



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 601 25 300 T2 2007.07.05**

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 195 955 B1**

(21) Deutsches Aktenzeichen: **601 25 300.0**

(96) Europäisches Aktenzeichen: **01 308 394.4**

(96) Europäischer Anmeldetag: **02.10.2001**

(97) Erstveröffentlichung durch das EPA: **10.04.2002**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **20.12.2006**

(47) Veröffentlichungstag im Patentblatt: **05.07.2007**

(51) Int Cl.⁸: **H04L 12/56 (2006.01)**
H04L 12/46 (2006.01)

(30) Unionspriorität:

237764 P	03.10.2000	US
805903	15.03.2001	US

(74) Vertreter:

Bosch, Graf von Stosch, Jehle
Patentanwaltsgesellschaft mbH, 80639 München

(73) Patentinhaber:

Altima Communications, Inc., Irvine, Calif., US

(84) Benannte Vertragsstaaten:

DE, FR, GB

(72) Erfinder:

Chien, Sheng Feng (William), I-Lan City 260, TW

(54) Bezeichnung: **Schaltungsanordnung zum Übertragen von Daten mit Datenpaketierung und-Depaketierung**

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

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

Beschreibung

HINTERGRUND DER ERFINDUNG:

Gebiet der Erfindung:

[0001] Die Erfindung betrifft ein Verfahren und ein System zum Einkapseln und Entkapseln von Quellidentifikationsinformationen in einem Datenpaket, das durch eine Vielzahl von Schaltanordnungen (Switches) übertragen wird.

[0002] Insbesondere betrifft die Erfindung eine neue Vermittlungsarchitektur, die auf leistungseffiziente und kostenempfindliche Märkte abgestellt ist, und die auf einem Halbleitersubstrat wie etwa einem Silizium-Chip implementiert werden kann.

Beschreibung des Standes der Technik:

[0003] Da sich die Computerleistung in den letzten Jahren immer mehr gesteigert hat, sind auch die Anforderungen an Computernetzwerke beträchtlich gestiegen; schnellere Computerprozessoren und höhere Speicherfähigkeiten erfordern Netzwerke mit höheren Bandbreitenfähigkeiten, um einen Hochgeschwindigkeits-transfer von beträchtlichen Mengen an Daten zu ermöglichen. Die allgemein bekannte Ethernet-Technologie, die auf zahlreichen IEEE Ethernet Standards beruht, ist ein Beispiel für eine Computervernetzungstechnologie, die modifiziert und verbessert werden konnte, um eine lebensfähige Computertechnologie zu bleiben. Eine vollständigere Diskussion von Netzwerksystemen aus dem Stand der Technik kann zum Beispiel in dem Dokument SWITCHED AND FAST ETHERNET von Breyer und Riley (Ziff-Davis, 1996) und in zahlreichen IEEE-Veröffentlichungen gefunden werden, die sich auf die IEEE 802 Standards beziehen. Auf der Grundlage des Open Systems Interconnect (OSI) 7-Schichten-Referenzmodells sind die Netzwerkfähigkeiten durch die Entwicklung von Zwischenverstärkern (repeaters), Brücken (bridges), Routern und neuerdings auch "Schaltanordnungen" bzw. "Vermittlungsstellen" ("Switches") gewachsen, die mit verschiedenen Arten von Kommunikationsmedien arbeiten. Grobdraht (thickwire), Dünndraht (thinwire), verdrehtes Leitungspaar und Glasfasern sind Beispiele für die Medien, die für Computernetzwerke verwendet worden sind und werden. Switches sind, wenn sie die Computervernetzung und das Ethernet betreffen, Vorrichtungen auf Hardwarebasis, die den Fluss von Datenpaketen oder Zellen auf der Grundlage von Zieladresseninformationen steuern, die in jedem Paket zur Verfügung stehen. Ein richtig ausgelegter und implementierter Switch sollte in der Lage sein, ein Paket zu empfangen und das Paket an einen geeigneten Ausgabeport mit einer Geschwindigkeit zu vermitteln, die als Leitungsgeschwindigkeit bezeichnet wird, welches die maximale Geschwindigkeit ist, zu der das spezielle Netzwerk in der Lage ist. Die Basis-Ethernet-Leitungsgeschwindigkeit beträgt bis zu 10 Megabit pro Sekunde, und beim Fast Ethernet beträgt diese bis zu 100 Megabit pro Sekunde. Ein Gigabit Ethernet ist in der Lage, Daten über ein Netzwerk mit einer Geschwindigkeit von bis zu 1.000 Megabit pro Sekunde zu übertragen. Da sich die Geschwindigkeiten erhöht haben, sind die Entwurfsbeschränkungen und Entwurfsanforderungen im Hinblick auf die nachfolgenden bestimmten Entwurfs- und Protokoll-Vorschriften und im Hinblick darauf, eine kostengünstige, kommerziell lebensfähige Lösung bereitzustellen, immer komplexer geworden. Zum Beispiel benötigt die Hochgeschwindigkeitsvermittlung einen Hochgeschwindigkeitsspeicher, um ein geeignetes Puffern von Paketdaten vorzusehen; ein herkömmliches dynamisches RAM (Dynamic Random Access Memory (DRAM)) ist relativ langsam und benötigt ein Hardware-gesteuertes Auffrischen. Deshalb führt die Geschwindigkeit von DRAMs als Pufferspeicher bei der Netzwerkvermittlung zum Verlust von wertvoller Zeit, und es wird beinahe unmöglich, den Switch oder das Netzwerk bei Leitungsgeschwindigkeit zu betreiben. Außerdem ist aufgrund der Tatsache, dass die Netzwerk-Switches immer komplizierter geworden sind im Hinblick darauf, dass sie Regeltabellen und eine Speichersteuerung erfordern, eine komplexe Multichip-Lösung notwendig, die einen logischen Schaltungsaufbau benötigt, der manchmal als Klebstoff-Logikschaltung (glue logic circuitry) bezeichnet wird, um zu ermöglichen, dass die verschiedenen Chips miteinander kommunizieren können. Außerdem sind Kosten-/Nutzenkompromisse im Hinblick auf teure, aber schnelle SRAMs gegenüber günstigen, aber langsamen DRAMs notwendig. Außerdem benötigen DRAMs aufgrund ihrer dynamischen Natur die Auffrischung der Speicherinhalte, um zu verhindern, dass diese verloren gehen. SRAMs leiden nicht an der Auffrischungs-Anforderung und weisen einen verringerten betrieblichen Overhead auf im Vergleich zu den DRAMs, wie etwa die Eliminierung von Seitenfehlschlägen, etc. Obwohl DRAMs eine adäquate Geschwindigkeit aufweisen, wenn sie auf Stellen auf der gleichen Seite zugreifen, wird die Geschwindigkeit verringert, wenn auf andere Seiten zugegriffen werden muss.

[0004] Unter Bezugnahme auf das vorher diskutierte OSI-7-Schichten-Referenzmodell weisen die höheren Schichten typischerweise mehr Informationen auf. Verschiedene Arten von Produkten stehen zur Durchfüh-

zung von vermittlungsbezogenen Funktionen auf verschiedenen Schichten des OSI-Modells zur Verfügung. Verteiler (Hubs) oder Zwischenverstärker arbeiten bei Schicht 1 und kopieren und "rundsenden" im Wesentlichen ankommende Daten zu einer Vielzahl von Speichen des Verteilers. Vermittlungsbezogene Vorrichtungen der Schicht 2 werden typischerweise als Multiport-Bridges bezeichnet und sind in der Lage, zwei separate Netzwerke zu überbrücken. Brücken können eine Tabelle von Abwicklungsregeln auf der Grundlage dessen errichten, welche MAC-(Medienzugangskontrolleinrichtungs-)Adressen an welchen Ports der Brücke existieren, und können Pakete, die für eine Adresse bestimmt sind, die sich auf einer gegenüberliegenden Seite der Brücke befindet, übermitteln. Brücken verwenden typischerweise das, was als der "Spannbaum"-Algorithmus bekannt ist, um potentielle Datenschleifen zu eliminieren; eine Datenschleife ist eine Situation, in der ein Paket in einem Netzwerk auf der Suche nach einer bestimmten Adresse endlos Runden dreht. Der Spannbaum-Algorithmus definiert ein Protokoll zur Verhinderung von Datenschleifen. Schicht-3-Switches, die manchmal auch als Router bezeichnet werden, können Pakete auf der Basis der Zielnetzwerkadresse weiterleiten. Schicht-3-Switches sind in der Lage, Adressen zu lernen und Tabellen davon zu verwalten, die den Port-Mappings entsprechen. Die Verarbeitungsgeschwindigkeit für Schicht-3-Switches kann verbessert werden, indem eine spezialisierte Hochleistungs-Hardware verwendet wird und die Host-CPU ausgeschaltet wird, so dass Befehlsentscheidungen nicht das Weiterleiten von Paketen verzögern.

[0005] Die US 5 081 612 beschreibt ein Verfahren und eine Vorrichtung zur Steuerung der Datenkommunikation in einem Multi-Netzwerk, wobei ein Identifikationscode zur Identifizierung eines Datenrahmens zu den Daten hinzugefügt wird, die von einer Station aus übertragen werden. Wenn eine Brücke, die Übertragungspfade miteinander verbindet, den Datenrahmen weitergeben soll, vergleicht sie den Identifikationscode mit dem Identifikationscode, der in einem Speicher gespeichert ist, unter einer Bedingung einer Aktivierung eines Timers und gibt nur den Datenrahmen weiter, der einen anderen Identifikationscode aufweist. Der Timer wird von einem zuerst kommenden Datenrahmen aktiviert und nach einem vorbestimmten Zeitintervall deaktiviert.

[0006] Die US 4 577 313 beschreibt ein Verfahren und eine entsprechende Vorrichtung zum Schutz der Integrität von Daten in einem Mehrschleifen-Netzwerk, in dem eine Quellstation einen Datenrahmen erzeugt und weiterleitet, der ein Rahmenprüfsequenzfeld (FSC-Feld) aufweist. Wenn sich ein solcher Rahmen an einer Quellbrücke befindet und nicht verworfen wird, wird die FSC bewahrt, indem sie in das Informationsfeld eingekapselt wird, welches ein spezielles Feld des Datenrahmens ist. Danach erzeugt die Quellbrücke eine neue FSC, hängt die neue FSC an, vergrößert z.B. die Länge des Rahmens, und leitet den abgeänderten Rahmen an die nächste Zwischenbrücke weiter.

[0007] Gemäß der Erfindung ist ein Verfahren und ein System zum Einkapseln und Entkapseln von Quellidentifikationsinformationen in einem Datenpaket bereitgestellt, wie diese jeweils in den unabhängigen Ansprüchen 1 und 10 definiert sind.

[0008] Vorteilhafte Merkmale der Erfindung sind in den abhängigen Unteransprüchen definiert.

KURZE BESCHREIBUNG DER ZEICHNUNGEN:

[0009] Die Aufgaben und Merkmale der Erfindung werden unter Bezugnahme auf die nachfolgende Beschreibung und die beigefügten Zeichnungen besser verständlich, in denen:

[0010] [Fig. 1](#) ein allgemeines Blockdiagramm von Elementen der vorliegenden Erfindung ist;

[0011] [Fig. 2](#) den Datenfluss in dem CPS-Kanal eines Netzwerk-Switch gemäß der vorliegenden Erfindung veranschaulicht;

[0012] [Fig. 3A](#) eine Verbundlistenstruktur des Paketpufferspeichers veranschaulicht;

[0013] [Fig. 3B](#) eine Verbundlistenstruktur des Paketpufferspeichers mit zwei Datenpaketen veranschaulicht;

[0014] [Fig. 3C](#) eine Verbundlistenstruktur des Paketpufferspeichers veranschaulicht, nachdem der Speicher, der von einem Datenpaket besetzt gewesen war, freigegeben ist;

[0015] [Fig. 3D](#) eine Verbundlistenstruktur des Paketpufferspeichers veranschaulicht, nachdem der Speicher, der von dem anderen Datenpaket besetzt gewesen war, freigegeben ist;

[0016] [Fig. 4](#) eine Veranschaulichung einer Vielzahl von Switches in einer gestapelten Konfiguration ist.

[0017] [Fig. 5](#) eine Veranschaulichung eines Datenpakets in einem Beispiel der Erfindung ist.

[0018] [Fig. 6](#) ein Flussdiagramm eines Beispiels der Erfindung ist.

[0019] [Fig. 7](#) eine Tabelle ist, die Informationen veranschaulicht, die zum Einkapseln und Entkapseln von Informationen aus einem Datenpaket verwendet werden.

[0020] [Fig. 8A](#) eine Veranschaulichung eines Beispiels einer Einkapselungsvorrichtung der Erfindung ist.

[0021] [Fig. 8B](#) eine Veranschaulichung eines Beispiels einer Entkapselungsvorrichtung der Erfindung ist.

AUSFÜHRLICHE BESCHREIBUNG DER ERFINDUNG

[0022] [Fig. 1](#) ist ein Beispiel eines Blockdiagramms einer Schaltanordnung/Vermittlungsstelle bzw. eines Switch **100** der vorliegenden Erfindung. In diesem Beispiel weist der Switch **100** 12 Ports **102(1)–102(12)** auf, die vollständig integrierte, IEEE-konforme Ports sein können. Alle diese 12 Ports **102(1)–102(12)** können 10BASE-T/100BASE-TX/FX-Ports sein, die jeweils ein physikalisches Element (PHY) aufweisen können, das den IEEE-Standards entsprechen kann. Jeder der Ports **102(1)–102(12)** weist in einem Beispiel der Erfindung eine Port-Geschwindigkeit auf, die zu einer bestimmten Konfiguration gezwungen werden kann oder die so gesetzt werden kann, dass die automatische Einstellung die optimale Geschwindigkeit für jeden Port einzeln festlegen wird. Jedes PHY jedes Ports kann mit einer verdrehten Leitungspaar-Schnittstelle, die TXOP/N und RXIP/N als Sende- und Empfangsprotokolle verwendet, oder mit einer Glasfaser-Schnittstelle verbunden sein, die FXOP/N und FXIP/N als Sende- und Empfangsprotokolle verwendet.

[0023] Jeder der Ports **102(1)–102(12)** weist eine Medienzugangskontrolleinrichtung (MAC; media access controller) auf, die mit jedem entsprechenden PHY verbunden ist. In einem Beispiel der Erfindung ist jede MAC eine vollständig konforme IEEE 802.3 MAC. Jede MAC kann bei 10 Mbps oder 100 Mbps arbeiten und unterstützt sowohl einen Vollduplexmodus, der eine Datenübertragung und einen Datenempfang zur gleichen Zeit erlaubt, als auch einen Halbduplexmodus, der es zwar erlaubt, dass die Daten entweder übertragen oder empfangen werden, aber nie beides zur gleichen Zeit.

[0024] Die Flusskontrolle wird von jeder der MACs bereitgestellt. Wenn die Flusskontrolle implementiert ist, wird der Fluss der ankommenden Datenpakete so verwaltet oder gesteuert, dass die Chancen, dass die Systemressourcen erschöpft werden, verringert werden. Obwohl das vorliegende Ausführungsbeispiel ein blockierungsfreier Leitungsgeschwindigkeits-Switch sein kann, kann es aufgrund des begrenzten Speicherraums, der zur Verfügung steht, zu Beschränkungen kommen. Zum Beispiel kann der verfügbare Speicher während Perioden von Paketüberflutungen (d.h., Paketrundsandanstürmen) ziemlich schnell erschöpft sein. Um die Funktionsfähigkeit des Switch in diesen Arten von Situationen zu verbessern, kann die vorliegende Erfindung zwei unterschiedliche Arten von Flusskontrolle implementieren. In dem Vollduplexmodus kann die vorliegende Erfindung zum Beispiel die IEEE 802.3x Flusskontrolle implementieren. In dem Halbduplexmodus kann die vorliegende Erfindung ein Kollisionsrückstauverfahren (collision back pressure scheme) implementieren.

[0025] In einem Beispiel der vorliegenden Erfindung weist jeder Port einen Latenzzeitblock auf, der mit der MAC verbunden ist. Jeder der Latenzzeitblöcke weist Sende- und Empfangs-FIFOs auf, die eine Schnittstelle zu dem Hauptpaketspeicher bereitstellen. In diesem Beispiel wird ein Paket dann, wenn ein Paket nicht erfolgreich von einem Port zu einem anderen Port innerhalb einer vorher festgesetzten Zeit übertragen wird, aus der Sendewarteschlange fallen gelassen.

[0026] Zusätzlich zu den Ports **102(1)–102(12)** kann eine Gigabit-Schnittstelle **104** an dem Switch **100** bereitgestellt sein. Die Gigabit-Schnittstelle **104** kann eine medienunabhängige Gigabit-Schnittstelle (GMII; Gigabit Media Independent Interface) und eine 10-Bit-Schnittstelle (TBI; Ten Bit Interface) unterstützen. Die GMII kann vollständig konform zu IEEE 802.3ab sein, was nur den Vollduplexbetrieb unterstützt. Die GMII kann Daten mit einer Übertragungsgeschwindigkeit von 8 Bit alle 8 ns weiterleiten, was zu einem Durchsatz von 2 Gbps führt, was sowohl Sende- als auch Empfangsdaten einschließt. Zusätzlich zu der GMII kann die Gigabit-Schnittstelle **104** so konfiguriert sein, dass sie eine TBI ist, was mit vielen Industrie-Standard-Fasertreibern kompatibel ist. Da in einigen Ausführungsbeispielen der Erfindung die MDIO/MDC-Schnittstellen (optische Schnittstellen) nicht unterstützt werden, wird die Gigabit-PHY (physikalische Schicht) von dem Systemdesigner in den richtigen Modus gesetzt.

[0027] Die Gigabit-Schnittstelle **104** weist wie die Ports **102(1)–102(12)** ein PHY, eine Gigabit-Medienzu-

gangskontrolleinrichtung (GMAC) und einen Latenzzeitblock auf. Die GMAC kann eine vollständig konforme IEEE 802.3z MAC sein, die nur bei 1 Gbps Vollduplex arbeitet, und kann durch das PHY mit einer vollständig konformen GMII- oder TBI-Schnittstelle verbunden sein. In diesem Beispiel stellt die GMAC **108** Vollduplex-Flusskontrollmechanismen und eine preisgünstige Stapellösung für entweder einen verdrehten Leitungspaar-Modus oder einen TBI-Modus bereit, der eine Inband-Signalisierung für die Verwaltung verwendet. Diese preisgünstige Stapellösung sieht eine Ringstruktur vor, um jeden Switch zu verbinden, wobei nur ein Gigabit-Port verwendet wird.

[0028] Eine CPU-Schnittstelle **106** ist an dem Switch **100** vorgesehen. In einem Beispiel der vorliegenden Erfindung ist die CPU-Schnittstelle **106** eine asynchrone 8- oder 16-Bit-E/A-Geräteschnittstelle. Durch diese Schnittstelle kann eine CPU interne Register lesen, Pakete empfangen, Pakete senden und Unterbrechungen (Interrupts) zulassen. Die CPU-Schnittstelle **106** erlaubt es auch, dass ein Spannbaum-Algorithmus implementiert werden kann. In einem Beispiel der vorliegenden Erfindung steht ein Bausteinauswahl-Pin zur Verfügung, der es erlaubt, dass eine einzige CPU zwei Switches steuert. In diesem Beispiel wird ein Unterbrechungs-Pin, wenn er niedrig angesteuert wird (d.h., so angesteuert wird, dass er in den aktiven Zustand übergeht), was einen Pull-Up-Widerstand benötigt, es erlauben, dass so viele Switch-Unterbrechungen miteinander verbunden werden können, wie dies gewünscht wird.

[0029] Eine Switching Fabric **108** ist in einem Beispiel der vorliegenden Erfindung ebenfalls an dem Switch **100** angeordnet. Die Switching Fabric **108** kann eine Betätigung aller Ports bei voller Leitungsgeschwindigkeit erlauben. Es kann auch eine gemeinsam genutzte hybride Speicherlösung implementiert werden, um die Bandbreiten- und Speicheranforderungen zu minimieren. Diese Architektur erlaubt eine effiziente Übertragung von Paketen mit einer niedrigen Latenzzeit durch den Switch und unterstützt auch Adresslern- und Alterungsmerkmale, VLAN, Portgruppierung und Portspiegelung.

[0030] Die Speicherschnittstellen **110**, **112** und **114** können an dem Switch **100** angeordnet sein und können die Trennung von Daten- und Steuerinformationen gestatten. Die Paketpufferspeicher-Schnittstelle (PBM; packet buffer memory interface) **110** handhabt eine Paketdatenspeicherung, während die Sendewarteschlangenspeicher-Schnittstelle (TXM; transmit queue memory interface) **112** eine Liste von Paketen bewahrt, die übertragen werden sollen, und eine Adresstabellen-/Steuerspeicher-Schnittstelle (ATM; address table/control memory interface) **114** handhabt die Adresstabelle und die Header-Informationen. Jede dieser Schnittstellen verwendet in einem Ausführungsbeispiel der Erfindung einen SSRAM, der mit verschiedenen Gesamtmengen und Chipgrößen konfiguriert sein kann.

[0031] Die PBM **110** ist an dem Switch **100** angeordnet und kann einen externen Paketpufferspeicher (nicht gezeigt) aufweisen, der dazu verwendet wird, das Paket während der Vermittlungsoperationen zu speichern. In einem Beispiel der Erfindung ist der Paketpufferspeicher aus mehreren 256-Byte-Puffern aufgebaut. Deshalb kann sich ein einziges Paket über mehrere Puffer in dem Speicher erstrecken. Diese Struktur erlaubt eine effiziente Speicherbenutzung und minimiert den Bandbreitenaufwand. Der Paketpufferspeicher kann so konfigurierbar sein, dass bis zu 4 MByte an Speicher pro Chip für eine Gesamtsumme von 8 MByte pro 24+2 Ports verwendet werden können. In diesem Beispiel wird eine effiziente Speicherbenutzung verwaltet, indem 256-Byte-Blöcke zugewiesen werden, was eine Speicherung von bis zu 32K Paketen gestattet. Die PBM **110** kann 64 Bit breit sein und kann entweder einen 64 Bit breiten Speicher oder einen 32 Bit breiten Speicher verwenden und kann bei 100 MHz betrieben werden.

[0032] Die TXM **112** ist an dem Switch **100** angeordnet und kann einen externen Sendewarteschlangenspeicher (nicht gezeigt) aufweisen. Die TXM **112** verwaltet in diesem Beispiel 4 Prioritätswarteschlangen pro Port und erlaubt 64K Pakete pro Chip und bis zu 128K Pakete pro System. Die TXM **112** kann mit einer Geschwindigkeit von bis zu 100 MHz betrieben werden.

[0033] Die ATM **114** kann an dem Switch **100** angeordnet sein und kann einen externen Adresstabellen-/Steuerspeicher (nicht gezeigt) aufweisen, der dazu verwendet wird, die Adresstabelle und die Header-Informationen zu speichern, die jedem 256-Byte-Abschnitt der PBM **110** entsprechen. Der Adresstabellen-/Steuerspeicher erlaubt bis zu 16K einzigartige Unicast-Adressen. Der restliche zur Verfügung stehende Speicher wird für Steuerinformationen verwendet. Die ATM **114** läuft in diesem Beispiel bei bis zu 133 MHz.

[0034] Der Switch **100** weist in einem Beispiel der Erfindung einen Flusskontroll-Manager **116** auf, der den Fluss der Paketdaten verwaltet. Während jeder Port mehr und mehr Daten zu dem Switch sendet, kann der Flusskontroll-Manager **116** den Betrag an Speicher, der von jedem Port **102(1)–102(12)** des Switch **100** und dem Switch insgesamt verwendet wird, überwachen. Wenn in diesem Beispiel einer der Ports **102(1)–102(12)**

oder der Switch als Ganzes zu viel Speicher benutzt, wird der Flusskontroll-Manager **116** über den ATM-Bus Befehle ausgeben, die den Port oder den Switch auffordern, langsamer zu werden und schließlich, falls notwendig, Pakete fallen zu lassen.

[0035] Zusätzlich zu dem Flusskontroll-Manager **116** weist der Switch **100** auch einen Startpunkt-Manager (SPM; start point manager) **118**, einen Weiterleit-Manager (FM, forwarding manager) **120** und einen Adress-Manager (AM) **122** auf.

[0036] Der Startpunkt-Manager (SPM) **118** verfolgt in einem Beispiel der vorliegenden Erfindung, welche Blöcke des Speichers in der PBM **110** verwendet werden und welche Blöcke des Speichers frei sind.

[0037] Der Weiterleit-Manager **120** kann zum Beispiel Paketdaten an entsprechende Ports zur Übertragung weiterleiten.

[0038] Der Adress-Manager (AM) **122** kann die Adresstabelle verwalten, was das Lernen von Quelladressen, das Zuweisen von Headern zu Paketen und das Verfolgen dieser Adressen umfasst. In einem Beispiel der Erfindung verwendet der AM **122** die Alterung, um Adressen aus der Adresstabelle zu entfernen, die für ein vorgeschriebenes Zeitintervall oder für eine spezifizierte Sequenz von Ereignissen nicht verwendet worden sind.

[0039] Ein Erweiterungsport **124** kann ebenfalls an dem Switch **100** vorgesehen sein, um zwei Switches miteinander zu verbinden. Dies erlaubt einen Betrieb bei voller Leitungsgeschwindigkeit auf fünfundzwanzig 100M Ports (was einen CPU-Port mit einschließt) und zwei Gigabit-Ports. Der Erweiterungsport **124** gestattet in diesem Beispiel, dass 4,6 Gbps an Daten zwischen Switches übertragen werden können.

[0040] Eine LED-Steuerinrichtung **126** kann ebenfalls an dem Switch **100** vorgesehen sein. Die LED-Steuerinrichtung **126** aktiviert entsprechende LEDs, um einem Benutzer notwendige Statusinformationen zu geben. Jeder Port der Ports **102(1)–102(12)** weist in einem Beispiel der Erfindung 4 separate LEDs auf, die Statusinformationen pro Port bereitstellen. Die LEDs sind vollständig programmierbar und sind aus Port-LEDs und anderen LEDs zusammengesetzt. Jede LED kann einen Standardzustand für jeden der vier Port-LEDs umfassen. Ein Beispiel des Standardbetriebs jeder der Port-LEDs ist unten gezeigt.

LED	STANDARD BETRIEB
0	<p>Geschwindigkeitsanzeige</p> <p>AUS = 10 Mbps oder keine Verbindung</p> <p>EIN = 100 Mbps</p>
1	Voll-/Halb-/Kollisionsduplex

	<p>AUS = Der Port ist im Halbduplex oder keine Verbindung</p> <p>BLINKEN = Der Port ist im Halbduplex und eine Kollision ist aufgetreten</p> <p>EIN = Der Port ist im Vollduplex</p>
2	<p>Verbindungs-/Aktivitätsanzeige</p> <p>AUS = Zeigt an, das der Port keine Verbindung hat</p> <p>BLINKEN = Verbindung ist vorhanden und eine Empfangs- oder Übertragungsaktivität findet auf den Medien statt</p> <p>EIN = Verbindung vorhanden ohne Aktivität</p>
3	<p>Warnzustand</p> <p>AUS = Keine Warnzustände, Port arbeitet normal</p> <p>EIN = Der Port hat einen Isolierzustand entdeckt</p>

[0041] Zusätzlich zu den Standardoperationen für die Port-LEDs kann jeder der Port-LEDs durch Register programmiert werden. Diese Register können in einem Beispiel der Erfindung von einer CPU errichtet werden. Dadurch, dass programmierbare Register vorhanden sind, die die LEDs steuern, kann eine volle Anpassung der Systemarchitektur an die kundenspezifische Situation realisiert werden, einschließlich der Programmierbarkeit der Blinkgeschwindigkeit.

[0042] Jede der LEDs weist eine Tabelle auf, in der die Registerbits R_{Ax} , R_{Bx} und R_{Cx} so gesetzt werden können, dass sie einen breiten Bereich von Informationen bereitstellen. Zum Beispiel können die Registerbits R_{Ax} , R_{Bx} und R_{Cx} so gesetzt werden, dass sie bestimmen, wann LED_{EIN} , LED_{BLINK} und LED_{AUS} aktiviert oder deaktiviert werden. Zusätzlich zu den Port-LEDs gibt es weitere LEDs, die den Status des Switch anzeigen.

[0043] Die Register **128** sind in diesem Beispiel der vorliegenden Erfindung in dem Switch **100** angeordnet. Die Register **128** sind volle Register, die ein Konfigurations-Status- und Remote-Monitoring-(RMON)-Management ermöglichen. In diesem Beispiel sind die Register **128** in Gruppen und Offsets angeordnet. Es gibt 32 Adressgruppen, von denen jede bis zu 64 Register enthalten kann.

[0044] [Fig. 2](#) ist eine Veranschaulichung eines Ausführungsbeispiels der Erfindung, die einen PBM-Bus, einen ATM-Bus und einen TXM-Bus für Kommunikationen mit anderen Abschnitten des Switch umfasst. In diesem Beispiel ist die PBM **110** mit dem PBM-Bus und einem externen PBM-Speicher verbunden; die TXM **112** ist mit dem TXM-Bus und einem externen TXM-Speicher verbunden; und die ATM **114** ist mit dem ATM-Bus und einem externen ATM-Speicher verbunden. Jeder dieser Sende-(TX)- und Empfangs-(RX)-Abschnitte der Ports **102(1)–102(12)** ist für Kommunikationszwecke mit dem PBM-Bus, dem ATM-Bus und dem TXM-Bus verbunden.

[0045] Der FM **120** ist mit jedem der Ports **102(1)–102(12)** direkt verbunden und ist auch mit dem ATM-Bus für Kommunikationen mit anderen Abschnitten des Switch verbunden. Der SPM **118** und der AM **122** sind ebenfalls mit dem ATM-Bus zum Zwecke von Kommunikationen mit anderen Abschnitten des Switch verbunden.

[0046] Der Betrieb des Switch **100** für die Übertragung eines Unicast-Pakets (d.h., eines Pakets, das für einen einzigen Port zur Ausgabe bestimmt ist) in einem Beispiel der Erfindung wird unter Bezugnahme auf [Fig. 2](#) wie folgt beschrieben.

[0047] In diesem Beispiel wird das System im Anschluss an die Freigabe eines Hardware-Rücksetz-Pin initialisiert. Eine Reihe von Initialisierungsschritten wird auftreten, einschließlich der Initialisierung des externen Pufferspeichers und der Adresstabelle. Alle Ports in dem Switch werden dann deaktiviert, und die CPU wird den Paketverkehr freigeben, indem sie ein Freigaberegister einstellt. Wenn Verbindungen an den Ports (Port **102(1)**–**102(12)** und Gigabit-Port **104**) verfügbar werden, wird ein SPT-Protokoll diese Ports bestätigen, und die Ports werden aktiviert. Nachdem der Initialisierungsprozess vollendet ist, kann der normale Betrieb des Switch **100** beginnen.

[0048] In diesem Beispiel wird dann, wenn ein Port initialisiert und aktiviert worden ist, ein PORT_ACTIVE-Befehl ausgegeben. Dies zeigt an, dass der Port bereit ist, Datenpakete zu übertragen und zu empfangen. Wenn aus irgendwelchen Gründen ein Port ausgeht oder deaktiviert wird, wird ein PORT_INACTIVE-Befehl ausgegeben.

[0049] Während der Unicast-Übertragung kann ein Paket zu dem Port **102(1)** von einer externen Quelle zu dem Empfangs-(RX)-PHY des Ports **102(1)** gesendet werden.

[0050] In diesem Beispiel wird die RX-MAC des Ports **102(1)** warten, bis ein Rahmenbeginn-Begrenzer (SFD; start of frame delimiter) für das Paket erfasst wird. Wenn der SFD von dem RX-MAC-Abschnitt des Ports **102(1)** erfasst wird, wird die RX-MAC das Paket in eine Empfangs-(RX)-FIFO des Latenzzeitblocks des Ports **102(1)** platzieren. Wenn die RX-FIFO voll besetzt wird, wird der Port **102(1)** einen freien Empfangspuffer anfordern, indem er eine Nachricht über den ATM-Bus zu dem Adress-Manager (AM) **122** sendet. Wenn der Zugriff auf den ATM-Bus gewährt ist, sendet der RX-FIFO-Latenzzeitblock des Ports **102(1)** Pakete, die in dem RX-FIFO empfangen worden sind, zu dem externen PBM-Speicher durch den PBM-Bus und die PBM **110**, bis das Ende des Pakets erreicht ist.

[0051] Der PBM-Speicher besteht in diesem Beispiel aus 256-Byte-Puffern. Deshalb kann sich ein Paket über mehrere Puffer in dem Paketpufferspeicher erstrecken, wenn die Paketgröße größer als 256 Byte ist. Verbindungen zwischen den Paketpuffern werden in einem Beispiel der vorliegenden Erfindung durch ein Verbundlistensystem aufrechterhalten. Ein Verbundlistensystem erlaubt eine effiziente Speicherbenutzung und einen minimierten Bandbreitenaufwand und wird unter Bezugnahme auf die [Fig. 3A–Fig. 3D](#) noch ausführlicher erläutert werden.

[0052] Zu der gleichen Zeit, in der Pakete zu dem externen PBM-Speicher gesendet werden, wird der Port auch die Quelladresse zu dem Adress-Manager (AM) **122** senden und eine Filtertabelle vom AM **122** anfordern.

[0053] Wenn das Paket "gut" ist, schreibt der Port die Header-Informationen durch den ATM-Bus und die ATM **114** in den ATM-Speicher. Der Port sendet auch einen RECEP_COMPL-Befehl über den ATM-Bus, der anzeigt, dass ein Paketempfang vollendet ist. Andere Informationen werden ebenfalls zusammen mit dem RECEP_COMPL-Befehl gesendet, wie etwa die Startadresse und die Filtertabelle, die anzeigt, auf welchen Ports das Paket hinausgesendet werden soll. Zum Beispiel würde eine Filtertabelle, die eine Zeichenfolge wie etwa "01111111111" aufweist, das Paket zu allen Ports außer zu Port 1 senden und würde eine Zahl von 11 aufweisen. Die Zahl ist einfach die Anzahl an Ports, zu denen das Paket gesendet werden soll.

[0054] Der Weiterleit-Manager (FM) **120** überwacht ständig den ATM-Bus, um festzustellen, ob ein RECEP_COMPL-Befehl ausgegeben worden ist. Wenn der FM **120** feststellt, dass ein RECEP_COMPL-Befehl ausgegeben worden ist, wird der Weiterleit-Manager (FM) **120** die Filtertabelle dazu verwenden, Pakete zu den entsprechenden Ports zu senden. Es sei angemerkt, dass ein Paket nicht weitergeleitet wird, wenn eine der nachfolgenden Bedingungen erfüllt ist:

- a. Das Paket enthält einen CRC-Fehler
- b. Das PHY signalisiert einen Empfangsfehler
- c. Das Paket umfasst weniger als 64 Byte
- d. Das Paket ist größer als 1518 Byte oder 1522 Byte, was von den Registereinstellungen abhängt
- e. Das Paket wird nur zu dem gleichen Port wie dem empfangenden Port weitergeleitet

[0055] Wenn der FM **120** einen RECEP_COMPL-Befehl auf dem ATM-Bus erfasst, wird der FM **120** entscheiden, ob das Paket dafür gedacht ist, zu einem seiner Ports gesendet zu werden. Der RECEP_COMPL-Befehl umfasst Informationen wie zum Beispiel eine Filtertabelle, einen Startzeiger, Prioritätsinformationen und andere mannigfaltige Informationen. Der FM **120** wird die Filtertabelle lesen, um festzustellen, ob das Paket von einem seiner Ports gesendet werden soll. Wenn bestimmt wird, dass das Paket von einem seiner Ports gesen-

det werden soll, wird der FM **120** die RECEP_COMPL-Befehlsinformation direkt zu dem Port senden. In diesem Fall wird die RECEP_COMPL-Befehlsinformation zu dem TX-FIFO des Ports **102(12)** gesendet.

[0056] Wenn der Port beschäftigt ist, wird die RECEP_COMPL-Befehlsinformation zu dem TXM-Speicher durch den TXM-Bus und die TXM **112** transferiert. Der TXM-Speicher ist einfach eine Warteschlange von Paketen, die übertragen werden sollen. Der TXM-Speicher wird auf einer pro Port Basis zugewiesen, so dass dann, wenn es zehn Ports gibt, zehn Warteschlangen in dem TXM-Speicher vorhanden sind, die jedem Port zugeordnet sind. Wenn jede der Sendevorrichtungen der Ports frei wird, wird jeder Port die nächste RECEP_COMPL-Befehlsinformation lesen, die in dem TXM-Speicher gespeichert ist. Der TX-FIFO des Ports **102(12)** wird als einen Teil der RECEP_COMPL-Befehlsinformation einen Startzeiger empfangen, der auf einen Header in dem ATM-Speicher über den ATM-Bus zeigen wird, der wiederum über den PBM-Bus auf die Speicherzelle eines Pakets in dem PBM-Speicher zeigen wird. Der Port wird an diesem Punkt anfordern, dass das Paket in den Sende-(TX)-FIFO des Ports **102(12)** geladen wird und wird es durch die MAC und das PHY des Ports **102(12)** hinaussenden.

[0057] Wenn der Port in einem Halbduplexmodus ist, ist es möglich, dass eine Kollision auftreten könnte und die Paketübertragung zwingen könnte, noch einmal zu beginnen. Wenn dies passiert, ruft der Port einfach den Bus-Master erneut ab und lädt das Paket erneut und beginnt wieder von vorn. Wenn aber die Anzahl an aufeinander folgenden Kollisionen übermäßig wird, dann wird das Paket aus der Übertragungswarteschlange fallen gelassen.

[0058] Wenn der Port ein Paket erfolgreich überträgt, wird der Port dem FM **120** mit dem gegenwärtigen Puffer signalisieren, dass dies erledigt ist. Der FM **120** wird dann einen Zähler dekrementieren, der anzeigt, wie viele weitere Ports das Paket noch übertragen müssen. Wenn zum Beispiel ein Paket für elf Ports zur Ausgabe bestimmt ist, wird der Zähler in diesem Beispiel auf 11 gesetzt. Jedes Mal dann, wenn ein Paket erfolgreich übertragen ist, dekrementiert der FM **120** den Zähler um Eins. Wenn der Zähler Null erreicht, wird dies anzeigen, dass alle spezifizierten Ports das Paket erfolgreich übermittelt haben. Der FM **120** wird dann einen FREE-Befehl über den ATM-Bus ausgeben, der anzeigt, dass der Speicher, der von dem Paket in dem PBM-Speicher belegt ist, nicht mehr länger benötigt wird und nun für eine andere Verwendung freigegeben werden kann.

[0059] Wenn der SPM **118** einen FREE-Befehl über den ATM-Bus entdeckt, werden Schritte unternommen, um anzuzeigen, dass der Raum, der von dem Paket eingenommen wurde, nun freier Speicher ist.

[0060] Multicast- und Broadcast-Pakete werden genauso behandelt wie Unicast-Pakete, mit der Ausnahme, dass ihre Filtertabellen anzeigen werden, dass alle Ports das Paket übertragen sollen. Dies wird die Weiterleit-Manager dazu zwingen, das Paket auf allen Ports auszugeben.

[0061] [Fig. 3A](#) ist eine Veranschaulichung einer PBM-Speicherstruktur in einem Beispiel der Erfindung. Die PBM-Speicherstruktur **300** ist eine Verbundliste von 256-Byte-Segmenten **302, 304, 306, 308, 310, 312, 314** und **316**. In diesem Beispiel ist das Segment **302** der free_head (freier Kopf), der den Beginn der Verbundliste des freien Speichers anzeigt, und das Segment **316** ist der free_tail (freies Ende), der das letzte Segment des freien Speichers anzeigt.

[0062] In [Fig. 3B](#) sind zwei Pakete empfangen und in dem PBM-Speicher gespeichert worden. Das Paket 1 belegt die Segmente **302, 306** und **308**, und das Paket 2 belegt das Segment **304**. Die Segmente **310, 312, 314** und **316** sind freier Speicher. Das Segment **310** ist der free_head, der den Beginn des freien Speichers anzeigt, und das Segment **316** ist der free_tail, der das Ende des freien Speichers anzeigt.

[0063] In [Fig. 3C](#) ist das Paket 1 vollständig übertragen worden und der Weiterleit-Manager (FM) hat einen FREE-Befehl ausgegeben. Da das Paket 1 bereits in einem Verbundlistenformat ist, ist es für den SPM leicht, den Speicher, der von dem Paket 1 belegt worden ist, zu der freien Speicher-Verbundliste zu addieren. Der free_head, das Segment **310**, bleibt gleich. Aber der free_tail wird geändert. Dies wird erreicht, indem das Segment **316** mit dem Beginn des Pakets 1, dem Segment **302**, verbunden wird, und das letzte Segment des Pakets 1, das Segment **308**, als der free_tail bezeichnet wird. Als eine Folge davon gibt es eine Verbundliste, die mit dem Segment **310** beginnt, das an das Segment **312** angeschlossen ist, das Segment **312** ist an das Segment **314** angeschlossen, das Segment **314** ist an das Segment **316** angeschlossen, das Segment **316** ist an das Segment **302** angeschlossen, das Segment **302** ist an das Segment **306** angeschlossen und das Segment **306** ist an das Segment **308** angeschlossen, wobei das Segment **308** der free_tail ist.

[0064] [Fig. 3D](#) veranschaulicht in diesem Beispiel einfach den PBM-Speicher, nachdem das Paket 2 erfolgreich übertragen worden ist und der Weiterleit-Manager einen FREE-Befehl über den ATM-Bus ausgegeben hat. Der SPM wird den FREE-Befehl entdecken und dann den Speicherraum, der von dem Paket 2 in dem PBM-Speicher belegt worden ist, zu der Verbundliste des freien Speichers hinzufügen. In diesem Beispiel ist das Segment **308** mit dem Speicher verbunden, der von dem Paket 2 belegt worden ist, nämlich mit dem Segment **304**, und das Segment **304** wird als der `free_tail` identifiziert.

[0065] [Fig. 4](#) ist eine Veranschaulichung eines Beispiels von gestapelten Switches der vorliegenden Erfindung. Der Switch **400**, SW1, weist einen Gigabit-Port **401** auf, der Switch **402**, SW2, weist einen Gigabit-Port **403** auf, der Switch **404**, SW3, weist einen Gigabit-Port **405** auf, und der Switch **406**, SW4, weist einen Gigabit-Port **407** auf. Die SW1, SW2, SW3 und SW4 sind miteinander in einer Simplex-Schleife verbunden, was einfach bedeutet, dass die Switches miteinander durch Ports so verbunden sind, dass sie eine Schleife bilden, so dass die Daten durch die Switches in einer Richtung fließen, wie dies in [Fig. 4](#) dargestellt ist. In diesem Beispiel ist der SW1 mit dem SW2 unter Verwendung des Gigabit-Ports **401** und des Gigabit-Ports **403** verbunden, der SW2 ist mit dem SW3 unter Verwendung des Gigabit-Ports **403** und des Gigabit-Ports **405** verbunden, der SW3 ist mit dem SW4 unter Verwendung des Gigabit-Ports **405** und des Gigabit-Ports **407** verbunden, und der SW4 ist mit dem SW1 unter Verwendung des Gigabit-Ports **407** und des Gigabit-Ports **401** verbunden.

[0066] Aber bei der Verwendung dieser Art von Schleife tritt ein Problem dahingehend auf, dass ein Paket endlos durch die gestapelten Switches Schleifen bzw. Runden drehen wird, es sei denn, dass irgendeine Art von Mechanismus implementiert wird, um dieses endlose Schleifendrehen zu verhindern. Eine Lösung dieses Problems liegt darin, einfach die Quell-ID den Daten hinzuzufügen, die durch die Switches übertragen werden. In diesem Beispiel könnte, wenn das Paket in dem SW1 empfangen wird, die Quell-ID für den SW1 den Daten hinzugefügt werden, die von dem SW1 zu dem SW2 zu dem SW3 zu dem SW4 und zurück zu dem SW1 transferiert werden. In jedem Switch würde eine Überprüfung durchgeführt werden, um festzustellen, ob dieser Switch der Ursprungs-Switch ist.

[0067] Wenn in diesem Beispiel der SW1 die Daten empfängt, die durch die Switches übertragen werden, wird die Quell-ID für den SW1 den Daten hinzugefügt, die zu dem SW2 übertragen werden. Der SW2 wird die Quell-ID, die zu dem SW2 übertragen wird, mit den Daten prüfen und wird sehen, dass die Quell-ID der SW1 ist. Da die Quell-ID nicht der SW2 ist, wird der SW2 fortfahren, die Daten zu verarbeiten und die Daten und die Quell-ID zu dem SW3 senden. Der SW3 wird die Quell-ID prüfen und sehen, dass die Quell-ID der SW1 ist, und deshalb wird er damit fortfahren, die Daten zu verarbeiten und die Daten und die Quell-ID zu dem SW4 senden.

[0068] Der SW4 wird die Quell-ID überprüfen und sehen, dass die Quell-ID der SW1 ist, und deshalb wird er die Daten verarbeiten und die Daten und die Quell-ID zu dem SW1 senden. Der SW1 wird die Quell-ID überprüfen und wird sehen, dass dies der SW1 ist. Da die Quell-ID die gleiche ist wie die Switch-ID, wird die Verarbeitung gestoppt, wodurch ein unnötiges Schleifendrehen der Daten verhindert wird.

[0069] Der Nachteil des oben beschriebenen Prozesses liegt darin, dass durch das Hinzufügen der Quell-ID zu den Daten die Bandbreite, die dazu benutzt werden kann, die Daten zu übertragen, reduziert wird, da ein Teil der Bandbreite dazu verwendet werden muss, die Quell-ID zu übertragen. Wenn zum Beispiel ein Port in der Lage wäre 64 Bits an Daten zu übertragen, aber 8 Bits verwenden müsste, um die Quell-ID anzuzeigen, wäre die tatsächlich zur Verfügung stehende Bandbreite zur Übertragung der Daten nur 56 Bits (64 – 8). Um die vollen 64 Bits an Bandbreite zu benutzen, kapselt die vorliegende Erfindung deshalb die Quell-ID in die Daten auf der Sendeseite ein und entkapselt die Quell-ID auf der Empfangsseite.

[0070] Es sei zum Beispiel angenommen, dass ein 64 Bit Datenpaket mit einer 8 Bit System-ID übertragen werden soll. Um die Daten zu übertragen, muss ein Port verwendet werden, der in der Lage ist, eine Bandbreite von 72 Bits zu handhaben (64 Bits für die Daten und 8 Bits für die System-ID), ansonsten müssten die Daten und die System ID von dem Port getrennt übertragen werden.

[0071] Aber wenn die System-ID in das Datenpaket eingekapselt wäre, dann würde nur eine Bandbreite von 64 Bits benötigt. Wenn zum Beispiel ein Datenpaket 0x 60_60_60_60 (d.h., 0x gibt an, dass eine hexadezimale Zahl folgen wird) mit einer System-ID von 0x 05 übertragen werden soll, dann könnte die System-ID in das Paket eingekapselt werden, indem eine Exklusiv-ODER-Funktion bei der System-ID und dem Datenpaket durchgeführt werden würde. Wenn das Datenpaket 0x 60_60_60_60 mit der System-ID 0x 05 einer Exklusiv-ODER-Funktion unterzogen werden würde, würde dies zu einem eingekapselten Paket 0x 60_60_60_65 führen. Somit wäre die System-ID in dem Paket eingekapselt und würde nicht die Länge des Pakets ändern.

Aber wenn die System-ID zu dem Paket hinzugefügt werden würde, dann wäre das Paket wie folgt 0x 60_60_60_60_05, was zusätzlich ein Extra von 8 Bits zu der Länge des Pakets hinzufügen würde. Deshalb ändert sich durch das Einkapseln der Information in das Paket die Länge des Pakets nicht und die volle Bandbreite eines Ports kann für die Datenübertragung verwendet werden.

[0072] [Fig. 5](#) ist ein Beispiel einer Datenstruktur, die durch die Switches SW1, SW2, SW3 und SW4 gesendet wird. Die Datenstruktur weist ein Zieladressenfeld (DA; destination address), ein Quelladressenfeld (SA; source address), ein Nutzdatenfeld (Payload) und ein Rahmenprüfsequenzfeld (FCS; frame check sequence) auf. Die FCS wird in diesem Beispiel durch die Durchführung einer zyklischen Redundanzprüfungs-(CRC)-Funktion bei der DA, der SA und den Nutzdaten der Datenstruktur ermittelt, bevor die Daten von dem SW1 zu dem SW2 übertragen werden. Wenn die Daten von dem SW2 empfangen werden, wird die gleiche CRC-Funktion bei der DA, SA und den Nutzdaten der Datenstruktur durchgeführt, um eine neu berechnete FCS zu bestimmen. Wenn die FCS gleich dem neu berechneten FCS-Wert ist, liegen keine Fehler in der Datenstruktur vor. Aber wenn der neue FCS-Wert nicht gleich dem FCS-Wert ist, gibt es Fehler in der Datenstruktur, und die Datenstruktur muss erneut gesendet werden.

[0073] Der IEEE 802.3 Standard definiert ein CRC-Polynom, um die FCS für den 802.3 MAC Frame zu berechnen. Das FCS-Feld, das in dem MAC Frame definiert ist, kann dazu verwendet werden, um mit anderen MAC-Frame-Feldern zu kooperieren, um eine eindeutige Rahmenprüfnummer M abzuleiten, indem das CRC-Polynom, das mit dem 802.3 Standard konform ist, verwendet wird, um das Paket zu qualifizieren. Bei allen guten CRC-Paketen sollte ihr individueller CRC-Wert die gleiche Rahmenprüfnummer M sein. Die Rahmenprüfnummer M kann bei unterschiedlichen Implementierungen variieren. Aber innerhalb eines gegebenen Systems sind die CRC und die Rahmenprüfnummer M gleich.

[0074] In einem Beispiel der vorliegenden Erfindung, wie es in [Fig. 6](#) dargestellt ist, wird die Dateneinkapselung/Datenentkapselung bei dem IEEE 802.3 Ethernet Paket durchgeführt, ohne dass die Paketlänge geändert wird, wodurch die Datenstromleistung aufrechterhalten wird. In diesem Beispiel wird die Rahmenprüfsequenz (FCS) des 802.3 Pakets mit der Quell-ID auf der Sendeseite eingekapselt, und die FCS des 802.3 Pakets wird auf der Empfangsseite entkapselt, um die eingekapselte Quell-ID zu identifizieren.

[0075] Im Schritt **600** empfängt ein erster Switch SW1 ein Datenpaket zur Übertragung. In einem Beispiel der Erfindung wird ein CRC-Polynom verwendet, um das FCS-Feld des Datenpakets zu berechnen. In diesem Beispiel wird das CRC-Polynom unter den IEEE Standards an das Datenpaket angelegt, um einen CRC-Wert abzuleiten. In diesem Beispiel wird der CRC-Wert unter den IEEE Standards als 0x C7_04_DD_7B (d.h., M = 0x C7_04_DD_7B) definiert. Deshalb ist die Rahmenprüfnummer 0x C7_04_DD_7B,

[0076] Im Schritt **610** wird die Information in das FCS-Feld des Datenpakets eingekapselt. In diesem Beispiel wird die Switch-ID in das FCS-Feld eingekapselt, um ein kontinuierliches Schleifendrehen des Pakets in einer Stapelumgebung zu verhindern.

[0077] Im Schritt **620** wird das Datenpaket, das in diesem Beispiel mit der Ursprungs-Switch-ID in dem FCS-Feld des Datenpaketes eingekapselt ist, zu einem zweiten Switch SW2 übertragen.

[0078] Im Schritt **630** empfängt der zweite Switch SW2 das Datenpaket. In diesem Beispiel wird der zweite Switch SW2 eine CRC bei dem Datenpaket durchführen. Wenn keine Information wie etwa die Ursprungs-System-ID in dem Datenpaket eingekapselt worden wäre, würde ein Wert von 0x C7_04_DD_7B zurückgesendet, wenn keine Fehler bei der Übertragung des Datenpakets vorliegen. Aber da eine Information in dem Datenpaket eingekapselt ist, muss die Information aus dem Datenpaket entkapselt werden, um zu bestimmen, ob das Datenpaket korrekt übertragen worden ist.

[0079] Im Schritt **640** werden die Daten aus dem Paket entkapselt. In diesem Beispiel wird eine Tabelle, wie sie in [Fig. 7](#) dargestellt ist, dazu verwendet, die Information in dem Datenpaket zu entkapseln.

[0080] [Fig. 7](#) ist eine Tabelle, die aus drei Spalten aufgebaut ist. Die erste Spalte wird mit Syndrom bezeichnet und stellt alle möglichen Werte dar, die sich aus einer CRC-Berechnung bei dem gesamten Datenpaket ergeben. Die zweite Spalte wird Daten genannt und stellt die Informationen dar, die mit dem Datenpaket übertragen werden sollen. Die dritte Spalte ist mit Distanzvektor bezeichnet und stellt die tatsächlichen Daten dar, die in das FCS-Feld des Datenpakets eingekapselt sind.

[0081] Wenn der Ursprungs-Switch in dem vorliegenden Beispiel der Switch 6 wäre, würden die Daten, die

mit dem Datenpaket übertragen werden sollen, 6 sein. Aus der Tabelle, die in [Fig. 7](#) dargestellt ist, sollte der Distanzvektor, der in dem FCS-Feld des Datenpakets eingekapselt werden soll, 0x 06_06_06_06 sein. Deshalb sollte, wenn ein CRC-Polynom an das gesamte Datenpaket angelegt wird, ein Syndrom von 0x 7C_80_26_02 zurückgesendet werden. Obwohl dieser Wert nicht der gemäß dem IEEE Standard definierte Wert von 0x_C7_04_DD_7B ist, ist der Wert von 0x 7C_80_26_02 korrekt, da das FCS-Feld des Datenpakets mit dem Distanzvektor 0x 06_06_06_06 eingekapselt ist, was anzeigt, dass der Ursprungs-Switch der Switch 6 ist.

[0082] Deshalb wird es, anstatt dass man nur einen CRC-Wert hat, eine Vielzahl von gültigen CRC-Werten geben. Wenn in diesem Beispiel 16 Switches miteinander gestapelt sind, gibt es 16 mögliche gültige CRC-Werte (Syndrome). Wenn in dem vorliegenden Beispiel der zweite Switch SW2 das Datenpaket empfängt und eine CRC bei dem gesamten Datenpaket durchführt, sollte der Wert einer der 16 möglichen Syndrome sein, die in [Fig. 7](#) aufgelistet sind, um als ein gültiger Pakettransfer ohne Fehler betrachtet zu werden. Anderenfalls wird bestimmt, dass das Paket kein gutes Paket ist, und es wird neu übertragen werden müssen.

[0083] [Fig. 8A](#) ist eine Veranschaulichung eines Beispiels einer Einkapselungsvorrichtung **800**. In diesem Beispiel weist die Einkapselungsvorrichtung **800** ein Exklusiv-ODER-Modul **802** und Codier-Codebuch **804** auf. In einem Beispiel der Erfindung werden Informationen, die in das FCS-Feld eines Datenpakets eingekapselt werden sollen, durch das Codier-Codebuch **804** eingegeben. Wenn die Informationen, die codiert werden sollen, Daten wären, wie etwa die Zahl 5, wäre der Distanzvektor 0x 05_05_05_05 (siehe [Fig. 7](#), in der ein Datenwert von 5 einen Distanzvektor von 0x 05_05_05_05 aufweist). In diesem Beispiel werden die Original-FCS und die Distanzvektoren als Eingaben zu dem Exklusiv-ODER-Modul **805** verwendet. Wenn in diesem Beispiel die Original-FCS 0x 60_60_60_60 wäre, wäre die Sende-FCS 0x 65_65_65_65. Die Information ist nun in die Sende-FCS eingekapselt.

[0084] [Fig. 8B](#) ist eine Veranschaulichung einer Entkapselungsvorrichtung **806**, die ein Decodier-Codebuch **808** und eine Abgleichschaltung (matching circuit) **810** aufweist. Die Abgleichschaltung **810** verwendet in diesem Beispiel das Decodier-Codebuch **808**, um die Sende-FCS-Eingabe und die CRC zu verarbeiten. Wenn die Sende-FCS empfangen wird, wird eine CRC ausgeführt und würde in diesem Fall einen Wert von 0x 0F_48_CD_47 zurücksenden. Aus [Fig. 7](#) ist es offensichtlich, dass die Daten oder Informationen, die übertragen werden, die 5 ist, dass der Distanzvektor 0x 05_05_05_05 ist und deshalb die Original-FCS 0x 60_60_60_60 ist (d.h., die Sende-FCS (0x 65_65_65_65) EXKLUSN-ODER der Distanzvektor (0x 05_05_05_05). Der CRC-Status würde anzeigen, dass das Datenpaket ohne irgendeinen Fehler übertragen worden ist. Wenn aber die CRC nicht den Wert von 0x 0F_48_CD_47 zurücksenden würde, würden wir wissen, dass es einen Fehler bei der Übertragung des Datenpakets gegeben hat, und der CRC-Status würde einen Fehler bei der Datenpaketübertragung anzeigen.

[0085] Durch das Einkapseln von Informationen in vorher vorhandene Datenfelder, wie dies oben beschrieben worden ist, wird die Information übertragen, ohne dass die Paketlänge verändert wird, wodurch die Datenstromleistung aufrechterhalten wird, indem die volle zur Verfügung stehende Bandbreite für die Datenübertragung benutzt wird. In der Vergangenheit wurde die Information an den Anfang oder das Ende eines Pakets angehängt, wodurch die Bandbreite für die reine Datenübertragung verringert worden ist.

[0086] Die oben diskutierte Konfiguration der Erfindung ist in einem bevorzugten Ausführungsbeispiel der Erfindung auf einem Halbleitersubstrat wie etwa Silizium mit geeigneten Halbleiterherstellungstechniken und auf der Basis eines Schaltungs-Layout verwirklicht, die den Fachleuten auf diesem Gebiet auf der Grundlage der oben diskutierten Ausführungsbeispiele offensichtlich sein sollten. Ein Fachmann auf dem Gebiet des Halbleiterdesigns und der Halbleiterherstellung wäre in der Lage, die verschiedenen Module, Schnittstellen und Tabellen, Puffer, etc. der vorliegenden Erfindung auf einem einzigen Halbleitersubstrat auf der Grundlage der oben erläuterten Architekturbeschreibung zu implementieren. Es würde auch im Rahmen der Erfindung liegen, die offenbarten Elemente der Erfindung in diskreten elektronischen Bauteilen zu implementieren, wobei die funktionalen Aspekte der Erfindung ausgenutzt würden, ohne die Vorteile durch die Verwendung eines einzigen Halbleitersubstrats zu maximieren.

Patentansprüche

1. Verfahren zum Einkapseln und Entkapseln von Quellidentifikationsinformationen in einem Datenpaket, das durch eine Vielzahl von Schaltanordnungen (**400–406**) übertragen wird, wobei das Verfahren die folgenden Schritte umfasst:

– Empfangen eines Datenpakets (**600**) in einer ersten Schaltanordnung (**400**) zur Übertragung zu einer zweiten

Schaltanordnung (402);

- Einkapseln der Quellidentifikationsinformationen in ein vorher vorhandenes Rahmenprüfsequenzfeld FSC des Datenpakets (610);
- Übertragen des Datenpakets (620), das die Quellidentifikationsinformationen aufweist, die in das vorher vorhandene Rahmenprüfsequenzfeld FSC des Datenpakets eingekapselt worden sind, zu der zweiten Schaltanordnung (402);
- Empfangen des Datenpakets (630), das die Quellidentifikationsinformationen aufweist, die in dem vorher vorhandenen Rahmenprüfsequenzfeld FSC des Datenpakets eingekapselt sind, in der zweiten Schaltanordnung (402); und
- Entkapseln der Quellidentifikationsinformationen, die in dem vorher vorhandenen Rahmenprüfsequenzfeld FSC des Datenpakets (640) eingekapselt sind.

2. Verfahren nach Anspruch 1, wobei der Schritt des Einkapselns der Quellidentifikationsinformationen in das vorher vorhandene Rahmenprüfsequenzfeld FSC des Datenpakets (610) des Weiteren den Schritt des Verwendens eines Codier-Codebuchs (804) zum Einkapseln der Quellidentifikationsinformationen in das vorher vorhandene Rahmenprüfsequenzfeld FSC des Datenpakets umfasst.

3. Verfahren nach Anspruch 1, wobei der Schritt des Einkapselns der Quellidentifikationsinformationen in das vorher vorhandene Rahmenprüfsequenzfeld FSC des Datenpakets (610) des Weiteren den Schritt des Durchführens einer Exklusiv-ODER-Funktion (802) umfasst, wobei das Datenpaket einer Exklusiv-ODER-Operation mit den Quellidentifikationsinformationen unterzogen wird.

4. Verfahren nach Anspruch 1, wobei der Schritt des Einkapselns der Quellidentifikationsinformationen in das vorher vorhandene Rahmenprüfsequenzfeld FSC des Datenpakets (610) des Weiteren den Schritt des Durchführens einer Exklusiv-ODER-Funktion (802) umfasst, wobei das vorher vorhandene Rahmenprüfsequenzfeld FSC des Datenpakets einer Exklusiv-ODER-Operation mit den Quellidentifikationsinformationen unterzogen wird.

5. Verfahren nach Anspruch 1, wobei der Schritt des Entkapselns der Quellidentifikationsinformationen, die in dem Datenpaket (640) eingekapselt sind, den Schritt des Verwendens eines Decodier-Codebuchs (808) umfasst, um die Quellidentifikationsinformationen zu entkapseln, die in dem vorher vorhandenen Rahmenprüfsequenzfeld FSC des Datenpakets eingekapselt sind.

6. Verfahren nach Anspruch 1, das außerdem den Schritt des Durchführens einer zyklischen Redundanzprüfung (CRC) bei dem Datenpaket umfasst, um zu bestimmen, ob es bei der Übertragung des Datenpakets einen Fehler gegeben hat.

7. Verfahren nach Anspruch 1, das des Weiteren den Schritt des Bestimmens umfasst, ob ein Fehler bei der Übertragung des Datenpakets aufgetreten ist.

8. Verfahren nach Anspruch 1, das des Weiteren den Schritt des Bestimmens, dass ein Fehler bei der Übertragung des Datenpakets aufgetreten ist, umfasst, wenn die Quellidentifikationsinformationen, die in dem vorher vorhandenen Rahmenprüfsequenzfeld FSC des Datenpakets eingekapselt sind, nicht bestimmt werden können.

9. Verfahren nach Anspruch 1, das des Weiteren den Schritt des Bestimmens, dass ein Fehler bei der Übertragung des Datenpakets aufgetreten ist, umfasst, wenn die Quellidentifikationsinformationen, die in dem vorher vorhandenen Rahmenprüfsequenzfeld FSC des Datenpakets eingekapselt sind, nicht in einem Decodier-Codebuch (808) identifiziert werden können.

10. System zum Einkapseln und Entkapseln von Quellidentifikationsinformationen in einem Datenpaket, das durch eine Vielzahl von Schaltanordnungen (400–406) übertragen wird, wobei das System Folgendes umfasst:

- eine erste Schaltanordnung (400) mit: einem Sender; und einem Einkapselungsmodul (800), das die Quellidentifikationsinformationen in ein vorher vorhandenes Rahmenprüfsequenzfeld FSC eines Datenpakets einkapselt; und
- eine zweite Schaltanordnung (402) mit: einem Empfänger, wobei der Sender der ersten Schaltanordnung (400) das Datenpaket, bei dem die Quellidentifikationsinformationen in dem vorher vorhandenen Rahmenprüfsequenzfeld FSC des Datenpakets eingekapselt worden sind, zu dem Empfänger der zweiten Schaltanordnung (402) überträgt, wobei der Empfänger

der zweiten Schaltanordnung (**402**) das Datenpaket empfängt, bei dem die Quellidentifikationsinformationen in dem vorher vorhandenen Rahmenprüfsequenzfeld FSC des Datenpakets eingekapselt sind;
und
ein Entkapselungsmodul (**806**), das die Quellidentifikationsinformationen, die in dem vorher vorhandenen Rahmenprüfsequenzfeld FSC des Datenpakets eingekapselt sind, entkapselt, um die Quellidentifikationsinformationen zu bestimmen, die in dem vorher vorhandenen Rahmenprüfsequenzfeld FSC des Datenpakets eingekapselt sind.

11. System nach Anspruch 10, wobei das Einkapselungsmodul (**800**) ein Codier-Codebuch (**804**) verwendet, um die Quellidentifikationsinformationen in das vorher vorhandene Rahmenprüfsequenzfeld FSC des Datenpakets einzukapseln.

12. System nach Anspruch 10, wobei das Einkapselungsmodul (**800**) die Quellidentifikationsinformationen in das vorher vorhandene Rahmenprüfsequenzfeld FSC des Datenpakets eingekapselt, indem es eine Exklusiv-ODER-Funktion (**802**) durchführt, wobei das Datenpaket einer Exklusiv-ODER-Operation mit den Quellidentifikationsinformationen unterzogen wird.

13. System nach Anspruch 10, wobei das Einkapselungsmodul (**800**) die Quellidentifikationsinformationen in das vorher vorhandene Rahmenprüfsequenzfeld FSC des Datenpakets eingekapselt, indem es eine Exklusiv-ODER-Funktion (**802**) bei der vorher vorhandenen Rahmenprüfsequenz des Datenpakets und den Quellidentifikationsinformationen durchführt, wobei das vorher vorhandene Rahmenprüfsequenzfeld FSC des Datenpakets einer Exklusiv-ODER-Operation (**802**) mit den Quellidentifikationsinformationen unterzogen wird.

14. System nach Anspruch 10, wobei das Entkapselungsmodul (**806**) die Quellidentifikationsinformationen entkapselt, die in dem vorher vorhandenen Rahmenprüfsequenzfeld FSC des Datenpakets eingekapselt sind, indem es ein Decodier-Codebuch (**808**) verwendet, um die Quellidentifikationsinformationen zu entkapseln, die in dem vorher vorhandenen Rahmenprüfsequenzfeld FSC des Datenpakets eingekapselt sind.

15. System nach Anspruch 10, wobei das Entkapselungsmodul (**806**) eine zyklische Redundanzprüfung (CRC) bei dem Datenpaket durchführt, um zu bestimmen, ob es einen Fehler bei der Übertragung des Datenpakets gegeben hat.

16. System nach Anspruch 10, wobei das Entkapselungsmodul (**806**) bestimmt, ob ein Fehler bei der Übertragung des Datenpakets aufgetreten ist.

17. System nach Anspruch 10, wobei das Entkapselungsmodul (**806**) bestimmt, dass ein Fehler in der Übertragung des Datenpakets aufgetreten ist, wenn die Quellidentifikationsinformationen, die in dem vorher vorhandenen Rahmenprüfsequenzfeld FSC des Datenpakets eingekapselt sind, nicht bestimmt werden können.

18. System nach Anspruch 10, wobei das Entkapselungsmodul (**806**) bestimmt, dass ein Fehler in der Übertragung des Datenpakets aufgetreten ist, wenn die Quellidentifikationsinformationen, die in dem vorher vorhandenen Rahmenprüfsequenzfeld FSC des Datenpakets eingekapselt sind, nicht in einem Decodier-Codebuch (**808**) identifiziert werden können.

Es folgen 10 Blatt Zeichnungen

100

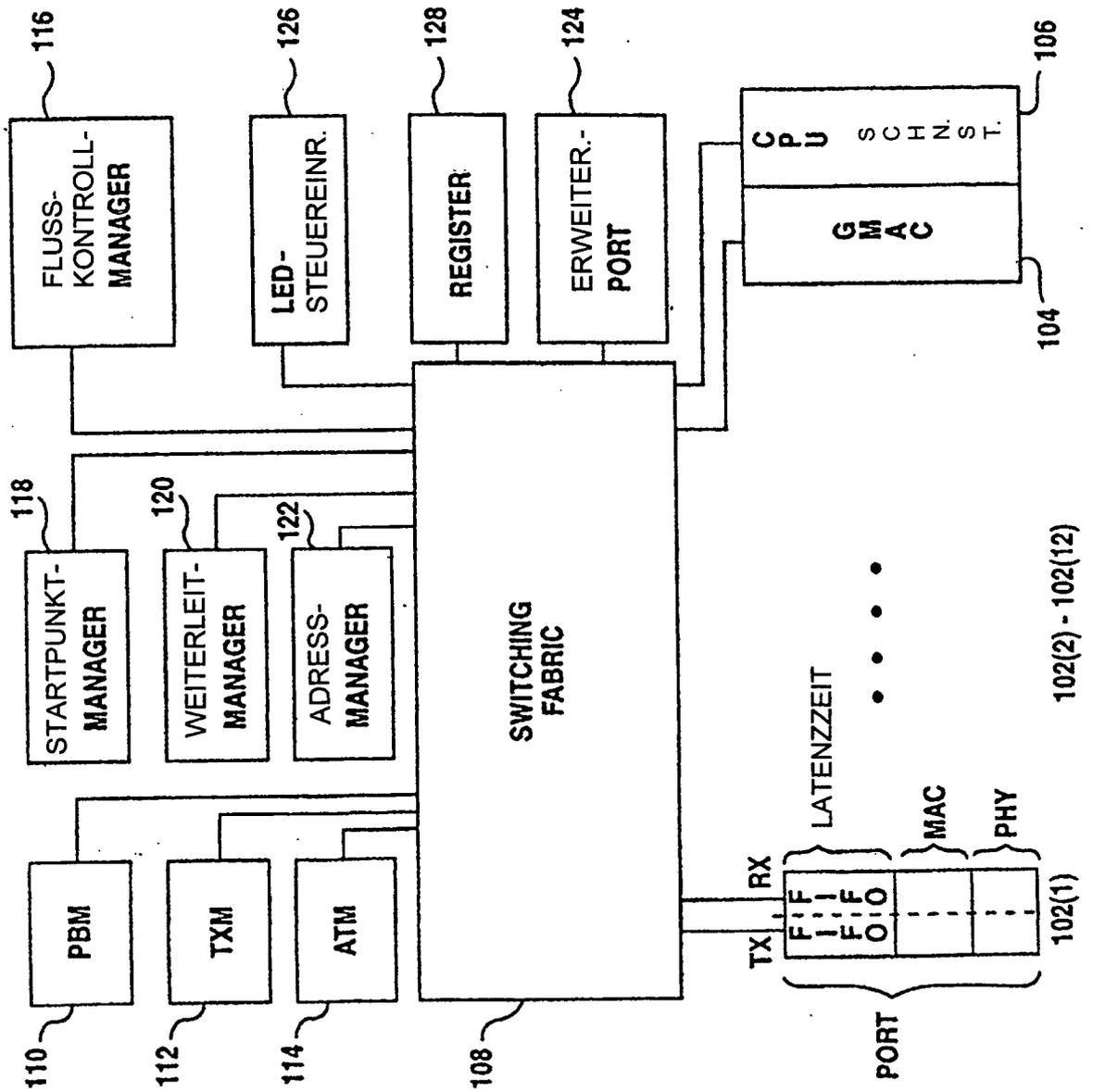


FIG.1

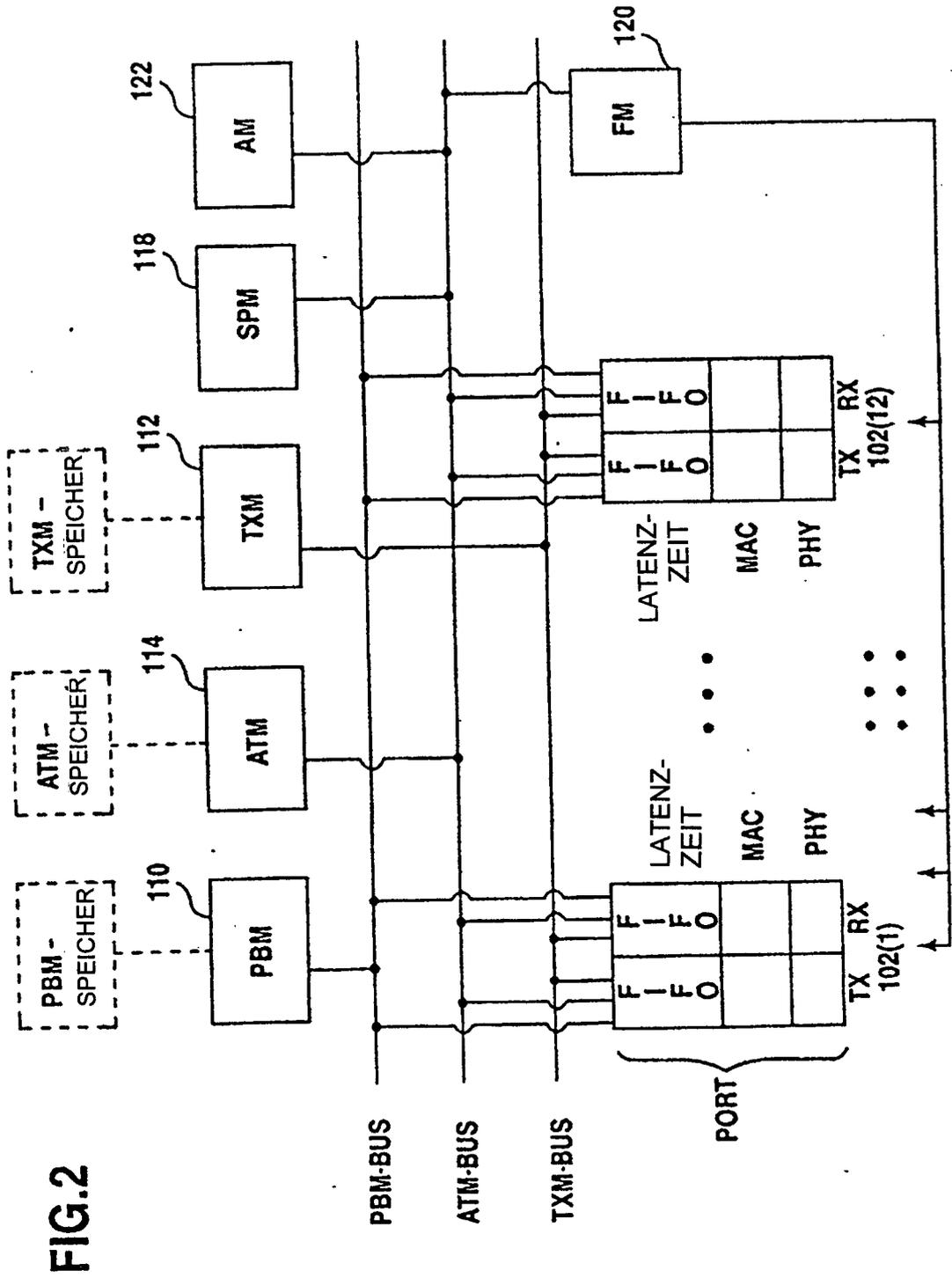


FIG.3A

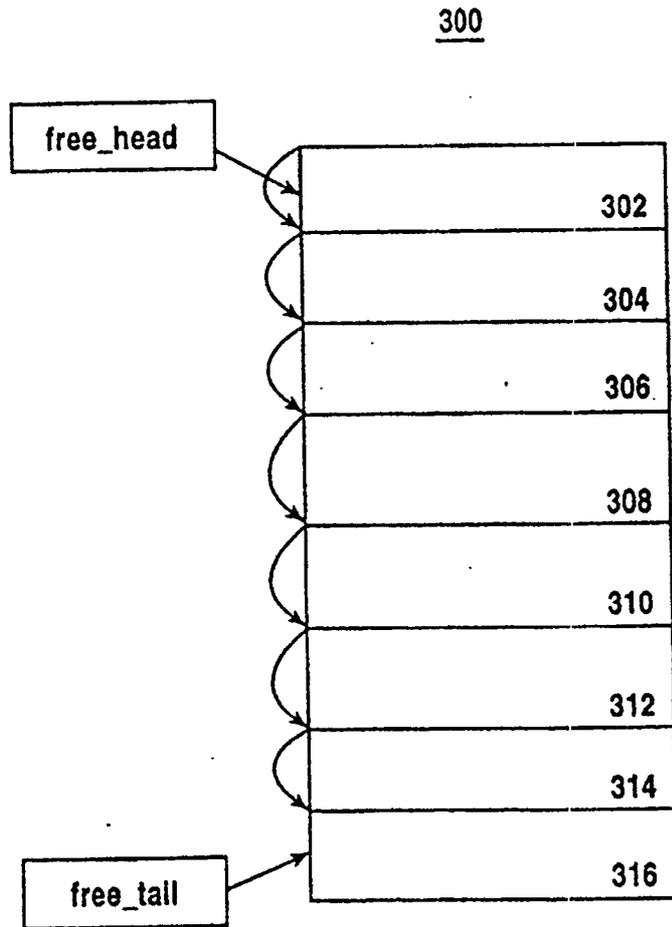


FIG.3B

300

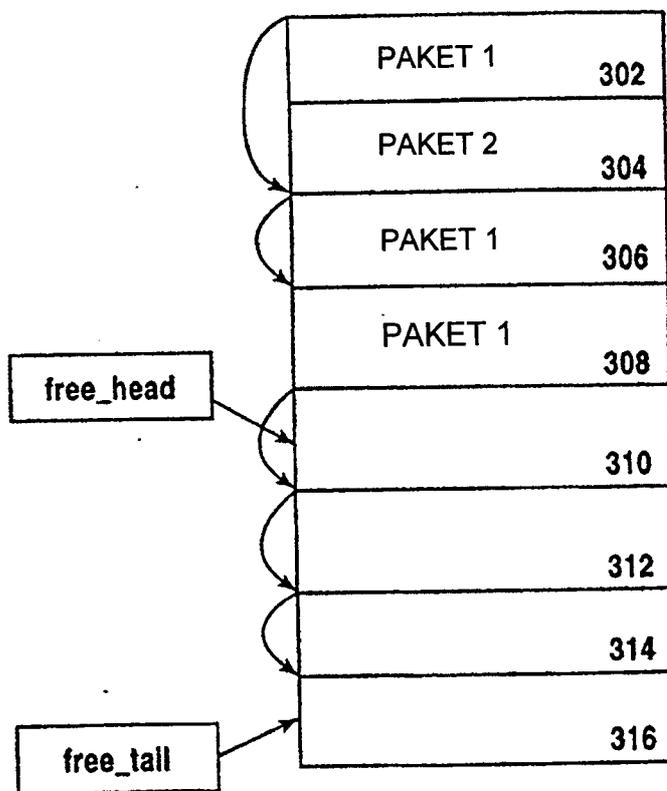


FIG.3C

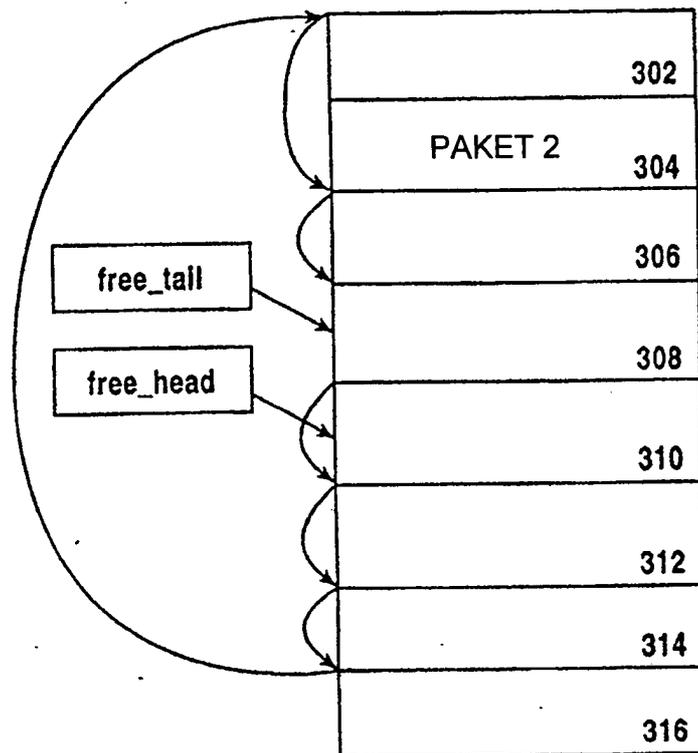


FIG.3D

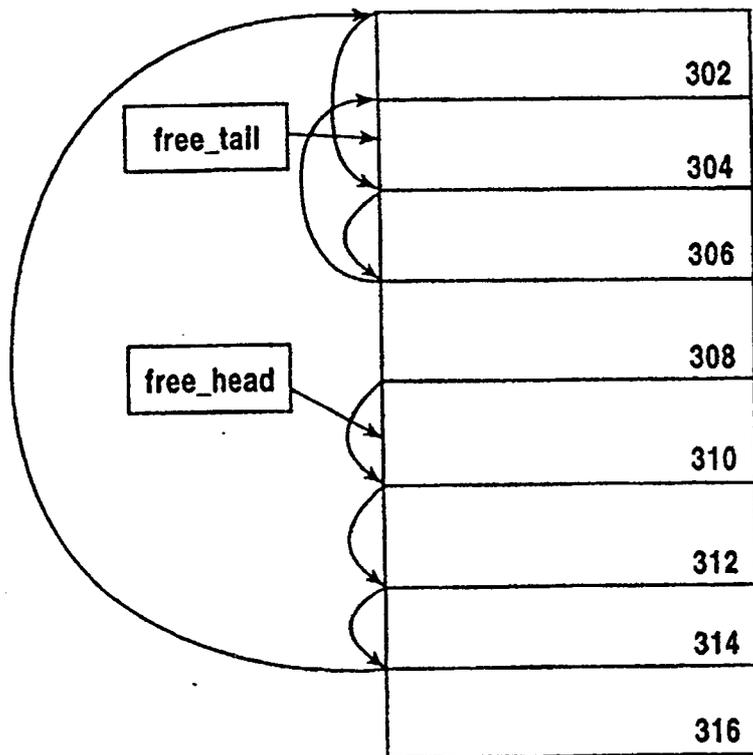


FIG.4

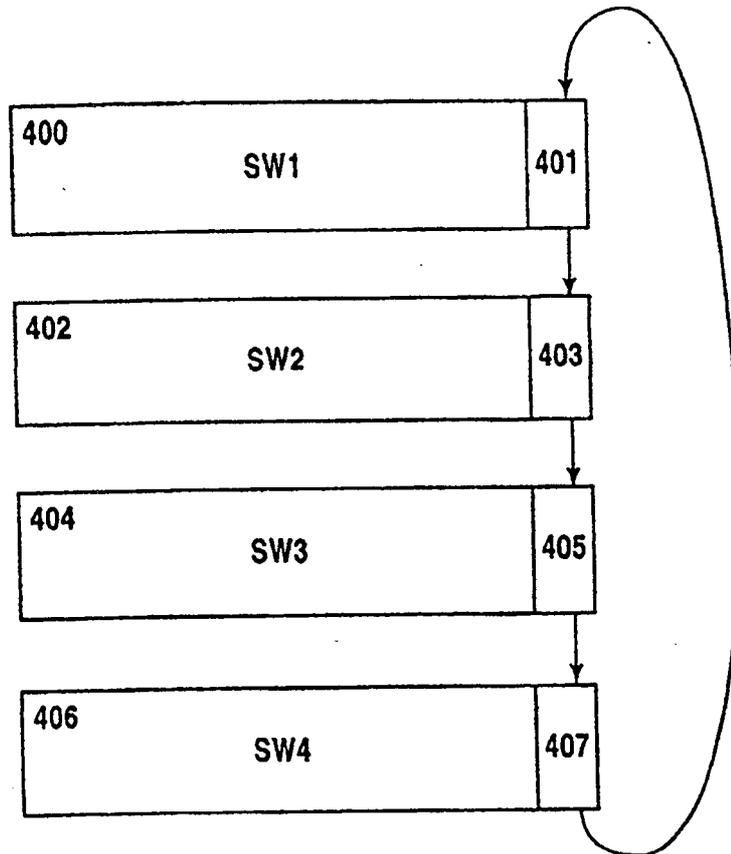


FIG.5

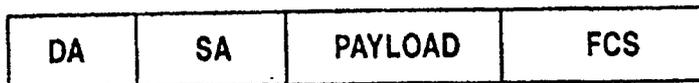


FIG.6

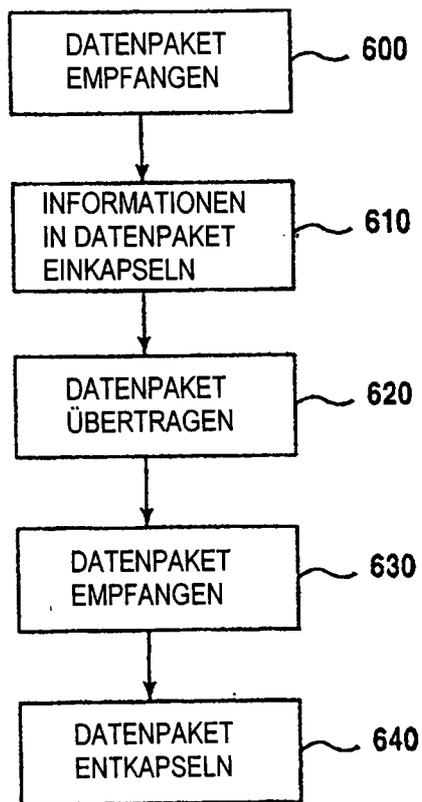


FIG.7

SYNDROM	DATEN	DISTANZVEKTOR
Ox C7_04_DD_7B	0	Ox 00_00_00_00
Ox 66_34_9B_90	1	Ox 01_01_01_01
Ox 15_FC_70_D5	2	Ox 02_02_02_02
Ox B4_CC_36_3E	3	Ox 03_03_03_03
Ox AE_78_8B_AC	4	Ox 04_04_04_04
Ox OF_48_CD_47	5	Ox 05_05_05_05
Ox 7C_80_26_02	6	Ox 06_06_06_06
Ox DD_B0_60_E9	7	Ox 07_07_07_07
Ox 71_DA_78_CB	8	Ox 08_08_08_08
Ox D0_EA_3E_20	9	Ox 09_09_09_09
Ox A3_22_D5_65	10	Ox 10_10_10_10
Ox 02_12_93_8E	11	Ox 11_11_11_11
Ox 18_A6_2E_1C	12	Ox 12_12_12_12
Ox B9_96_68_F7	13	Ox 13_13_13_13
Ox CA_5E_83_B2	14	Ox 14_14_14_14
Ox 6B_6E_C5_59	15	Ox 15_15_15_15

FIG.8A

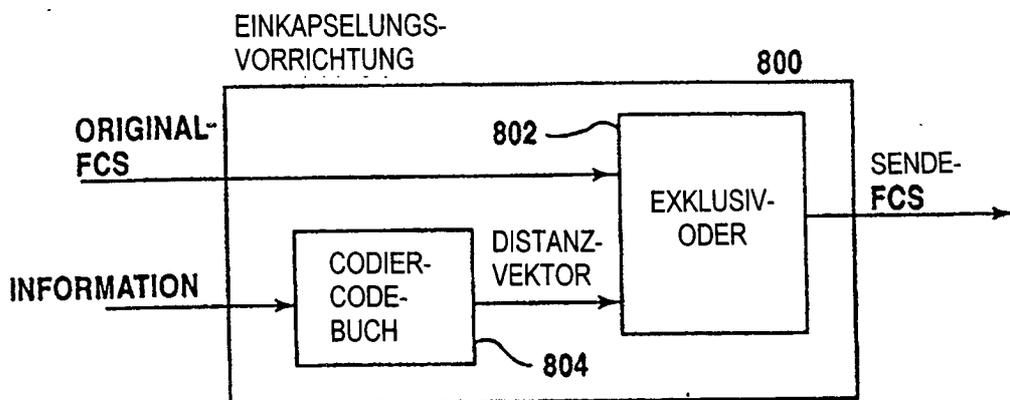


FIG.8B

