



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) DE 602 17 221 T2 2007.10.04

(12)

Übersetzung der europäischen Patentschrift

(97) EP 1 261 173 B1

(21) Deutsches Aktenzeichen: 602 17 221.7

(96) Europäisches Aktenzeichen: 02 253 533.0

(96) Europäischer Anmeldetag: 20.05.2002

(97) Erstveröffentlichung durch das EPA: 27.11.2002

(97) Veröffentlichungstag

der Patenterteilung beim EPA: 03.01.2007

(47) Veröffentlichungstag im Patentblatt: 04.10.2007

(51) Int Cl.⁸: H04L 12/44 (2006.01)

H04L 29/06 (2006.01)

(30) Unionspriorität:
861188 18.05.2001 US

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:
Broadcom Corp., Irvine, Calif., US

(72) Erfinder:
Hayter, Mark D., Menlo Park, California 94025, US;
Desai, Shailendra S., San Jose, California 95123,
US; Dobberpuhl, Daniel W., Menlo Park, California
94025, US; Chui, Kwong-Tak A., Cupertino,
California 94014, US

(74) Vertreter:
Bosch, Graf von Stosch, Jehle
Patentanwaltsgesellschaft mbH, 80639 München

(54) Bezeichnung: Ein-Chip System zur Paketverarbeitung

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 eingelebt, 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] Die vorliegende Erfindung betrifft das Gebiet der Verarbeitung und insbesondere – aber nicht ausschließlich – Paketverarbeitung, Vernetzung, und ein Ein-Chip-System.

[0002] Die Vernetzung von Computersystemen und anderen elektronischen Vorrichtungen setzt sich auf allen Ebenen des Netzes immer mehr durch. Computersysteme in Büros und manchmal sogar Haushalten werden in lokalen Netzen (LANs) vernetzt. Geografisch entfernte Computersysteme können in Weitbereichsnetzen (WANs) vernetzt werden. Das Internet kann als ein Beispiel für ein WAN angesehen werden.

[0003] Typischerweise kommunizieren Systeme auf einem Netz mit einander unter Verwendung von Paketen. Das Paket weist einen Header und Daten auf. Der Header identifiziert die Sende- und Empfangssysteme und kann andere Informationen wie etwa die Länge des Paketes, einen Pakettyp usw. aufweisen. Je nach der Topologie des Netzes kann es nötig sein, dass Pakete von dem Sendesystem zu dem Empfangssystem geroutet werden. Routing beinhaltet im Allgemeinen eine oder mehr Vorrichtungen in dem Netz, welche das Paket überprüfen, um seinen Zielort zu bestimmen und das Paket an eine andere Vorrichtung zu übertragen, bis das Paket an der Zielvorrichtung ankommt. Zusätzlich können Pakete während der Übertragung überprüft werden, um Merkmale wie etwa Quality of Service (QoS), Cache-Speichern von Informationen usw. zur Verfügung zu stellen. Daher ist eine leistungsfähige Paketverarbeitung für die Leistungsfähigkeit des Netzes insgesamt von entscheidender Wichtigkeit.

[0004] Es wird ein beispielhaftes Paketverarbeitungssystem beschrieben. Ein Prozessor, ein Cache-Speicher, ein Speichercontroller und mindestens eine Paketschnittstellenschaltung können in eine einzelne integrierte Schaltung integriert sein. Bei einer beispielhaften Implementierung kann das Design die Integrierung nutzen, um eine enge Kopplung zwischen dem Prozessor, dem Cache-Speicher, dem Speichercontroller und der Paketschnittstellenschaltung zu bilden. Beispielsweise können Interrupts eng mit dem Prozessor gekoppelt werden und dabei eine geringe Latenzzeit zu dem Interrupt sowie einen Zugriff auf Interrupt-Status-/Ursachenregister mit einer geringen Latenzzeit besitzen. Die Speicherkohärenz unter den Vorrichtungen kann aufrechterhalten werden, was die Latenzzeit, bis ein Schreiben durch eine Vorrichtung für andere Vorrichtungen sichtbar wird, reduzieren kann. Die Speicherkohärenz und der Zugriff mit geringer Latenzzeit auf Speicher durch die Paketschnittstellenschaltung kann eine enge Koppelung für Deskriptoren zur Verfügung stellen, die verwendet werden, um Paketdaten zu speichern.

[0005] Bei einer beispielhaften Ausführungsform (die in integrierten oder nicht integrierten Systemen verwendet werden kann) ist die Paketschnittstellenschaltung dazu konfiguriert, eine Zuweisung im Cache-Speicher eines Teils eines empfangenen Paketes zu veranlassen. Insbesondere kann der Teil Bytes umfassen, von denen erwartet wird, dass sie von den Prozessoren verarbeitet werden, während die übrigen Paketdaten im Speicher gespeichert werden können. Die Prozessoren können auf Cache-Speicher-Latzenzeiten für Zugriff auf die Paketdaten treffen, und eine Belastung des Cache-Speichers durch Daten, auf die von den Prozessoren nicht zugegriffen wird, kann vermieden werden.

[0006] Bei einer beispielhaften Ausführungsform (die in integrierten oder nicht integrierten Systemen verwendet werden kann) kann der Speichercontroller so konfiguriert sein, dass er Speichertransaktionen selektiv blockiert. Insbesondere kann der Speichercontroller mindestens zwei Blocksignale – eines für die Paketschnittstellenschaltung und eines für andere Vorrichtungen – implementieren. Die Blocksignale können zum Steuern der Initiierung von Speichertransaktionen verwendet werden, wenn sich die Eingabewarteschlange des Speichercontrollers dem vollen Zustand annähert. Insbesondere kann das Blocksignal für die anderen Vorrichtungen aktiviert werden, wenn die Warteschlange noch übrige Einträge hat (z.B. bei einem Füllungsgrad von 75%, obgleich der Schwellwert auf jedes Niveau programmiert werden kann), um die Latenzzeit zu reduzieren, auf die Lesen-Speichertransaktionen von der Paketschnittstellenschaltung (zum Lesen von Paketdaten, die auf der externen Paketschnittstelle übertragen werden sollen) treffen können, wenn sich die Warteschlange mit Transaktionen füllt. Bei einigen Ausführungsformen kann ein Misserfolg des kontinuierlichen Übertragens eines vollen Paketes auf der externen Paketschnittstelle dazu führen, dass die Paketübertragung fehlschlägt und eine Übertragungswiederholung durchgeführt werden muss. Durch Verringern der Latenzzeit kann die Wahrscheinlichkeit verringert werden, dass darauf folgende Cache-Speicherblöcke des Paketes bei Bedarf in der Paketschnittstellenschaltung nicht verfügbar sind.

[0007] Allgemein gesprochen wird eine beispielhafte Vorrichtung in Betracht gezogen, die folgendes aufweist: einen Prozessor, der mit einer Zwischenverbindung gekoppelt ist; einen Cache-Speicher, der mit der Zwischenverbindung gekoppelt ist; einen Speichercontroller, der mit der Zwischenverbindung gekoppelt ist; und eine Paketschnittstellenschaltung, welche Pakete von einer Paketschnittstelle empfängt und veranlasst, dass die Pakete zum Speichern auf der Zwischenverbindung übertragen werden. Der Prozessor, der Cache-Speicher, der Speichercontroller und die Paketschnittstellenschaltung sind in ein einzelnes Halbleitersubstrat integriert. Der Prozessor

kann programmiert sein, um die von der Paketschnittstelle empfangenen Pakete zu verarbeiten.

[0008] Ferner wird ein beispielhaftes Verfahren in Betracht gezogen. Ein Paket wird von einer Paketschnittstelle in einer Paketschnittstellenschaltung empfangen. Das Paket wird zum Speichern auf einer Zwischenverbindung übertragen, mit der ein Cache-Speicher, ein Speichercontroller und ein Prozessor gekoppelt sind. Der Prozessor, der Cache-Speicher, der Speichercontroller und die Paketschnittstellenschaltung sind in ein einzelnes Halbleitersubstrat integriert.

[0009] Die Schrift EP 0 498 201 beschreibt eine einschlägige Adapterarchitektur mit großer Bandbreite, die einen Paketspeicher aufweist. Pakete, die an einem Anschluss eintreffen, werden in dem Paketspeicher gespeichert. Eine Paketspeicherschnittstelle speichert ein empfangenes Paket zusammen mit einer Datenheader-Steuerinformation in den Paketspeicher.

[0010] Die Schrift US 5 778 414 beschreibt eine Rahmenverarbeitungs-Engine zum Empfangen und Verarbeiten eines Datenrahmens mit einem Header und einer Nutzlast.

[0011] Es ist eine Aufgabe der vorliegenden Erfindung, eine Vorrichtung und ein Verfahren zum engen Koppeln von Interrupts mit dem Prozessor zur Verfügung zu stellen, mit einer geringen Latenzzeit zu dem Interrupt sowie einem Zugriff mit geringer Latenzzeit auf Interrupt-Status-/Ursachenregister.

[0012] Diese Aufgabe wird durch eine Vorrichtung gemäß den Angaben in dem unabhängigen Patentanspruch 1, ein Trägermedium (Patentanspruch 9), ein Computerprogramm (Patentanspruch 14) und ein Computerprogramm-Trägermedium (Patentanspruch 15) gelöst.

[0013] Vorteilhafte Ausführungsformen der Erfindung sind in den Unteransprüchen definiert.

[0014] Die nachfolgende ausführliche Beschreibung soll nur beispielhaft sein und bezieht sich auf die beigefügte Zeichnung, von der nun eine kurze Beschreibung gegeben wird.

[0015] [Fig. 1](#) ist ein Blockdiagramm einer Ausführungsform eines Systems.

[0016] [Fig. 2](#) ist ein Blockdiagramm einer Ausführungsform eines Paketes.

[0017] [Fig. 3](#) ist ein Diagramm, welches das Vormerknen ("targeting") von Paketheader-Daten in dem L2-Cachespeicher veranschaulicht.

[0018] [Fig. 4](#) ist ein Zeitablaufdiagramm, das eine Ausführungsform des Vormerkens von Paketheader-Daten in dem L2-Cachespeicher veranschaulicht.

[0019] [Fig. 5](#) ist ein Blockdiagramm einer Ausführungsform von Deskriptoren für eine in [Fig. 1](#) gezeigte Paketschnittstellenschaltung.

[0020] [Fig. 6](#) ist ein Ablaufdiagramm zur Veranschaulichung des Betriebs einer Ausführungsform der in [Fig. 1](#) gezeigten Paketschnittstellenschaltung während des Paketempfangs.

[0021] [Fig. 7](#) ist ein Ablaufdiagramm zur Veranschaulichung des Betriebs einer Ausführungsform eines Prozessors, der unterbrochen wurde.

[0022] [Fig. 8](#) ist ein Blockdiagramm einer Ausführungsform verschiedener Einheiten aus [Fig. 1](#), welches zwischen den Einheiten vorgesehene Blocksignale zum Blockieren der Initiierung von Transaktionen veranschaulicht.

[0023] [Fig. 9](#) ist ein Ablaufdiagramm zur Veranschaulichung des Betriebs einer Ausführungsform des in [Fig. 8](#) gezeigten Speichercontrollers zum Aktivieren der Blocksignale.

[0024] [Fig. 10](#) ist ein Blockdiagramm einer Ausführungsform eines Trägermediums.

[0025] Während die Erfindung verschiedenen Modifikationen und alternativen Formen zugänglich ist, werden konkrete Ausführungsformen davon beispielhaft in der Zeichnung gezeigt und werden vorliegend ausführlich beschrieben. Es dürfte jedoch verständlich sein, dass die Zeichnung und deren ausführliche Beschreibung nicht dazu bestimmt sind, die Erfindung auf die besondere beschriebene Form zu einschränken, sondern es ist im Gegenteil beabsichtigt, dass die Erfindung alle Modifikationen, Äquivalente und Alternativen abdeckt, die in den Schutzbereich der vorliegenden Erfindung fallen.

[0026] Unter Bezugnahme auf [Fig. 1](#) ist dort ein Blockdiagramm einer Ausführungsform eines Systems **10** gezeigt. Andere Ausführungsformen sind möglich und werden in Betracht gezogen. Bei der Ausführungsform von [Fig. 1](#) weist das System **10** Prozessoren **12A–12B**, einen L2-Cachespeicher **14**, einen Speichercontroller **16**, eine Ein/Ausgabe (I/O)-Bridge **20**, eine Gruppe von Paketschnittstellenschaltungen **22A–22C** und einen Interrupt-Mapper **28** auf. Das System **10** kann einen Bus **24** aufweisen, der die verschiedenen Komponenten des Systems **10** unter einander verbindet. Wie in [Fig. 1](#) veranschaulicht ist, sind die Prozessoren **12A–12B**, der L2-Cachespeicher **14**, der Speichercontroller **16** und die I/O-Bridge **20** jeweils mit dem Bus **24** gekoppelt.

Somit können alle Prozessoren **12A–12B**, der L2-Cachespeicher **14**, der Speichercontroller **16** und die I/O-Bridge **20** jeweils ein Agent auf dem Bus **24** für die veranschaulichte Ausführungsform sein. Der Interrupt-Controller **28** ist auch mit dem Bus **24** gekoppelt gezeigt, um mit den Prozessoren **12A–12B** zu kommunizieren (die Prozessoren **12A–12B** können z.B. Status-/Ursachenregister in dem Interrupt-Controller **28** lesen). Die I/O-Bridge **20** ist mit den Paketschnittstellenschaltungen **22A–22C** gekoppelt, die ferner mit jeweiligen Bitübertragungsschicht-Schaltungen (PHYs) **30A–30C** gekoppelt sind. Der Interrupt-Mapper **28** ist so gekoppelt, dass er Interrupts für die Prozessoren **12A–12B** zur Verfügung stellt, und ist ferner so gekoppelt, dass er Interrupt-Anforderungen von den Paketschnittstellenschaltungen **22A–22C** (die Kopplung der Paketschnittstellenschaltung **22A** ist beispielhaft in [Fig. 1](#) veranschaulicht) und von anderen Vorrichtungen in dem System **10** (nicht gezeigt) empfängt. Der L2-Cachespeicher **14** ist mit dem Speichercontroller **16** gekoppelt, der ferner mit einem Speicher **26** gekoppelt ist. Die Paketschnittstellenschaltung **22A** ist in [Fig. 1](#) in mehr Detail gezeigt, und die Paketschnittstellenschaltungen **22B–22C** können ähnlich konfiguriert sein. Insbesondere weist die Paketschnittstellenschaltung **22A** in der Ausführungsform von [Fig. 1](#) eine Tx/Rx-Steuerschaltung **32**, einen Rx-FIFO **34**, einen Tx-FIFO **36**, einen DMA-Controller **38** und ein oder mehr Deskriptor-Konfigurationsregister **40** auf, sowie ein Konfigurationsregister **42**, das einen L2-Sollwert speichert. Der DMA-Controller **38** ist mit der Tx/Rx-Steuerschaltung **32**, dem Rx-FIFO **34**, dem Tx-FIFO **36**, den Deskriptor Konfigurationsregistern **40** und dem Konfigurationsregister **42** gekoppelt. Die Tx/Rx-Steuerschaltung **32** ist ferner mit dem Tx-FIFO **36** und dem Rx-FIFO **34** gekoppelt.

[0027] Bei einer Implementierung können das System **10** (das bei der dargestellten Ausführungsform durch den gestrichelten Kasten begrenzt ist, welcher die Prozessoren **12A–12B**, den L2-Cachespeicher **14**, den Speichercontroller **16**, den Interrupt-Mapper **28**, die I/O-Bridge **20**, die Paketschnittstellenschaltungen **22A–22C** und den Bus **24** umschließt) in eine einzelne integrierte Schaltung in einer Ein-Chip-Systemkonfiguration integriert sein. Das System **10** kann so entworfen sein, dass es die Integration mittels enger Kopplung der Komponenten nutzt, um eine leistungsfähige Paketverarbeitung zur Verfügung zu stellen. Konkret gesprochen können die Prozessoren **12A–12B**, der L2-Cachespeicher **14**, der Speichercontroller **16** und die Paketschnittstellenschaltungen **22A–22C** (durch die I/O-Bridge **20**) eng mit empfangen Paketen gekoppelt werden, diese verarbeiten, und die Pakete weiter leiten (falls nötig).

[0028] Die enge Kopplung kann sich auf mehrere Weisen zeigen. Beispielsweise können die Interrupts eng gekoppelt sein. Eine I/O-Vorrichtung (z.B. die Pa-

ketschnittstellenschaltungen **22A–22C**) kann einen Interrupt anfordern, der (über den Interrupt-Mapper **28**) auf einen der Prozessoren **12A–12B** gemappt wird. Die Übertragung des Interrupt zu dem Prozessor kann schnell sein, da die Signale mit der Taktfrequenz der integrierten Schaltung übertragen werden können, welche das System **10** umfasst (im Gegensatz zu der Zwischenverbindung von separaten integrierten Schaltungen). Wenn der Prozessor **12A–12B** die Interrupt Service Routine ausführt, werden typischerweise ein oder mehr Statusregister in dem Interrupt-Mapper **28** und/oder der unterbrechenden Vorrichtung gelesen. Diese Statusregister-Lesevorgänge können mit einer relativ geringen Latenzzeit über den Bus **24** und die I/O-Bridge **20** stattfinden (z.B. im Vergleich mit einem peripheren Bus mit hoher Latenzzeit wie etwa PCI). Die Latenzzeit der Statusregister-Lesevorgänge kann bei einigen Ausführungsformen eine oder mehr Größenordnungen weniger betragen als die eines peripheren Busses wie etwa PCI.

[0029] Eine weitere Auswirkung der engen Kopplung kann die Verwendung von Hardwarebasierten Speicherkohärenz unter den integrierten Vorrichtungen sein. Der Bus **24** kann kohärente Transaktionen unterstützen, so dass im Cache gespeicherte Kopien von Daten in den Agenten (und im Falle der I/O-Bridge **20** in den mit diesem Agent gekoppelten Vorrichtungen) auf geeignete Weise gehandhabt werden, so dass ein Lesevorgang von bestimmten Daten die neueste Aktualisierung für die bestimmten Daten erhält. Die erzwungene Kohärenz kann die enge Kommunikation zwischen den Vorrichtungen verbessern, da von einer Vorrichtung durchgeföhrte Aktualisierungen schnell für die Vorrichtungen sichtbar sind, welche die aktualisierten Daten lesen.

[0030] Eine Paketverarbeitung kann in dem System **10** im Allgemeinen wie folgt stattfinden: Eine Paketschnittstellenschaltung **22A–22C** kann ein Paket von der externen Schnittstelle **44A–44C** an die Paketschnittstellenschaltung (z.B. die Schnittstelle von den PHY-Schaltungen **30A–30C**) empfangen. Die Paketschnittstellenschaltung ist mit einem oder mehr Deskriptoren versehen, welche Datenpuffer im Speicher für das Speichern von Paketen angeben, und wählt einen Deskriptor für das Paket. Die Paketschnittstellenschaltung leitet das Paket per DMA (direkter Speicherzugriff) in den Speicher durch die I/O-Bridge **20** und den Speichercontroller **16**. Wenn das Paket vollständig ist (bzw. wenn bei einigen Ausführungsformen mehrere Pakete vollständig sind), kann die Paketschnittstellenschaltung einen der Prozessoren **12A–12B** unterbrechen. Als Alternative können einer oder beide der Prozessoren **12A–12B** die Statusinformationen in den Deskriptoren überwachen. Die Paketschnittstellenschaltungen können die Deskriptoren aktualisieren, wenn ein Paket vollständig ist, und die Prozessoren **12A–12B** können die Ak-

tualisierung erfassen, da die Transaktion zum Aktualisieren des Deskriptors kohärent ist. Der Prozessor **12A–12B** verarbeitet das Paket oder die Pakete und kann, falls gewünscht, das Paket (durch eine der Paketschnittstellenschaltungen **22A–22C** oder eine andere I/O-Vorrichtung oder Schnittstelle) erneut übertragen. Das System **10** kann auch Pakete im Ansprechen auf Anwendungen erzeugen, die auf den Prozessoren **12A–12B** laufen, und diese Pakete können ebenso übertragen werden. Um ein Paket durch eine der Paketschnittstellenschaltungen zu übertragen, kann der Prozessor **12A–12B** das Paket in einen von der gewünschten Paketschnittstellenschaltung verwendeten Deskriptor kopieren, und ein Statusbit in dem Deskriptor kann geändert werden, um anzugeben, dass das Paket für die Übertragung bereit ist. Da die Änderung kohärent ist, kann die Paketschnittstellenschaltung die Änderung schnell empfangen und kann mit dem Übertragen des Pakets beginnen.

[0031] Die Tx/Rx-Steuerschaltung **32** kann eine Media Access Controller (MAC)-Schaltung gemäß der Ethernet-Spezifikation aufweisen. Insbesondere bei einer Implementierung kann die Tx/Rx-Steuerschaltung **32** den MAC-Teil der Institute for Electrical and Electronic Engineers (IEEE)-Spezifikation 802.3 für Raten von 10 Megabits, 100 Megabit/s und 1 Gigabit/s erfüllen. Bei einer Implementierung kann die Tx/Rx-Steuerschaltung **32** auch konfigurierbar sein, um als ein generischer Paketempfänger/-sender zu arbeiten. Beispielsweise kann der Anfang oder das Ende von Paketen unter Verwendung von Steuersignalen an der externen Schnittstelle **44A–44C** markiert werden, und die zwischen aufeinander folgenden Anfängen oder Enden übertragenen Daten können als ein Paket behandelt werden. Eine Implementierung schließlich kann es ermöglichen, dass die drei Paketschnittstellenschaltungen **22A–22C** als zwei 16Bit-Paketempfänger/-sender betrieben werden.

[0032] Die Schnittstelle zwischen jeder Paketschnittstellenschaltung **22A–22C** und externen Vorrichtungen (Bezugszeichen **44A–44C** in [Fig. 1](#)) können die Media Independent Interface (MII) oder die Gigabit MII (GMII)-Schnittstellen zu den PHY-Schaltungen **30A–30C** sein. Als Alternative kann die externe Schnittstelle eine generische Paketschnittstelle sein, in der entweder der Anfang oder das Ende von Paketen unter Verwendung von Steuersignalen markiert wird. Bei wieder einer anderen Alternative können die drei externen Schnittstellen als zwei breitere Paketschnittstellen betrieben werden (z.B. die vorausgehend erwähnten 16Bit-Schnittstellen). Die Paketschnittstellenschaltungen können konfigurierbar sein (z.B. während des Rücksetzens), um die Schnittstelle in einer beliebigen der erwähnten Betriebsarten zu betreiben.

[0033] Die Paketschnittstellenschaltung **22A** ist für eine Ausführungsform in [Fig. 1](#) ausführlicher veran-

schaulicht. Wie bereits erwähnt wurde, können die anderen Paketschnittstellenschaltungen **22B–22C** auf ähnliche Weise konfiguriert sein. Die Tx/Rx-Steuerschaltung **32** kann die Schaltungen zum Implementieren einer Ethernet MAC-Operation aufweisen und kann ferner Schaltungen zum Implementieren der generischen Paketschnittstellen aufweisen, wie vorausgehend beschrieben wurde. Der Rx-FIFO **34** ist ein first-in, first-out Puffer zum Empfangen von Paketen von der Schnittstelle **44A** zum Speichern im Speicher. Der Tx-FIFO **36** ist auf ähnliche Weise ein FIFO zum Empfangen von Paketen aus dem Speicher für die Übertragung auf der Schnittstelle **44A**. Im Allgemeinen kann jeder der FIFOs **34** und **36** eine Mehrzahl von Einträgen zum Speichern von Paketdaten aufweisen (die einen Header, die Datennutzlast und einen Trailer umfassen können). Die FIFOs sind möglicherweise nicht groß genug, um ein ganzes Paket auf einmal zu speichern, können aber im Allgemeinen verwendet werden, um das System **10** von den Leitungsralten der externen Schnittstelle **44A** von dem Bus **24** abzupuffern. Während Paketdaten an der Schnittstelle **44A** empfangen werden, speichert die Tx/Rx-Steuerschaltung **32** die Daten im Allgemeinen in dem Rx-FIFO **34**. Zusätzlich liest die Tx/Rx-Steuerschaltung **32** Daten aus dem Tx-FIFO **36** für die Übertragung an der Schnittstelle **44A** aus.

[0034] Der DMA-Controller **38** ist dazu vorgesehen, die Paketdaten zu und von dem Speicher (und/oder dem L2-Cachespeicher **14**) zu übertragen. Die Deskriptor-Konfigurationsregister **40** lokalisieren die Deskriptoren, die der Paketschnittstellenschaltung **22A** zugeordnet sind. Jeder Deskriptor weist einen Zeiger auf mindestens einen Datenpuffer im Speicher auf (einen reservierten Bereich im Speicher, der dem Deskriptor durch Software zugewiesen ist, welche auf den Prozessoren **12A–12B** läuft). Der Deskriptor kann auch Statusinformationen aufweisen, die dem Datenpuffer entsprechen. Der DMA-Controller leitet Daten per DMA zu dem Tx-FIFO **36** (für die Übertragung an der Schnittstelle **44A**) von einem Datenpuffer, der ein zu übertragendes Paket speichert, und leitet Daten per DMA von dem Rx-FIFO **34** zu einem Datenpuffer im Ansprechen auf das Empfangen von Paketen. Der DMA weist eine oder mehr Transaktionen auf dem Bus **24** auf. Der Bus **24** kann so konfiguriert sein, dass er bis zu einen Cache-Speicherblock Daten pro Transaktion überträgt, so dass die Anzahl von Transaktionen pro Paket von der Anzahl der Cache-Speicherblöcke in dem Paket abhängt. Im Allgemeinen kann der DMA-Controller **38** es ermöglichen, dass ein Cache-Speicherblock von Paketdaten sich in dem Rx-FIFO **34** ansammelt, und kann dann eine Transaktion auf dem Bus **24** initiieren, um den Cache-Speicherblock zu übertragen. Wenn das letzte Byte eines Paketes empfangen wurde und ein unvollständiger Cache-Speicherblock übrig bleibt, kann der DMA-Controller **38** den unvollständigen Cache-Speicherblock übertragen (z.B. als einen weni-

ger-als-Cache-Speicherblock übertragen oder zuerst durch Lesen des Blocks, Modifizieren des Blocks mit den empfangenen Daten, und Schreiben des Cache-Speicherblocks zurück in den Speicher). Auf ähnliche Weise, falls der Tx-FIFO **36** genügend freie Einträge aufweist, um einen Cache-Speicherblock eines in der Übertragung befindlichen Paketes zu speichern, kann der DMA-Controller **38** eine Transaktion zum Lesen des nächsten Cache-Speicherblocks des Paketes und Speichern des Blocks in dem Tx-FIFO **36** initiieren.

[0035] Für empfangene Pakete kann die Tx/Rx-Steuerschaltung **32** so konfiguriert sein, dass sie den Anfang des Paketes in dem Rx-FIFO **34** anzeigen. Bei einer Implementierung kann der Rx-FIFO **34** beispielsweise Statusinformationen für jeden Eintrag aufweisen. Die Statusinformationen können anzeigen, ob der Eintrag der Anfang eines Paketes ist. Als Alternative kann die Rx/Tx Steuerschaltung **32** direkt mit dem DMA-Controller **38** kommunizieren, um anzusehen, welche Einträge der Anfang eines Paketes sind.

[0036] Der DMA-Controller **38** kann bei einer Ausführungsform so konfiguriert sein, dass er einen Teil eines empfangenen Paketes zum Speichern im L2-Cachespeicher **14** vormerkt. Konkret gesprochen kann es wünschenswert sein, dass der Header eines Paketes in dem L2-Cachespeicher **14** gespeichert wird, während die Datennutzlast im Speicher gespeichert werden kann. Der Prozessor **12A–12B**, der das Paket verarbeitet, kann typischerweise an den Header-Informationen operieren, und kann nicht an der Datennutzlast operieren. Falls der Header in dem L2-Cachespeicher **14** gespeichert ist, kann die Latenzzeit reduziert sein, die der Prozessor **12A–12B** beim Lesen des Headers erfährt. Während viele Paketverarbeitungsfunktionen nur Zugriff auf den Header erfordern können, wird es üblicher, beim Verarbeiten eines Paketes einen Teil der Datennutzlast ebenfalls zu verarbeiten (z.B. um komplexere Merkmale wie etwa Web-Caching zu versuchen). Daher kann es wünschenswert sein, auch einen Teil der Datennutzlast eines Paketes im L2-Cachespeicher **14** zu speichern. Es kann jedoch wünschenswert sein, Teile des Paketes zu speichern, an denen nicht von den Prozessoren **12A–12B** im Speicher operiert wird, um zu verhindern, dass der L2-Cachespeicher **14** mit Daten belastet wird, auf die von den Prozessoren **12A–12B** nicht zugegriffen wird. Somit umfasst die Paketschnittstellenschaltung **22A** das Konfigurationsregister **42**, das mit einer Größe des Teils am Anfang eines Paketes programmiert wird, das im L2-Cachespeicher **14** gespeichert werden soll. Die übrigen Bytes können im Speicher gespeichert werden. Zusätzliche Details bezüglich des Speicherns eines Teils des Paketes im L2-Cachespeicher **14** werden weiter unten angegeben. Es wird angemerkt, dass zwar die Größe des Teils des Paketes, der im L2-Ca-

chespeicher **14** gespeichert werden soll, bei der dargestellten Ausführungsform in das Konfigurationsregister **42** programmiert werden kann, dass die Größe aber nach Wunsch bei anderen Ausführungsformen in der Hardware festgelegt sein kann. Die Größe kann als eine Anzahl von Bytes (die der DMA-Controller **38** in eine Anzahl von Cache-Speicherblöcken konvertieren kann, welche die Bytes umfassen), eine Anzahl von Cache-Speicherblöcken usw. programmiert werden.

[0037] Wie bereits erwähnt wurde, kann es sein, dass der Tx-FIFO **36** nicht groß genug ist, um ein gesamtes Paket für die Übertragung zu speichern. Um ein Paket zu übertragen, kann der DMA-Controller **38** Transaktionen zum Lesen der Paketdaten aus dem Speicher durchführen und kann die Daten in dem Tx-FIFO **36** speichern. Wenn der DMA-Controller **38** den Tx-FIFO **36** füllt, kann der DMA-Controller **38** weitere Transaktionen verzögern, bis die Tx/Rx-Steuerschaltung **32** genügend Daten aus dem Tx-FIFO **36** heraus transferiert, um Platz für einen anderen Cache-Speicherblock der Paketdaten zu schaffen. Sobald auf dem Ethernet ein Paket mit der Übertragung beginnt (und falls es keine Paketkollisionen gibt), wird das Paket kontinuierlich übertragen. Falls das Paket nicht kontinuierlich übertragen werden kann, wird es vom Empfänger fallen gelassen und muss erneut übertragen werden. Somit kann es wünschenswert sein, zu verhindern, dass der Tx-FIFO **36** leer wird, bis das Paket vollständig übertragen wurde.

[0038] In einigen Fällen kann die Kombination aus Wettbewerb um den Bus **24** und der Anzahl von Speichertransaktionen, die im Speichercontroller **16** aufgereiht sind, den DMA-Controller **38** lange genug verzögern, damit der Tx-FIFO **36** leer wird. Um diesem Effekt zumindest teilweise entgegen zu wirken, kann der Speichercontroller **16** selektiv Transaktionen von anderen Agenten blockieren, wenn sich der Eingangspuffer des Speichercontrollers **16** dem vollen Zustand annähert. Beispielsweise kann das System **10** ein Transaktionsblockierverfahren anwenden, bei dem der Speichercontroller **16** mindestens zwei Blocksignale verwendet: eines für die Paketschnittstellenschaltungen **22A–22C** und ein anderes für die anderen Agenten. Die Blocksignale werden verwendet, um die Ausgabe neuer Transaktionen an den Speichercontroller zu blockieren (z.B. weil der Eingangspuffer des Speichercontrollers voll ist). Eine Quelle, die ein aktiviertes Blocksignal von dem Speichercontroller **16** empfängt, ist daran gehindert, Transaktionen an den Speichercontroller **16** zu initiieren. Die Quelle kann eine Arbitration für den Bus unterbinden, wenn die Quelle nur Speichertransaktionen durchzuführen hat.

[0039] Das Blocksignal an die anderen Agenten kann aktiviert werden, bevor der Eingangspuffer tatsächlich voll ist (z.B. wenn der Puffer zu ca. 75% voll

ist), wodurch die anderen Agenten abgehalten werden und es den Paketschnittstellenschaltungen **22A–22C** ermöglicht wird, die Durchführung von Transaktionen an den Speichercontroller **16** fortzusetzen, wenn sich der Speichercontroller **16** mit Transaktionen füllt, die abgearbeitet werden müssen. Das Blocksignal für die Paketschnittstellenschaltungen **22A–22C** wiederum kann aktiviert werden, wenn der Eingangspuffer annähernd voll ist. Die Verwendung von separaten Blocksignalen kann den Effekt haben, dass die maximale Latenzzeit verringert wird, welche die Paketschnittstellenschaltungen **22A–22C** erfahren, wenn der Speichercontroller **16** große Anzahlen von Anforderungen empfängt. Da ferner die anderen Agenten eine Arbitration unterbinden können, wenn sie nur Speichertransaktionen durchzuführen haben, kann die Verwendung der Blocksignale den Effekt haben, dass es den Paketschnittstellenschaltungen **22A–22C** ermöglicht wird, den Arbitrationsmechanismus zu umgehen und während solcher Zeiten mehr Busbandbreite auf dem Bus **24** zu erhalten, als der Fall wäre, wenn es allen anderen Agenten erlaubt wäre zu arbitrieren. Beispielsweise wenn der Arbitrationsmechanismus ein Round-Robin-Prioritätsverfahren wäre (bei dem die Arbitrationspriorität eines Agenten zu der niedrigsten Priorität geändert wird, wenn der Agent die Arbitration für den Bus **24** gewinnt, und der Agent, der die höchste Priorität anfordert, die Arbitration gewinnt), kann den Paketschnittstellenschaltungen **22A–22C** der Bus einmal gewährt werden und dann nicht mehr wieder gewährt werden, bis der Bus jedem anderen arbitrierenden Agenten gewährt worden ist. Diese zusätzliche Latenzzeit könnte den Tx-FIFO **36** weiterhin verlassen, sich vor dem Abschluss des Transfer eines Paketes zu entleeren. Da alle anderen arbitrierenden Agenten (wenn sie nur Speichertransaktionen durchzuführen haben) eine Arbitration im Ansprechen auf das Blocksignal unterbinden, das sie von dem Speichercontroller empfangen, umgehen die Paketschnittstellenschaltungen **22A–22C** effektiv die Arbitration und können die Arbitration häufiger gewinnen.

[0040] Bei einer Ausführungsform kann der DMA-Controller einen oder mehr DMA-Sendekanäle und einen oder mehr DMA-Empfangskanäle aufweisen. Die Größe des Teils des Paketes, das im L2-Cachespeicher **14** gespeichert werden soll, kann auf einer Kanal-um-Kanal-Basis programmierbar sein. Falls gewünscht, kann jeder Kanal ferner mit separaten Gruppen von Deskriptoren über die Deskriptor-Konfigurationsregistern **40** versehen werden.

[0041] Es wird angemerkt, dass zwar Ethernet MACs bei der vorliegenden Ausführungsform verwendet werden, dass aber andere Ausführungsformen jeglichen Typ von Paketschnittstelle unterstützen können. Beispielsweise unterstützen die oben erwähnten Ausführungsformen bei einigen Betriebs-

arten eine generische Paketschnittstelle. Andere Paketschnittstellen können X.25, Frame Relay, Asynchronous Transfer Mode (ATM) usw. umfassen. So mit werden Ausführungsformen des Systems **10** erwogen, die jeglichen Typ von Paketschnittstellenschaltung umfassen. Eine Paketschnittstellenschaltung ist eine Schaltung, die dazu konfiguriert ist, Pakete von einer Paketschnittstelle zu empfangen und die Pakete zum Verarbeiten zu speichern. Bei verschiedenen Implementierungen können Paketschnittstellen Pakete auf der Grundlage von vorgegebenen Filterkriterien (z.B. Adressen) filtern, Overhead aus den Paketdaten entfernen, der dem Pegel der Paketschnittstellenschaltung entspricht (z.B. den MAC-Overhead), das Paket auf Fehler überprüfen (z.B. CRC-Fehler, Größenfehler usw.).

[0042] Der Interrupt-Mapper **28** kann jeglichen Mappingmechanismus anwenden. Bei einer Ausführungsform kann der Interrupt-Mapper **28** ein Maskierbit und ein Mapregister für jede Interrupt-Anforderung und für jeden Prozessor **12A–12B** aufweisen. Das Maskierbit zeigt an, ob der Interrupt für den bestimmten Prozessor **12A** oder **12B**, maskiert ist und das Mapregister zeigt an, auf welche der Interrupt-Leitungen auf dem bestimmten Prozessor **12A** oder **12B** die Interrupt-Anforderung gemappt ist. Falls eine Ausführungsform der Prozessoren **12A–12B** nur eine Interrupt-Leitung aufweist, können die Mapregister gelöscht werden. Im Allgemeinen, wenn eine Interrupt-Anforderung empfangen wird und nicht maskiert ist, aktiviert der Interrupt-Mapper **28** das Interrupt-Signal gemäß den Angaben im Mapregister auf den Prozessor **12A** oder **12B**.

[0043] Die Prozessoren **12A–12B** können für jede Befehlssatzarchitektur entworfen sein und können Programme ausführen, die für diese Befehlssatzarchitektur geschrieben sind. Veranschaulichende Befehlssatzarchitekturen können die MIPS-Befehlssatzarchitektur (einschließlich der MIPS-3D und MIPS MDMX anwendungsspezifischen Erweiterungen), die von Intel Corp. entwickelten Befehlssatzarchitekturen IA-32 oder IA-64, die PowerPC-Befehlssatzarchitektur, die Alpha-Befehlssatzarchitektur, die ARM-Befehlssatzarchitektur, oder jegliche andere Befehlssatzarchitektur umfassen. Während das System **10**, das in [Fig. 1](#) gezeigt ist, zwei Prozessoren aufweist, können andere Ausführungsformen nach Wunsch einen Prozessor oder mehr als zwei Prozessoren aufweisen.

[0044] Der L2-Cachespeicher **14** ist ein Hochgeschwindigkeits-Cachespeicher. Der L2-Cachespeicher **14** wird als "L2" bezeichnet, weil die Prozessoren **12A–12B** internen Level 1 ("L1")-Cache-Speicher verwenden können. Falls in den Prozessoren **12A–12B** keine L1-Cachespeicher enthalten sind, kann der L2-Cachespeicher **14** ein L1-Cachespeicher sein. Wenn ferner mehrere Caching-Level in

den Prozessoren **12A–12B** enthalten sind, kann der L2-Cachespeicher **14** ein Cache-Speicher mit einem anderen Level als L2 sein. Der L2-Cachespeicher **14** kann jegliche Organisation verwenden, einschließlich direkt abbildende, satzassoziative und vollassoziative Organisationen. Bei einer bestimmten Implementierung kann der L2-Cachespeicher **14** ein satzassoziativer Cache-Speicher (allgemein N-fach, wobei N eine ganze Zahl ist, obgleich bei einer Implementierung ein 4-Weg-Cachespeicher verwendet werden kann) mit 32Byte-Cache-Speicherblöcken sein. Ein satzassoziativer Cache-Speicher ist ein Cache-Speicher, der in mehreren Sätzen angeordnet ist, wobei jeder Satz zwei oder mehr Einträge aufweist. Ein Teil der Adresse (der "Index") wird dazu verwendet, einen der Sätze auszuwählen (d.h. jede Codierung des Index wählt einen anderen Satz). Die Einträge in dem gewählten Satz sind verfügbar, um den Cache-Speicherblock zu speichern, auf den durch die Adresse zugegriffen wird. Jeder der Einträge in dem Satz wird als ein "Weg" des Satzes bezeichnet. Derjenige Teil der Adresse, der nach dem Entfernen des Index (und des Versatzes in dem Cache-Speicherblock) übrig bleibt, wird als "Tag" bezeichnet und wird in jedem Eintrag gespeichert, um den Cache-Speicherblock in diesem Eintrag zu identifizieren. Die gespeicherten Tags werden mit dem entsprechenden Tagteil der Adresse einer Speichertransaktion verglichen, um zu bestimmen, ob die Speichertransaktion in dem Cache-Speicher ein Treffer oder ein Fehlschlag ist, und wird dazu verwendet, den Weg zu wählen, auf welchem der Treffer erfasst wird (falls ein Treffer erfasst wird).

[0045] Der Speichercontroller **16** ist dazu konfiguriert, im Ansprechen auf Speichertransaktionen, die auf dem Bus **24** empfangen wurden, auf den Speicher **26** zuzugreifen. Der Speichercontroller **16** empfängt ein Treffersignal vom L2-Cachespeicher **14**, und falls ein Treffer im L2-Cachespeicher **14** für eine Speichertransaktion erfasst wird, reagiert der Speichercontroller **16** nicht auf diese Speichertransaktion. Andere Ausführungsformen können den L2-Cachespeicher **14** nicht aufweisen, und der Speichercontroller **16** kann auf jede Speichertransaktion reagieren. Falls ein Fehlschlag von dem L2-Cachespeicher **14** erfasst wird, oder die Speichertransaktion nicht im Cache speicherbar ist, kann der Speichercontroller **16** auf den Speicher **26** zugreifen, um die Lese- oder Schreiboperation durchzuführen. Der Speichercontroller **16** kann dazu entworfen sein, auf jeglichen von einer Vielzahl von Speichertypen zuzugreifen. Beispielsweise kann der Speichercontroller **16** für Synchronous Dynamic Random Access-Speicher (SDRAM) und insbesondere für Double Data Rate (DDR)-SDRAM entworfen sein. Als Alternative kann der Speichercontroller **16** für DRAM, Rambus DRAM (RDRAM), SRAM oder jegliche andere geeignete Speichervorrichtung entworfen sein.

[0046] Die I/O-Bridge **20** verbindet die Paketschnittstellenschaltungen **22A–22C** (und potenziell eine oder mehr andere I/O-Vorrichtungen oder I/O Schnittstellen) mit dem Bus **24**. Die I/O-Bridges **20** können dazu dienen, die elektrische Last auf dem Bus **24** zu reduzieren. Im Allgemeinen führt die I/O-Bridge **20** Transaktionen auf dem Bus **24** für die Paketschnittstellenschaltungen **22A–22C** durch und leitet Transaktionen, welche die Paketschnittstellenschaltungen **22A–22C** vormerken, von dem Bus **24** auf diese Paketschnittstellenschaltung **22A–22C** weiter. Auf ähnliche Weise kann die I/O-Bridge **20** Transaktionen zu oder von anderen I/O-Vorrichtungen oder Schnittstellen routen, mit denen sie gekoppelt ist. Bei einer Implementierung kann eine zweite I/O-Bridge (nicht gezeigt) eine Bridge zu einer PCI Schnittstelle und zu einem Lightning Data Transport (LDT) I/O-Fabric sein, das von Advanced Micro Devices, Inc. entworfen wurde. Die LDT-Schnittstelle wird auch als die HyperTransport-Schnittstelle bezeichnet, obgleich LDT vorliegend verwendet werden kann. Der PCI-Bus und das LDT-Fabric können außerhalb des Systems **10** zum Anschließen von PCI- oder LDT-peripheren Vorrichtungen vorgesehen sein. Andere I/O-Vorrichtungen/Schnittstellen (einschließlich der Paketschnittstellenschaltungen **22A–22C**) können von der I/O-Bridge **20** überbrückt werden. Die anderen I/O-Schnittstellen (nicht gezeigt) können eine oder mehr serielle Schnittstellen, Personal Computer Memory Card International Association (PCM-CIA)-Schnittstellen usw. umfassen.

[0047] Der Bus **24** kann bei einer Ausführungsform ein Split Transaction-Bus sein. Der Bus **24** kann bei einer Ausführungsform ein verteiltes Arbitrationsverfahren anwenden. Bei einer Ausführungsform kann der Bus **24** pipelined sein. Der Bus **24** kann jegliche geeignete Signalgebungstechnik anwenden. Beispielsweise kann bei einer Ausführungsform eine Differentialsignalgebung für die Hochgeschwindigkeits-Signalübermittlung verwendet werden. Andere Ausführungsformen können jegliche andere Signalgebungstechnik (z.B. TTL, CMOS, GTL, HSTL usw.) anwenden.

[0048] Es wird angemerkt, dass das System **10** (und insbesondere die Prozessoren **12A–12B**, der L2-Cachespeicher **14**, der Speichercontroller **16**, die Paketschnittstellenschaltungen **22A–22C**, die I/O-Bridge **20**, der Interrupt-Mapper **26** und der Bus **24**) bei der oben beschriebenen Ausführungsform zwar auf eine einzelne integrierte Schaltung als eine Ein-Chip-Systemkonfiguration integriert sein können, dass aber andere Ausführungsformen eine oder mehr der Vorrichtungen als separate integrierte Schaltungen implementieren können. Bei einer anderen Konfiguration kann der Speicher **26** ebenfalls integriert sein. Als Alternative können eine oder mehr der Komponenten als separate integrierte Schaltungen implementiert sein, oder nach Wunsch können

alle Komponenten separate integrierte Schaltungen sein. Jegliches Integrationsniveau kann verwendet werden.

[0049] Es wird angemerkt, dass die veranschaulichte Ausführungsform zwar einen Split Transaction-Bus mit separater Arbitration für die Adress- und Datenbusse anwendet, dass aber andere Ausführungsformen Non-Split Transaction-Busse anwenden können, die mit einer einzigen Arbitration für Adresse und Daten arbitriert werden, und/oder einen Split Transaction-Bus, bei dem der Datenbus nicht ausdrücklich arbitriert ist. Entweder ein zentrales Arbitrationsverfahren oder ein verteiltes Arbitrationsverfahren kann gemäß der Entwurfswahl verwendet werden. Falls gewünscht, kann der Bus **24** ferner nicht pipelined sein. Während bei der vorliegenden Ausführungsform ein geteilter Bus verwendet wird, kann bei anderen Ausführungsformen jegliche Art von Zwischenverbindung verwendet werden (z.B. Kreuzschienenverbindungen, Punkt-zu-Punkt-Verbindungen in einer Ring-, Stern- oder jeglichen anderen Topologie, Maschen, Cube o. dgl.). Im Allgemeinen ist eine Zwischenverbindung jede Art von Kommunikationsmedium.

[0050] Es wird angemerkt, dass [Fig. 1](#) zwar die Paketschnittstellenschaltungen **22A–22C** als mittels der I/O-Bridge **20A** mit dem Bus **24** gekoppelt veranschaulicht, dass aber andere Ausführungsformen, falls gewünscht, Paketschnittstellenschaltungen **22A–22C** aufweisen können, die direkt mit dem Bus **24** gekoppelt sind. Ferner kann die Anzahl von Paketschnittstellenschaltungen **22A–22C** je nach Wunsch von einer bis zu vielen variiert werden.

[0051] Der vorliegend verwendete Ausdruck Transaktion bezieht sich auf eine Kommunikation zwischen zwei oder mehr Agenten auf einer Zwischenverbindung (z.B. dem Bus **24**). Ein Agent, der die Transaktion initiiert, kann der anfordernde Agent oder die Quelle sein, und ein Agent, der auf die Transaktion anspricht, ist ein antwortender Agent oder ein antwortendes Ziel. Der Ausdruck "Speichertransaktion" bezieht sich auf eine Kommunikation zwischen einer Vorrichtung und dem Speichersystem. Die Speichertransaktion umfasst eine Adresse, welche ein Speicherstelle in dem Speicher identifiziert. Bei der veranschaulichten Ausführungsform wird die Speichertransaktion auf dem Bus **24** an den Speichercontroller **16** übertragen (und kann von dem L2-Cachespeicher **14** an Stelle des Speichercontrollers **16** abgearbeitet werden). Speichertransaktionen können Lese- transaktionen (einen Transfer von Daten vom Speicher zur Vorrichtung) und Schreibtransaktionen (einen Transfer von Daten von der Vorrichtung zum Speicher) umfassen. Allgemeiner noch kann eine Lese- transaktion ein Transfer von Daten von einem Ziel der Transaktion zu der Quelle (bzw. dem Initiator) der Transaktion sein, und eine Schreibtransaktion kann

ein Transfer von Daten von einer Quelle der Transaktion zu dem Ziel der Transaktion sein.

[0052] Unter Bezugnahme auf [Fig. 2](#) ist dort ein Blockdiagramm eines beispielhaften Paketes **50** gezeigt. Andere Ausführungsformen sind möglich und werden in Betracht gezogen. Bei der Ausführungsform von [Fig. 2](#) umfasst das Paket **50** einen Header **52**, eine Datennutzlast **54** und einen optionalen Trailer **56**. Jeder von dem Header **52**, der Datennutzlast **54** und dem Trailer **56** kann jegliche Anzahl von Bytes aufweisen. Der Header **52** ist in einer auseinander gezogenen Ansicht für ein Ethernet Paket gezeigt, das verwendet wird, um ein Transportsteuerungsprotokoll/Internetprotokoll (TCP/IP)-Paket zu übertragen, und umfasst einen Ethernet-Header **58**, einen IP-Header **60** und einen TCP-Header **62**.

[0053] Der Header **52** beschreibt das Paket **50**, damit der Empfänger des Paketes das Paket richtig interpretieren kann. Beispielsweise kann der Header eines oder mehr der folgenden aufweisen: Quell- und Bestimmungsadressen für das Paket, die Größe des Paketes, die Größe des Headers, und Status-/Steuerinformationen für das Paket. Die Status-/Steuerinformationen können Fehlerprüfinformationen, Prioritäts- informationen, Pakettyp usw. umfassen.

[0054] Wie in der auseinander gezogenen Ansicht veranschaulicht ist, kann der Header **52** tatsächlich mehrere Header umfassen, die verschiedenen Schichten der Networking Protocol Suite entsprechen. Im Allgemeinen können in jeder Schicht zusätzliche Informationen verwendet werden, um das Paket zu beschreiben. Die zusätzlichen Informationen verkapseln die Informationen von den höheren Schichten, so dass eine Kommunikation auf dieser Schicht das Format der Informationen von den höheren Schichten nicht zu wissen braucht.

[0055] Bei dem Beispiel von [Fig. 2](#) umfasst der Header **52** den Ethernet-Header **58** mit 14 Bytes, den IP-Header **60** mit 20 Bytes, und den TCP-Header **62** mit 20 Bytes. Daher kann der Header **52** bei diesem Beispiel 54 Bytes groß sein. Der Ethernet-Header **58** kann zwei 6Byte-Ethernet-Adressen (eine Quell- adresse und eine Bestimmungsadresse) und ein Längenfeld mit zwei Bytes aufweisen. Der IP-Header **60** kann zwei 4Byte-IP-Adressen (wiederum Quell- und Bestimmungsadresse) sowie eine Vielzahl anderer Informationen einschließlich der Länge sowohl des Headers als auch des IP-Pakets, eine IP- Versionsnummer, einen Typ von Dienstfeld, ein Identifikationsfeld, ein Fragment Offset, ein Time to Live-Feld, ein Protokollfeld und eine Header-Prüfsumme aufweisen. Optional können zusätzliche Optionsfelder vorgesehen sein (welche die Größe des IP-Headers **60** vergrößern). Der TCP-Header **62** kann Quell- und Bestimmungsanschlusnummern, eine Sequenz- nummer, welche das Paket in einer Sequenz von Pa-

keten identifiziert, eine Bestätigungsnummer, eine Headerlänge, einen Satz von Flagbits, eine Prüfsumme, einen Urgent Pointer und eine Fenstergröße aufweisen. Der TCP-Header kann nach Wunsch wiederum optional erweitert werden, um zusätzliche Optionen für ein Paket zu enthalten.

[0056] Je nach der durchzuführenden Paketverarbeitung kann der Prozessor **12A** oder **12B** an einem Teil des Headers, dem gesamten Header, oder sogar dem gesamten Header und einem Teil der Datennutzlast operieren. Wenn das Paket beispielsweise einfach auf einem Ethernet geroutet wird, kann an dem Ethernet-Header **58** operiert werden. Wenn das Paket auf einem IP-Netz geroutet wird (d.h. das System **10** ist Teil eines IP-Routers), kann sowohl an dem Ethernet-Header **58** als auch an dem IP-Header **60** operiert werden, oder nur an dem IP-Header **60** kann operiert werden, usw.

[0057] Die Datennutzlast **54** sind die mit dem Paket übertragenen Daten. Bei einigen Paketen kann die Datennutzlast **54** nicht enthalten sein. Beispielsweise sind einige TCP-Pakete Bestätigungspakete, die gesendet werden, um den Empfang eines vorherigen Paketes zu bestätigen. Ein solches Paket braucht keine Daten zu enthalten.

[0058] Der optionale Trailer **56** kann ähnlich wie der Header **52** zusätzliche Paketbeschreibungsinformationen enthalten. Beispielsweise können Ethernet-Pakete einen Cyclical Redundancy Check (CRC)-Wert in dem Trailer **56** enthalten.

[0059] Es wird angemerkt, dass das veranschaulichte Paket **50** zwar ein Ethernet-Paket ist, das ein TCP/IP-Paket trägt, dass aber andere Ethernet-Pakete andere Typen von Daten tragen können. Ferner können andere Typen von Paketen das TCP/IP-Paket verkapseln.

[0060] Es wird angemerkt, dass das Beispiel von [Fig. 3](#) zwar den gesamten Header im L2-Cachespeicher **14** und die gesamte Datennutzlast im Speicher **26** veranschaulicht, dass aber andere Beispiele einen Teil des im L2-Cachespeicher **14** gespeicherten Header aufweisen können (z.B. falls dieser Teil nicht in dem L2-Zählerwert enthalten ist, der in die Paketschnittstellenschaltungen **22A–22C** programmiert ist), oder einige der im L2-Cachespeicher **14** gespeicherten Daten (z.B. falls erwartet wird, dass die Daten verarbeitet werden, ist der Header des Paketes kleiner als andere Pakete, oder die Daten befinden sich in dem gleichen Cache-Speicherblock wie der Header).

[0061] Unter Bezugnahme auf [Fig. 3](#) ist dort ein Blockdiagramm gezeigt, das ein Beispiel für das Vormerken des Paketheaders im L2-Cachespeicher **14** veranschaulicht. In [Fig. 3](#) sind der Prozessor **12A**,

der L2-Cachespeicher **14**, der Speicher **26** und die Paketschnittstellenschaltung **22A** veranschaulicht. Es werden Pfeile verwendet, um den Fluss von Informationen zwischen den in [Fig. 3](#) gezeigten Blöcken zu veranschaulichen. Zwei Cache-Speicherblock-Speicherstellen **70** und **72** sind im L2-Cachespeicher **14** veranschaulicht, und zwei Speicherstellen (mit der Größe eines Cache-Speicherblocks) **74** und **76** sind in dem Speicher **26** veranschaulicht.

[0062] Bei dem Beispiel von [Fig. 3](#) empfängt die Paketschnittstellenschaltung **22A** ein Paket, das von dem System **10** verarbeitet werden soll. Die Paketschnittstellenschaltung **22A** ist bei diesem Beispiel dazu programmiert (unter Verwendung des Konfigurationsregisters **42**), die ersten zwei Cache-Speicherblöcke des Paketes im L2-Cachespeicher **14** (und die übrigen Blöcke des Paketes in dem Speicher **26**) zu speichern. Wenn die ersten zwei Cache-Speicherblöcke des Paketes übertragen werden, veranlasst die Paketschnittstellenschaltung **22A** daher, dass die Cache-Speicherblöcke im L2-Cachespeicher **14** (Pfeil **78**) gespeichert werden. Im Allgemeinen liefert die Paketschnittstellenschaltung **22A** Informationen in den Transaktionen, um anzusehen, welche Cache-Speicherblöcke im L2-Cachespeicher **14** gespeichert werden sollen. Ein Beispiel ist nachfolgend in [Fig. 4](#) gezeigt. Der L2-Cachespeicher **14** speichert die Blöcke (gezeigt als "Header" in [Fig. 3](#)).

[0063] Beim Übertragen der übrigen Cache-Speicherblöcke des Paketes gibt die Paketschnittstellenschaltung **22A** nicht an, dass die Cache-Speicherblöcke im L2-Cachespeicher (Pfeil **80**) gespeichert werden sollen. Der Speicher **26** speichert die übrigen Blöcke (gezeigt als "Daten" in [Fig. 3](#)).

[0064] Im Anschluss an die Übertragung des Paketes an den Speicher kann die Paketschnittstellenschaltung **22A** den Prozessor **12A** unterbrechen. Als Alternative kann die Paketschnittstellenschaltung **22A** den dem Paket entsprechenden Deskriptor aktualisieren, um anzusehen, dass das Paket vollständig ist, und der Prozessor **12A** kann den Deskriptor abfragen, um die Aktualisierung zu erfassen. In jedem dieser Fälle kann der Prozessor **12A** das Paket verarbeiten. Da derjenige Teil des Paketes, von dem erwartet wird, das der Prozessor an ihm operiert, **14** während des DMA des Paketes durch die Paketschnittstellenschaltung **22A** in den L2-Cachespeicher gespeichert wird, kann der Prozessor **12A** auf L2-Cachespeichertreffer für den Zugriff auf die Paketdaten treffen (Pfeil **82**). Die Latenzzeit der Zugriffe kann somit verkürzt werden, was zu einer insgesamt kürzeren benötigten Zeit für die Verarbeitung des Paketes führen kann.

[0065] Wie bereits erwähnt wurde, liefert die Paketschnittstellenschaltung **22A** Informationen bei den

Transaktionen zum Übertragen eines empfangenen Paketes an den Speicher, welche angeben, welche der Cache-Speicherblöcke im L2-Cachespeicher **14** gespeichert werden sollen und welche nicht. Konkret gesprochen kann ein während der Adressphasen der Transaktionen übertragenes Steuersignal (im Nachfolgenden als L2CA bezeichnet) als die Angabe verwendet werden. Andere Ausführungsformen können Befehlsinformationen oder jegliche andere Signalgebung verwenden, um die Angabe zu übertragen, oder können die Angabe während anderer Phasen der Transaktion übertragen. Während der Adressphase der ersten N Transaktionen eines Paketes (wobei N durch den Wert im Konfigurationsregister **42** bestimmt ist) kann die Paketschnittstellenschaltung **22A** das L2CA-Signal aktivieren, um den L2-Cachespeicher **14** zu veranlassen, die Daten zu speichern. Während der übrigen Transaktionen kann die Paketschnittstellenschaltung **22A** das L2CA-Signal deaktivieren.

[0066] Der L2-Cachespeicher **14** spricht auf das L2CA-Signal für eine Transaktion folgendermaßen an: Wenn der Cache-Speicherblock ein Fehlschlag im L2-Cachespeicher **14** ist und das L2CA-Signal aktiviert ist, weist der L2-Cachespeicher **14** eine Cache-Speicherblock-Speicherstelle zu, um den Cache-Speicherblock zu speichern. Im Allgemeinen kann die Zuweisung das Auswählen einer Cache-Speicherblock-Speicherstelle zum Speichern des Blocks, das Entfernen des gegenwärtig darin gespeicherten Blocks (falls vorhanden), und das Aktualisieren der Cache-Speicherblock-Speicherstelle mit dem Cache-Speicherblock umfassen. Bei einer Ausführungsform kann der Speichercontroller **16** den Speicher **26** nicht mit dem L2-Cachespeicher **14** zugewiesenen Cache-Speicherblöcken aktualisieren, obgleich der Speicher **26** bei anderen Ausführungsformen auch aktualisiert werden kann. Falls andererseits der Cache-Speicherblock ein Fehlschlag im L2-Cachespeicher **14** ist und das L2CA-Signal deaktiviert ist, kann der L2-Cachespeicher **14** keinen Speicher für den Cache-Speicherblock zuweisen. Der Cache-Speicherblock kann in diesem Fall von dem Speichercontroller **16** nur im Speicher **26** gespeichert werden. Somit kann eine Cache-Speicherbelastung bei Daten vermieden, bei denen es nicht wahrscheinlich ist, dass von dem Prozessor **12A-12B** während der Verarbeitung des Paketes auf sie zugegriffen wird. Falls der Cache-Speicherblock ein Treffer im L2-Cachespeicher **14** ist, wird bei der vorliegenden Ausführungsform die den Treffer darstellende Cache-Speicherblock-Speicherstelle mit den während der Transaktion übertragenen Daten aktualisiert (unabhängig davon, ob das L2CA-Signal aktiviert oder deaktiviert ist).

[0067] Es wird angemerkt, dass ein Signal als aktiviert definiert werden kann, wenn es auf einer ersten Spannung angetrieben wird, die für eine logische

Eins steht, oder als Alternative, wenn es auf einer zweiten Spannung angetrieben wird, die für eine logische Null steht. Das Signal kann als deaktiviert definiert sein, wenn es auf dem Pegel angetrieben wird, der dem aktivierte Pegel entgegengesetzt ist. Bei Ausführungsformen, bei denen eine Differentialsignalgebung verwendet wird, ist das Signal aktiviert, wenn ein erstes von dem Differentialpaar niedriger als das zweite ist, und ist deaktiviert, wenn das zweite von dem Differentialpaar niedriger als das erste ist.

[0068] Unter Bezugnahme auf [Fig. 4](#) ist dort ein Zeitablaufdiagramm gezeigt, das mehrere Adressphasen von Transaktionen veranschaulicht, die von der Paketschnittstellenschaltung **22A** (konkret von dem DMA-Controller **38** durch die I/O-Bridge **20**) durchgeführt werden, um ein Paket zu übertragen. Es sind mehrere Taktzyklen des Taktsignals CLK gezeigt, das dem Bus **24** entspricht. Jeder Taktzyklus ist oben in [Fig. 4](#) benannt (CLK0, CLK1 usw.). Phasen der Taktzyklen sind durch vertikale gestrichelte Linien begrenzt.

[0069] [Fig. 4](#) veranschaulicht ein Beispiel, bei dem die ersten fünf Cache-Speicherblöcke eines empfangenen Pakets an den Speicher/L2-Cachespeicher übertragen werden. Die während der Adressphasen der fünf Transaktionen übertragenen Adressen sind horizontal von dem Etikett "Addr" veranschaulicht. Der Befehl für die Transaktion ist horizontal von dem Etikett "CMD" veranschaulicht. Für jede Transaktion ist der Befehl ein Schreiben. Konkret gesprochen kann der Befehl ein Schreiben mit Devalidieren sein, wodurch veranlasst wird, dass alle im Cache gespeicherten Kopien des Cache-Speicherblocks deaktiviert werden (außer im L2-Cachespeicher **14**, der täglich ist, um die Latenzzeit des Speichers **26** zu verringern und keinem spezifischen Agenten zugeordnet ist). Falls im Cache gespeicherte Kopien des Cache-Speicherblock in Agenten-Cache-Speichern vorhanden sind, sind die Daten veraltet, da die Daten mit den Paketdaten überschrieben werden. Das L2CA-Signal schließlich ist horizontal von dem Etikett L2CA veranschaulicht. Das L2CA-Signal ist bei dieser Ausführungsform als aktiviert hoch und deaktiviert niedrig veranschaulicht.

[0070] Bei diesem Beispiel ist die Paketschnittstellenschaltung **22A** dazu programmiert (über das Konfigurationsregister **42**), die ersten zwei Cache-Speicherblöcke eines empfangenen Paketes im L2-Cachespeicher **14** zu speichern. Somit aktiviert die Paketschnittstellenschaltung **22A** das L2CA-Signal für die ersten zwei Transaktionen, um das empfangene Paket auf dem Bus **24** zu übertragen, und deaktiviert das L2CA-Signal für die übrigen Transaktionen. Es wird angemerkt, dass die Paketschnittstellenschaltung **22A** programmiert sein kann, keinen der Cache-Speicherblöcke im L2-Cachespeicher, einen der Cache-Speicherblöcke, oder mehr als zwei

der Cache-Speicherblöcke zu speichern, gemäß dem Teil des Paketes, von dem erwartet wird, dass es von den Prozessoren **12A–12B** verarbeitet wird.

[0071] Die erste Transaktion ist der erste Cache-Speicherblock des Paketes (Taktzyklus CLK0). Die Adresse der Transaktion ist die Adresse des Datenpuffers, der im Ansprechen auf den Empfang des Paketes von dem Deskriptor ausgewählt wird (Adresse "A" in [Fig. 4](#) – Bezugszeichen **90**). Da die erste Transaktion der erste Cache-Speicherblock des Paketes ist, wird zusätzlich das L2CA-Signal für diese Transaktion aktiviert (Bezugszeichen **92**). Der L2-Cachespeicher **14** weist eine Cache-Speicherblock-Speicherstelle zum Speichern des durch Adresse "A" adressierten Cache-Speicherblocks zu, falls die Adresse ein Fehlschlag ist, oder aktualisiert den Cache-Speicherblock, falls die Adresse ein Treffer ist. Auf ähnliche Weise ist die zweite Transaktion der zweite Cache-Speicherblock des Paketes und wird daher unmittelbar an dem ersten Cache-Speicherblock des Paketes in dem Speicherraum gespeichert. Daher ist die Adresse der zweiten Transaktion die Adresse "A + 1", wobei "A + 1" die Adresse "A" plus einen Cache-Speicherblock ist. Die zweite Transaktion ist in Taktzyklus CLK1 gezeigt (Bezugszeichen **94**). Das L2CA-Signal ist auch für die zweite Transaktion aktiviert, wodurch veranlasst wird, dass der Block im L2-Cachespeicher **14** gespeichert wird.

[0072] Die übrigen drei Transaktionen an aufeinander folgenden Cache-Speicherblöcken (Adressen "A + 2", "A + 3", und "A + 4" an den Bezugszeichen **98**, **100** bzw. **102**) sollen nicht im L2-Cachespeicher **14** zugewiesen werden. Daher wird das L2CA-Signal für diese Transaktionen deaktiviert (Bezugszeichen **104**, **106** bzw. **108**). Falls die Adressen zufällig einen Treffer im L2-Cachespeicher **14** darstellen, aktualisiert der L2-Cachespeicher **14** mit dem Cache-Speicherblock. Wenn die Adressen jedoch einen Fehlschlag im L2-Cachespeicher darstellen, werden die Cache-Speicherblöcke von dem Speichercontroller **16** im Speicher **26** gespeichert, und der L2-Cachespeicher **14** wird nicht aktualisiert. Während der L2-Cachespeicher den Cache-Speicherblock für einen Cache-Speichertreffer aktualisiert, falls das L2CA-Signal deaktiviert ist, können andere Ausführungsformen, falls gewünscht, den Cache-Speicherblock devalidieren, wenn das L2CA-Signal deaktiviert ist.

[0073] Es wird angemerkt, dass Adressphasen bei der dargestellten Ausführungsform während einer Phase des Taktsignals CLK vorkommen (z.B. während der hohen Phase des Taktsignals, obgleich auch die niedrige Phase verwendet werden kann). Bei der veranschaulichten Ausführungsform können Signale auf dem Bus **24** während einer Phase des Taktzyklus angetrieben werden, und die andere Phase kann im Leerlauf sein. Genauer gesagt kann die andere Pha-

se bei einer Ausführungsform dazu verwendet werden, die Differentialsignalleitungen vorzuladen. So mit können die Signalleitungen im Ansprechen auf eine Flanke des Taktsignals CLK (z.B. die ansteigende Flanke) antreiben und können die Signalleitungen im Ansprechen auf die abfallende Flanke abtasten. Andere Ausführungsformen können wie viele externe Busse im Ansprechen auf die gleichen Flanken antreiben und abtasten, oder können an beiden Flanken des Taktes antreiben und abtasten, wie etwa quellsynchrone Entwürfe. Das L2CA-Signal ist während der niedrigen Phase des Taktes in [Fig. 4](#) gepunktet gezeigt, um anzugeben, dass es während dieser Phase nicht aktiv angetrieben wird (and vorgeladen werden kann).

[0074] Es wird angemerkt, dass in [Fig. 4](#) zwar aufeinander folgende Taktzyklen zum Übertragen von Cache-Speicherblöcken eines Paketes gezeigt sind, dass jedoch andere Transaktionen allgemein zwischen jeder Cache-Speicherblocktransaktion (z.B. von anderen Agenten auf dem Bus **24**) vorkommen können. Ferner können Verzögerungen zwischen Cache-Speicherblock-Transaktionen auftreten, während die aufeinander folgenden Cache-Speicherblöcke von der Paketschnittstellenschaltung **22A** empfangen werden. Somit brauchen die Transaktionen nicht aufeinander folgend stattzufinden, wie in [Fig. 4](#) gezeigt ist.

[0075] Nachdem die in [Fig. 4](#) veranschaulichten fünf Transaktionen abgeschlossen sind (unter der Annahme keiner weiteren Transaktionen, und dass die Adressen A, A + 1, A + 2, A + 3 und A + 4 auf verschiedene Cache-Speicherblock-Speicherstellen im L2-Cachespeicher **14** mappen), würden dann die den Adressen A und A + 1 entsprechenden Cache-Speicherblöcke im Cache gespeichert. Die Adressen A + 2, A + 3 und A + 4 können im Cache gespeichert werden, wenn sie vor dem Taktzyklus CLK0 im Cache gespeichert wurden, aber im Ansprechen auf die in [Fig. 4](#) veranschaulichten Transaktionen nicht zugewiesen wurden.

[0076] Unter Bezugnahme auf [Fig. 5](#) ist ein Blockdiagramm einer Ausführungsform eines Satzes von Deskriptoren gezeigt, der einem DMA-Controller **38** zugewiesen ist. Andere Ausführungsformen sind möglich und werden in Betracht gezogen. In [Fig. 5](#) sind Konfigurationsregister **40A**, **40B**, **40C** und **40D** gezeigt sowie ein Satz von Deskriptoren **110** (einschließlich der Deskriptoren **110A**, **110B**, **110C**, **110D** und **110E**). Der Deskriptor **110D** ist in mehr Detail gezeigt. Es ist auch ein von dem Deskriptor **110D** angezeigter Datenpuffer **112** in [Fig. 5](#) gezeigt.

[0077] Die Konfigurationsregister **40A–40D** können Teil der in [Fig. 1](#) gezeigten Konfigurationsregister **40** sein. Das Konfigurationsregister **40A** speichert eine Deskriptorbasisadresse, welche einen Speicherbe-

reich angibt, der die Deskriptoren **110** speichert. Das Konfigurationsregister **40B** speichert einen Deskriptorzählerwert, der die Gesamtanzahl von Deskriptoren angibt. In Kombination mit der Deskriptorbasisadresse kann der Deskriptorzählerwert das Ausmaß des Speicherbereichs angeben, der die Deskriptoren speichert. Bei dieser Ausführungsform sind die Deskriptoren als ein Ring angeordnet. Der nächste Deskriptor in dem Ring, der von der Paketschnittstellenschaltung **22A** verwendet wird, ist durch die gegenwärtige Deskriptoradresse angegeben, die im Konfigurationsregister **40C** gespeichert ist. Ein gegenwärtiger Deskriptorzählerwert (der im Konfigurationsregister **40D** gespeichert ist) gibt die Anzahl von Deskriptoren an, die gegenwärtig für die Verwendung durch die Paketschnittstellenschaltung **22A** verfügbar sind. Die Paketschnittstellenschaltung **22A** kann keine Deskriptoren anwenden, abgesehen von den Deskriptoren zwischen dem gegenwärtigen Deskriptor und dem von dem gegenwärtigen Deskriptorzählerwert angegebenen Deskriptor.

[0078] Im Allgemeinen kann der Ring von Deskriptoren **110** durch Software erstellt werden, die auf dem System **10** ausgeführt wird, und kann der Paketschnittstellenschaltung **22A** durch Aktualisieren der Konfigurationsregister **40** zugewiesen werden. Ähnliche Ringe von Deskriptoren können den anderen Paketschnittstellenschaltungen **22B–22C** zugewiesen werden. Ein separater Satz von Deskriptoren **110** kann jedem DMA-Kanal innerhalb jeder Paketschnittstellenschaltungen **22A–22C** zugewiesen werden. Beispielsweise kann bei einer Ausführungsform jeder DMA-Controller **38** zwei DMA-Empfangskanäle (zum Übertragen empfangener Pakete an den Speicher) und zwei DMA-Sendekanäle (zum Übertragen von Paketen von dem Speicher zu der Paketschnittstellenschaltung) aufweisen. Somit können Sätze von Konfigurationsregistern ähnlich den Konfigurationsregistern **40A–40D** für jeden DMA-Kanal enthalten sein.

[0079] Der Deskriptor **110D** ist in mehr Detail gezeigt, und andere Deskriptoren können ähnlich sein. Der Deskriptor **110D** weist eine Flag (Bezugszeichen **114**), eine Paketgröße (Bezugszeichen **116**), eine Datenpuffergröße (Bezugszeichen **118**) und einen Datenpufferzeiger (Bezugszeichen **120**) auf. Der Datenpufferzeiger **120** ist die Adresse eines Datenpuffers, in den die Paketdaten gespeichert werden sollen. Die Datenpuffergröße **118** ist die Größe des Datenpuffers. Die Paketgröße **116** ist die Größe des in dem Datenpuffer **112** gespeicherten Paketes (und möglicherweise in Datenpuffern nachfolgender Deskriptoren, falls die Paketgröße größer als die Datenpuffergröße ist, wie nachfolgend in mehr Detail beschrieben ist). Die Flag **114** ist eine Angabe darüber, ob ein Paket in dem Datenpuffer **112** ist oder nicht. Beispielsweise kann die Flag **114** ein Bit sein, das angibt, dass der Deskriptor ein Paket speichert, wenn

es gesetzt ist, und das angibt, dass der Deskriptor kein Paket speichert, wenn es nicht gesetzt ist. Als Alternative können der gesetzte und nicht gesetzte Zustand des Bit auf die entgegengesetzte Weise definiert sein.

[0080] Es wird angemerkt, dass zwar die veranschaulichte Ausführungsform einen Ring von Deskriptoren **110** zeigt, dass aber andere Ausführungsformen die Deskriptoren auf andere Weisen anordnen können. Beispielsweise kann eine Verknüpfungsliste von Deskriptoren verwendet werden. Bei einer solchen Ausführungsform kann die Deskriptorbasisadresse in dem Konfigurationsregister **40A** der Kopf der Verknüpfungsliste sein, und der Deskriptorzählerwert in dem Konfigurationsregister **40B** kann die Anzahl von Deskriptoren in der Verknüpfungsliste sein. Die gegenwärtige Deskriptoradresse in dem Konfigurationsregister **40C** kann auf den gegenwärtige Deskriptor in der Verknüpfungsliste weisen, und der gegenwärtige Deskriptorzählerwert in dem Konfigurationsregister **40D** kann, beginnend mit dem gegenwärtigen Deskriptor, die Anzahl von Deskriptoren in der Verknüpfungsliste angeben, die für die Verwendung durch die Paketschnittstellenschaltung **22A** verfügbar sind. Da die Mitglieder einer Verknüpfungsliste nicht in benachbarten Adressen gespeichert zu werden brauchen, können die Deskriptoren ein zusätzliches Zeigerfeld (zusätzlich zu dem in [Fig. 5](#) gezeigten Datenpufferzeiger **120**) aufweisen, um auf den nächsten Deskriptor zu weisen. Bei einer konkreten Ausführungsform kann das System sowohl den Ring als auch die Verknüpfungslistenmodi unterstützen. Bei einer solchen Ausführungsform kann der zweite Zeiger dazu verwendet werden, im Ringmodus auf einen zweiten Datenpuffer zu weisen (d.h. die Deskriptoren können jeweils zwei Datenpuffer angeben).

[0081] Es wird angemerkt, dass je nach Wunsche zusätzlich zu den in [Fig. 5](#) gezeigten Informationen eine Vielzahl von anderen Informationen in den Deskriptoren **110** enthalten sein kann. Beispielsweise können Statusinformationen, während des Empfangs erfasste Fehler usw. für empfangene Pakete enthalten sein.

[0082] Unter Bezugnahme auf [Fig. 6](#) wird als nächstes ein Ablaufdiagramm gezeigt, das den Betrieb der Paketschnittstellenschaltung **22A** (oder einer jeden der anderen Paketschnittstellenschaltungen **22B–22C**) während des Empfangs eines Paketes von der externen Paketschnittstelle zum Verarbeiten in dem System **10** veranschaulicht. Andere Ausführungsformen sind möglich und werden in Betracht gezogen. Die in [Fig. 6](#) gezeigten Blöcke sind für ein vereinfachtes Verständnis in einer bestimmten Reihenfolge veranschaulicht, jedoch kann je nach Wunsch jede andere Reihenfolge verwendet werden.

[0083] Die Paketschnittstellenschaltung **22A** erfasst, ob ein Anfang eines Paketes an der Paketschnittstelle **44A** empfangen wird oder nicht (Entscheidungsblock **130**). Falls kein Anfang eines Paketes empfangen wird, kann die Paketempfang-Logikschaltung im Leerlauf sein. Je nach der Betriebsart kann ein Anfang eines Paketes auf eine Vielzahl von Weisen erfasst werden. Wenn die Paketschnittstellenschaltung **22A** als Ethernet-MAC arbeitet, wird der Anfang eines Paketes gemäß der Ethernet-Spezifikation erfasst. Wenn die generische Paketschnittstelle verwendet wird, können Steuersignale, welche den Anfang des Paketes (oder das Ende des vorherigen Paketes) markieren, zum Erfassen des Anfangs eines Pakets verwendet werden.

[0084] Wenn ein Anfang eines Paketes empfangen wird, wählt die Paketschnittstellenschaltung **22A** (insbesondere der DMA-Controller **38**) den nächsten Deskriptor **110**, an den das Paket übertragen wird (Block **132**). Der von der gegenwärtigen Deskriptoradresse angegebene Deskriptor wird ausgewählt. Die Paketschnittstellenschaltung **22A** sammelt Paketbytes in dem Rx-FIFO **34** (Block **134**), bis mindestens ein Cache-Speicherblock für die Übertragung bereit ist (Entscheidungsblock **136**). Falls gewünscht, kann die Paketschnittstellenschaltung **22A** bei einigen Ausführungsformen warten, bis mehr als ein Cache-Speicherblock für die Übertragung bereit ist.

[0085] Falls ein Cache-Speicherblock für die Übertragung bereit ist, überträgt der DMA-Controller **38** den Cache-Speicherblock zu dem Speicher **26** oder dem L2-Cachespeicher **14**. Genauer gesagt bestimmt der DMA-Controller, ob die in dem Konfigurationsregister **42** angegebene Anzahl von Cache-Speicherblöcken (gemessen ab dem Anfang des Paketes) übertragen wurde oder nicht im Hinblick auf das Paket, das empfangen wird (Entscheidungsblock **138**). Der DMA-Controller **38** kann die Anzahl von übertragenen Blöcken auf einem bestimmten DMA-Empfangskanal verfolgen, um zu bestimmen, ob die Anzahl von angegebenen Cache-Speicherblöcken schon übertragen wurde, und kann den Zählerwert mit der angegebenen Anzahl von Blöcken vergleichen. Als Alternative kann der DMA-Controller **38** einen Zähler mit der Anzahl von angegebenen Cache-Speicherblöcken initialisieren und den Zähler dekrementieren, während Blöcke übertragen werden.

[0086] Wenn die Anzahl von übertragenen Cache-Speicherblöcken immer noch weniger als oder gleich dem L2-Zählerwert ist (Entscheidungsblock **138**), überträgt der DMA-Controller **38** den Cache-Speicherblock in einer Transaktion mit dem L2CA-Signal aktiviert (Block **140**). Wenn andererseits die Anzahl von übertragenen Blöcken größer als der L2-Zählerwert ist, überträgt der DMA-Controller den Cache-Speicherblock in einer Transaktion mit dem L2CA-Signal deaktiviert (Block **142**).

[0087] Die Paketschnittstellenschaltung **22A** ist auch dazu konfiguriert, das Ende des Paketes zu erfassen (Entscheidungsblock **144**). Das Paketende kann durch die Länge des Paketes angegeben werden (bei Ethernet-Paketen in dem Paket angegeben) oder kann von Steuersignalen angegeben werden (wie etwa bei der generischen Paketschnittstelle). Wenn das Ende des Paketes nicht erfasst wird, kann die Paketschnittstellenschaltung **22A** bestimmen, ob das Ende des Datenpuffers erreicht worden ist (Entscheidungsblock **146**). Die von den Deskriptoren angegebenen Datenpuffer können oder können nicht groß genug sein, um das gesamte Paket zu enthalten. Wenn die Datenpuffer so groß wie das größte Paket sind, kann die von dem Entscheidungsblock **146** dargestellte Prüfung weg gelassen werden. Wenn das Ende des Datenpuffers erreicht worden ist (wie durch die Datenpuffergröße **118** in dem Deskriptor angegeben ist), kann der DMA-Controller **38** den nächsten Deskriptor auswählen und damit beginnen, das Paket in dem Datenpuffer zu speichern, der von diesem Deskriptor angegeben wird (Block **132**). Wenn andererseits das Ende des gegenwärtigen Datenpuffers nicht erreicht worden ist, kann die Paketschnittstellenschaltung **22A** mit dem gegenwärtigen Datenpuffer fortfahren.

[0088] Wenn das Ende des Paketes erreicht worden ist, kann der DMA-Controller **38** den Deskriptor aktualisieren (Block **148**). Konkret gesprochen kann der DMA-Controller **38** die Paketgröße **116** aktualisieren, um die Größe des Paketes in dem von dem Deskriptor angegebenen Datenpuffer anzugeben. Zusätzlich kann der DMA-Controller **38** das Flagbit **114** setzen, um anzugeben, dass das Paket in dem Datenpuffer für die Verarbeitung verfügbar ist. Wie bereits erwähnt wurde, kann ein Paket bei einigen Ausführungsformen je nach der Größe einen oder mehr Datenpuffer einnehmen, die einem oder mehr Deskriptoren entsprechen. Bei einer Ausführungsform kann der erste Deskriptor, der für das Paket gewählt wird, aktualisiert werden, wenn mehr als ein Deskriptor verwendet wird.

[0089] Der DMA-Controller **38** kann zusätzlich das Konfigurationsregister **40C** aktualisieren, um den nächsten unbenutzten Deskriptor anzugeben, und kann den gegenwärtigen Deskriptorzählerwert in dem Konfigurationsregister **40D** dekrementieren, um die Verwendung von einem oder mehr Deskriptoren zum Speichern des Paketes zu reflektieren. Falls der gegenwärtige Deskriptorzählerwert Null ist (oder unter einem Schwellwert liegt, je nach der Ausführungsform), kann der DMA-Controller **38** den Prozessor unterbrechen, damit mehr Deskriptoren für die Verwendung validiert werden.

[0090] Optional kann die Paketschnittstellenschaltung **22A** einen Prozessor unterbrechen (Block **150**). Der Prozessor kann dadurch in Kenntnis gesetzt wer-

den, dass das Paket für die Verarbeitung verfügbar ist. Als Alternative können die Prozessoren die Deskriptoren periodisch abfragen, was dem Erfassen durch DMA-Empfangskanäle entspricht, ob das Flagbit gesetzt ist.

[0091] Es wird angemerkt, dass die Paketschnittstellenschaltung **22A** bestimmte Fehlerprüfungen an empfangenen Paketen durchführen kann (z.B. Erfassung einer Paketkollision an dem empfangenen Paket, CRC-Fehler usw.), was in [Fig. 6](#) nicht veranschaulicht ist. Falls ein Paketfehler erfasst wird, kann das Paket fallen gelassen werden, und der Fehler kann (bei einigen Ausführungsformen) geloggt werden.

[0092] Es wird angemerkt, dass die Erfassung von Paketanfang und -ende und die Akkumulation von Paketdaten von der Tx/Rx-Steuerschaltung **32** unabhängig und parallel zu der Deskriptorverwaltung und dem Übertragen von Blöcken durchgeführt werden kann, die von dem DMA-Controller **38** behandelt werden können. Die in [Fig. 6](#) gezeigten Blöcke können somit eine vereinfachte Ansicht zur Veranschaulichung der L2-Cachespeicherzuweisung sein.

[0093] Unter Bezugnahme auf [Fig. 7](#) ist als nächstes ein Ablaufdiagramm zur Veranschaulichung des Betriebs eines Prozessors **12A–12B** im Ansprechen auf seine Unterbrechung gezeigt. Andere Ausführungsformen sind möglich und werden in Betracht gezogen. Es wird angemerkt, dass die Blöcke von [Fig. 7](#) in Software implementiert werden können, die auf dem Prozessor **12A–12B** läuft (z.B. eine Interrupt Service Routine), als Hardware, oder als eine Kombination von diesen. Während die gezeigten Blöcke für ein vereinfachtes Verständnis in einer bestimmten Reihenfolge veranschaulicht sind, können andere Reihenfolgen verwendet werden.

[0094] Der Prozessor liest eine oder mehr Ressourcen in dem System **10**, um die Quelle des Interrupt zu bestimmen (Block **160**). Beispielsweise kann der Interrupt-Mapper **28** ein Quellenregister umfassen, das die Quelle eines Interrupt angibt. Falls die Quelle nicht eine Paketschnittstellenschaltung (oder PIC in [Fig. 7](#)) (Entscheidungsblock **162**) ist, kann der Prozessor innerhalb der Interrupt-Verarbeitung auf jegliche beliebige Weise fortfahren (Block **164**). Wenn die Quelle eine Paketschnittstellenschaltung ist, kann der Prozessor ferner ein Paketschnittstellenschaltung-Ursachenregister lesen, um die Ursache für den Interrupt zu bestimmen (Block **166**). Das Ursachenregister könnte sich je nach Wunsch in dem Interrupt-Mapper **28** oder in der Paketschnittstellenschaltung **22A–22C** befinden.

[0095] Falls die Ursache für den Interrupt eine andre ist, als dass ein oder mehr Pakete für die Verarbeitung bereit sind (Entscheidungsblock **168**), kann der

Prozessor die anderen Ursachen auf jede beliebige Weise verarbeiten (Block **170**). Eine Ursache kann beispielsweise sein, dass die Paketschnittstellenschaltung alle verfügbaren Deskriptoren verwendet hat (oder weniger als eine Schwellwertanzahl von Deskriptoren für die Verwendung übrig hat). Der Prozessor kann Datenpuffer zuweisen, Deskriptoren in der Gruppe von Deskriptoren, die der Paketschnittstellenschaltung (und dem DMA-Kanal in der Paketschnittstellenschaltung) zugewiesen sind, auf die Datenpuffer weisen, und die Konfigurationsregister **40** aktualisieren, um die zugewiesenen Datenpuffer und aktualisierten Deskriptoren zu reflektieren. Bei einer Ausführungsform wird der gegenwärtige Deskriptorzähler aktualisiert, indem die Anzahl von neu hinzugefügten Deskriptoren in das Konfigurationsregister **40D** geschrieben wird. Der DMA-Controller **38** kann die Anzahl im Ansprechen auf das Schreiben zu dem gegenwärtigen Deskriptorzähler hinzufügen. Eine Vielzahl von anderen Ursachen für einen Interrupt durch die Paketschnittstellenschaltungen kann nach Wunsch unterstützt werden.

[0096] Falls die Ursache für den Interrupt ist, dass ein oder mehr Pakete bereit sind, kann der Prozessor den Deskriptor lesen, der das Paket speichert (oder den Anfang des Paketes speichert, falls Datenpuffer von mehr als einem Deskriptor zum Speichern des Paketes verwendet werden), um die Adresse des Anfangs des Paketes zu erhalten (Block **172**). Beispielsweise kann der Prozessor das Konfigurationsregister **40C** lesen und in Rückwärtsrichtung durch den Ring nach dem ersten Deskriptor suchen, der ein gesetztes Flagbit **114** aufweist. Als Alternative kann die Paketschnittstellenschaltung die Deskriptoradresse des ersten für die Verarbeitung bereiten Deskriptors in ein Register liefern, das von dem Prozessor gelesen werden kann. Bei wieder einer anderen Alternative kann der Prozessor das Deskriptorbasisadresse-Register **40A** lesen und in Vorwärtsrichtung suchen, bis ein Deskriptor mit einem gesetzten Flagbit **114** lokalisiert wird. Der Prozessor kann das von dem Deskriptor lokalisierte Paket verarbeiten (Block **174**). Wenn das L2CA-Signal dazu verwendet wurde, einen Teil des Paketes während der Übertragung von der Paketschnittstellenschaltung zu dem L2-Cachespeicher **14**/Speicher **26** in den L2-Cachespeicher **14** zuzuweisen, kann der Prozessor L2-Cachespeicher auf Treffer für den zu verarbeitenden Teil des Paketes stoßen, was die Latenzzeit des Zugriffs auf das Paket verringern kann (und daher auch die Leistungsfähigkeit der Paketverarbeitung verbessern kann).

[0097] Wie bereits erwähnt wurde, wenn eine bestimmte Paketschnittstellenschaltung **22A–22C** ein Paket von dem System **10** an der Paketschnittstelle überträgt, kann die Übertragung des Paketes fehlgeschlagen (das Paket kann von dem Empfänger fallen gelassen werden und eine Übertragungswiederholung erfordern), falls das Paket an der Paketschnitt-

stelle nicht kontinuierlich übertragen wird. Eine mögliche Quelle für eine Verzögerung bei der Übertragung des Paketes, welche zu einem Fehlschlag bei der Übertragung führen kann, ist die Latenzzeit beim Auslesen des Paketes aus dem Speicher. Die Latenzzeit kann teilweise auf Transaktionen zurückgehen, die von anderen Agenten initiiert werden (z.B. den Prozessoren **12A–12B**, dem L2-Cachespeicher **14** zum Entfernen von Rückkopien, oder Les-/Schreibtransaktionen anderer I/O-Vorrichtungen). Ferner können die Paketschnittstellenschaltungen während Zeiten einer starken Konkurrenz um den Bus **24** durch die Round Robin-Prioritätsmethode für die Arbitration verzögert werden. Der Speichercontroller **16** kann ein anderes Blocksignal für die Paketschnittstellenschaltungen **22A–22C** als für die anderen Agenten implementieren, um das Problem des Fallenlassens übertragener Pakete zu mildern. Genauer gesagt können die Blocksignale bei einer Ausführungsform Teil eines allgemeinen Blockierverfahrens für eine Flusssteuerung auf dem Bus **24** sein, wie in [Fig. 8](#) veranschaulicht ist. Als Alternative können andere Ausführungsformen, falls gewünscht, Blocksignale nur vom Speichercontroller **16** implementieren. Eine Beschreibung des allgemeinen Blockierverfahrens folgt.

[0098] Im Allgemeinen können die Prozessoren **12A–12B**, der L2-Cachespeicher **14** und/oder die I/O-Bridge **20** Transaktionen auf dem Bus **24** initiieren. Das System **10** unterstützt verschiedene Typen von Transaktionen auf dem Bus **24**, einschließlich Speichertransaktionen (wie vorausgehend beschrieben) und I/O Transaktionen, welche eine I/O-Vorrichtung (z.B. eine durch die I/O-Bridge **20** oder eine andere, nicht gezeigte I/O-Bridge gekoppelte Vorrichtung) vormerken. Jeder von den Prozessoren **12A–12B**, dem L2-Cachespeicher **14**, dem Speichercontroller **16** oder der I/O-Bridge **20** kann ein Teilnehmer an den Transaktionen sein. Im Allgemeinen liegt die "Teilnahme" eines Agenten an einer Transaktion vor, wenn von dem Agenten während der Transaktion eine Aktion erforderlich ist, damit die Transaktion ordnungsgemäß abgeschlossen wird. Der Agent nimmt an einer Transaktion "nicht teil", falls von dem Agenten während der Transaktion keine Aktion erforderlich ist, damit die Transaktion ordnungsgemäß abgeschlossen wird. Auf ähnliche Weise ist der Agent ein "Teilnehmer" an der Transaktion, wenn es erforderlich ist, dass der Agent während der Transaktion eine Aktion unternimmt, damit die Transaktion ordnungsgemäß abgeschlossen wird. Ein Agent wird selbst dann als ein Teilnehmer an einer bestimmten Transaktion bezeichnet, wenn diese Transaktion noch nicht auf dem Bus **24** initiiert worden ist, falls dieser Agent während der bestimmten Transaktion eine Aktion ergreifen würde, sobald diese bestimmte Transaktion durchgeführt wird. Der Agent ist ein "Nichtteilnehmer" an der Transaktion, falls es nicht erforderlich ist, dass der Agent während der Transaktion eine Ak-

tion ergreift, damit die Transaktion ordnungsgemäß abgeschlossen wird. Ein Agent wird selbst dann als ein Nichtteilnehmer an einer bestimmten Transaktion bezeichnet, wenn diese Transaktion noch nicht auf dem Bus **24** initiiert worden ist, falls dieser Agent während der bestimmten Transaktion keine Aktion ergreifen würde, sobald diese bestimmte Transaktion durchgeführt wird. Der durch die Transaktion vorgenommene Agent nimmt an der Transaktion teil. Zusätzlich können andere Agenten an verschiedenen Transaktionen teilnehmen. Beispielsweise können die Prozessoren **12A–12B** (und bei einigen Ausführungsformen die I/O-Bridge **20**) an kohärenten Speichertransaktionen teilnehmen, indem sie eine Aktion ergreifen, um die Kohärenz sicher zu stellen (z.B. Snooping von internen Cache-Speichern und Angeben des Ergebnisses des Snoop). Der L2-Cachespeicher **14** kann an Cache-speicherbaren Speichertransaktionen teilnehmen, indem er bestimmt, ob die Transaktion ein Treffer ist, und im Ansprechen auf die Transaktion die Daten liefert oder die darin gespeicherten Daten aktualisiert. Gemäß der vorliegenden Verwendung bezieht sich der Ausdruck "Snoop" oder "Snooping" auf die Überprüfung eines Cache-Speichers, um zu bestimmen, ob Daten, die einer von einer Transaktion betroffenen Speicherstelle entsprechen, in dem Cache-Speicher vorhanden sind, und das Signalisieren des Zustands der Daten, um sicher zu stellen, dass die Kohärenz der Daten gewahrt bleibt.

[0099] Ein adressierbarer Bereich des Systems **10** ist durch die Größe der Adressen definiert, die auf dem Bus **24** übertragen werden können. Der adressierbare Bereich kann in mehrere Adressräume unterteilt sein, einschließlich eines Speicheradressraums und verschiedener I/O-Adressräume. Bei dieser Ausführungsform kann der Adressraum, der die Adresse einer Transaktion enthält, den Typ der Transaktion identifizieren. Somit gibt eine Adresse in dem Speicheradressraum an, dass die Transaktion eine Speichertransaktion ist, und die Adresse identifiziert eine vorgemerkte Speicherstelle in dem Speicher **26** (weshalb der Speichercontroller **16** und der L2-Cachespeicher **14** auf die Transaktion ansprechen können, und die Prozessoren **12A–12B** durch Snooping an der Transaktion teilnehmen können). Eine Adresse in den I/O-Adressräumen gibt an, dass die Transaktion eine I/O-Transaktion und die Adresse eine I/O-Vorrichtung vormerkt. Die I/O-Vorrichtungen können Paketschnittstellenschaltungen **22A–22C**, andere I/O-Vorrichtungen, oder durch die I/O-Bridge **20** mit dem Bus **24** gekoppelte Schnittstellen, und jegliche durch andere I/O-Bridges (nicht gezeigt) oder Schnittstellen gekoppelte Vorrichtungen aufweisen. Jeder der I/O Adressräume kann der I/O-Bridge zugeordnet werden, mit der die entsprechende I/O-Vorrichtung oder Bridge gekoppelt ist.

[0100] Während Adressräume verwendet werden,

um zu bestimmen, welcher Typ von Transaktion durchgeführt wird (und das Ziel der Transaktion zu bestimmen), können andere Ausführungsformen diese Attribute auf andere Weisen bestimmen. Beispielsweise können verschiedene Instruktionen verwendet werden, um verschiedene Typen von Transaktionen zu veranlassen oder um verschiedene Vorfällungen vorzumerken. Steuersignale auf dem Bus können verwendet werden, um das Ziel zu identifizieren.

[0101] Unter Bezugnahme auf [Fig. 8](#) ist dort ein Blockdiagramm gezeigt, das zusätzliche Details einer Ausführungsform der Prozessoren **12A–12B**, des L2-Cachespeichers **14**, des Speichercontrollers **16** und der I/O-Bridge **20** veranschaulicht. [Fig. 8](#) veranschaulicht auch eine Ausführungsform einer bestimmten Zwischenverbindung zwischen den oben erwähnten Agenten. Andere Ausführungsformen sind möglich und werden in Betracht gezogen. Bei der Ausführungsform von [Fig. 8](#) weist der Prozessor **12A** eine Anforderungsschaltung **230A**, eine Blockschaltung **232A**, eine Ausgabewarteschlange **234A** und eine Eingabewarteschlange **236A** auf. Die Anforderungsschaltung **230A** ist gekoppelt, um ein Blocksignal von jedem anderen Agenten auf dem Bus **24** zu empfangen (spezifisch das Block_MC1-Signal von dem Speichercontroller **16**), und ist mit der Ausgabewarteschlange **234A** gekoppelt. Die Blockschaltung **232A** ist mit der Eingabewarteschlange **236A** gekoppelt und dazu konfiguriert, ein Block_P0-Signal zu erzeugen. Der Prozessor **12B** weist eine ähnliche Gruppe von Elementen auf, einschließlich der Anforderungsschaltung **230B**, der Blockschaltung **232B**, der Ausgabewarteschlange **234B** und der Eingabewarteschlange **236B**. Die I/O-Bridge **20** ist auch auf ähnliche Weise konfiguriert und weist die Anforderungsschaltung **230C**, die Blockschaltung **232C**, die Ausgabewarteschlange **234C** und die Eingabewarteschlange **236C** auf. Der L2-Cachespeicher **14** kann auch auf ähnliche Weise konfiguriert sein und eine Anforderungsschaltung **230E**, eine Blockschaltung **232E**, eine Ausgabewarteschlange **234E** und eine Eingabewarteschlange **236E** aufweisen. Der Speichercontroller **16** weist eine Blockschaltung **232D** und eine Eingabewarteschlange **236D** sowie ein Konfigurationsregister **238** auf. Die Blockschaltung **232D** ist mit der Eingabewarteschlange **236D** und dem Konfigurationsregister **238** gekoppelt.

[0102] Im Allgemeinen erfasst jeder Agent die Transaktionen, bei der dieser Agent ein Teilnehmer ist. Die Transaktionen werden in der Eingabewarteschlange **236A–236E** dieses Agenten erfasst. Der Agent kann die Transaktion aus seiner Eingabewarteschlange **236A–236E** entnehmen, um die Transaktion zu verarbeiten. Die Größe jeder Eingabewarteschlange **236A–236E** kann eine Größe (d.h. die Anzahl von bestimmten Einträgen) auf der Grundlage der erwarteten Latenzzeit zum Verarbeiten der

Transaktionen in dem teilnehmenden Agenten und der Anzahl von zusätzlichen, während dieser Latenzzeitperiode erwarteten Transaktionen usw. besitzen. Es ist typischerweise nicht wünschenswert, eine Warteschlange mit genügend Einträgen vorzusehen, um die maximale Anzahl von Transaktionen zu speichern, die stattfinden könnten, da häufig viele der Warteschlangeneinträge ungenutzt bleiben und somit Schaltungsfäche in dem Agenten vergeuden würden. Daher können die Eingabewarteschlangen für die wahrscheinlichste Anzahl von Transaktionen während gewöhnlicher Transaktionsabfolgen bemessen sein, und die Warteschlange kann in einigen Fällen voll werden. Beispielsweise tendieren I/O-Transaktionen zu einer längeren Latenzzeit als Speichertransaktionen, kommen aber auch tendenziell weniger oft vor. Daher kann die Eingabewarteschlange **236C** relativ wenige Einträge aufweisen. Andererseits können Speichertransaktionen mit einer geringeren Latenzzeit häufiger vorkommen, und daher können die Eingabewarteschlangen **236A–236B** und **236E** eine größere Anzahl von Einträgen aufweisen. Jede der Eingabewarteschlangen **236A–236E** kann jedoch voll werden, weshalb der die volle Eingabewarteschlange enthaltende Agent möglicherweise nicht in der Lage ist, an zusätzlichen Transaktionen teil zu nehmen, bis Eingabewarteschlangeneinträge für das Speichern zusätzlicher Transaktionen frei sind.

[0103] Die Blockschaltungen **232A–232E** sind vorgesehen, um zu bestimmen, ob die entsprechende Eingabewarteschlange **236A–236E** voll wird, und um ein Blocksignal zu erzeugen, falls die entsprechende Eingabewarteschlange **236A–236E** voll wird. Bei der veranschaulichten Ausführungsform erzeugt jede Blockschaltung **232A–232E** ein separates Blocksignal wie folgt: Die Blockschaltung **232A** erzeugt Block_P0; die Blockschaltung **232B** erzeugt Block_P1; die Blockschaltung **232C** erzeugt Block_IO; die Blockschaltung **232D** erzeugt Block_MC1 und Block_MC2; und die Blockschaltung **232E** erzeugt Block_L2. Das Blocksignal kann aktiviert werden, um anzugeben, dass der entsprechende Agent für die Teilnahme an zusätzlichen Transaktionen nicht verfügbar ist, und deaktiviert werden, um anzugeben, dass der entsprechende Agent für die Teilnahme an zusätzlichen Transaktionen verfügbar ist.

[0104] Jeder anfordernde Agent (z.B. die Prozessoren **12A–12B**, die I/O-Bridge **20** und der L2-Cachespeicher **14**) weist eine Anforderungsschaltung **230A–230E** auf, die Blocksignale, sowie eine Ausgabewarteschlange **234A–234E**. Der anfordernde Agent kann Transaktionen, die auf dem Bus **24** initiiert werden sollen, in die entsprechende Ausgabewarteschlange **234Am234E** setzen. Die Anforderungsschaltungen **230A–230E** sind mit den jeweiligen Ausgabewarteschlangen **234A–234E** gekoppelt

und bestimmen, ob es dem entsprechenden Agenten erlaubt ist, Transaktionen aus der jeweiligen Ausgabewarteschlange im Ansprechen auf die Blocksignale auf dem Bus **24** zu initiieren. Genauer gesagt blockiert die Anforderungsschaltung **230A–230E** die Initiierung einer ersten Transaktion in der entsprechenden Ausgabewarteschlange **234A–234E**, wenn irgendein Agent, der ein Teilnehmer an der ersten Transaktion ist, sein Blocksignal aktiviert. Somit kann die Quelle einer jeden Transaktion selektiv die Initiierung der Transaktion in Abhängigkeit davon blockieren oder nicht blockieren, ob der/die andere(n) Teilnehmer an der Transaktion gegenwärtig zu einer Teilnahme in der Lage ist/sind. Bei der veranschaulichten Ausführungsform kann der L2-Cachespeicher **14** ein anfordernder Agent nur für Speichertransaktionen sein, und kann daher nur die für Speichertransaktionen verwendeten Signale empfangen (z.B. Block_P0, Block_P1, und Block_MC1). Andere Ausführungsformen können das Block_IO-Signal ebenfalls aufweisen (z.B. wenn die I/O-Bridges vorübergehend Daten Cache-speichern können und daher Speichertransaktionen aufspüren können).

[0105] Auf diese Weise werden Transaktionen, die nicht abgeschlossen werden können, weil ein teilnehmender Agent nicht in der Lage ist, teilzunehmen, nicht auf dem Bus **24** initiiert. Andererseits ist es zugelassen, dass Transaktionen, für die kein Teilnehmer ein Blocksignal aktiviert, auf dem Bus **24** initiiert werden. Mit anderen Worten, eine gegebene Transaktion wird nicht von einem Agenten blockiert, der ein Nichtteilnehmer an dieser Transaktion ist. Daher können Transaktionen auf Grund der Granularität des Blockierens (wobei z.B. jeder Agent unabhängig seine Fähigkeit zur Teilnahme anzeigt), selbst dann durchgeführt werden, wenn ein nicht-teilnehmender Agent diese Transaktionen blockiert.

[0106] Die Anforderungsschaltungen **230A–230E** können die Initiierung von Transaktionen auf eine Anzahl von Weisen blockieren. Wenn beispielsweise ein Blocksignal für einen teilnehmenden Agenten aktiviert wird, bevor der entsprechende Agent die Arbitration von dem Bus **24** gewinnt, um die blockierte Transaktion durchzuführen, kann die Anforderungsschaltung **230A–230E** die Arbitration (oder eine weitere Arbitration) unterbinden, um die blockierte Transaktion durchzuführen, bis das Blocksignal deaktiviert ist. Falls gewünscht, kann der entsprechende Agent arbitrieren, um andere, nicht-blockierte Transaktionen durchzuführen. Wenn andererseits das Blocksignal nicht aktiviert wird, bevor der entsprechende Agent die Arbitration gewinnt, kann die Anforderungsschaltung **230A–230E** den Agenten dazu veranlassen, eine ungültige Befehlscodierung auf dem Bus anzusteuern. In diesem Fall wird die Bandbreite nicht gewahrt, aber die Transaktion wird nicht durchgeführt, weshalb kein Neuversuch der Transaktion stattfindet.

[0107] Während die meisten der in [Fig. 8](#) veranschaulichten Teilnehmer ein einzelnes Blocksignal erzeugen, weist der Speichercontroller **16** zwei Blocksignale auf: Block_MC1 und Block_MC2. Die Anforderungsschaltungen **230A**, **230B**, und **230E** sind jeweils gekoppelt, um das Block_MC1-Signal zu empfangen, und die Anforderungsschaltung **230C** ist gekoppelt, um das Block_MC2-Signal zu empfangen. Bei einigen Ausführungsformen (z.B. Ausführungsformen, bei denen die I/O-Bridge **20** mit anderen I/O-Vorrichtungen gekoppelt ist), kann die Anforderungsschaltung **230C** auch gekoppelt sein, um das Block_MC1-Signal zur Verwendung beim Blockieren von Anforderungen von den anderen I/O-Vorrichtungen zu empfangen.

[0108] Jede der Anforderungsschaltungen **230A–230E** verwendet das von dem Speichercontroller **16** empfangene Blocksignal, um Speichertransaktionen zu blockieren (falls das Blocksignal aktiviert ist). Somit können die Prozessoren **12A–12B** und der L2-Cachespeicher **14** Speichertransaktionen blockieren, wenn das Block_MC1-Signal aktiviert ist, während die I/O-Bridge **20** Transaktionen blockieren kann, wenn das Block_MC2-Signal aktiviert ist. Somit können andere Agenten blockiert werden, und es kann den Paketschnittstellenschaltungen **22A–22C** ermöglicht werden, das Initiiieren von Speichertransaktionen fortzusetzen, indem der Speichercontroller **16** das Block_MC1-Signal aktiviert und das Block_MC2-Signal deaktiviert. Insbesondere kann das Block_MC1-Signal aktiviert werden, wenn die Anzahl von Transaktionen in der Eingabewarteschlange **236D** einen ersten Schwellwert übersteigt, und das Block_MC2-Signal kann aktiviert werden, wenn die Eingabewarteschlange **236D** voll wird (oder voll ist), ähnlich den anderen Blocksignalen von den anderen Agenten. Im Allgemeinen kann der erste Schwellwert weniger als die Anzahl von Transaktionen in der Eingabewarteschlange **236D** sein, bei der das Block_MC2-Signal aktiviert wird.

[0109] Bei einer Ausführungsform kann der Schwellwert zum Aktivieren des Block_MC1-Signals in dem Konfigurationsregister **238** programmierbar sein. Das Konfigurationsregister **238** kann eine Angabe des ersten Schwellwerts speichern. Die Angabe kann die Anzahl von Einträgen sein, die für das Speichern von Transaktionen frei sein müssen, bevor das Block_MC1-Signal deaktiviert wird, die Anzahl von Einträgen, die, wenn sie durch Transaktionen besetzt sind, die Aktivierung des Block_MC1-Signals veranlassen, ein Prozentsatz der Gesamtanzahl von Einträgen, welche die Aktivierung oder Deaktivierung des Block_MC1-Signals veranlassen, usw. Falls gewünscht, kann der erste Schwellwert als Alternative festgelegt sein.

[0110] Wie bereits erwähnt wurde, kann eine Blockschaltung **232A–232E** das entsprechende Blocksig-

nal aktivieren, wenn die entsprechende Eingabewarteschlange **236A–236E** voll wird. Genauer gesagt kann die Blockschaltung **232A–232E** das entsprechende Blocksignal aktivieren, wenn die Anzahl von freien (gegenwärtig unbenutzten) Einträgen auf einen vorgegebenen Schwellwert abfällt. Der Schwellwert kann Null sein, wenn das Blocksignal früh genug erzeugt werden kann, um jegliche nachfolgende Transaktion zu blockieren, die eine Teilnahme durch den entsprechenden Agenten erfordern kann. Bei einer Ausführungsform sind Arbitration und Transaktionsinitiierung jedoch pipelined. Daher kann die Initiierung einer Transaktion, die den letzten freien Eingabewarteschlangeneintrag füllen würde, parallel mit der Arbitration für eine Transaktion stattfinden, die ebenfalls einen Eingabewarteschlangeneintrag in diesem Agenten erfordern würde. Somit kann der Schwellwert bei einer solchen Ausführungsform Eins sein. Im Ansprechen auf eine Transaktion, welche den vorletzten freien Warteschlangeneintrag füllt und einen Warteschlangeneintrag frei lässt, aktiviert die entsprechende Blockschaltung **232A–232E** ihr Blocksignal.

[0111] Es wird angemerkt, dass jede der Eingabewarteschlangen **236A–236E** eine geeignete Anzahl von Einträgen aufweisen kann, die von der Anzahl von Einträgen in anderen Eingabewarteschlangen **236A–236E** verschieden sein kann. Auf ähnliche Weise kann jede der Ausgabewarteschlangen **234A–234E** eine geeignete Anzahl von Einträgen aufweisen, die von der Anzahl von Einträgen in anderen Ausgabewarteschlangen **234A–234E** verschieden sein kann.

[0112] Es wird ferner angemerkt, dass sich die oben stehende Beschreibung zwar auf Szenarios mit vollen Eingabewarteschlangen als Ursache für die Nichtverfügbarkeit zur Teilnahme bei zusätzlichen Transaktionen bezieht, dass es aber nach Wunsch bei einem gegebenen Agenten andere Ursachen dafür geben kann, dass er für die Teilnahme an zusätzlichen Transaktionen nicht verfügbar ist. Die Blocksignale können wie vorausgehend beschrieben auch für solche Szenarios verwendet werden.

[0113] Unter Bezugnahme auf [Fig. 9](#) ist als nächstes ein Ablaufdiagramm gezeigt, das den Betrieb einer Ausführungsform des in [Fig. 8](#) gezeigten Speichercontrollers **16** veranschaulicht. Andere Ausführungsformen sind möglich und werden in Betracht gezogen. Die in [Fig. 9](#) gezeigten Blöcke sind für ein vereinfachtes Verständnis in einer bestimmten Reihenfolge veranschaulicht. Es kann jedoch jede Reihenfolge verwendet werden. Konkret gesprochen sind die Blöcke **180** und **182** als Gruppe und die Blöcke **184** und **186** als Gruppe unabhängig und können in jeder Reihenfolge oder parallel ausgeführt werden. Konkret gesprochen kann die kombinatorische Logikschaltung in dem Speichercontroller **16** die Gruppen

parallel ausführen.

[0114] Falls die Anzahl von Transaktionen in der Eingabewarteschlange **236D** größer als der in dem Konfigurationsregister **238** (Entscheidungsblock **180**) eingestellte Schwellwert ist, aktiviert der Speichercontroller **16** das Block_MC1-Signal (Block **182**). Ansonsten ist das Block_MC1-Signal deaktiviert. Auf äquivalente Weise kann die Prüfung in Block **180** sein, dass die Anzahl von freien Einträgen in der Eingabewarteschlange **236D** weniger als der im Konfigurationsregister **238** eingestellte Schwellwert ist.

[0115] Wenn die Eingabewarteschlange **236D** voll ist (oder voll wird, wie vorausgehend beschrieben wurde) (Entscheidungsblock **184**), aktiviert der Speichercontroller das Block_MC2-Signal (Block **186**). Ansonsten ist das Block_MC2-Signal deaktiviert.

[0116] Unter Bezugnahme auf [Fig. 10](#) ist dort als nächstes ein Blockdiagramm eines Trägermediums **300** mit einer Datenbank gezeigt, das repräsentativ für das System **10** ist. Allgemein gesprochen kann ein Trägermedium Speichermedien wie etwa magnetische oder optische Medien, z.B. Disk oder CD-ROM, flüchtige oder nichtflüchtige Speichermedien wie etwa RAM (z.B. SDRAM, RDRAM, SRAM usw.), ROM usw., sowie Übertragungsmedien oder Signale wie etwa elektrische, elektromagnetische oder digitale Signale aufweisen, die über ein Kommunikationsmedium wie etwa ein Netz und/oder eine drahtlose Verbindung übertragen werden.

[0117] Im Allgemeinen kann die Datenbank des auf dem Trägermedium **300** getragenen Systems **10** eine Datenbank sein, die von einem Programm direkt oder indirekt gelesen und verwendet werden kann, um die Hardware herzustellen, welche das System **10** aufweist. Beispielsweise kann die Datenbank eine Behavioral Level-Beschreibung oder Register-Transfer Level (RTL)-Beschreibung der Hardware-Funktionalität in einer High Level-Entwurfssprache (HDL) wie etwa Verilog oder VHDL sein. Die Beschreibung kann von einem Synthesetool gelesen werden, das die Beschreibung synthetisieren kann, um eine Netlist zu erzeugen, die eine Liste von Gates von einer Synthesebibliothek aufweist. Die Netlist weist eine Gruppe von Gates auf, die ebenfalls die Funktionalität der Hardware darstellen, welche das System **10** aufweist. Die Netlist kann dann platziert und geroutet werden, um einen Datensatz zu erzeugen, der auf Masken anzuwendende geometrische Formen beschreibt. Die Masken können dann in verschiedenen Halbleiterherstellungsschritten verwendet werden, um eine Halbleiterschaltung oder Halbleiterschaltungen herzustellen, die dem System **10** entsprechen. Als Alternative kann die Datenbank auf dem Trägermedium **300** auf Wunsch die Netlist (mit oder ohne die Synthesebibliothek) oder der Datensatz sein.

[0118] Während das Trägermedium **300** eine Darstellung des Systems **10** trägt, können andere Ausführungsformen nach Wunsch eine Darstellung eines jeglichen Teils des Systems **10** tragen, einschließlich jeder Gruppe von Agenten (z.B. der Prozessoren **12A–12B**, des L2-Cachespeichers **14**, des Speichercontrollers **16** und/oder der I/O-Bridge **20**), Paketschnittstellenschaltungen **22A–22C** oder Teile davon, den Bus **24** usw. Die oben erwähnten Datenbanken können eine Schaltung aufweisen, die einen Mechanismus für das System **10** oder Teile davon definiert.

[0119] Zahlreiche Variationen und Modifikationen ergeben sich für den Fachmann bei einer gründlichen Würdigung der oben stehenden Beschreibung. Es ist beabsichtigt, dass die nachfolgenden Patentansprüche so interpretiert werden, dass sie alle solchen Variationen und Modifikationen mit umfassen.

[0120] Soweit vorausgehend beschriebene Ausführungsformen der Erfindung zumindest teilweise unter Verwendung einer Software-gesteuerten programmierbaren Verarbeitungsvorrichtung wie etwa eines Digitalen Signalprozessors, Mikroprozessors, anderer Verarbeitungsvorrichtungen, eines Datenverarbeitungsgeräts oder Computersystems implementierbar sind, dürfte es ersichtlich sein, dass ein Computerprogramm oder Programmelement zum Konfigurieren einer programmierbaren Vorrichtung, eines programmierbaren Geräts oder Systems zum Implementieren der vorausgehend beschriebenen Verfahren, Mechanismen und/oder Module als ein Aspekt der vorliegenden Erfindung in Betracht gezogen wird. Das Computerprogramm oder Programmelement kann als Quellcode ausgeführt sein und einer Kompilierung für die Implementierung auf einer Verarbeitungsvorrichtung, einem Gerät oder System unterzogen werden, oder kann beispielsweise als Objektcode ausgeführt sein. Für den Fachmann dürfte problemlos ersichtlich sein, dass der Ausdruck Computer/Rechner in seiner grundlegenden Bedeutung programmierbare Vorrichtungen wie die oben erwähnten, sowie Datenverarbeitungsgeräte und Computersysteme mit umfasst.

[0121] Auf geeignete Weise ist das Computerprogramm oder Programmelement auf einem Trägermedium in einer durch eine Maschine oder Vorrichtung lesbaren Form gespeichert, z.B. in Festkörperspeicher, optischem oder magneto-optischem Speicher wie etwa einer lesbaren und/oder beschreibbaren Disk, z.B. einer CD und einer DVD, oder Magnetspeicher wie Disc oder Band, und die Verarbeitungsvorrichtung verwendet das Programm, Programmelement oder einen Teil davon, um es für den Betrieb zu konfigurieren. Das Computerprogramm oder Programmelement kann von einer entfernten Quelle geliefert werden, die in einem Kommunikationsmedium wie etwa einem elektronischen Signal ausgeführt ist,

einschließlich Funkfrequenzträgerwelle oder optischer Trägerwelle. Solche Trägermedien werden ebenfalls als Aspekte der vorliegenden Erfindung in Betracht gezogen.

[0122] Im Hinblick auf die voraus gegangene Beschreibung bestimmter Ausführungsformen der Erfindung dürfte es für einen Fachmann ersichtlich sein, dass verschiedene Hinzufügungen, Modifikationen und Alternativen dazu vorgesehen werden können.

[0123] Im Hinblick auf die beigefügten Patentansprüche können Merkmale aus Unteransprüchen mit Merkmalen aus den nebengeordneten Ansprüchen kombiniert werden, und Merkmale aus jeweiligen nebengeordneten Ansprüchen können auf geeignete Weise und nicht nur in den in den Ansprüchen aufgezählten konkreten Kombinationen kombiniert werden.

[0124] Um Unsicherheiten zu vermeiden, sollte der Ausdruck "aufweisend", der in der Beschreibung und den Patentansprüchen verwendet wird, nicht so ausgelegt werden, dass er ausschließlich "bestehend aus" bedeutet.

Patentansprüche

1. Vorrichtung (**10**), welche aufweist:
einen Prozessor (**12A, 12B**), der mit einer Zwischenverbindung (**24**) gekoppelt ist;
einen Cache-Speicher (**14**), der mit der Zwischenverbindung (**24**) gekoppelt ist;
einen Speichercontroller (**16**), der mit der Zwischenverbindung (**24**) gekoppelt ist; und
eine Paketschnittstellenschaltung (**22A, 22B, 22C**), welche Pakete (**50**) von einer Paketschnittstelle (**44A, 44B, 44C**) empfängt und veranlasst, dass die Pakete (**50**) zum Speichern auf der Zwischenverbindung (**24**) übertragen werden;
wobei der Prozessor (**12A, 12B**) für das Verarbeiten der von der Paketschnittstelle (**44A, 44B, 44C**) empfangenen Pakete (**50**) programmierbar ist,
dadurch gekennzeichnet, dass
der Prozessor (**12A, 12B**), der Cache-Speicher (**14**), der Speichercontroller (**16**) und die Paketschnittstellenschaltung (**22A, 22B, 22C**) in ein einzelnes Halbleitersubstrat integriert sind;
die Paketschnittstellenschaltung (**22A, 22B, 22C**), im Ansprechen auf das Empfangen eines Paketes (**50**) von der Paketschnittstelle (**44A, 44B, 44C**), dazu konfiguriert ist, zu veranlassen, dass ein erster Abschnitt des Paketes (**50**) im Cache-Speicher (**14**) gespeichert wird, und ein restlicher Abschnitt des Paketes (**50**) in einem Speicher (**26**) gespeichert wird, mit dem der Speichercontroller (**16**) gekoppelt ist; und
der Speichercontroller (**16**) so gekoppelt ist, dass er ein erstes Blocksignal für die Paketschnittstellenschaltung (**22A, 22B, 22C**) und ein zweites Blocksignal für andere, mit der Zwischenverbindung (**24**) ge-

koppelte Vorrichtungen zur Verfügung stellt, und wobei die Initiierung von Speichertransaktionen von der Paketschnittstellenschaltung (22A, 22B, 22C) auf der Zwischenverbindung (24) im Ansprechen auf eine Aktivierung des ersten Blocksignals gesperrt ist, und wobei die anderen Vorrichtungen so konfiguriert sind, dass sie die Initiierung von Speichertransaktionen auf der Zwischenverbindung (24) im Ansprechen auf eine Aktivierung des zweiten Blocksignals sperren.

2. Vorrichtung nach Anspruch 1, wobei der erste Abschnitt mindestens einen Abschnitt eines Headers (52) des Paketes (50) aufweist.

3. Vorrichtung nach Anspruch 1 oder 2, wobei die Paketschnittstellenschaltung (22A, 22B, 22C) ein Konfigurationsregister (40) aufweist, das für das Speichern einer Größe des ersten Abschnitts des Paketes (50) programmierbar ist.

4. Vorrichtung nach einem der Ansprüche 1 bis 3, wobei die Paketschnittstelle (44A, 44B, 44C) einen Ethernet Media Access Controller aufweist.

5. Vorrichtung nach einem der Ansprüche 1 bis 4, wobei der Speichercontroller (16) eine Eingabewarteschlange mit einer Mehrzahl von Einträgen für Transaktionen aufweist, und wobei der Speichercontroller (16) dazu konfiguriert ist, das zweite Blocksignal im Ansprechen darauf zu aktivieren, dass eine Anzahl der Mehrzahl von Einträgen, die verfügbar sind, weniger als eine erste Anzahl ist.

6. Vorrichtung nach Anspruch 5, wobei der Speichercontroller (16) ferner dazu konfiguriert ist, das erste Blocksignal im Ansprechen darauf zu aktivieren, dass die Anzahl der Mehrzahl von Einträgen, die verfügbar sind, weniger als eine zweite Anzahl beträgt, wobei die zweite Anzahl weniger als die erste Anzahl beträgt.

7. Vorrichtung nach Anspruch 5 oder 6, wobei der Speichercontroller (16) ein Konfigurationsregister aufweist, das für das Speichern einer Angabe der ersten Anzahl programmierbar ist.

8. Vorrichtung nach einem der Ansprüche 1 bis 7, wobei die Paketschnittstellenschaltung (22A, 22B, 22C) mit der Zwischenverbindung (24) durch eine Busbrücke (20) gekoppelt ist.

9. Trägermedium (300) zum Tragen von einer oder mehr Datenbanken, welches die Vorrichtung (10) nach einem der Ansprüche 1 bis 8 darstellt und durch ein Programm zum Herstellen von Hardware, welche die Vorrichtung (10) aufweist, lesbar ist.

10. Verfahren, welches umfasst:
Empfangen eines Paketes (50) von einer Paketschnittstelle (44A, 44B, 44C) in einer Pa-

ketschnittstellenschaltung (22A, 22B, 22C); und Übertragen des Paketes (50) zum Speichern auf einer Zwischenverbindung (24), mit der ein Cache-Speicher (14), ein Speichercontroller (16) und ein Prozessor (12A, 12B) gekoppelt sind; dadurch gekennzeichnet, dass der Prozessor (12A, 12B), der Cache-Speicher (14), der Speichercontroller (16) und die Paketschnittstellenschaltung (22A, 22B, 22C) in ein einzelnes Halbleitersubstrat integriert sind; wobei das Verfahren ferner die Schritte umfasst: Speichern eines ersten Abschnitts des Paketes (50) in dem Cache-Speicher (14) im Ansprechen auf den Empfang des Paketes (50) von der Paketschnittstelle (44A, 44B, 44C); Speichern eines restlichen Abschnitts des Paketes (50) in einem Speicher (26), mit dem der Speichercontroller (16) gekoppelt ist; Vorsehen durch den Speichercontroller (16) eines ersten Blocksignals für die Paketschnittstellenschaltung (22A, 22B, 22C) und eines zweiten Blocksignals für andere Vorrichtungen, die mit der Zwischenverbindung (24) gekoppelt sind; Sperren der Initiierung von Speichertransaktionen auf der Zwischenverbindung (24) von der Paketschnittstellenschaltung (22A, 22B, 22C) im Ansprechen auf eine Aktivierung des ersten Blocksignals; und Sperren der Initiierung von Speichertransaktionen auf der Zwischenverbindung (24) durch die anderen Vorrichtungen im Ansprechen auf eine Aktivierung der zweiten Blockangabe.

11. Verfahren nach Anspruch 10, wobei der erste Abschnitt mindestens einen Abschnitt eines Headers (52) des Paketes (50) aufweist.

12. Verfahren nach Anspruch 11, welches ferner die Aktivierung des zweiten Blocksignals durch den Speichercontroller (16) im Ansprechen darauf umfasst, dass eine Anzahl einer Mehrzahl von Einträgen in einer Eingabewarteschlange des Speichercontrollers (16), die verfügbar sind, weniger als eine erste Anzahl beträgt.

13. Verfahren nach Anspruch 12, welches ferner die Aktivierung des ersten Blocksignals durch den Speichercontroller (16) im Ansprechen darauf umfasst, dass die Anzahl der Mehrzahl von Einträgen, die verfügbar sind, weniger als eine zweite Anzahl beträgt, wobei die zweite Anzahl weniger als die erste Anzahl beträgt.

14. Computerprogramm mit maschinen- oder computerlesbaren Programmelementen, das zum Konfigurieren eines Computers betreibbar ist, um eines der Verfahren der Ansprüche 10 bis 13 zu implementieren.

15. Computerprogramm-Trägermedium (300),

das dazu konfiguriert ist, ein Computerprogramm nach Anspruch 14 zu tragen.

16. Computerprogramm-Trägermedium **(300)**
nach Anspruch 15, wobei der Träger mindestens eines aus der folgenden Gruppe von Medien aufweist:
ein Signal einschließlich eines optischen, elektronischen oder HF-Trägersignals, oder Magnetplatte oder -band, Festkörperspeicher, eine optische lesbare und/oder beschreibbare Platte, eine Compact Disk (CD) und eine Digital Versatile Disk (DVD).

Es folgen 9 Blatt Zeichnungen

Anhängende Zeichnungen

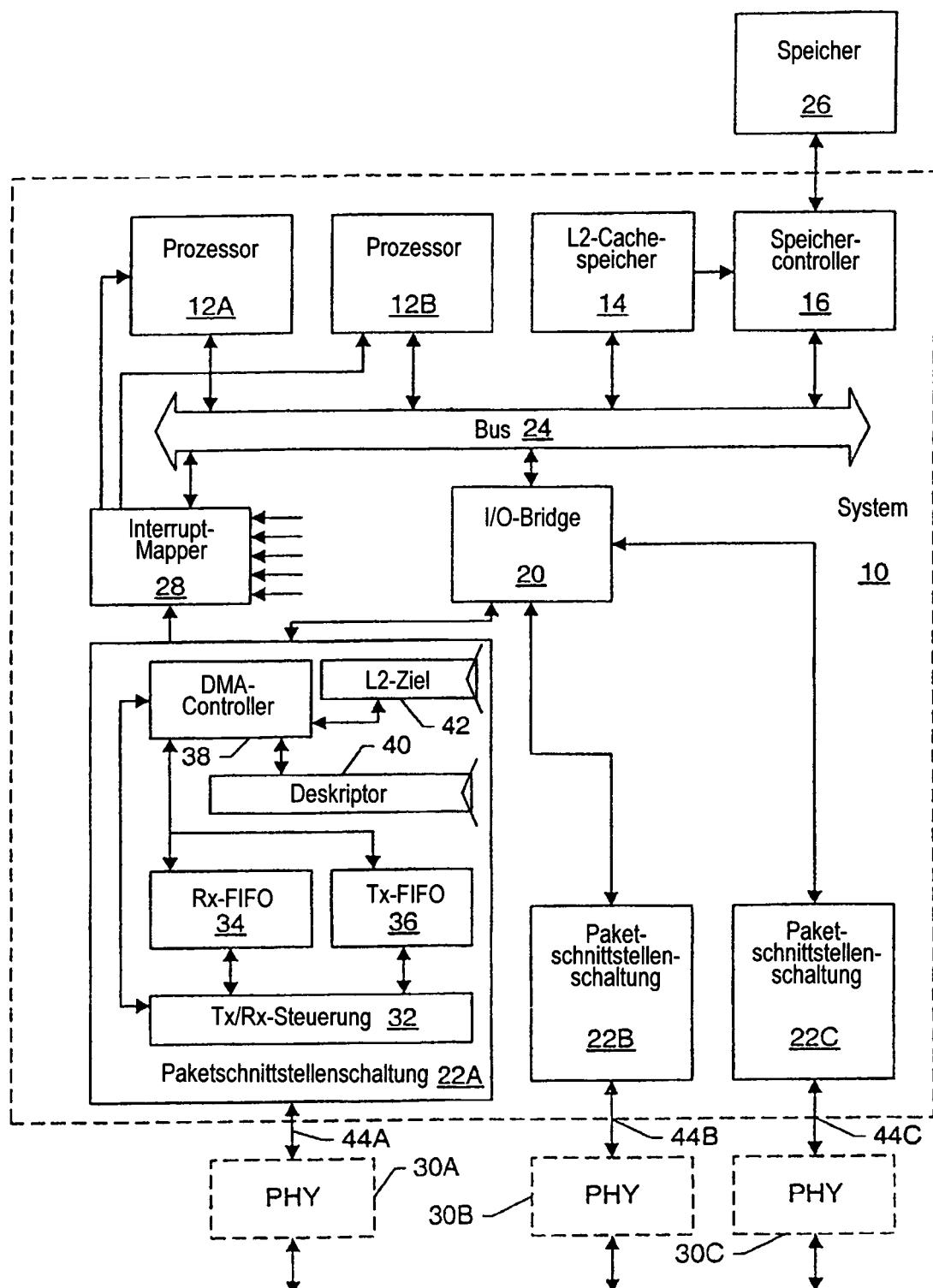


FIG. 1

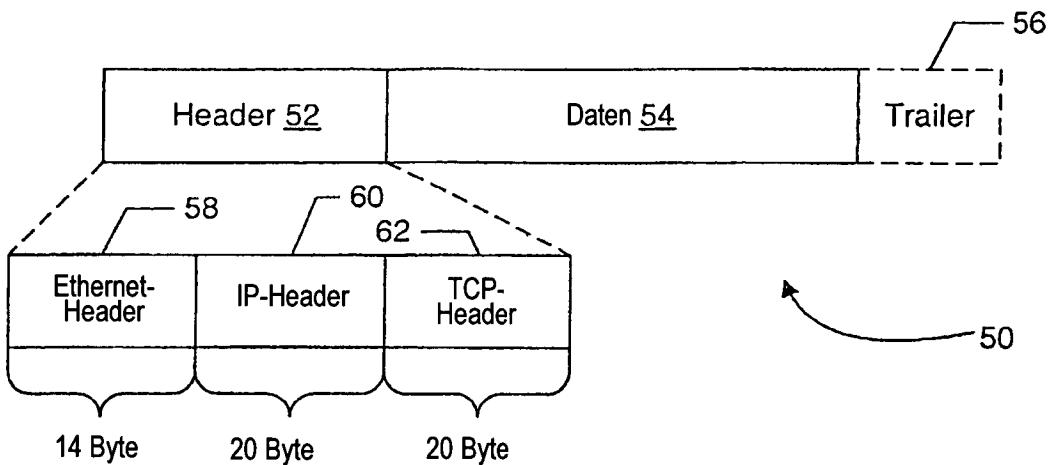


FIG. 2

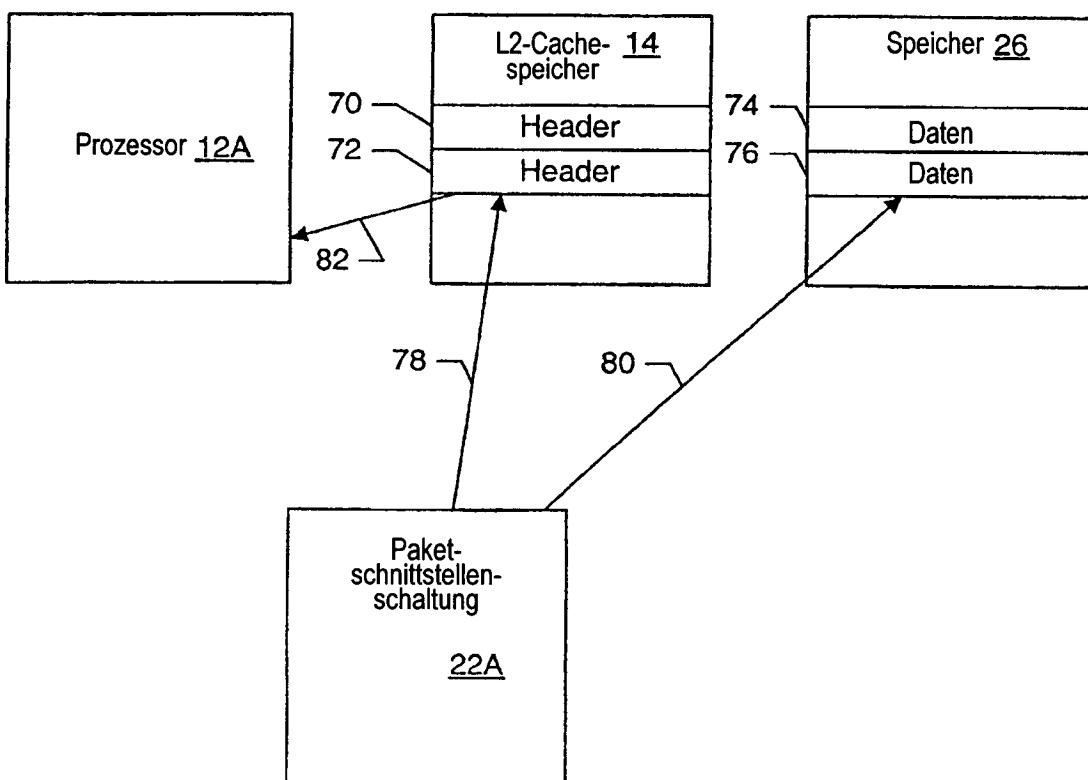


FIG. 3

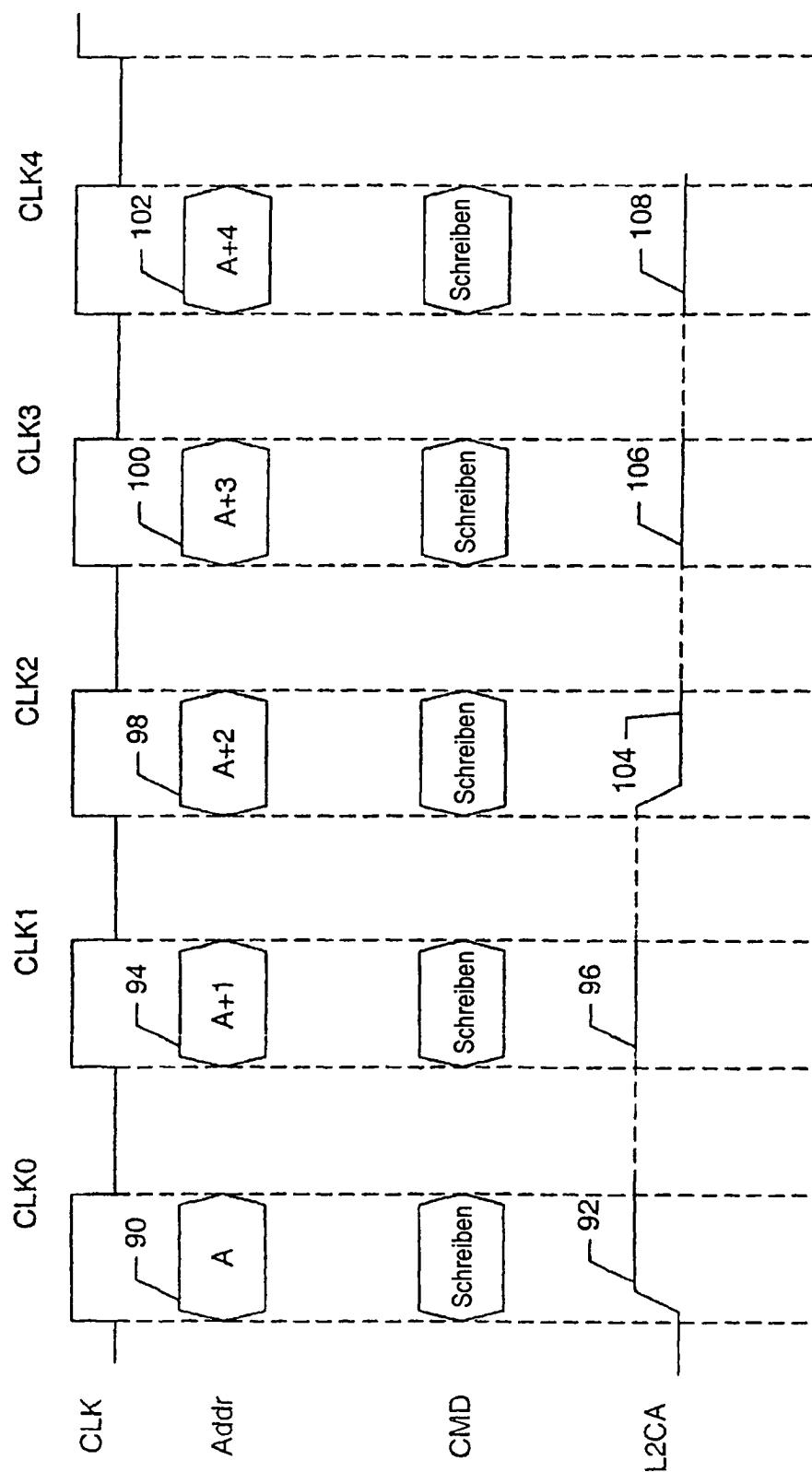


FIG. 4

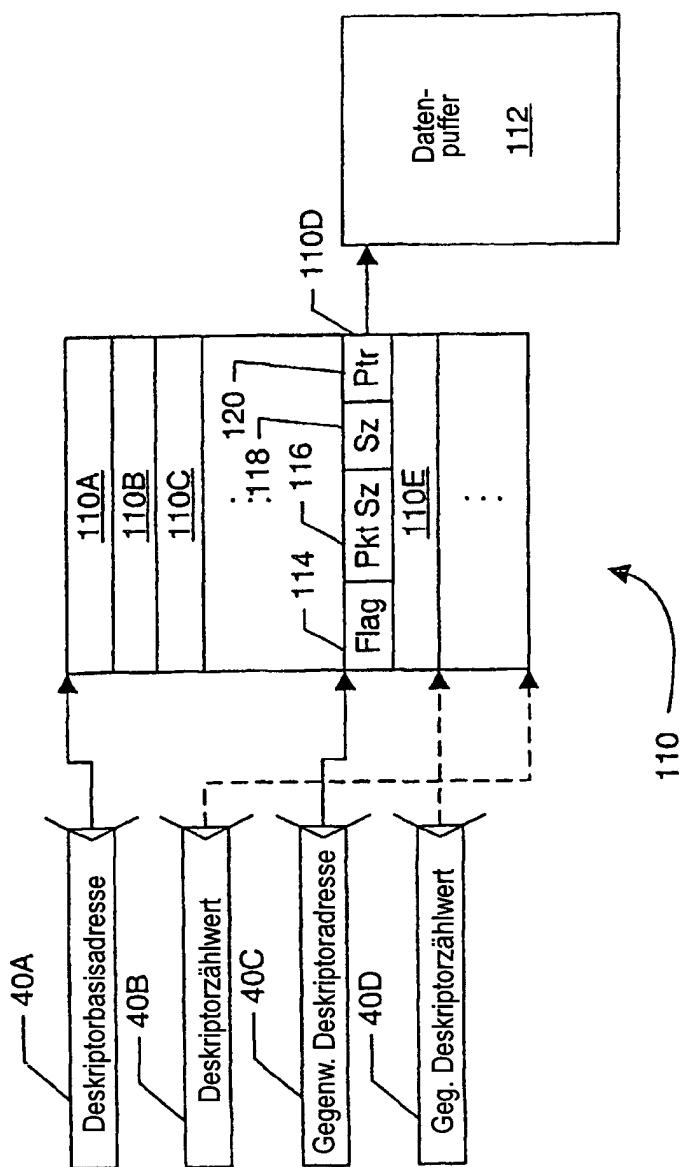


FIG. 5

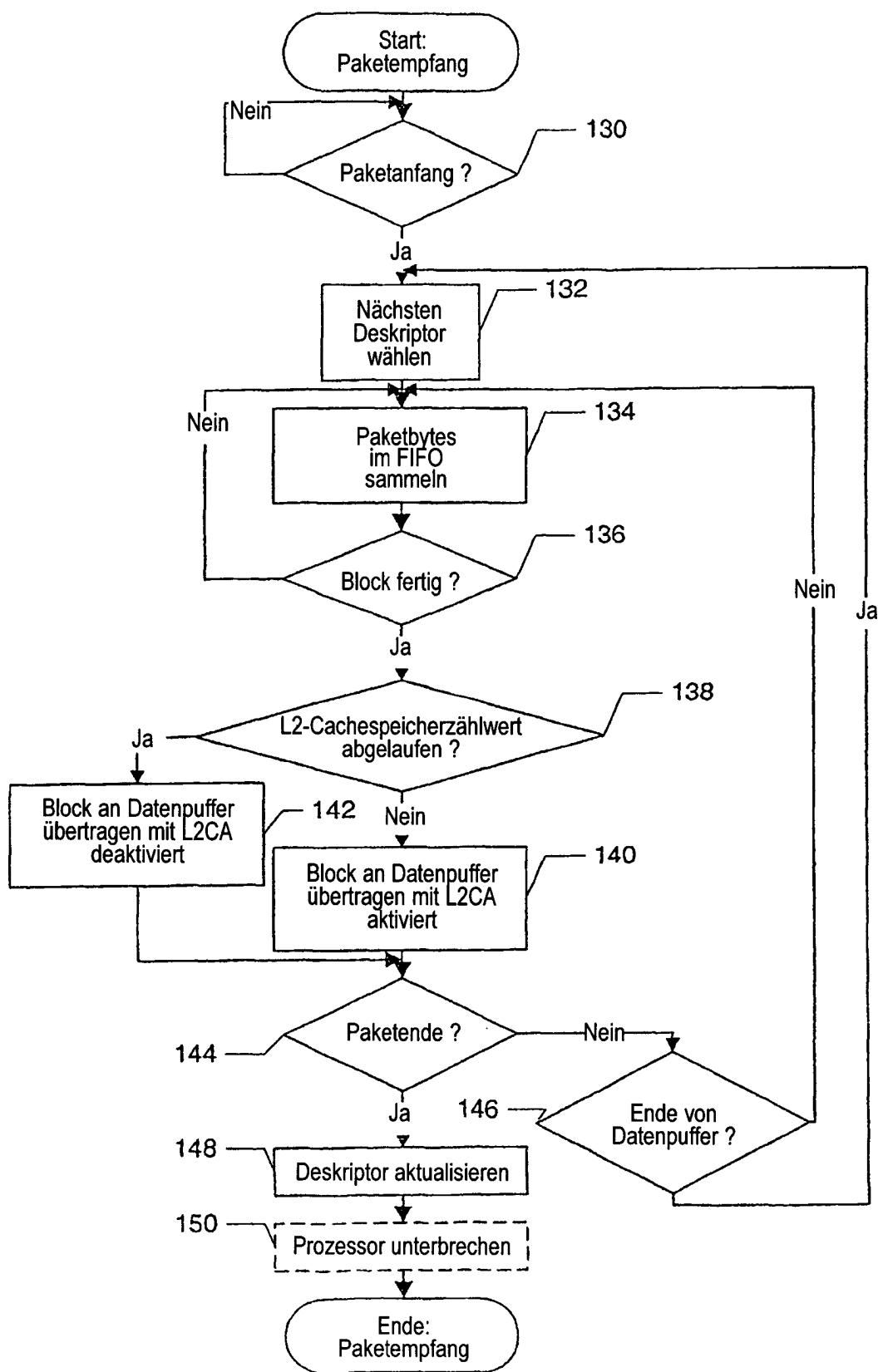


FIG. 6

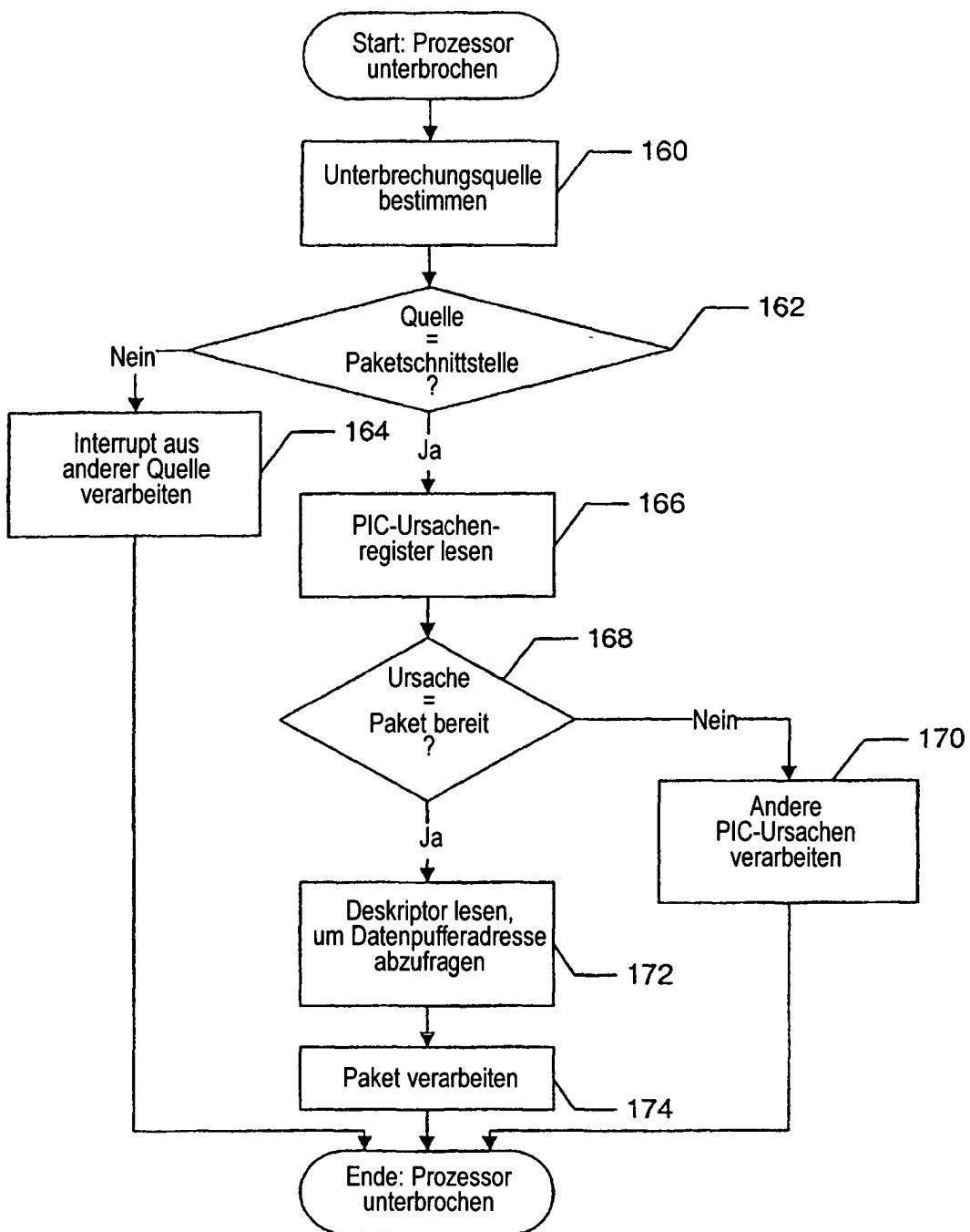


FIG. 7

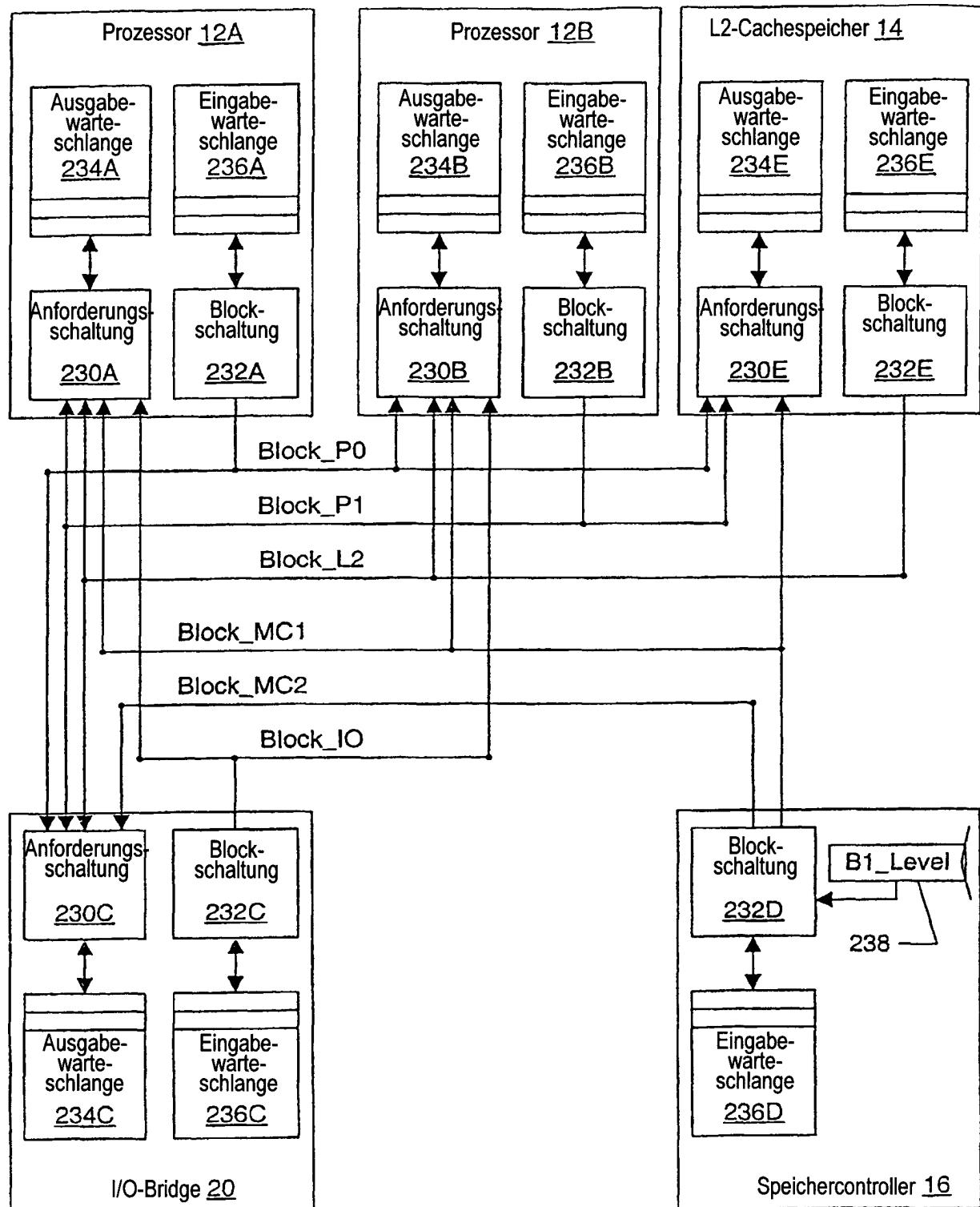


FIG. 8

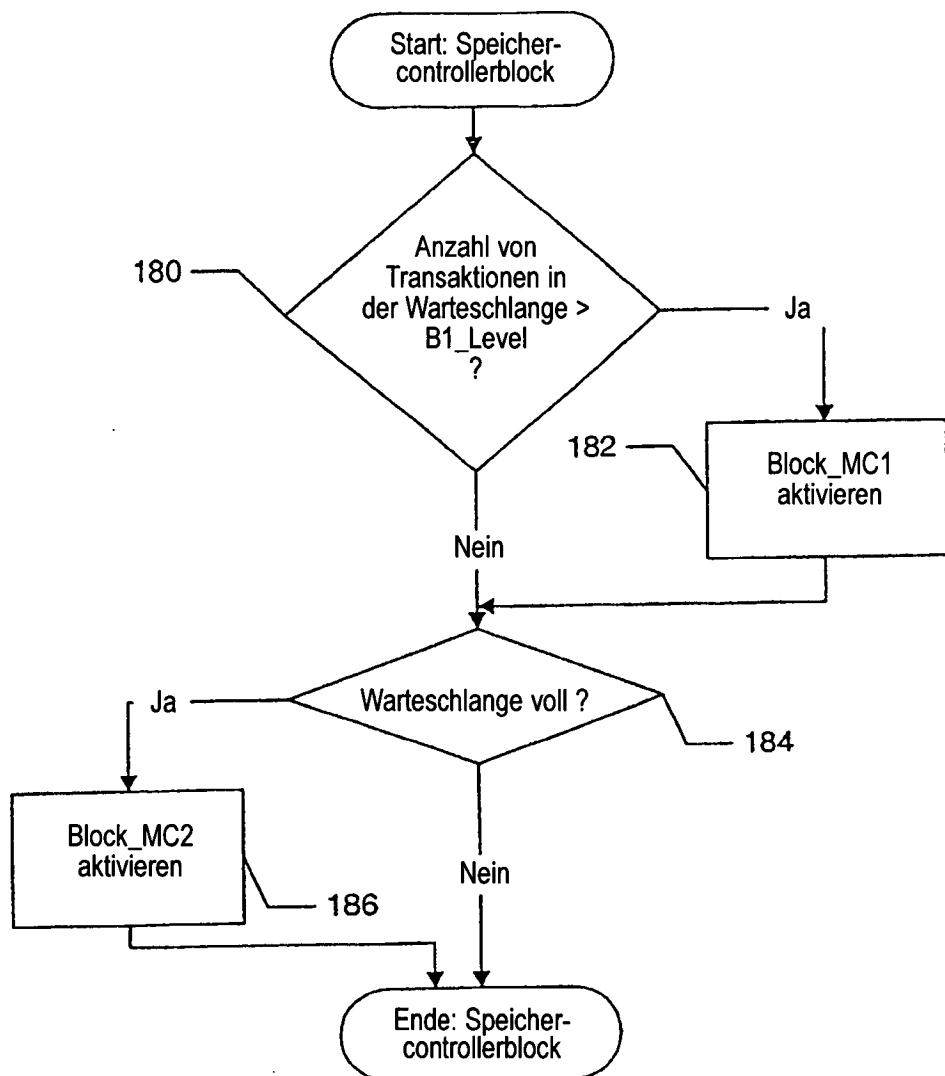


FIG. 9

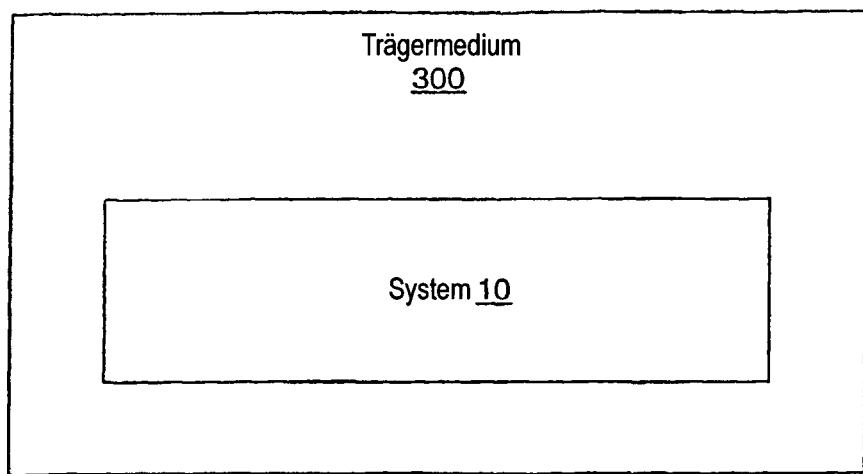


FIG. 10