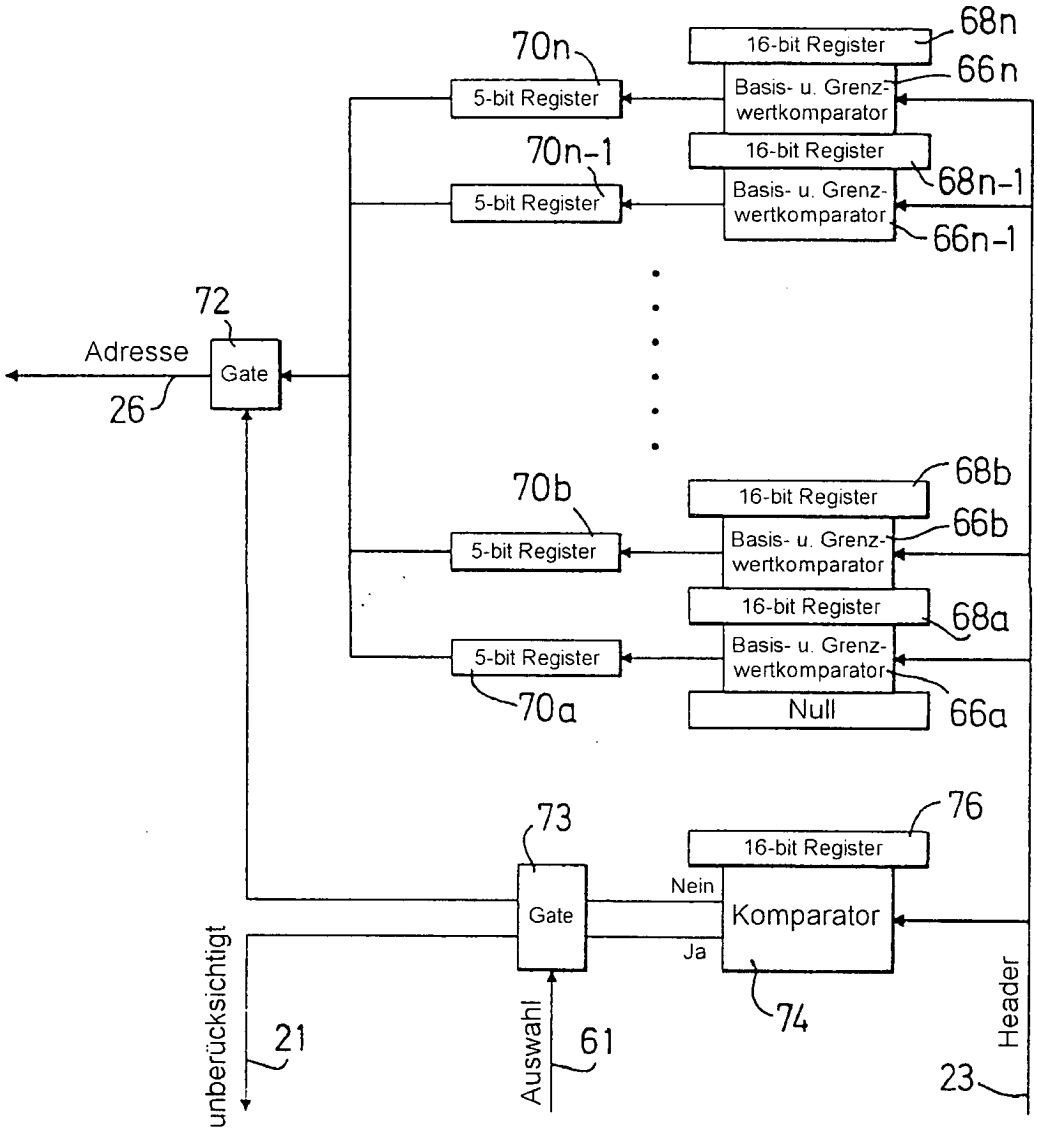

16-bit Register
68n
70n
5-bit Register
Basis- u. Grenz-
wertkomparator
66n
16-bit Register
70n-1
68n-1
5-bit Register
Basis- u. Grenz-
wertkomparator
66n-1
72
Adresse
Gate
26
16-bit Register
68b
70b
5-bit Register
Basis- u. Grenz-
wertkomparator
66b
16-bit Register
68a
5-bit Register
Basis- u. Grenz-
wertkomparator
70a
Null
66a
73
16-bit Register
76
Nein
Gate
Komparator
Ja
74
unberücksichtigt
21
Auswahl
61
Header
23

FIG. 4.

FIG.5.

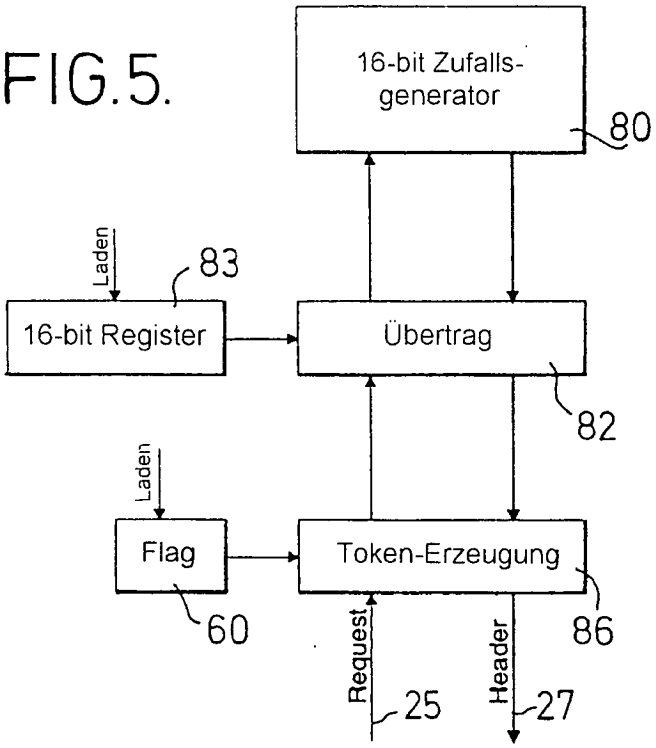

16-bit Zufalls-
generator
80
Laden
83
16-bit Register
Übertrag
82
Laden
Flag
Token-Erzeugung
60
86
Request
25
Header
27

FIG.6.

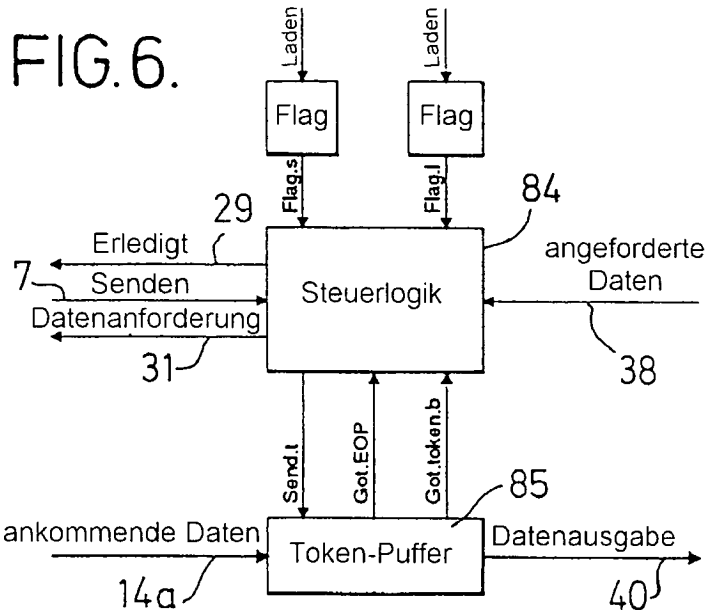
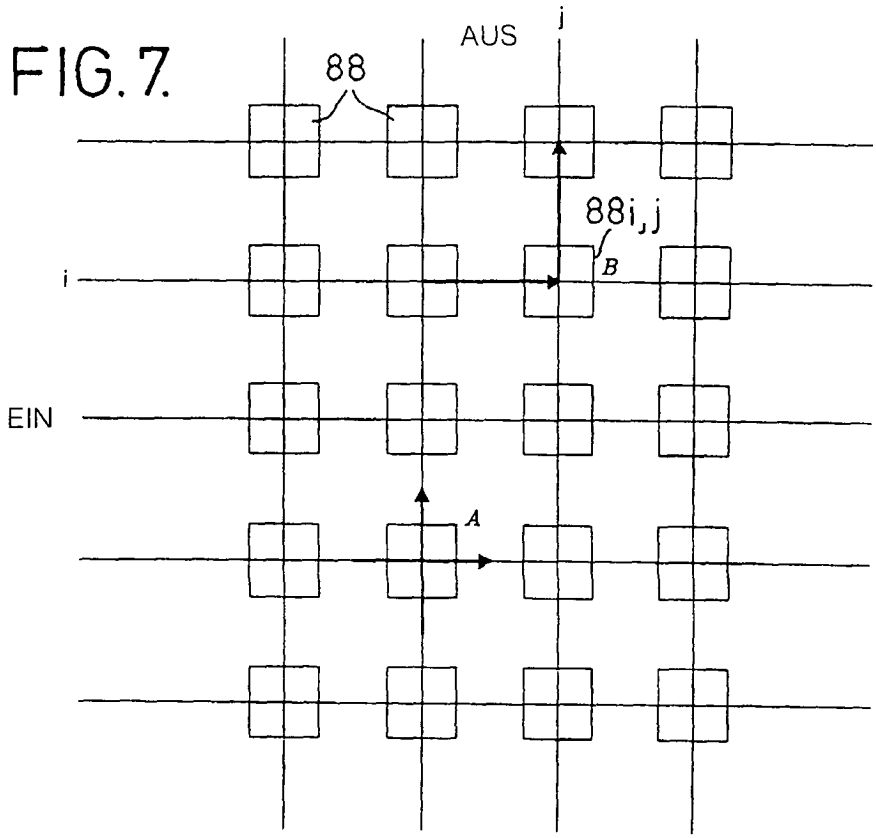

Laden
Laden
Flag
Flag
Flag.s
Flag.l
29
84
Erledigt
7
Senden
Datenanforderung
31
Steuerlogik
angeforderte
Daten
38
Send.t
Got.EOP
Got.token.b
85
ankommende Daten
14a
Token-Puffer
Datenausgabe
40

FIG. 7.


AUS
j
88
88i,j
B
i
EIN
A

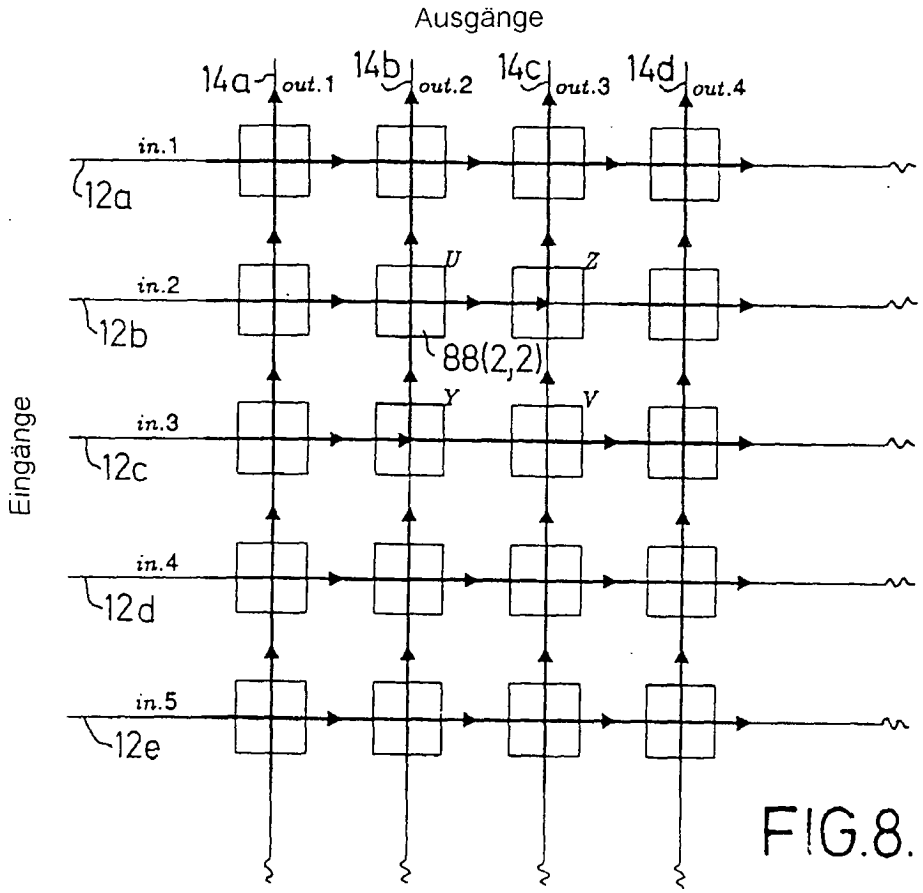

Ausgänge
14a
out.1
14b
out.2
14c
out.3
14d
out.4
in.1
12a
in.2
12b
in.3
12c
in.4
12d
in.5
12e
Eingänge
U
Z
88(2,2)
Y
V

FIG. 8.

FIG.9.

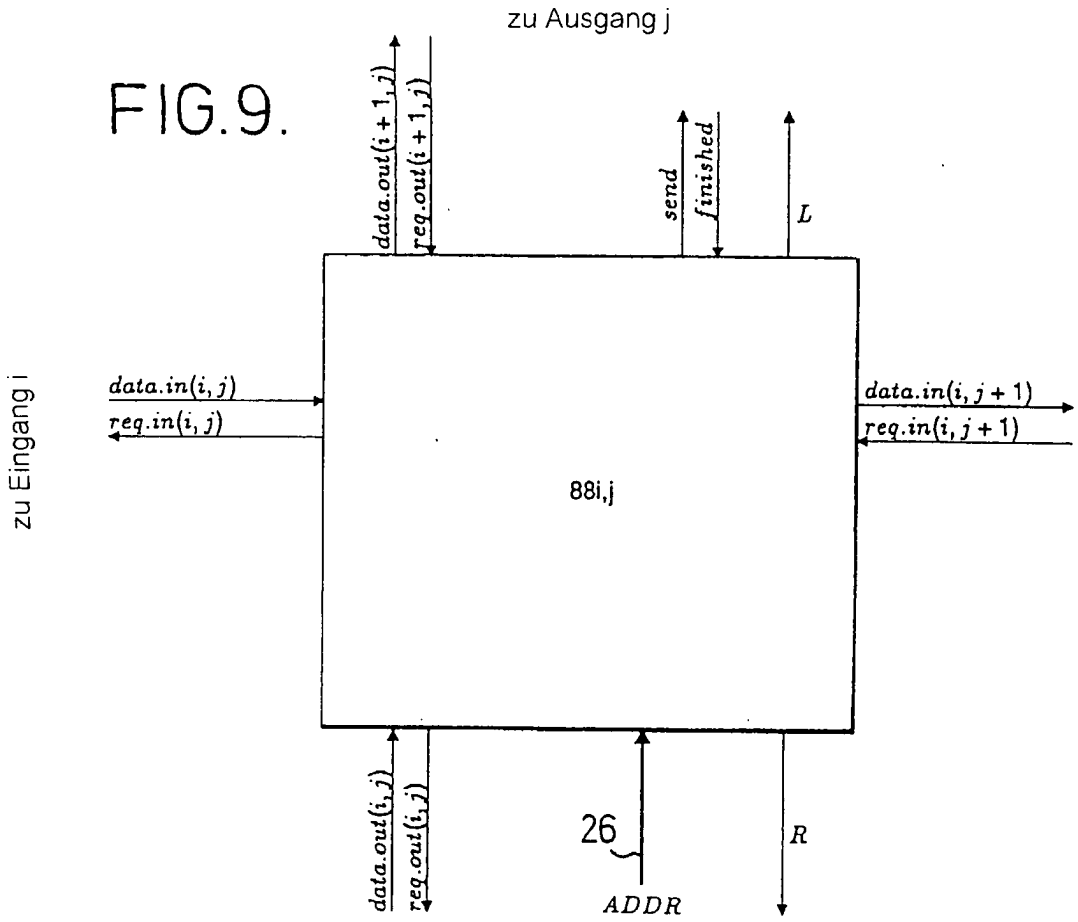

zu Ausgang j
zu Eingang i
data.out(i+1,j)
req.out(i+1,j)
send
finished
L
data.in(i,j)
req.in(i,j)
88i,j
data.in(i,j+1)
req.in(i,j+1)
data.out(i,j)
req.out(i,j)
26
ADDR
R

FIG.10.

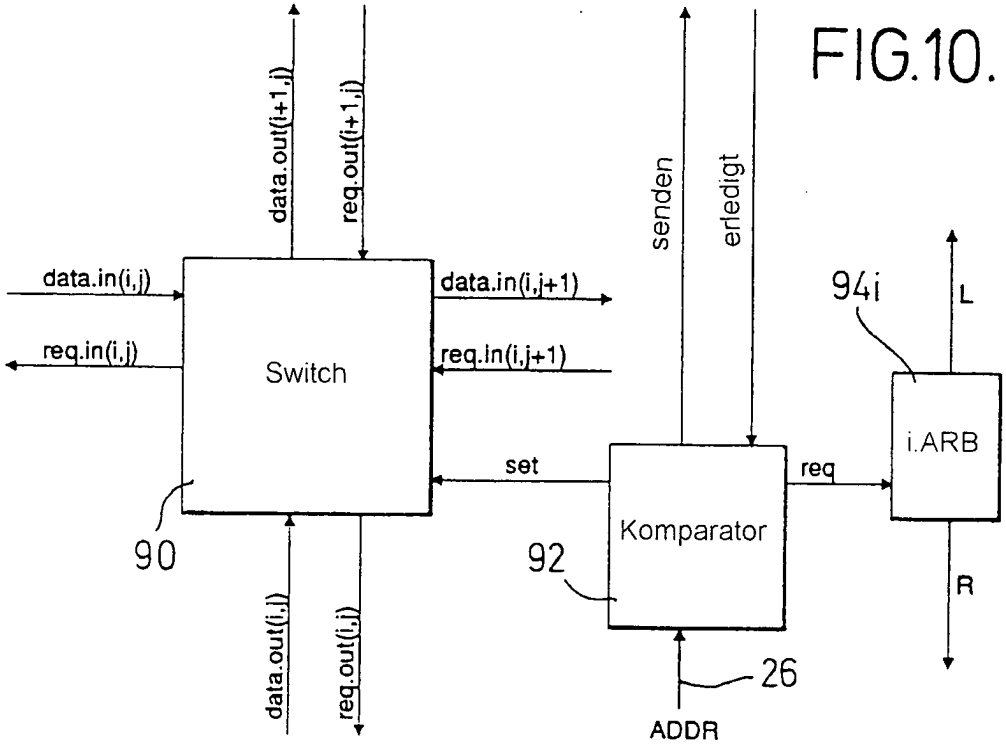
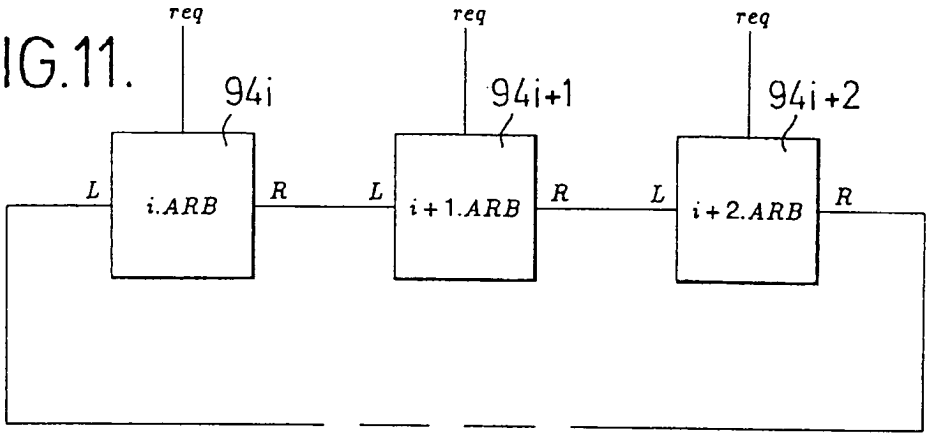

data.out(i+1,j)
req.out(i+1,j)
senden
erledigt
94i
L
data.in(i,j)
req.in(i,j)
Switch
data.in(i,j+1)
req.in(i,j+1)
i.ARB
set
req
Komparator
90
92
R
26
ADDR
data.out(i,j)
req.out(i,j)

FIG.11.



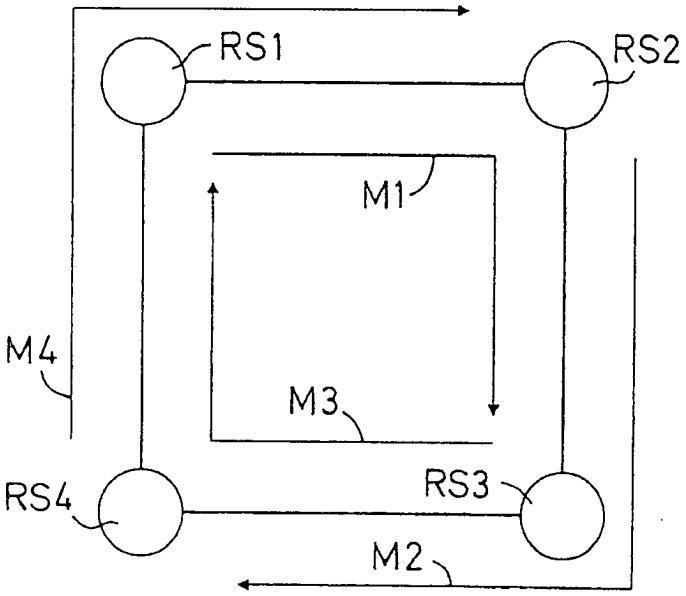
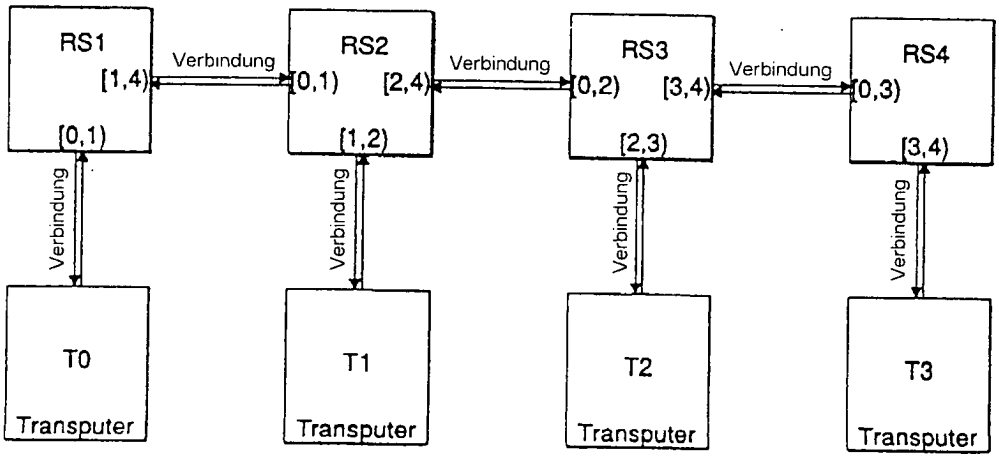


FIG.12.

FIG.14.



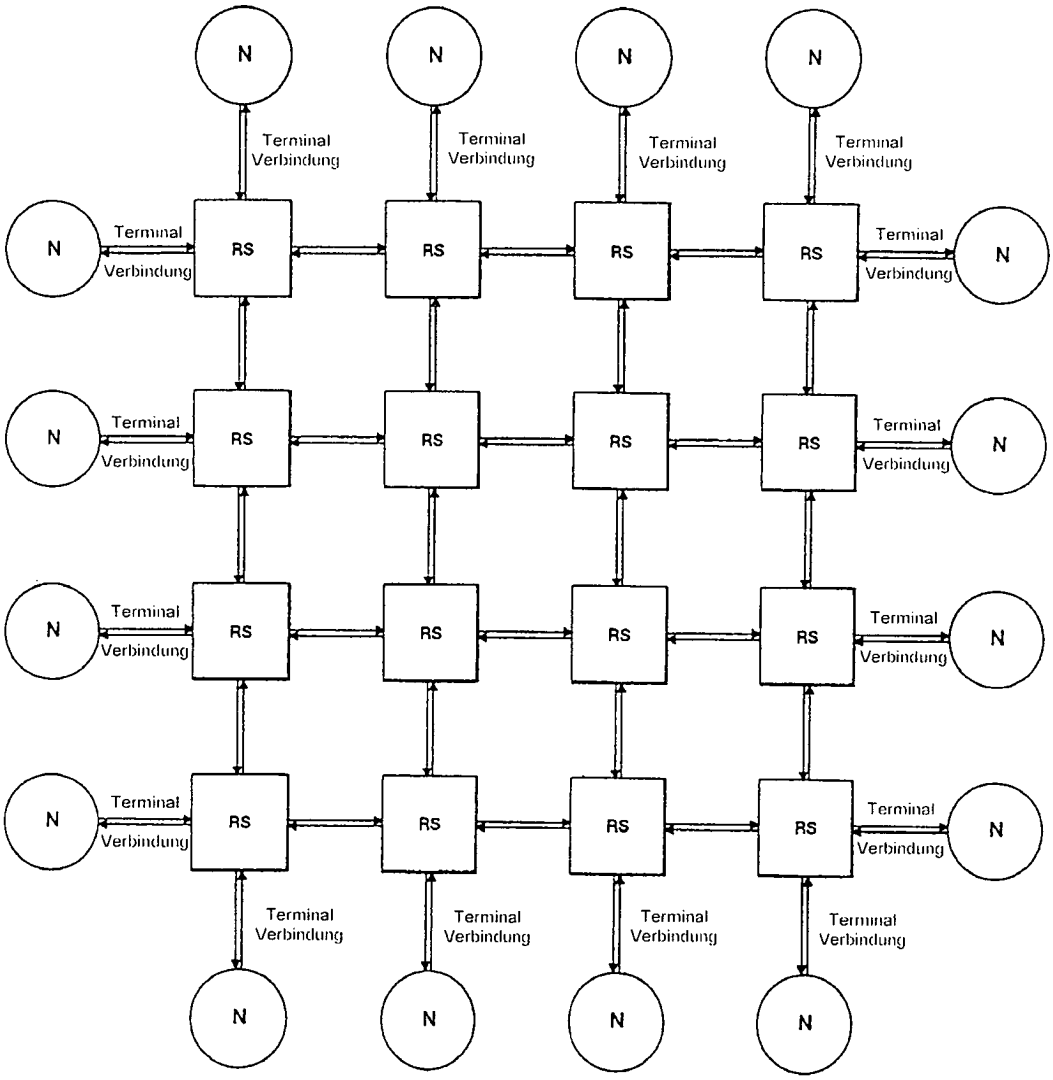


FIG.13.