



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) **DE 600 26 539 T2** 2006.08.17

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 208 443 B1**

(51) Int Cl.<sup>8</sup>: **G06F 13/40** (2006.01)

(21) Deutsches Aktenzeichen: **600 26 539.0**

(86) PCT-Aktenzeichen: **PCT/US00/22833**

(96) Europäisches Aktenzeichen: **00 955 748.9**

(87) PCT-Veröffentlichungs-Nr.: **WO 2001/016772**

(86) PCT-Anmeldetag: **18.08.2000**

(87) Veröffentlichungstag  
der PCT-Anmeldung: **08.03.2001**

(97) Erstveröffentlichung durch das EPA: **29.05.2002**

(97) Veröffentlichungstag  
der Patenterteilung beim EPA: **08.03.2006**

(47) Veröffentlichungstag im Patentblatt: **17.08.2006**

(30) Unionspriorität:  
**385209                      30.08.1999                      US**

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

(73) Patentinhaber:  
**Intel Corp., Santa Clara, Calif., US**

(72) Erfinder:  
**NAYYAR, Raman, Hillsboro, OR 97123, US;**  
**MORAN, R., Douglas, Beaverton, OR 97007, US;**  
**CROSS, W., Leonard, Portland, OR 97229, US**

(74) Vertreter:  
**Patentanwälte Hauck, Graalfs, Wehnert, Döring,**  
**Siemons, Schildberg, 80339 München**

(54) Bezeichnung: **EIN-/AUSGABEADRESSÜBERSETZUNG IN EINER BRÜCKE IN UNMITTELBARER NÄHE EINES  
LOKALEN EIN-/AUSGABEBUSES**

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

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

**Beschreibung**

## STAND DER TECHNIK

## Gebiet der Erfindung

**[0001]** Die vorliegende Erfindung betrifft allgemein die Busprotokollüberbrückung. Im Besonderen betrifft die vorliegende Erfindung die Ein-Ausgabe-Umsetzung (E/A-Umsetzung), wie etwa die neue Abbildung eines durchgängigen Logikadressraums auf einen potenziell unterbrochenen physikalischen Adressraum in einer Brücke.

## Beschreibung des Stands der Technik

**[0002]** Ein-Ausgabe-Peripheriegeräte (E/A-Peripheriegeräte) arbeiten für gewöhnlich effizienter, wenn sie mit einer großen durchgängigen Ansicht des Systemspeichers (Hauptspeichers) präsentiert werden. Dies ermöglicht den Vorrichtungen einen wahlfreien bzw. direkten Zugriff auf einen großen Speicherbereich, der die Daten aufweist, an denen die Vorrichtungen bzw. Geräte interessiert sind. Eine zusammenhängende Betrachtung bzw. ein zusammenhängendes Bild des Speichers ist besonders wichtig für Grafikkarten, die häufig eine ganze Texturabbildung abrufen müssen, um eine Szene bzw. ein Motiv auf einer Anzeigevorrichtung bereitzustellen. Im Sinne der vorliegenden Anmeldung wird der Adressraum, in dem das E/A-Peripheriegerät arbeitet, welcher die Wahrnehmung eines großen durchgängigen bzw. zusammenhängenden Adressraums vermittelt, als ein „virtueller Speicherraum“ bezeichnet.

**[0003]** Da das Peripheriegerät einen anderen Adressraum verwendet als der Hauptspeicher des Host-Computersystems, ist in jedem Fall ein Prozess erforderlich, der als „Adressumsetzung“ bezeichnet wird, um die durch das Peripheriegerät bereitgestellten virtuellen Adressen auf die entsprechenden potenziell unterbrochenen bzw. nicht durchgängigen physikalischen Adressen in dem Hauptspeicher abzubilden.

**[0004]** Ein Beispiel für eine aktuelle Technik für die Adressumsetzung wird nachstehend in Bezug auf die Abbildung aus [Fig. 1](#) beschrieben. Die Abbildung aus [Fig. 1](#) zeigt ein Blockdiagramm auf höherer Ebene eines allgemeinen Computersystems **100**, das die Umsetzung von Grafikadressen in physikalische Adressen veranschaulicht. Das Computersystem **100** weist Verarbeitungseinrichtungen auf, wie zum Beispiel einen oder mehrere Prozessoren **105** und **110**, einen Chipsatz **120**, einen Hauptspeicher **125**, einen Speicherbus **124**, einen Accelerated Graphics Port (AGP) Bus **130** und einen PCI-Bus **140**. In dem vorliegenden Beispiel weist der Hauptspeicher eine Graphics Address Relocation Tabelle (GART) auf, wel-

che Einträge zur Abbildung virtueller Adressen, die von einem Peripheriegerät wie etwa einem Grafikbeschleuniger **135** verwendet werden, in physikalische Adressen in dem Hauptspeicher **125** aufweisen.

**[0005]** Der Chipsatz **120** stellt Überbrückungsfunktionen zwischen dem RGP-Bus **130**, dem PCI-Bus **140** und dem Speicherbus **124** bereit. Als Teil der Überbrückungsfunktion übersetzt der Chipsatz **120** Grafikadressen aus eingehenden Lese- und Schreibtransaktionen (z.B. Lese- und Schreibtransaktionen, die von dem Grafikbeschleuniger **135** stammen) unter Verwendung der GART **126** in entsprechende physikalische Adressen in dem Hauptspeicher **125**. In einem gemäß der Abbildung aus [Fig. 1](#) konfigurierten Computersystem erfordert die Verarbeitung von Grafikdaten, wie etwa von Texturabbildungen (Texture-Maps), mehrere Zugriffe auf den Hauptspeicher **125**. Erstens muss der Chipsatz **120** einen Eintrag aus der GART **126** abrufen, der der in der ausgeführten Transaktion spezifizierten Grafikadresse entspricht. Als nächstens muss der Chipsatz **120**, nachdem er die gewünschte physikalische Adresse bestimmt hat, einen zweiten Zugriff auf den Hauptspeicher **125** ausführen, um die durch die Transaktion spezifizierten Daten zu lesen oder zu schreiben.

**[0006]** Einer der großen Vorteile des AGP ist es, dass er das Videoteilsystem von dem Rest des Computers isoliert, so dass über die E/A-Bandbreite nicht annähernd so viel Konkurrenz existiert wie bei PCI. Aus der vorstehenden Beschreibung sollte es aber auch deutlich werden, dass die aktuelle Art der Behandlung von AGP-Transaktionen auch mit einer Reihe von Ineffizienzen verbunden ist. Erstens ist der Zugriff auf den Hauptspeicher **125**, der für gewöhnlich aus dynamischen Direktzugriffsspeichern (DRAM als englische Abkürzung von Dynamic Random Access Memory) besteht, verhältnismäßig langsam im Vergleich zum Beispiel zu Zugriffen auf statische Direktzugriffsspeicher (SRAM als englische Abkürzung von Static Random Access Memory). Zweitens müssen Zugriffe auf den Hauptspeicher **125** zum Zwecke der Adressumsetzung mit anderen Zugriffsanforderungen konkurrieren, wie etwa mit Anforderungen, die von dem Prozessor **105** oder **110** stammen und über den Systembus **115** empfangen werden, oder mit denen, die an dem PCI-Bus **140** empfangen werden. Folglich werden AGP-Umsetzungen durch konkurrierenden Verkehr gebremst. Wenn die Umsetzungseinheit (nicht abgebildet) schließlich mehrere Arten von Verkehr behandelt bzw. bearbeitet, wie zum Beispiel Systembusverkehr, PCI-Verkehr und AGP-Verkehr, dann gestaltet sich die Situation noch komplexer bzw. komplizierter, da eine Arbitrierungslogik benötigt wird, um die Konkurrenz um die Umsetzungseinheit zwischen den verschiedenartigen ein- und ausgehenden Transaktionen aufzulösen.

**[0007]** Das U.S. Patent US-A-5.335.329 offenbart ein Erweiterungs-Host-Board zur Verbindung eines Erweiterungsgehäuses mit einem SBus. Das Erweiterungs-Board weist eine Speicherverwaltungseinheit (MMU als englische Abkürzung von Memory Management Unit) auf, welche die erforderliche Umsetzung in dem zusätzlichen Level der Adressanzeige vorsieht, die von dem Erweiterungssystem verwendet wird, und mit einem ID- und Programm-RAM, der vorzugsweise einen programmierbaren Adressbereich aufweist, der die folgende neue Zuweisung bzw. Zuordnung zu Überlagerungsabschnitten eines PROM 30 ermöglicht. Der ID- und Programm-RAM wird zum Speichern einer variablen Definition einer ersten Instanz von Variablen sowie zum Speichern eines Längenwertes und eines Versatzwertes der Länge eines Registers oder eines Speicherplatzes des ID-PROM verwendet, auf welches sich der Versatz bezieht. Das ID- und Programm-RAM weist ferner den Inhalt jedes der ID-PROMs der installierten Zusatzfunktionen in dem Erweiterungsgehäuse auf. Die MMU weist ferner einen RAM auf, um Lese-/Schreibeingaben zu empfangen und um als eine Lese-/Schreib-Verweistabelle zum Umsetzen einer physikalischen SBus-Adresse in eine physikalische Adresse für eine Erweiterungsvorrichtung bzw. einen Erweiterungsbaustein in dem Erweiterungsgehäuse zu fungieren.

**[0008]** EP-A-0 392 657 offenbart eine Schnittstellenschaltung, welche ausgesuchte erste Busadressen in entsprechende zweite Busadressen abbildet, so dass bei einem versuchten Zugriff auf eine abgebildete erste Busadresse ein Zugriff auf eine entsprechende Adresse in einem Speicher an dem zweiten Bus erfolgt. Somit kann ein mit einem VMEbus verbundener Computer auf einen mit einem Futurebus verbundenen Speicher zugreifen, indem die Schnittstellenschaltung einen Teil des Adressraums des VMEbus in den Adressraum des Futurebus abbildet und vice versa. In der Schnittstellenschaltung existiert eine F-V-Umsetzungsschaltung mit einem RAM. Der RAM-Speicher weist lokale Adressbits als Adressierungseingabe auf und speichert und liest VMEbus-Adressmodifikationsbits, Adressbits und ein Bit „QUADLET“ aus.

**[0009]** Somit wäre es wünschenswert, eine Technik zur Umsetzung von E/A-Adressen bereitzustellen, welche die Ausführung von E/A-Transaktionen beschleunigt und eine vereinfachte Implementierung der Adressumsetzlogik ermöglicht. Darüber hinaus wäre es wünschenswert, die erforderliche E/A-Adressumsetzung in eine Vorrichtung auszuführen, die von andersartigem Verkehr isoliert ist, wie etwa von Systembusverkehr oder Verkehr von anderen E/A-Vorrichtungen.

**[0010]** Vorgesehen ist gemäß einem ersten Aspekt der vorliegenden Erfindung ein Chipsatz mit einer

Ein-Ausgabe-Erweiterungsbrücke (E/A-Erweiterungsbrücke) gemäß dem gegenständlichen Anspruch 1.

**[0011]** Vorgesehen ist gemäß einem zweiten Aspekt der vorliegenden Erfindung ein Verfahren zum Umsetzen von Ein-Ausgabe-Transaktionen (E/A-Transaktionen) in einer Brücke gemäß dem gegenständlichen Anspruch 11.

**[0012]** Beschrieben werden ein Verfahren und eine Vorrichtung zur Ausführung einer Adressumsetzung in einer Ein-Ausgabe-Erweiterungsbrücke (E/A-Erweiterungsbrücke). Die E/A-Erweiterungsbrücke weist eine erste Schnittstelleneinheit, eine zweite Schnittstelleneinheit und eine Adressumsetzungseinheit auf. Die erste Schnittstelleneinheit ist so konfiguriert, dass sie über einen oder mehrere E/A-Ports mit einer Systemspeicher- und E/A-Steuereinheit gekoppelt werden kann. Die erste Schnittstellenschaltung ermöglicht Datenübertragungen über den einen oder die mehreren E/A-Ports zu oder von dem Hauptspeicher eines Computersystems. Die zweite Schnittstelleneinheit stellt Bussteuersignale und Adressen bereit, um Datenübertragungen über einen Bus zu oder von einem Peripheriegerät zu ermöglichen. Die Adressumsetzungseinheit ist mit der ersten Schnittstelleneinheit und der zweiten Schnittstelleneinheit gekoppelt. Die Adressumsetzungseinheit setzt die Adressen um, die den an der zweiten Schnittstelle empfangenen Transaktionen zugeordnet sind, indem sie auf einen lokalen Speicher zugreift, der physikalische Adressen von Seiten in dem Hauptspeicher des Computersystems enthält.

**[0013]** Weitere Merkmale und Vorteile der vorliegenden Erfindung werden aus den beigefügten Zeichnungen sowie der genauen Beschreibung deutlich.

#### KURZE BESCHREIBUNG DER VERSCHIEDENEN ANSICHTEN DER ZEICHNUNGEN

**[0014]** Die vorliegende Erfindung ist beispielhaft und ohne einzuschränken in den Abbildungen der beigefügten Zeichnungen veranschaulicht, in denen die gleichen Elemente mit den gleichen Bezugsziffern bezeichnet sind. In den Zeichnungen zeigen:

**[0015]** [Fig. 1](#) ein Blockdiagramm, das einen dem Stand der Technik entsprechenden Ansatz für die Umsetzung von Grafikadressen veranschaulicht;

**[0016]** [Fig. 2](#) ein Blockdiagramm, das eine Ein-Ausgabe-Adressumsetzung gemäß einem Ausführungsbeispiel der vorliegenden Erfindung veranschaulicht;

**[0017]** [Fig. 3](#) ein Blockdiagramm, das eine Grafikadressumsetzung gemäß einem Ausführungsbeispiel der vorliegenden Erfindung veranschaulicht;

**[0018]** [Fig. 4](#) ein Blockdiagramm, das einen beispielhaften Chipsatz veranschaulicht, in dem Ausführungsbeispiele der vorliegenden Erfindung implementiert werden können;

**[0019]** [Fig. 5A](#) ein vereinfachtes Blockdiagramm, das eine High-Level-Zwischenverbindung zwischen den funktionalen Einheiten einer Grafikerweiterungsbrücke gemäß einem Ausführungsbeispiel der vorliegenden Erfindung veranschaulicht;

**[0020]** [Fig. 5B](#) ein vereinfachtes Blockdiagramm, das eine genauere Ansicht einer Grafikerweiterungsbrücke gemäß einem Ausführungsbeispiel der vorliegenden Erfindung veranschaulicht

**[0021]** [Fig. 6](#) ein High-Level-Flussdiagramm, das konzeptuell die Transaktionsverarbeitung gemäß einem Ausführungsbeispiel der vorliegenden Erfindung veranschaulicht;

**[0022]** [Fig. 7](#) ein Flussdiagramm der eingehenden PCI-Schreibtransaktionsverarbeitung gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

**[0023]** [Fig. 8](#) ein Flussdiagramm der abgehenden PCI-Schreibtransaktionsverarbeitung gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

**[0024]** [Fig. 9](#) konzeptuell die Umsetzung von Grafikadressen (virtuelle Adressen) in einen physikalischen Adressraum gemäß einem Ausführungsbeispiel der vorliegenden Erfindung;

**[0025]** [Fig. 10A](#) ein beispielhaftes virtuelles Adressenformat, das eingesetzt werden kann, um sich auf Daten zu beziehen, die sich auf Seiten von 4 KByte in dem Hauptspeicher befinden;

**[0026]** [Fig. 10B](#) ein beispielhaftes virtuelles Adressenformat, das eingesetzt werden kann, um sich auf Daten zu beziehen, die sich auf Seiten von 4 MByte in dem Hauptspeicher befinden;

**[0027]** [Fig. 11A](#) ein beispielhaftes GART-Eintragsformat zur Unterstützung der Adressumsetzung in Seiten von 4 KByte in dem Hauptspeicher; und

**[0028]** [Fig. 11B](#) ein beispielhaftes GART-Eintragsformat zur Unterstützung der Adressumsetzung in Seiten von 4 MByte in dem Hauptspeicher

#### GENAUE BESCHREIBUNG DER ERFINDUNG

**[0029]** Beschreiben werden ein Verfahren und eine Vorrichtung zur Ausführung einer Adressumsetzung in einer E/A-Erweiterungsbrücke. Im weiteren Sinne sollen die Ausführungsbeispiele der vorliegenden Erfindung eine Technik zur Umsetzung von E/A-Adressen bereitstellen, wodurch die Ausführung von

E/A-Transaktionen beschleunigt wird, und zwar unter anderem durch das Isolieren der E/A-Transaktionsvorrichtungen, wie z.B. einer E/A-Erweiterungsbrücke von andersartigem Verkehr, wie etwa von Systembustransaktionen. Gemäß einem Merkmal der vorliegenden Erfindung kann die GART in einem lokalen SRAM anstatt über den Speicherbus gespeichert werden. Da die Schnittstelle lokal ist und Zugriff zur Erfassung von GART-Einträgen nicht mit andersartigem Verkehr konkurrieren müssen, können Transaktionen schneller durchgeführt werden und die Implementierung der Transaktionsvorrichtung lässt sich vereinfachen.

**[0030]** In der folgenden Beschreibung sind zu Erläuterungszwecken zahlreiche besondere Einzelheiten ausgeführt, um ein umfassendes Verständnis der vorliegenden Erfindung zu vermitteln. Für den Fachmann auf dem Gebiet ist es jedoch ersichtlich, dass die vorliegende Erfindung auch ohne einige dieser besonderen Einzelheiten ausgeführt werden kann. In anderen Fällen sind allgemein bekannte Strukturen und Vorrichtungen in Blockdiagrammform dargestellt.

**[0031]** Die vorliegende Erfindung umfasst verschiedene Schritte, die nachstehend beschrieben sind. Die Schritte der vorliegenden Erfindung können durch Hardwarekomponenten ausgeführt werden oder in maschinell ausführbaren Anweisungen ausgeführt sein, die eingesetzt werden können, um es zu bewirken, dass Allround-Prozessoren oder besondere Prozessoren oder Logikschaltungen, die mit den Anweisungen programmiert sind, diese Schritte ausführen. Alternativ können die Schritte durch eine Kombination aus Hardware und Software ausgeführt werden.

**[0032]** Es ist wichtig anzuführen, dass die Ausführungsbeispiele der vorliegenden Erfindung zwar in Bezug auf den Accelerated Graphics Port (AGP) und eine Graphics Address Relocation Tabelle (GART) beschrieben sind, auf welche eine Grafikerweiterungsbrücke (GXB als englische Abkürzung von Graphics Expansion Bridge) lokal zugreifen kann, um verschiedene AGP- und PCI-Transaktionen umzusetzen, sind das hierin beschriebene Verfahren und die Vorrichtung gleichermaßen anwendbar auf andere Arten von E/A-Transaktionen und E/A-Erweiterungsbrücken oder andere Arten von Bussen, wobei die Vorrichtung an dem E/A-Bus eine zusammenhängende Ansicht bzw. Betrachtung des Hauptspeichers erfordert. Zum Beispiel gelten die hierin beschriebenen Techniken als nützlich in Verbindung mit Transaktionen, welche einen VESA Local Bus (VLB) und/oder den Peripheral Component Interconnect (PCI) Bus umfassen.

#### Terminologie

**[0033]** Vor der Beschreibung einer Beispielumge-

bung, in der verschiedene Ausführungsbeispiele der vorliegenden Erfindung implementiert werden können, werden einige Begriffe kurz definiert, die in dieser Anwendung verwendet werden.

**[0034]** Der Begriff „Bus“ betrifft allgemein ein physikalisches Kommunikationsmedium oder einen Kanal, über das bzw. den Informationen zwischen zwei oder mehr Vorrichtungen fließen. Während ein Bus mit nur zwei daran angeordneten Vorrichtungen an anderer Stelle teilweise als ein „Port“ bezeichnet wird, umfasst der hierin verwendete Begriff Bus auch diese Ports.

**[0035]** Die Begriffe „Prozessorbus“, „Systembus“ oder „Host-Bus“ beziehen sich für gewöhnlich auf den Bus der höchsten Ebene der Bushierarchie eines Computersystems. Dieser Bus wird für gewöhnlich von dem Chipsatz des Systems eingesetzt, um Informationen zu und von einem oder mehreren Prozessoren des Computersystems zu übertragen.

**[0036]** Der Begriff „Speicherbus“ bezeichnet den Bus, der das Speicherteilsystem mit dem Systemchipsatz und dem bzw. den Prozessor(en) verbindet. Hiermit wird festgestellt, dass der Prozessorbus und der Speicherbus in einigen Computersystemen ein und derselbe Bus sein können.

**[0037]** Ein „lokaler E/A-Bus“ ist für gewöhnlich ein Hochgeschwindigkeits-Ein-Ausgabe-Bus (E/A), der zur Verbindung verschiedenartiger Peripheriegeräte mit dem Speicher, dem Chipsatz des Systems und/oder dem bzw. den Prozessor(en) des Computersystems verwendet wird. Zum Beispiel sind für gewöhnlich Videokarten, Plattenspeichervorrichtungen, Hochgeschwindigkeits-Netzwerkschnittstellen bzw. Netzwerkkarten und andere für die Leistung entscheidende Peripheriegeräte mit einem derartigen Bus gekoppelt. Zu den herkömmlichen E/A-Bussen zählen der VESA Local Bus (VLB), der Peripheral Component Interconnect (PCI) Bus und der Accelerated Graphics Port (AGP) Bus. Im Sinne der vorliegenden Offenbarung gelten ebenfalls ein Bus, der so gestaltet ist, dass er von mehr als zwei Vorrichtungen gemeinsam genutzt wird, und ein Port, der zur Verwendung durch nur zwei Vorrichtungen ausgelegt ist, wie etwa ein E/A-Port oder AGP, beide als Busse.

**[0038]** „Accelerated Graphics Port“ (AGP) ist eine Hochleistungs-Busspezifikation gemäß der Beschreibung in Accelerated Graphics Port Interface Specification, Revision 2.0, 4. Mai 1998 sowie folgender Überarbeitungen, der für eine schnelle, hochwertige Anzeige von 3D-Grafiken und Videobildern ausgelegt ist.

**[0039]** Eine „PCI-Transaktion“ bezeichnet allgemein eine PCI-Transaktion mit AGP-Semantik, die auf dem AGP-Bus ausgegeben wird.

**[0040]** Der hierin verwendete Begriff „Brücke“ bezeichnet allgemein die Logik und Schaltkreisanordnung, die in einem Chipsatz ausgeführt sind, wie zum Beispiel, um zwei Busse miteinander zu verbinden und es der bzw. den Vorrichtung(en) an einem Bus zu ermöglichen, mit der bzw. den Vorrichtung(en) an dem anderen Bus zu kommunizieren und vice versa. Zum Beispiel handelt es sich bei einer typischen Brücke, die in Computersystemen eingesetzt wird, um eine PCI-ISA-Brücke, die für gewöhnlich Bestandteil des Chipsatzes des Systems ist. Der PCI-Bus kann auch eine Brücke zu dem Prozessorbus aufweisen, die als PCI-Erweiterungsbrücke (PXB) bezeichnet wird.

**[0041]** Im Zusammenhang mit der vorliegenden Anmeldung umfasst eine „Transaktion“ eine oder mehrere Anforderungsphase (Adresse) und Datenphase. Eine Transaktion weist für gewöhnlich eine Startadresse, einen Transaktionstyp und eine Anzeige der zu übertragenden Datenmenge auf.

**[0042]** Die Richtung für eine bestimmte Transaktion wird aus der Perspektive des bzw. der Prozessor(en) bestimmt. Somit betrifft eine „eingehende Transaktion“ allgemein eine durch ein Peripheriegerät, wie etwa eine Videokarte, eine Plattenspeichervorrichtung oder eine Hochgeschwindigkeits-Netzwerkschnittstelle, an das Computersystem gerichtete Lese- oder Schreib Anforderung. Im anderen Fall betrifft eine „abgehende Transaktion“ allgemein eine von dem Computersystem an ein Peripheriegerät gerichtete Lese- oder Schreib Anforderung.

#### Ein-Ausgabe-Adressumsetzung

**[0043]** Verschiedene Merkmale der vorliegenden Erfindung gelten als allgemein nützlich in Verbindung mit der Ausführung der Adressumsetzung in einer E/A-Erweiterungsbrücke. Um diesen weiteren Aspekt der vorliegenden Erfindung in Kontext zu bringen, wird nachstehend in Bezug auf die Abbildung aus [Fig. 2](#) eine beispielhafte Architektur für ein Computersystem beschrieben, in der ein Ausführungsbeispiel der vorliegenden Erfindung implementiert werden kann. Die Abbildung aus [Fig. 2](#) veranschaulicht ein Universal-Computersystem **200**, in dem ein Schritt oder mehrere Schritte eines Systems gemäß einem Ausführungsbeispiel der vorliegenden Erfindung ausgeführt werden können, und in dem eine Vorrichtung gemäß einem Ausführungsbeispiel der vorliegenden Erfindung realisiert werden kann. Das Computersystem **200** weist eine Verarbeitungseinrichtung auf, wie etwa einen oder mehrere Prozessoren **205** und **210**, einen Chipsatz **220**, einen Hauptspeicher **225**, einen Speicherbus **224** und einen lokalen E/A-Bus **230**. Gemäß dem vorliegenden Ausführungsbeispiel weist der Chipsatz **220** eine Adressumsetzungstabelle **226** auf. Die Adressumsetzungstabelle enthält Einträge (nicht abgebildet), welche die



Umsetzung von einem Peripheriegerät **235** empfangener virtueller Adressen in physikalische Adressen in dem Hauptspeicher **225** erleichtert. Wie dies nachstehend im Text beschrieben ist, kann der Chipsatz **220** eine oder mehrere integrierte Schaltungen aufweisen, auf welche die Chipsatzfunktionen verteilt werden können, wie zum Beispiel die Systembusarbitrierung, die E/A-Adressumsetzung, etc. In dem vorliegenden Beispiel fungiert der Chipsatz **220** als eine Brücke zwischen dem lokalen E/A-Bus **230**, dem Systembus **215** und dem Speicherbus **224**. Als Bestandteil der Überbrückungsoperationen greift der Chipsatz **220** auf die Umsetzungstabelle **226** für lokale Adressen zu, um den durchgängigen E/A-Adressraum, der von dem Peripheriegerät **235** verwendet wird, in den physikalischen Adressraum des Hauptspeichers **225** abzubilden. Wie dies nachstehend im Text näher beschrieben wird, ist es von Bedeutung, dass durch die Anordnung der Adressumsetzungstabelle **226** lokal im Verhältnis zu dem Chipsatz **220** (z.B. in einem dedizierten auf dem Chip oder außerhalb des Chips vorgesehenen SRAM) eingehende Transaktionen von dem Peripheriegerät **235** effizienter ausgeführt werden können.

#### Grafikadressumsetzung

**[0044]** Verschiedene Merkmale der vorliegenden Erfindung gelten als besonders nützlich für die Ausführung der Adressumsetzung aus einem Grafikadressraum, wie dieser etwa von AGP-kompatiblen Grafikvorrichtungen verwendet wird, in den physikalischen Adressraum des Hauptspeichers **225**. Die Abbildung aus [Fig. 3](#) zeigt ein Blockdiagramm auf hoher Ebene, das ein Universal-Computersystem **300** veranschaulicht, in dem eine verbesserte Umsetzungstechnik für virtuelle Adressen gemäß einem Ausführungsbeispiel der vorliegenden Erfindung implementiert werden kann. Das Computersystem **300** weist Verarbeitungseinrichtungen auf, wie etwa einen oder mehrere Prozessoren **305** und **310**, einen Chipsatz **320**, einen Hauptspeicher **325**, einen Speicherbus **324**, einen Accelerated Graphics Port (AGP) Bus **330** und einen PCI-Bus **340**. In dem dargestellten Ausführungsbeispiel weist der Chipsatz **320** eine Graphics Address Relocation Tabelle (GART) **326** auf. Die GART **326** weist GART-Einträge (nachstehend beschrieben) auf, die Informationen aufweisen, um die Umsetzung der virtuellen Adressen, die von einem Grafikbeschleuniger **335** empfangen werden, in physikalische Adressen des Hauptspeichers **325** zu erreichen.

**[0045]** Gemäß dem vorliegenden Ausführungsbeispiel ist die GART **326** lokal zugänglich für den Chipsatz **320** und nicht über den Speicherbus **324** in dem Hauptspeicher **325** angeordnet. Der Chipsatz **320** stellt eine Überbrückungsfunktion bereit zwischen dem AGP-Bus **320**, dem PCI-Bus **340** und dem Speicherbus **324**. Als Teil der Überbrückungsfunktion

setzt der Chipsatz **320** Grafikadressen aus eingehenden Lese- und Schreibtransaktionen (z.B. Lese- und Schreibtransaktionen, die ihren Ursprung in dem Grafikbeschleuniger **335** haben) unter Verwendung der lokalen GART **326** in entsprechende physikalische Adressen um. Wie dies vorstehend im Text in Bezug auf die Abbildung aus [Fig. 2](#) beschrieben worden ist, können eingehende Transaktionen von Peripheriegeräten, wie etwa von dem Grafikbeschleuniger **335**, durch die Anordnung der GART **326** in dem Chipsatz **320** effizienter verarbeitet und ausgeführt werden. Eine Erfassung von dem lokalen SRAM kann in bedeutender Weise in ungefähr 30 Nanosekunden oder weniger ausgeführt werden anstatt in ungefähr 300 bis 500 Nanosekunden, die für eine Erfassung von Daten aus dem Hauptspeicher **325** (z.B. DRAM) erforderlich sein können.

#### Beispiel für einen Chipsatz

**[0046]** Nach der kurzen Beschreibung von Beispielen von Umgebungen, in denen die vorliegende Erfindung eingesetzt werden kann, wird nachstehend eine Implementierung eines Beispiels für einen Chipsatz in Bezug auf die Abbildung aus [Fig. 4](#) beschrieben. Die Abbildung aus [Fig. 4](#) zeigt ein Beispiel für die Zuordnung von Funktionen an eine Mehrzahl von Einheiten (z.B. Chips) gemäß einem Ausführungsbeispiel der vorliegenden Erfindung. In dem vorliegenden Beispiel weist der Chipsatz **320** eine Systemadressen-Steuereinheit (SAC als englische Abkürzung von System Address Controller) **415**, eine Systemdaten-Steuereinheit (SDC als englische Abkürzung von System Data Controller) **420**, eine PCI-Erweiterungsbrücke (PXB) **425**, eine Grafikerweiterungsbrücke (GXB) **410** und einen statischen RAM (SRAM) **405** auf.

**[0047]** Die SAC **415** ist mit Adress- und Steuerabschnitten (nicht abgebildet) des Systembusses **315** verbunden. Die SAC ist für die Umsetzung und Anweisung von Zugriffen des Systembusses **315** auf den entsprechenden E/A-Port oder Hauptspeicher **325** zuständig. Sie fungiert auch als Routing-Agent für eingehenden Verkehr, wobei dieser zu einem Peer-E/A-Port **412** oder **427**, dem Hauptspeicher **325** oder dem Systembus **315** geleitet wird.

**[0048]** Die SDC **420** verbindet die Datensignale (nicht abgebildet) an dem Systembus **315** und ferner mit dem Hauptspeicher **325** über den Speicherbus **324** und die SAC **415**. Die Daten für alle Übertragungen verlaufen durch die SDC, ob von dem Systembus **315** zu dem Hauptspeicher **325**, von einem der E/A-Ports **412** oder **427** zu dem Speicher oder Peer-to-Peer-E/A-Port **412** oder **427**. Die Verbindung zwischen der SAC **415** und der SDC **420** ermöglicht gleichzeitige Datenübertragungen zu dem Systembus **315** und den E/A-Ports **412** oder **427**, wodurch die für jeden Systemagenten verfügbare Bandbreite

maximiert wird. In alternativen Ausführungsbeispielen kann die Funktionalität der SAC **415** und der SDC **420** in einem einzigen Systemspeicher und der E/A-Steuereinheit **421** kombiniert oder über weitere integrierte Schaltungen verteilt werden.

**[0049]** Die PXB **425** stellt eine Busprotokoll-Überbrückungsfunktion zwischen dem E/A-Port **427** wie etwa einem F16 Bus und dem PCI-Bus **340** bereit. Eine einzelne PXB **425** ist für gewöhnlich mit einem E/A-Port **427** verbunden und erzeugt zwei 32-Bit-PCI-Busse mit 33 MHz. Alternativ kann die PXB **425** einen einzelnen 64-Bit-PCI-Bus unterstützen. Ein Computersystem kann mehrer PXBs aufweisen (für gewöhnlich 1 bis 4).

**[0050]** Die GXB **410** erzeugt eine Brücke zwischen den E/A-Ports **412**, wie zum Beispiel zwei F16 Bussen und dem AGP-Bus **330**, indem eine Busprotokoll-Überbrückungsfunktion zwischen den E/A-Ports **412** und dem AGP-Bus **330** bereitgestellt wird. Auf diese Weise fügt die GXB **410** dem Computersystem einen dedizierten Hochleistungs-Grafikananschluss hinzu. Mehrere GXBs können in einem Computersystem eingesetzt werden, um eine Workstation mit mehreren Grafikports zu erzeugen. Die primäre Funktion der GXB **510** ist es, eine Brücke zu erzeugen zwischen den E/A-Ports **412** und dem AGP-Bus **330**. Kurz gesagt leistet die GXB **410** dies, indem abgehende Transaktionen von den E/A-Ports **412** empfangen, diese intern gespeichert und danach über den AGP-Bus **330** unter Verwendung des richtigen Signalgebungsprotokolls weiter übertragen werden. Im anderen Fall akzeptiert die GXB **410** eingehende Transaktionen, die an dem AGP-Bus **330** ausgegeben werden, wobei diese wiederum intern gespeichert werden, wobei den eingehenden Transaktionen zugeordnete Grafikadressen übersetzt und die übersetzten eingehenden Transaktionen an den bzw. die E/A-Port(s) **412** ausgegeben werden. wie dies nachstehend im Text näher beschrieben ist, führt die GXB **410** eine Adressumsetzung aus, indem auf die GART **326** zugegriffen wird, die in einem in dem Chip integrierten oder in einem SRAM **405** außerhalb des Chips gespeichert ist. Verschiedene Vorteile können durch die Chipsatz-Architektur aus [Fig. 4](#) erreicht werden. Zuerst wird die GXB **410** von dem Systembus **315** und zugeordnetem Systemverkehr isoliert, wodurch eine Konkurrenzquelle für die Ressourcen der GXB **410** eliminiert wird und auf eine zugeordnete Arbitrierungslogik verzichtet werden kann. Da sich die GART **326** in dem SRAM **405** befindet, beeinträchtigt die Adressumsetzungsverarbeitung nicht den Verkehr des Hauptspeichers, und der Speicherbus **324** kann effizienter genutzt werden. Darüber hinaus sind die Zugriffe des lokalen SRAM auf die GART **326** ungefähr zehnmal schneller als wenn die GXB **410** den Weg über den Speicherbus **324** nehmen müsste. Folglich handelt es sich bei dem Endergebnis um eine verbesserte Protokollbrücke zur Be-

arbeitung bzw. Behandlung von Transaktionen an dem AGP-Bus **330**. Die verbesserte Protokollbrücke ist einfacher, schneller und effizienter als die vorstehend beschriebenen aktuellen Überbrückungstechniken. Nachstehend im Text werden Beispiele für interne Partitionen der GXB **410** beschrieben.

**[0051]** In Anbetracht der aktuellen Integrationswelle ist es von Bedeutung, dass es möglich ist, dass eine oder mehrere Komponenten der GXB **410** in einer einzigen integrierten Schaltung (IS) integriert werden können. In alternativen Ausführungsbeispielen können ferner eine oder mehrere der Komponenten der GXB **410** mit einem Prozessor und/oder Speicher auf einer einzigen integrierten Schaltung integriert werden.

Funktionale Aufteilung der Grafikerweiterungsbrücke

**[0052]** Ein Beispiel für die Aufteilung der verschiedenen Funktion der GXB **410** in mehrere Einheiten wird nachstehend in Bezug auf die Abbildung aus [Fig. 5A](#) beschrieben. Die Abbildung aus [Fig. 5A](#) zeigt ein vereinfachtes Blockdiagramm einer Zwischenverbindung auf hoher Ebene unter den verschiedenen funktionalen Einheiten einer Grafikerweiterungsbrücke gemäß einem Ausführungsbeispiel der vorliegenden Erfindung. In dem vorliegenden Beispiel weist die GXB **410** eine AGB-Busschnittstelleneinheit **505**, eine Datenpuffereinheit **510**, eine Transaktionswarteschlangeneinheit **515**, eine GART-Einheit **520** und eine E/A-Port-Schnittstelleneinheit **525** auf. Die AGP-Busschnittstelleneinheit **505** kann auf einem 32-Bit-PCI-Bus mit 66 MHz basieren. In der Praxis und zur leichteren Implementierung kann die AGP-Busschnittstelleneinheit **505** weiter aufgeteilt werden in den Kern-PCI-Bus und die inkrementale Logik, welche die AGP-Funktionen hinzufügt. Auf der AGP-Seite weist die AGP-Busschnittstelleneinheit **505** eine Arbitrierungslogik (nicht abgebildet) zur Verwaltung der Herrschaft zwischen eingehenden und abgehenden Transaktionen des AGP-Busses **330** auf. Darüber hinaus werden Bussteuersignale und Adressen durch die AGP-Busschnittstelleneinheit **505** bereitgestellt, um AGP-Datenübertragungen über den AGP-Bus **330** zu ermöglichen. Die AGP-Busschnittstelleneinheit **505** weist ferner eine interne Transaktionswarteschlange und Datenpuffer (nicht abgebildet) zur Ausführung etwaiger erforderlicher neuer Anordnungen innerhalb des Stroms sowie zum Halten einer Transaktion auf, bis entsprechende Ressourcen zur Behandlung in der Transaktionswarteschlange **515** zur Verfügung stehen.

**[0053]** Wie dies nachstehend im Text näher beschrieben ist, implementiert die Transaktionswarteschlangeneinheit **515** eine oder mehrere interne Transaktionswarteschlangen für eingehende und abgehende Transaktionen. Ferner kann auch Puffer-

raum in der GXB **410** implementiert werden, um verschiedene Datenpufferungsanforderungen zu unterstützen. Zum Beispiel kann die Datenpuffereinheit **510** Datenpuffer zum temporären Speichern von Daten in der GXB **410** aufweisen. Die Datenpuffer können als separate Puffer für jeden Transaktionstyp in jede Richtung oder als ein gemeinsamer Puffer implementiert werden.

**[0054]** Wie dies bereits vorstehend im Text beschrieben worden ist, ist die GART **126** in aktuellen Umsetzungsmechanismen für Grafikadressen in dem Hauptspeicher **125** gespeichert. Bei derartigen Implementierungen wird die Funktionalität der GART somit proximal zu dem Hauptspeicher **125** ausgeführt. Im Gegensatz dazu wird die GART-Funktionalität gemäß dem vorliegenden Ausführungsbeispiel in der GXB **410** durch die GART-Einheit **520** ausgeführt. Gemäß einem Ausführungsbeispiel implementiert die GART-Einheit ein Einspeicherungsmodell und setzt Transaktionen immer dann um, wenn sich eine gültige Transaktion in der bzw. in einer der Transaktionswarteschlange(n) befinden. Diesbezüglich ist die GART-Einheit **520** für die Annahme eingehender Transaktionen von der bzw. den Transaktionswarteschlange(n) sowie für die Umsetzung der Transaktionen je nach Bedarf zuständig. In dem abgebildeten Ausführungsbeispiel ist die GART-Einheit **520** mit einem lokalen, außerhalb des Chips vorgesehenen SRAM **405** gekoppelt, der die GART **326** aufweist. Alternativ kann die GART **326** in einem in dem Chip integrierten SRAM gespeichert werden. Die GART-Einheit **520** ist ferner für die Programmierung des Inhalts des SRAM **405** zuständig. Der SRAM **405** sollte als ungültig oder programmiert werden, bevor etwaige Zugriffe beginnen. Die GART-Einheit **520** kann so ausgerichtet sein, dass der SRAM **405** dynamisch immer dann programmiert wird, wenn zum Beispiel eine neue Anwendung beginnt. Die Programmierung kann durch Ausgabe von Programmierungszugriffsanforderungen von dem Systembus **315** erreicht werden.

**[0055]** Die E/A-Port-Schnittstelleneinheit **525** stellt eine Schnittstelle zu einem oder mehreren E/A-Ports bereit, indem Anforderungen und Datenübertragungen über einen oder mehrere E/A-Ports ermöglicht bzw. freigegeben werden. Zum Beispiel stellt die E/A-Port-Schnittstelleneinheit **525** gemäß einem Ausführungsbeispiel eine Schnittstelle zwischen zwei F16 Bussen bereit. Die E/A-Port-Schnittstelleneinheit **525** entfernt Transaktionen aus der bzw. den eingehenden Transaktionswarteschlange(n), die in der Transaktionswarteschlangeneinheit **515** verwaltet werden, und zwar auf der Basis der Verfügbarkeit und der Strompriorität, und sie überträgt die Transaktionen über den entsprechenden E/A-Port **412**. Wenn es sich bei der Anforderung um einen Schreibvorgang handelt, so werden die Daten, die aus dem entsprechenden Puffer in der Datenpuffereinheit **510** ge-

lesen werden, auch übertragen. Bei abgehenden Transaktionen wird ein Eintrag der entsprechenden abgehenden Transaktionswarteschlange hinzugefügt, die in der Transaktionswarteschlangeneinheit **515** verwaltet wird. Sofern dies erforderlich ist, werden einer abgehenden Transaktion zugeordnete Daten dem entsprechenden Puffer in der Datenpuffereinheit **510** zugeführt.

**[0056]** Im Sinne der vorliegenden Beschreibung wurden die Transaktionen und die damit verbundenen Daten in separate Einheiten aufgeteilt. Allerdings sind auch verschiedene alternative Ausführungsbeispiele denkbar. Zum Beispiel können gemäß einem Ausführungsbeispiel verwandte Datenpuffer und Transaktionen in Gruppen zusammengefasst werden. Somit ist keine der Offenbarungen hierin so auszulegen, als würde die vorliegende Erfindung dadurch auf ein Ausführungsbeispiel auf der Basis einer bestimmten Aufteilung der GXB-Funktionalität beschränkt werden.

**[0057]** Die Abbildung aus [Fig. 5B](#) zeigt ein vereinfachtes Blockdiagramm einer genaueren Ansicht der Grafikerweiterungsbrücke **410** gemäß einem Ausführungsbeispiel der vorliegenden Erfindung. Gemäß dem vorliegenden Beispiel werden die hohen und niedrigen AGP- und PCI-Ströme im Sinne der AGP-Spezifikation als separate Strukturen bis zu der E/A-Schnittstelleneinheit **525** verwaltet. Die Transaktionswarteschlangeneinheit **515** weist eine AGP-Warteschlange **560** mit niedriger Priorität, eine AGP-Warteschlange **565** mit hoher Priorität, eine eingehende PCI-Warteschlange **570**, eine eingehende SAC-Steuerlogik **50**, eine abgehende SAC-Steuerlogik **585** und eine abgehende PCI-Warteschlange **590** auf. Die AGP-Busschnittstelleneinheit **505** empfängt eingehende AGP-Transaktionen von dem AGP-Bus **330** und fügt diese der entsprechenden Warteschlange **560**, **565** oder **570** hinzu. Zum Beispiel werden eingehende PCI-Transaktionen von dem AGP-Bus **330** durch die AGP-Busschnittstelleneinheit **505** zu der eingehenden PCI-Warteschlange **570** geleitet.

**[0058]** Die abgehende SAC-Steuerlogik **585** empfängt abgehende Transaktionen von den E/A-Ports **412** und stellt diese in die abgehende PCI-Warteschlange **590**. Die abgehende PCI-Warteschlange **590** wiederum wird durch die AGP-Busschnittstelleneinheit **505** entladen. Da abgehende Transaktionen keine Adressumsetzung erfordern, ist es wichtig festzustellen, dass sie nicht durch die GART-Einheit **520** verlaufen müssen.

**[0059]** Gemäß dem vorliegenden Ausführungsbeispiel weist die GART-Einheit **520** eine Arbitrierungseinheit **530**, eine Ausführungseinheit **535** und eine Writeback-Einheit **540** auf. Die Arbitrierungseinheit **530** führt zwei Hauptfunktionen aus: (1) die Arbitrierung zwischen den AGP- und PCI-Zugriffen und



SRAM-Zugriffen (Programmierzugriffen), die von der Konfigurationssteuereinheit **595** stammen; und (2) die Erzeugung von SRAM-Adressen (GART-Indizes). Die Arbitrierungsfunktion der Arbitrierungseinheit **530** umfasst die Entscheidung, welche Transaktion der Transaktionen am Kopf der Transaktionswarteschlangen **560**, **565** und **570** als nächstes umgesetzt werden. Die virtuellen Adressen (die in einem Ausführungsbeispiel eine Breite von 40 Bits aufweisen), die den eingehenden Transaktionen in den Transaktionswarteschlangen **560**, **565** und **570** zugeordnet sind, sind zu breit bzw. groß, um direkt dem SRAM **405** zugeführt zu werden. Somit bestimmt die SRAM-Adresserzeugungsfunktion (GART-Indexerzeugungsfunktion) der Arbitrierungseinheit **530** eine SRAM-Adresse auf der Basis der virtuellen Adresse der Transaktion, welche die Arbitrierung erfolgreich abgeschlossen hat. Die Arbitrierungseinheit **530** kann danach der Ausführungseinheit **535** mitteilen, dass sie die entsprechenden Steuersignale für die SRAM-Operation aktivieren soll (z.B. Lesen oder Schreiben). Auf diese Weise kann die Ausführungseinheit **535** direkt einen SRAM-Zyklus ausgeben, ohne die Adresse weiter verarbeiten zu müssen. Alternativ kann die Arbitrierungseinheit **530** an die Ausführungseinheit **535** die SRAM-Adresse bereitstellen, und die Ausführungseinheit **535** kann ein ganzes Paket zusammenstellen und es an den SRAM **405** ausgeben.

**[0060]** Im Idealfall arbeite der SRAM **405** in dem Pipeline-Modus und jeder Takt der Arbitrierungseinheit **530** sollte in der Lage sein, die Adresse für die nächste umzusetzende Adresse bereitzustellen und die Verfügbarkeit der nächsten Transaktion für die Ausführungseinheit **535** zu übertragen.

**[0061]** In einem Ausführungsbeispiel initiiert und steuert die Ausführungseinheit **535** verschiedene SRAM-Operationen, indem das SRAM-Steuersignal zu entsprechend geeigneten Zeitpunkten aktiviert und deaktiviert wird. In einem alternativen Ausführungsbeispiel kann die Ausführungseinheit **535** eine Ausführungsstufe ausführen, welche die Verarbeitung einer eingehenden SRAM-Adresse von der Arbitrierungseinheit **530** und deren Ausführung durch Senden an den SRAM **405** aufweist. Wenn GART-Einträge von der Ausführungseinheit **535** empfangen werden, werden sie zur Ausführung der Adressumsetzung eingesetzt oder entsorgt. Wenn sich die Adresse der eingehenden Transaktion, die dem GART-Eintrag entspricht, innerhalb der GART-Öffnung befindet, so wird der GART-Eintrag zur Ausführung der Adressumsetzung verwendet. Wenn die eingehende Transaktionsadresse jedoch außerhalb des GART-Bereichs liegt (z.B. wenn sie eine physikalische Adresse darstellt), so wird der GART-Eintrag nicht verwendet und es wird keine Umsetzung ausgeführt. In einem alternativen Ausführungsbeispiel kann die Ausführungseinheit **535** alte

Transaktionen speichern oder einen Umsetzungs-Look-Aside-Puffer verwenden, um die Adressumsetzung zu erleichtern. In jedem Fall wird die „umgesetzte bzw. übersetzte Adresse“, wie zum Beispiel die Adresse einer eingehenden Transaktion oder die physikalische Adresse, an die Writeback-Einheit **540** bereitgestellt.

**[0062]** Die Writeback-Einheit **540** empfängt SRAM-Daten und formatiert unter Verwendung von Informationen, die von der Ausführungseinheit **535** und der Konfigurationssteuereinheit **595** bereitgestellt werden, bereitgestellte Daten und leitet diese zu dem richtigen Ziel (z.B. einen Eintrag einer Warteschlange in der Transaktionswarteschlangeneinheit **515**). Gemäß dem dargestellten Ausführungsbeispiel ist die Writeback-Einheit **540** zum Beispiel dafür zuständig, die virtuellen Adressen der eingehenden Transaktionen mit der durch die Ausführungseinheit bereitgestellten und umgesetzten Adresse zu überschreiben.

#### High-Level-Transaktionsverarbeitung

**[0063]** Die Abbildung aus [Fig. 6](#) zeigt ein High-Level-Flussdiagramm, das konzeptuell die Transaktionsverarbeitung gemäß einem Ausführungsbeispiel der vorliegenden Erfindung veranschaulicht. In dem vorliegenden Beispiel wird in dem Schritt **610** eine Transaktion empfangen. In dem Schritt **620** wird danach auf der Basis des Transaktionstyps bestimmt, ob eine Adressumsetzung ausgeführt werden soll oder nicht. Wenn es sich bei der Transaktion um eine eingehende Transaktion handelt (z.B. eine von dem AGP-Bus **330** empfangene Transaktion), so fährt die Transaktionsverarbeitung mit dem Schritt **630** fort. Wenn es sich im anderen Fall bei der Transaktion um eine abgehende Transaktion handelt (z.B. eine von einem oder mehreren E/A-Ports **412** empfangene Transaktion), so fährt die Verarbeitung mit dem Schritt **640** fort. In dem Schritt **630** wird die Verarbeitung der Umsetzung ausgeführt. Aus praktischen Gründen können die GART-Einträge zwar spekulativ aus dem SRAM **410** erfasst werden, allerdings werden nicht alle eingehenden Transaktionen umgesetzt. Zum Beispiel werden eingehende Transaktionen, welche eine virtuelle Adresse außerhalb des AGP-Bereichs spezifizieren (z.B. des GART-Bereichs), werden nicht umgesetzt. In dem Schritt **640** umgehen abgehende Transaktionen die Verarbeitung der Umsetzung. Schließlich wird in dem Schritt **650** die Transaktion an dem entsprechenden Bus ausgeführt (z.B. werden eingehende Transaktionen an einem oder mehreren der E/A-Ports **412** ausgeführt, und abgehende Transaktionen werden an dem AGP-Bus **330** ausgeführt).

Verarbeitung eingehender PCI-Schreibtransaktionen

**[0064]** Gemäß der Abbildung aus [Fig. 5B](#) ist die

GXB **410** vorzugsweise so konfiguriert, dass sie an dem AGP-Bus **330** empfangene AGP-Transaktionen und PCI-Transaktionen so verarbeitet, dass ein Maß der Rückwärtskompatibilität bereitgestellt wird. Der Verarbeitungsablauf durch die GXB **410** ist für jeden der Ströme, den PCI-Strom, den AGP-Strom mit niedriger Priorität und den AGP-Strom mit hoher Priorität ähnlich. Somit wird hierin nur die Verarbeitung für eine beispielhafte PCI-Transaktion beschrieben.

[0065] Die Abbildung aus [Fig. 7](#) ist ein Flussdiagramm, das die Verarbeitung eingehender PCI-Schreibtransaktionen gemäß einem Ausführungsbeispiel der vorliegenden Erfindung veranschaulicht. In dem dargestellten Beispiel läuft in dem Schritt **710** ein PCI-Zyklus an dem AGP-Bus **330** ab, wobei die PCI-Transaktion der GXB **410** zugeführt wird, gefolgt von den Daten, die der PCI-Transaktion zugeordnet sind. In dem Schritt **720** wird die PCI-Transaktion in dem eingehenden PCI-Puffer **570** zwischengespeichert. Nach erfolgreicher Arbitrierung wird die der PCI-Transaktion zugeordnete virtuelle Adresse in dem Hauptspeicher **325** in eine physikalische Adresse umgesetzt, wie dies nachstehend näher beschrieben ist (Schritt **730**). Die virtuelle Adresse der PCI-Transaktion wird mit der umgesetzten Adresse in dem Schritt **740** überschrieben. Aufgrund der spekulativen Beschaffenheit des Abrufs der GART-Einträge, kann, wenn die Umsetzung nicht ausgeführt wird, die umgesetzte Adresse der virtuellen Adresse entsprechen. In jedem Fall wird die umgesetzte PCI-Transaktion in dem Schritt **750** in dem eingehenden SAC-Puffer **580** gepuffert, bis die Transaktion an dem entsprechenden E/A-Port **412** ausgeführt werden kann. Schließlich wird die umgesetzte PCI-Transaktion durch Ausführung eines Schreibzyklus an dem E/A-Port **412** ausgeführt.

Verarbeitung abgehender PCI-Schreibtransaktionen

[0066] Wie dies bereits vorstehend im Text beschrieben worden ist, müssen abgehende Transaktionen nicht von der GXB **410** umgesetzt werden. Um dies weiter zu veranschaulichen wird nachstehend in Bezug auf die Abbildung aus [Fig. 8](#) die Verarbeitung beschrieben, die einer abgehenden PCI-Schreibtransaktion zugeordnet ist. Gemäß diesem Beispiel wird eine Schreibanforderung an einem E/A-Port **412** ausgeführt, und eine PCI-Anforderung wird empfangen und zu der abgehenden SAC-Steuerlogik **585** geleitet (Schritte **810** und **820**). Die der PCI-Anforderung zugeordneten Daten werden zu der Datenpufferreinheit **510** übertragen. Nach der Ermittlung eines Zeitschlitzes der Arbitrierung wird in dem Schritt **840** die abgehende PCI-Schreibtransaktion ausgeführt, indem ein Schreibzyklus an dem AGP-Bus **330** ausgeführt wird. Nachdem die abgehende PCI-Schreibtransaktion erfolgreich vollständig ausgeführt worden ist, kann eine Anzeige eines abgeschlossenen Schreibvorgangs bereitgestellt werden (Schritte **850**

und **860**).

## Übersicht über die Adressumsetzung

[0067] Die Abbildung aus [Fig. 9](#) veranschaulicht konzeptuell eine Umsetzung grafischer (virtueller) Adressen in den physikalischen Adressraum gemäß einem Ausführungsbeispiel der vorliegenden Erfindung. In diesem Beispiel umfasst eine virtuelle Adresse **915** kurz ausgedrückt eine AGP-Adresse **920** und einen Versatz **925**. Die virtuelle Adresse **915** stellt eine Adresse in dem durchgängigen logischen Adressraum dar, der zum Beispiel durch den Grafikbeschleuniger **335** eingesetzt wird. Die AGP-Adresse **920**, ein Teil dieser oder andere aus der AGP-Adresse **920** abgeleitete Informationen werden zum Abrufen eines entsprechenden Eintrags aus der GART **905** verwendet. Die Informationen in dem GART-Eintrag identifizieren eine physikalische Seite in dem Hauptspeicher **325**, welche die der virtuellen Adresse **915** entsprechende physikalische Adresse aufweist. Die durch den GART-Eintrag identifizierte physikalische Seite wird mit dem Versatz **925** von der virtuellen Adresse **915** kombiniert, so dass die Hauptspeicheradresse **930** gebildet wird.

## Unterstützung von Formaten mehrerer Seiten

[0068] Gemäß einem Ausführungsbeispiel unterstützt die GXB **410** zwei verschiedene Seitenformate in dem Hauptspeicher **325**. Die Umsetzung der virtuellen Adresse **915** ist von dem Format der Seite abhängig, auf welcher sich die entsprechende physikalische Adresse befindet. In einem Ausführungsbeispiel wird eine feste Seitengröße (z.B. 4 KByte oder 4 MByte) für alle Seiten innerhalb des GART-Bereichs verwendet, und eine Anzeige der Seitengröße wird in einem Steuerregister der GXB **410** während dem Systemstart gespeichert. Gegenwärtig ist das AGP-Programmierungsmodell für die Verwendung von Seiten von 4 KByte für GART-Einträge ausgelegt. Indem größere Seitengrößen (z.B. 2 MByte, 4 MByte, etc.) zugelassen werden, ist die GXB **410** kompatibel mit Hochleistungs-Computersystemen.

[0069] Während gemäß den vorstehenden Ausführungen alle Seiten in dem GART-Bereich eine bestimmte Größe aufweisen, kann der GART-Bereich in einem alternativen Ausführungsbeispiel Seiten unterschiedlicher Größen aufweisen. In dem vorliegenden Ausführungsbeispiel muss die Adressumsetzung gemäß der nachstehenden Beschreibung dynamisch gesteuert werden.

[0070] Die Abbildung aus [Fig. 10A](#) veranschaulicht ein Beispiel für ein Format virtueller Adressen, das eingesetzt werden kann, um auf Daten zu verweisen, die sich auf 4 KByte Seiten in dem Hauptspeicher befinden. Gemäß diesem Ausführungsbeispiel weist die virtuelle Adresse **915** eine Breite von 40 Bit auf. Die

AGP-Adresse **1020** umfasst 28 Bit und der Versatz **1025** umfasst 12 Bit.

**[0071]** Die Abbildung aus [Fig. 10B](#) veranschaulicht ein Beispiel für ein Format der virtuellen Adressen, das eingesetzt werden kann, um auf Daten zu verweisen, die sich auf 4 MByte Seiten in dem Hauptspeicher befinden. Gemäß diesem Ausführungsbeispiel ist die virtuelle Adresse **915** wiederum 40 Bit breit. Allerdings wird die AGP-Adresse **1021** auf 18 Bit reduziert, während der Versatz **1025** auf 22 Bit erhöht wird. Verschiedene andere Seiten sind jedoch ebenfalls möglich.

**[0072]** Die Abbildung aus [Fig. 11A](#) veranschaulicht ein Beispiel für das Format **1105** für einen GART-Eintrag zur Unterstützung der Adressumsetzung in Seiten von 4 KByte in dem Hauptspeicher. In diesem Beispiel ist der Eintrag **1105** 32 Bit breit und weist ein Paritätsbit **1115**, ein Kohärenzbit **1120**, ein Gültigkeitsbit **1125**, eine neue Seitenadresse **1130** und einen umgekehrten Bereich **1110** auf. Jeder Eintrag deckt eine Seite von 4 KByte ab. Ein Megabyte reicht aus, um 256 K Einträge oder insgesamt 1 GB des umsetzbaren Raums für Seiten von 4 KByte zu speichern.

**[0073]** Das Paritätsbit **1115** deckt den ganzen GART-Eintrag **1105** ab. Es kann durch Hardware erzeugt und so berechnet werden, dass die Anzahl an 1ern insgesamt, einschließlich des Paritätsbits **1115**, eine gerade oder eine ungerade Zahl ergibt.

**[0074]** Das Kohärenzbit **1120** zeigt an, ob der Verkehr von der Grafikkarte mit dem Systembus **315** kohärent sein muss. Die Kohärenz wird nachstehend im Text erörtert.

**[0075]** Das Gültigkeitsbit **1125** zeigt an, ob der SRAM **405** gültige Daten für diesen Eintrag aufweist. Wenn das Gültigkeitsbit **1125** anzeigt, dass die Daten für den Eintrag gültig sind, so kann dieser Eintrag für die Adressumsetzung verwendet werden.

**[0076]** In dem vorliegenden Beispiel ist die neue Seitenadresse **1130** 24 Bit breit. Die neue Seitenadresse **1130** umfasst eine Startadresse einer physikalischen Seite von 4 KByte in dem Hauptspeicher **325**, welche die gewünschte physikalische Adresse aufweist.

**[0077]** Die Abbildung aus [Fig. 11B](#) veranschaulicht ein Beispiel für das Format eines GART-Eintrags zur Unterstützung der Adressumsetzung in Seiten von 4 MByte in dem Hauptspeicher. In dem vorliegenden Beispiel ist der Eintrag **1106** wie vorstehend 32 Bit breit und weist ein Paritätsbit **1116**, ein Kohärenzbit **1121**, ein Gültigkeitsbit **1126**, eine neue Seitenadresse **1131** und umkehrte Bereiche **1111** und **1136** auf. Jeder Eindruck deckt eine Seite von 4 MByte ab.

**[0078]** Wie vorstehend deckt das Paritätsbit **1116** den ganzen GRRT-Eintrag **1106** ab und kann so berechnet werden, dass die gesamte Anzahl der 1er, einschließlich des Paritätsbits **1116**, insgesamt eine gerade oder eine ungerade Zahl ergibt.

**[0079]** Das Kohärenzbit **1121** zeigt an, ob Verkehr von der Grafikkarte mit dem Systembus **315** kohärent sein muss. Die Kohärenz wird nachstehend im Text erörtert.

**[0080]** Das Gültigkeitsbit **1126** zeigt an, ob der SRAM **405** gültige Daten für den betreffenden Eintrag aufweist. Wenn das Gültigkeitsbit **1126** anzeigt, dass die Daten für den Eintrag gültig sind, so kann der Eintrag für die Adressumsetzung verwendet werden.

**[0081]** In dem vorliegenden Beispiel ist die neue Seitenadresse **1131** 14 Bit breit. Die neue Seitenadresse **1131** umfasst eine Startadresse einer physikalischen Seite von 4 MByte in dem Hauptspeicher **325**, welcher die gewünschte physikalische Adresse aufweist.

**[0082]** In alternativen Ausführungsbeispielen kann die relative Positionierung der Bits und Felder sich von den veranschaulichten Beispielen der Abbildungen der [Fig. 11A](#) und [Fig. 11B](#) unterscheiden. In alternativen Ausführungsbeispielen kann das Eintragsformat auch ein Seitenmodusbit aufweisen, um die Zielseitengröße (z.B. 4 KByte oder 4 MByte) anzuzeigen, und um die dynamische Steuerung des Prozesses der Adressumsetzung vorzusehen. Auf diese Weise können gleichzeitig mehr als eine Seitengröße unterstützt werden.

#### Kohärenz

**[0083]** Für den Verkehr von der Grafikkarte (z.B. dem Grafikkbeschleuniger **335**) kann es gewollt sein, dass eine Kohärenz mit dem Systembus **325** gegeben ist oder nicht. Gemäß einem Ausführungsbeispiel kann die Kohärenz Seite für Seite ausgewählt werden. Im Sinne der vorliegenden Offenbarung bedeutet „Kohärenz“, dass die Adressen an dem Systembus **325** erscheinen bzw. auftreten, so dass der eine oder die mehreren Prozessoren **305** und **310** ihre Cache-Speicher für die Adressen einem Snooping unterziehen können. Wenn die Texturabbildung oder andere Bilddaten hinsichtlich einer Schreibkombination (WC) durch den Prozessor markiert sind, so sind die Daten nicht kohärent. Die Adressen an dem Systembus **325**, die einen WC-Puffer des Prozessors treffen, werden keinem Snooping unterzogen. Alternativ kann die Anwendung wissen, dass die Daten in dem Hauptspeicher **325** nicht von dem Prozessor verwendet worden sind (z.B. wenn sie direkt von der Platte stammten), und sie kann es wünschen, dass die Grafikkarte die Daten erfasst, ohne die Bandbreite des Systembusses **325** zu verwenden, so dass es

erzwungen wird, dass deren Zugriff nicht kohärent ist.

**[0084]** Wie dies bereits vorstehend im Text beschrieben worden ist, existiert für alle AGP-Zugriffe, welche innerhalb den AGP-Bereich fallen, ein Bit je GART-Eintrag, das bestimmt, ob die Adresse kohärent ist oder nicht. Für AGP-Zugriffe außerhalb des AGP-Bereichs können Informationen (z.B. ein Kohärenzindikator für ein Bit) aus einem Konfigurationsregister der GXB **410** gelesen werden, um die Kohärenz zu bestimmen. Es ist von Bedeutung, dass die Kohärenz oder die fehlende Kohärenz nur für den Zugriff unter Verwendung des AGP-Protokolls gilt. Zugriffe unter Verwendung des PCI-Protokolls werden immer kohärent ausgeführt, unabhängig davon, ob sie innerhalb der Umsetzungstabelle liegen oder nicht.

**[0085]** In der vorstehenden Beschreibung wurde die Erfindung in Bezug auf besondere Ausführungsbeispiele der Erfindung beschrieben. Es ist jedoch offensichtlich, dass diesbezüglich verschiedene Modifikationen und Abänderungen möglich sind, ohne dabei vom beanspruchten Umfang der vorliegenden Erfindung abzuweichen. Die Beschreibung und die Zeichnungen dienen somit Veranschaulichungszwecken und schränken die Erfindung nicht ein.

### Patentansprüche

1. Chipsatz (**330**) mit einer Ein-Ausgabe-Erweiterungsbrücke (**410**), wobei der Chipsatz folgendes umfasst:  
eine erste Schnittstelleneinheit (**525**), die so konfiguriert ist, dass sie über einen oder mehrere E/A-Ports (**412**) mit einem Hauptspeicher (**325**) und einer E/A-Steuereinheit (**421**) gekoppelt werden kann, wobei die erste Schnittstelleneinheit (**505**) Datenübertragungen über den einen oder die mehreren E/A-Ports (**412**) zu oder von dem Hauptspeicher (**325**) eines Computersystems (**300**) ermöglicht;  
eine zweite Schnittstelleneinheit (**505**), die Bussteuersignale und Adressen zur Freigabe von Datenübertragungen über einen Bus (**330**) zu oder von einem Peripheriegerät (**335**) bereitstellt;  
gekennzeichnet durch:  
eine Adressumsetzeinheit (**520**), die mit der ersten Schnittstelleneinheit (**525**) und der zweiten Schnittstelleneinheit (**505**) gekoppelt ist, wobei die Adressumsetzeinheit den an der zweiten Schnittstelleneinheit (**505**) empfangenen Transaktionen zugeordneten Adressen umsetzt, indem auf einen lokalen Speicher (**405**) zugegriffen wird, der physikalische Adressen (**326**) von Seiten in dem Hauptspeicher (**325**) des Computersystems (**300**) aufweist.

2. Chipsatz nach Anspruch 1, wobei der Bus (**330**) einen Accelerated Graphics Port Bus umfasst.

3. Chipsatz nach Anspruch 2, wobei der lokale

Speicher (**405**) darin eine Graphics Address Relocation Tabelle GART (**326**) aufweist.

4. Chipsatz nach Anspruch 3, wobei der lokale Speicher (**405**) einen in dem Chip integrieren statischen Direktzugriffsspeichers umfasst.

5. Chipsatz nach Anspruch 3, wobei der lokale Speicher (**405**) einen außerhalb des Chips vorgesehenen statischen Direktzugriffsspeichers umfasst.

6. Chipsatz nach Anspruch 3, wobei die GART (**326**) eine Mehrzahl von Einträgen aufweist, die jeweils einer Seite in dem Hauptspeicher (**325**) zugeordnet sind, und die eine Anzeige aufweisen, die identifiziert, ob Adressen auf der Seite auf einem Prozessorbus (**115**) des Computersystems einem Snooping unterzogen werden sollen oder nicht.

7. Chipsatz nach Anspruch 3, wobei die Adressumsetzeinheit (**520**) mehrere Seitengrößen unterstützt, indem Einträge in der GART (**326**) gemäß einem ersten Format oder einem zweiten Format interpretiert werden.

8. Chipsatz nach Anspruch 7, wobei die mehreren Seitengrößen zwei oder mehr Seiten mit 4 KByte, Seiten mit 2 MByte und 4 MByte aufweisen.

9. Chipsatz nach einem der Ansprüche 1 bis 8, wobei der Chipsatz ferner folgendes aufweist:  
eine Steuereinheitseinrichtung (**421**), die mit der Ein-Ausgabe-Erweiterungsbrücke (**410**) gekoppelt und so konfiguriert ist, dass sie mit einem Systembus (**315**) und einem Speicherbus (**324**) des Computersystems (**300**) gekoppelt werden kann, um Datenübertragungen zu und von dem Hauptspeicher (**325**) im Namen der Ein-Ausgabe-Erweiterungsbrücke (**410**) auszuführen.

10. Chipsatz nach Anspruch 9, wobei die Steuereinheitseinrichtung folgendes umfasst:  
eine Systemspeicher- und E/A-Steuereinheit (**421**), die mit der Ein-Ausgabe-Erweiterungsbrücke (**410**) gekoppelt und so konfiguriert ist, dass sie mit einem Systembus (**315**) und einem Speicherbus (**324**) des Computersystems (**300**) gekoppelt werden kann, wobei die Systemspeicher- und E/A-Steuereinheit (**421**) so konfiguriert ist, dass sie Datenübertragungen zu und von dem Hauptspeicher (**325**) im Namen der Ein-Ausgabe-Erweiterungsbrücke (**410**) ausführt.

11. Verfahren zum Umsetzen von Ein-Ausgabe-Transaktionen in einem Chipsatz, der einen lokalen Speicher (**405**) und eine Brücke (**410**) aufweist, wobei das Verfahren die folgenden Schritte umfasst:  
das Empfangen einer E/A-Transaktion auf einem lokalen E/A-Bus (**330**) an der Brücke (**410**);  
das Bestimmen einer Adresse und eines Versatzes, welche der E/A-Transaktion zugeordnet sind;



das Abrufen eines Eintrags aus einer Adressumsetzungstabelle (326) auf der Basis der Adresse, wobei die Adressumsetzungstabelle, die in dem genannten lokalen Speicher (405) gespeichert ist, direkt mit der Brücke (410) gekoppelt ist; und

das Bilden einer physikalischen Adresse, welche eine Position in einem Hauptspeicher (325) eines Computersystems (300) auf der Basis des Eintrags und des Versatzes identifiziert.

12. Verfahren nach Anspruch 11, wobei der lokale E/A-Bus einen Accelerated Graphics Port Bus (330) umfasst, und wobei der Schritt des Empfangens einer E/A-Transaktion auf einem lokalen E/A-Bus durch die Brücke (410) es umfasst, dass die Brücke eine AGP-Transaktion an dem AGP-Bus empfängt.

13. Verfahren nach Anspruch 11 oder 12, wobei der lokale Speicher (405) einen statischen Direktzugriffsspeicher umfasst.

14. Verfahren nach einem der Ansprüche 11 bis 13, wobei das Verfahren ferner die folgenden Schritte umfasst:

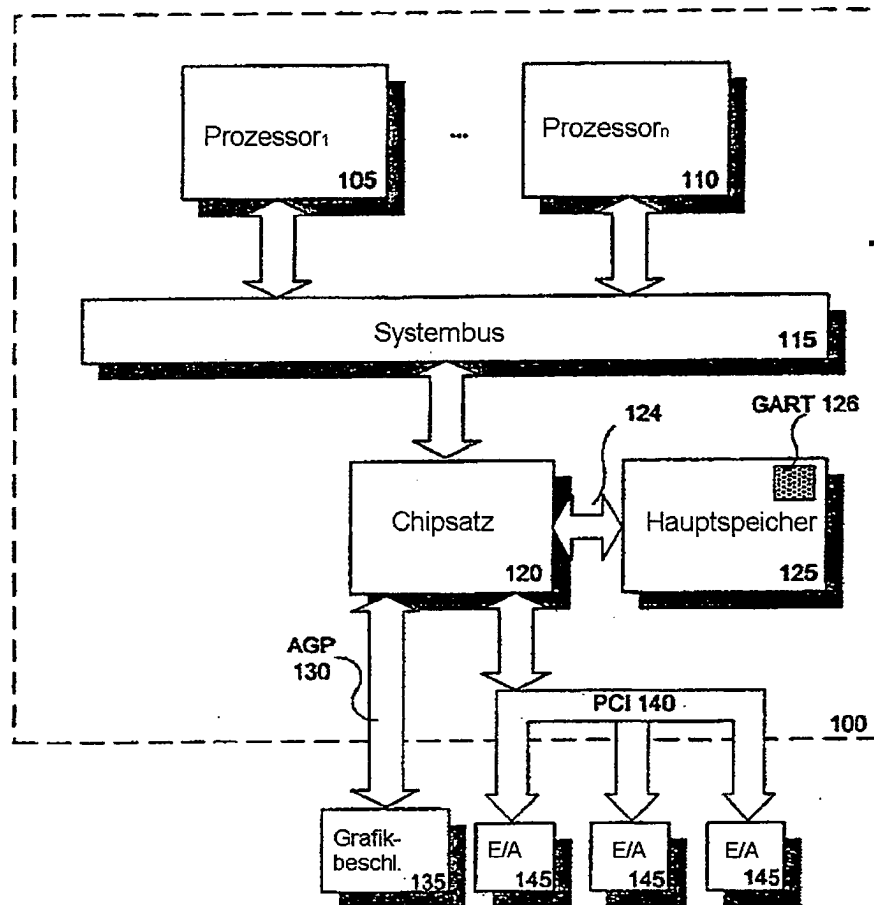
das Bestimmen, ob sich die Position innerhalb einer Seite von 4 KByte oder einer Seite von 4 MByte in dem Hauptspeicher (325) befindet; und

das Interpretieren des Eintrags auf der Basis der Ergebnisse des Bestimmungsschrittes.

15. Verfahren nach einem der Ansprüche 11 bis 14, wobei das Verfahren ferner den Schritt des Bestimmens umfasst, ob der Zugriff auf die physikalische Adresse kohärent oder nicht kohärent erfolgen soll.

Es folgen 10 Blatt Zeichnungen

## Anhängende Zeichnungen

**Figure 1** (Stand der Technik)

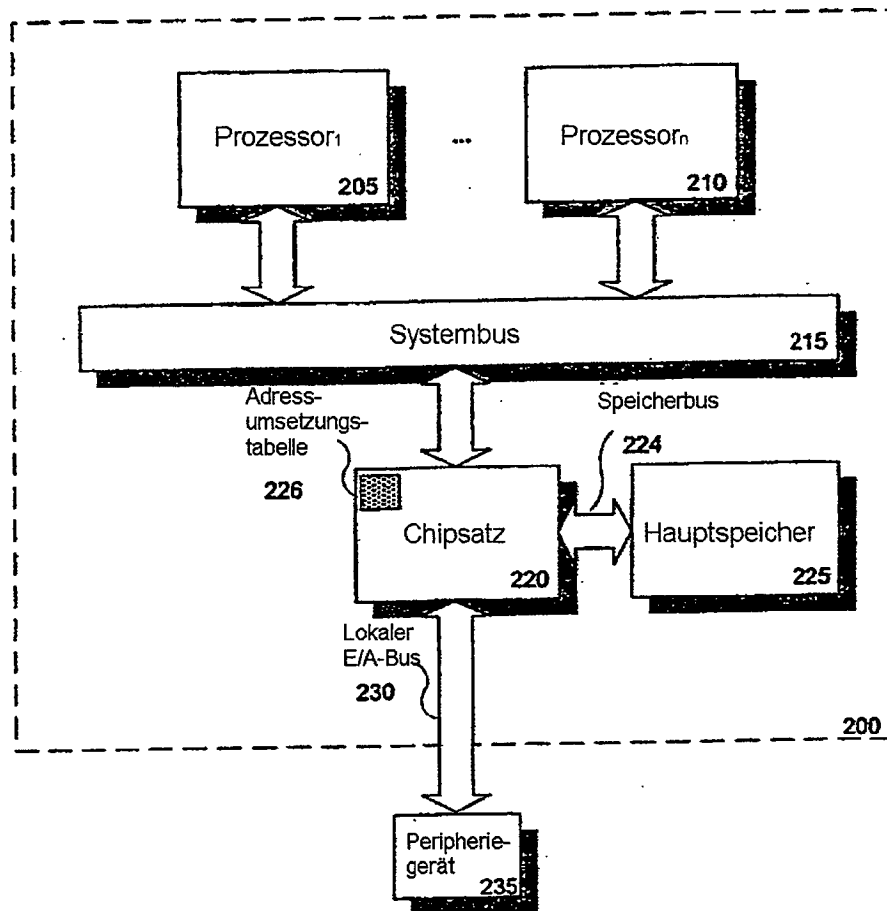


Figure 2

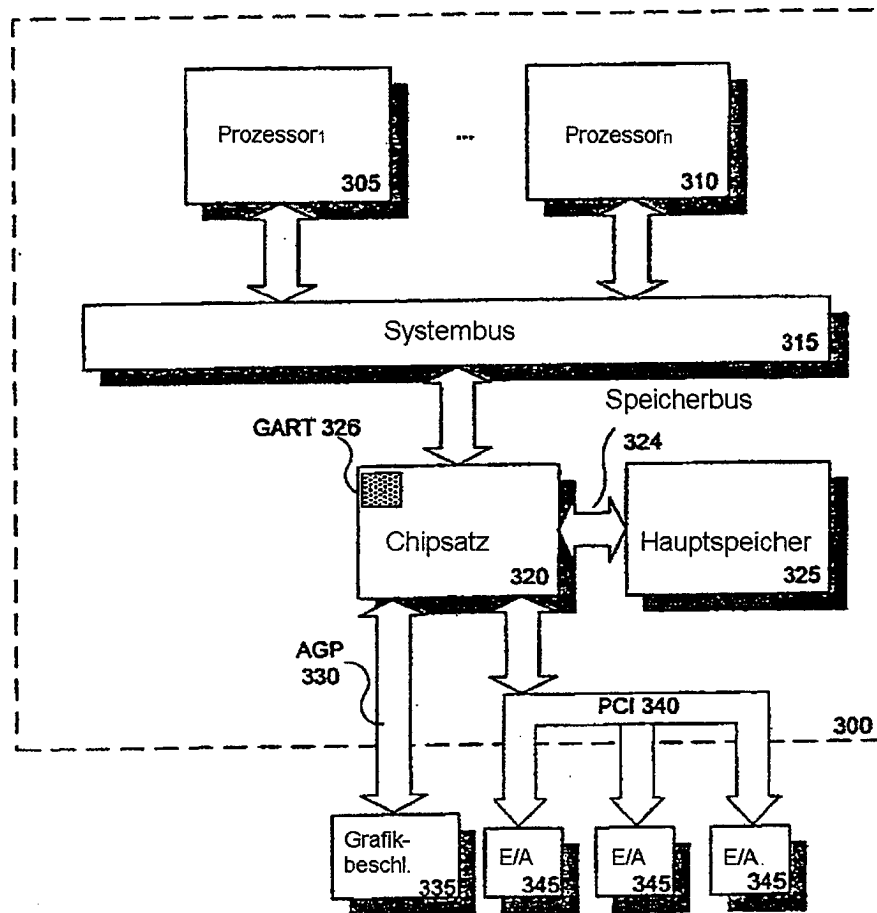


Figure 3



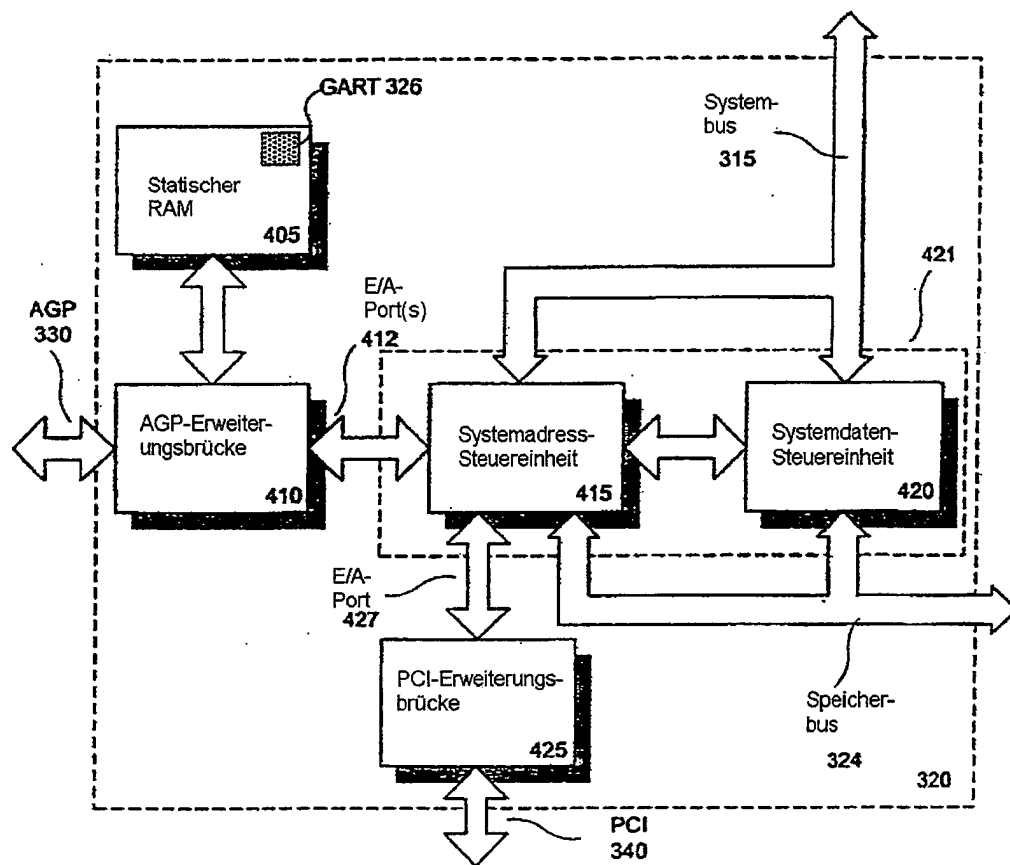


Figure 4

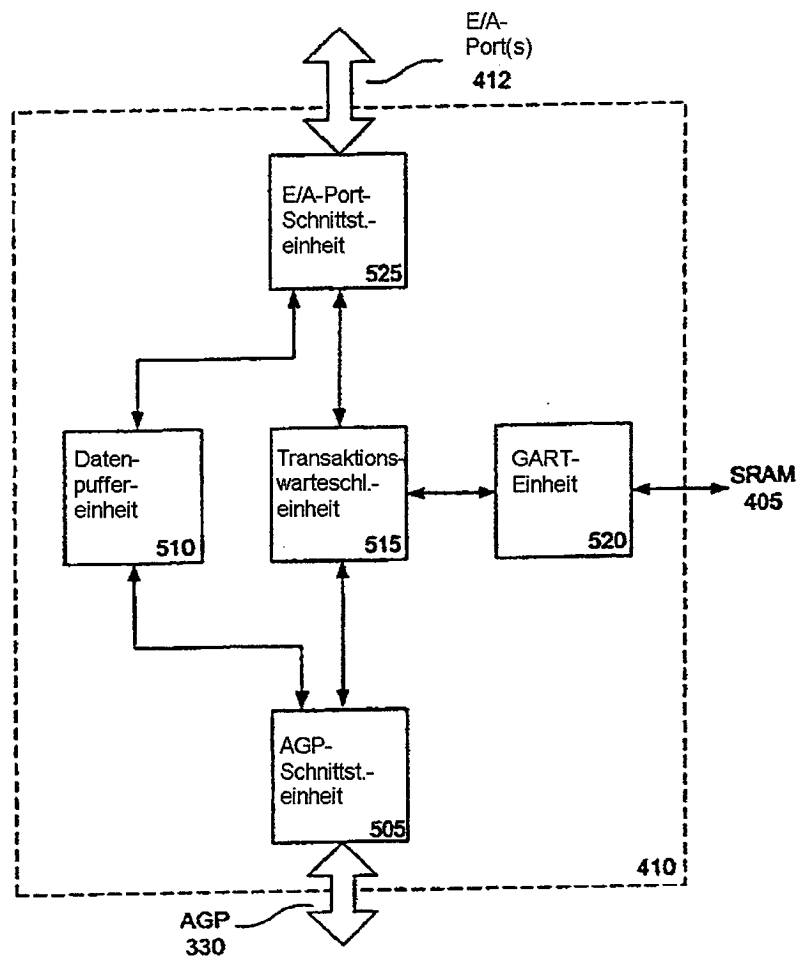


Figure 5A

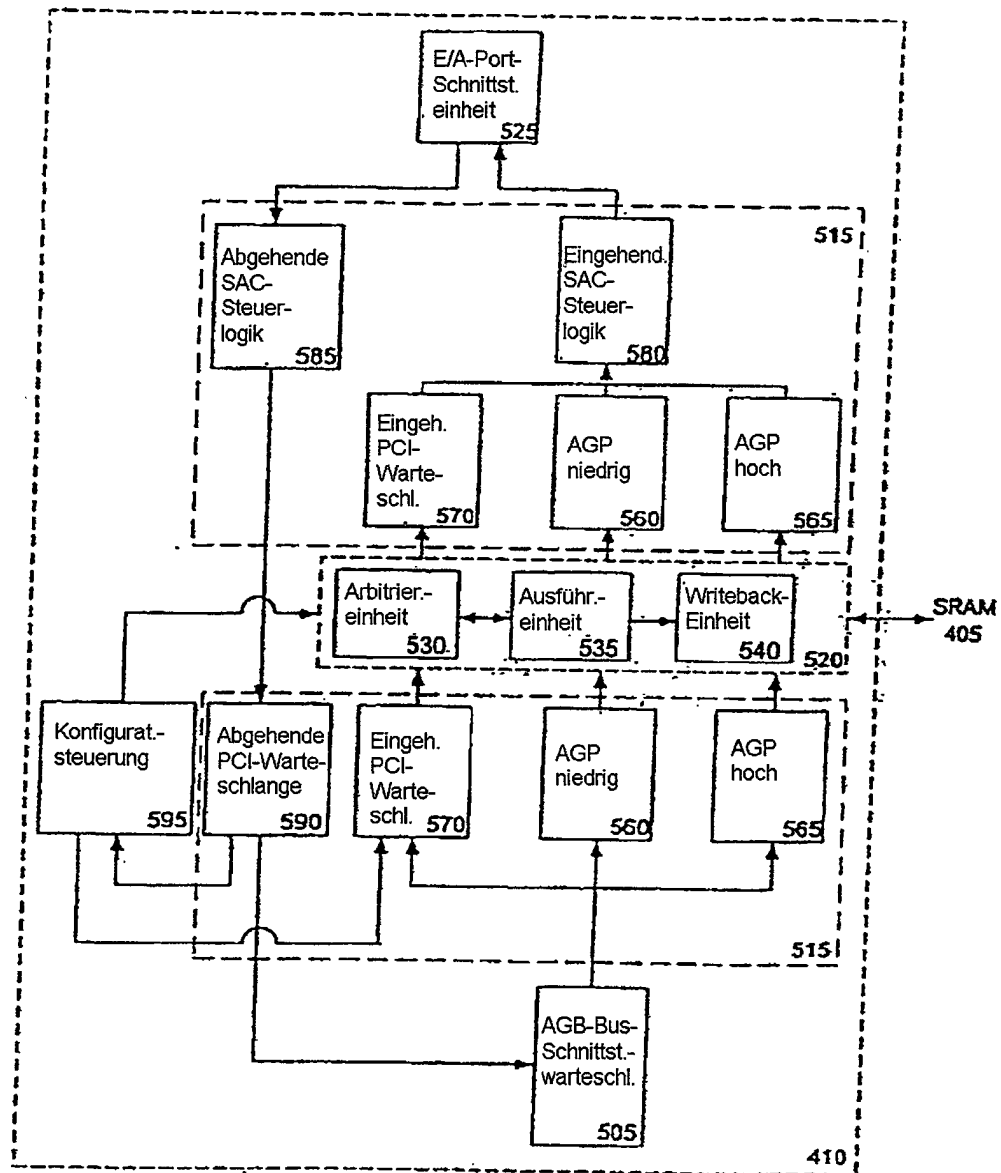
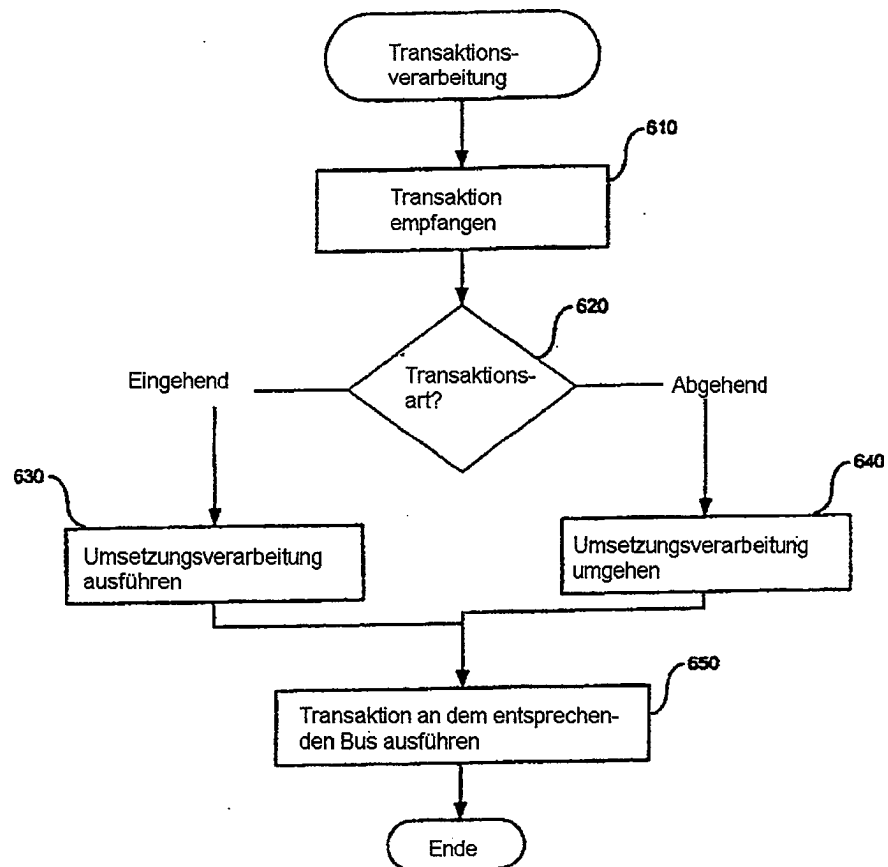


Figure 5B

**Figure 6**



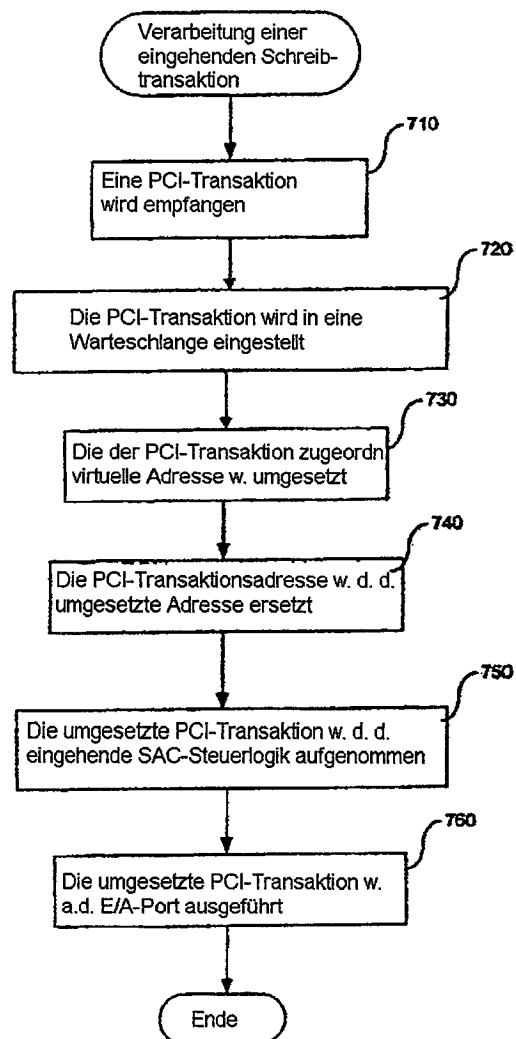
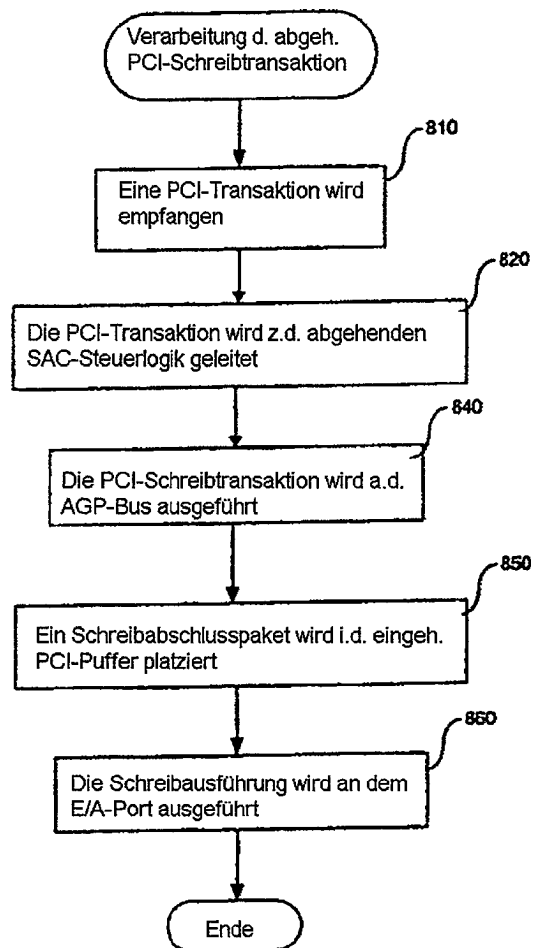


Figure 7



**Figure 8**

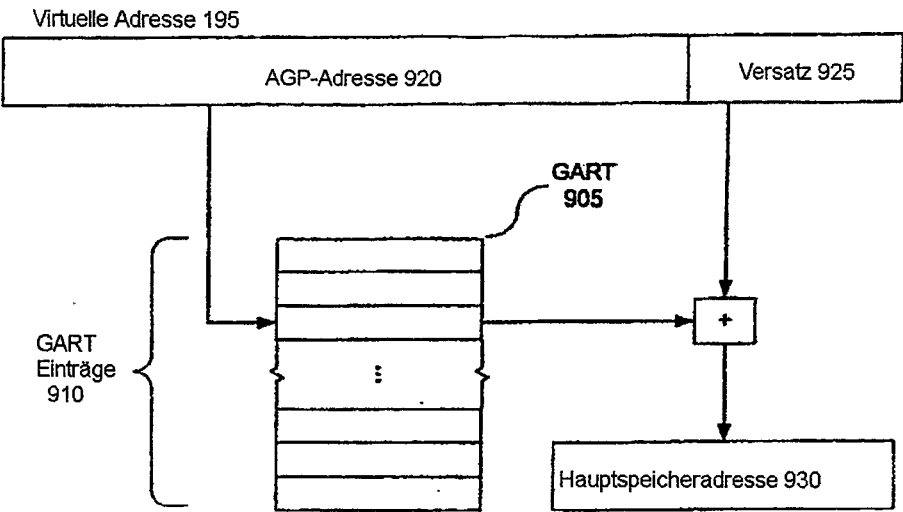


Figure 9

Figure 10A

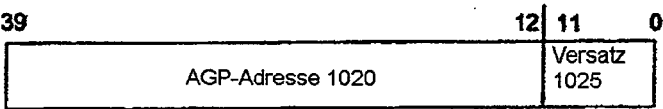


Figure 10B

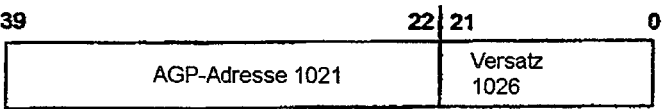


Figure 11A

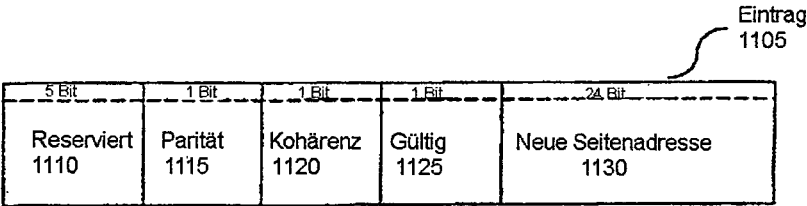


Figure 11B

