



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) DE 699 36 679 T2 2008.04.03

(12)

Übersetzung der europäischen Patentschrift

(97) EP 0 944 001 B1

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

(21) Deutsches Aktenzeichen: **699 36 679.8**

(96) Europäisches Aktenzeichen: **99 302 101.3**

(96) Europäischer Anmeldetag: **18.03.1999**

(97) Erstveröffentlichung durch das EPA: **22.09.1999**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **01.08.2007**

(47) Veröffentlichungstag im Patentblatt: **03.04.2008**

(30) Unionspriorität:
44660 19.03.1998 US

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:
**Hewlett-Packard Development Co., L.P., Houston,
Tex., US**

(72) Erfinder:
**Das Sharma, Debendra, Santa Clara, California
95051, US**

(74) Vertreter:
**Schoppe, Zimmermann, Stöckeler & Zinkler, 82049
Pullach**

(54) Bezeichnung: **Vorrichtung zur Transaktionsleitweglenkung in einer Busbrücke**

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] Diese Anmeldung bezieht sich im Allgemeinen auf Computersysteme und insbesondere auf einen Mechanismus zum Routen von Transaktionen mit Ordnungseinschränkungen. Genauer gesagt bezieht sie sich auf eine Busbrücke, die die Übertragung einer Sequenz von Transaktionen ermöglicht.

[0002] Bei einem verbunden Computernetzwerk kommunizieren die Vorrichtungen durch Transaktionen. Eine Transaktion weist ein oder mehrere Pakete auf und übermittelt einige bedeutende Informationen. Zum Beispiel kann eine CPU in einem Multiprozessornetzwerk auf eine Schnüffel-Anforderung mit einer „Schnüffel-Antwort“-Transaktion antworten. Im Allgemeinen besteht ein Paket aus einer festen Anzahl von Übertragungen oder Zyklen in der Verbindungsstruktur. Unterschiedliche Transaktionen können eine unterschiedliche Anzahl von Paketen erfordern.

[0003] Der Fluss der unterschiedlichen Typen von Transaktionen wird durch Ordnungseinschränkungen gesteuert. Diese Einschränkungen spezifizieren die Ordnungsbeziehung jeder Transaktion im Hinblick auf andere. Ein bestimmter Transaktionstyp kann erforderlich sein, um andere Transaktionstypen zu umgehen, um einen Vorrätsfortschritt zu garantieren, und es kann verboten sein, dass die bestimmten Transaktionstypen einige andere Transaktionstypen umgehen, um eine Ordnung zu garantieren. Üblicherweise werden diese Ordnungsregeln durch eine Ordnungstabelle bereitgestellt, die die Ordnungsbeziehung zwischen jedem Paar aus Transaktionstypen spezifiziert.

[0004] Für jedes Paar von Transaktionstypen P und Q spezifiziert die Ordnungstabelle, ob P Q umgehen sollte, oder ob P Q nicht umgehen sollte, oder ob es egal ist, ob P Q umgeht. Auf ähnliche Weise würde ein anderer Eintrag in der Tabelle spezifizieren, ob Q P umgehen könnte. Ein Beispiel von Ordnungseinschränkungen findet sich in der Ordnungstabelle in der PCI-Bus-Spezifikation, PCI Local Bus Specification, Rev. 2.1, PCI Special Interest Group, Portland, OR, Juni 1995. Diese Ordnungsregeln werden üblicherweise durch Laden aller eingehenden Transaktionen in eine gemeinsame Warteschlange implementiert. Eine umgehbarbare Transaktion wird nachfolgend in eine separate Umgehungswarteschlange geladen (oder ein System aus Warteschlangen), wenn sie sich zu dem Kopf der gemeinsamen Warteschlange bewegt.

[0005] [Fig. 4](#) zeigt eine typische Anordnung **40** gemäß dem Stand der Technik zum Implementieren eines Teilsatzes aus PCI-Transaktionen auf einem PCI zu einer PCI-Brücke **41**, die zwei PCI-Busse verbindet, Bus A **42** und B **43**. Diese Anordnung würde zwei Typen von PCI-Transaktionen handhaben, die Abge-

sendet-Speicher-Schreiben (PMW; posted memory write) und die Verzögert-Lesen-Anforderung (DRR; delayed read request). Eine PMW-Transaktion ist erforderlich, um frühere DRR-Transaktionen zu umgehen, um eine Blockierung zu vermeiden. Eine DRR-Transaktion darf jedoch keine PMW-Transaktionen umgehen, um das Ordnungsmodell durchzusetzen. Transaktionen, die an der PCI-Busbrücke **41** von dem PCI-Bus A **42** ankommen, müssen auf solche Weise in die Warteschlange gestellt werden, dass diese Ordnungseinschränkungen beibehalten werden. Somit verwendet dieser Lösungsansatz zwei Warteschlangen, eine Umgehungswarteschlange **44** ausschließlich für DRR-Transaktionen und eine gemeinsame Warteschlange **45**, in die sowohl PMW- als auch DRR-Transaktionen geladen sind. Sowohl DRR- als auch PMW-Transaktionen werden in die gemeinsame Warteschlange **45** von dem PCI-Bus A gespeichert. Wenn eine DRR-Transaktion den Kopf einer gemeinsamen Warteschlange **45** erreicht, wird sie dann in die Umgehungs-DRR-Warteschlange **44** geladen. Wenn somit jegliche PMWs nach der DRR vorliegen, können sie die DRR umgehen und über den PCI-Bus B **43** ausgesendet werden. Die DRR wird ausgesendet, nachdem alle PMWs ausgesendet wurden.

[0006] Ein Problem bei dieser Anordnung ist, dass, obwohl sie die Ordnungsanforderungen der PCI-Spezifikation erfüllt, sie das Verhalten dadurch verschlechtert, dass sie die verfügbare Bandbreite des PCI-Busses B **43** nicht vollständig ausnutzt. Die Unter-Nutzung des PCI-Busses B entsteht, wenn eine DRR-Transaktion in die Umgehungswarteschlange aus der gemeinsamen Warteschlange geladen wird. Da es eine FIFO-Struktur ist, kann die gemeinsame Warteschlange eine nachfolgende PMW-Transaktion nicht auf den PCI-Bus B entladen, wenn eine DRR-Transaktion an ihrem Kopf in die Umgehungswarteschlange entladen wird. Somit ist die Anordnung gemäß dem Stand der Technik nicht in der Lage, die maximale Bandbreite zu erreichen, die auf dem PCI-Bus B verfügbar ist.

[0007] Ein weiteres Problem bei dieser Anordnung ist, dass die gemeinsame Warteschlange eine große Größe aufweisen muss. Bestimmte Typen großer Transaktionen erfordern einen großen Betrag eines Speicherraums. Idealerweise würden diese großen Transaktionen nicht in die Warteschlange gestellt werden, sondern würden stattdessen direkt über den Bus gesendet werden. Um jedoch die Ordnung beizubehalten, müssen diese Transaktionen in der Lage sein, in die gemeinsame Warteschlange zusammen mit den anderen Transaktionen einzupassen.

[0008] Das Dokument EP 0811927 offenbart ein System, bei dem individuelle Warteschlangen verwendet werden, um entsprechende PMW- und DRR-Transaktionen zu empfangen, die dann durch

eine von oder beide einer Transaktionslaufwarteschlange und einer Transaktionsordnungswarteschlange zu einem zweiten Bus geroutet werden.

[0009] Daher besteht ein Bedarf in der Technik nach einem Mechanismus, der diesen Verhaltensengpass beseitigt, während er trotzdem eine Ordnung beibehält und die erforderliche Größe der gemeinsamen Warteschlange reduziert.

[0010] Gemäß einem Aspekt der vorliegenden Erfindung wird eine Busbrücke gemäß Anspruch 1 geschaffen. Gemäß einem weiteren Aspekt der vorliegenden Erfindung wird ein Verfahren gemäß Anspruch 6 geschaffen. Ein Mechanismus wird verwendet, der die Umgehungswarteschlange (oder System aus Warteschlangen) mit umgehbar Transaktionen an der Quelle belädt, und ohne dieselben durch die gemeinsame Warteschlange passieren zu lassen. Der Mechanismus behält die Ordnung mit einer zusätzlichen Steuerlogik bei. Daher wird die gemeinsame Warteschlange von der Umgehungswarteschlange abgegabelt, d. h. die gemeinsame Warteschlange führt nicht in die Umgehungswarteschlange.

[0011] Zur Zeit des Ladens der Transaktion bestimmt die Steuerlogik, ob die Transaktion umgehbar ist, und wenn ja, platziert sie die Transaktion direkt in die Umgehungswarteschlange. Wenn die Transaktion nicht umgehbar ist, dann lädt die Steuerlogik die Transaktion in die gemeinsame Warteschlange. Der Mechanismus verwendet zusätzliche Felder innerhalb der Warteschlangen, um eine Ordnung zwischen den unterschiedlichen Transaktionen beizubehalten, die in den unterschiedlichen Warteschlangen gespeichert sind. Die Felder bestimmen, ob eine bestimmte Transaktion vor oder nach einer anderen Transaktion kam. Jede Transaktion ist mit einem zusätzlichen Bit gespeichert, das verwendet wird, um die Ordnung beizubehalten. Das zusätzliche Bitfeld liegt sowohl in der gemeinsamen als auch der Umgehungswarteschlange vor, mit einem Feld für jede Transaktion.

[0012] Es ist ein technischer Vorteil der Erfindung, das Verhalten durch Gabeln der gemeinsamen Warteschlange und der Umgehungswarteschlange zu steigern. Es ist ein weiterer technischer Vorteil der Erfindung, die Umgehungswarteschlange direkt von dem Bus zu laden. Es ist ein weiterer technischer Vorteil der Erfindung, dass umgehbar Transaktionen nur einmal geladen werden, was die Verwendung der Ressourcen verbessert, in die diese Warteschlangen zugeführt werden. Es ist ein weiterer technischer Vorteil der Erfindung, dass die Gemeinsame-Warteschlange-Speicherung reduziert werden kann, da sie keine umgehbar Transaktionen aufweist, was somit zu einer potentiellen Verbesserung bei dem Verhalten aufgrund von weniger Flusssteuerung führt.

[0013] Durch die vorangehende Ausführung wurden die Merkmale und die technischen Vorteile der vorliegenden Erfindung relativ umfassend ausgeführt, so dass die nachfolgende detaillierte Beschreibung der Erfindung besser verständlich ist. Zusätzliche Merkmale und Vorteile der Erfindung werden hierin nachfolgend beschrieben, die den Gegenstand der Ansprüche der Erfindung bilden. Fachleute auf dem Gebiet sollten erkennen, dass das offenbare Konzept und das spezifische Ausführungsbeispiel ohne weiteres als eine Basis zum Modifizieren oder Entwerfen anderer Strukturen zum Ausführen derselben Zwecke der vorliegenden Erfindung verwendet werden können. Fachleute auf dem Gebiet sollten ferner erkennen, dass solche entsprechenden Konstruktionen nicht von dem Schutzbereich der Erfindung abweichen, wie er in den beiliegenden Ansprüchen ausgeführt ist.

[0014] Für ein umfassenderes Verständnis der vorliegenden Erfindung und der Vorteile derselben wird nun Bezug auf die nachfolgenden Beschreibungen in Verbindung mit den beiliegenden Zeichnungen genommen, in denen:

[0015] [Fig. 1](#) die erfindungsgemäße Anordnung zum Implementieren von PCI-Transaktionen auf einer PCI-zu-PCI-Brücke zeigt, die zwei PCI-Busse verbindet;

[0016] [Fig. 2A–Fig. 2G](#) eine Sequenz zum Laden von Transaktionen in eine Ein-Bit-Version der Anordnung aus [Fig. 1](#) zeigen;

[0017] [Fig. 3A–Fig. 3N](#) eine Sequenz zum Laden und Entladen von Transaktionen in eine und aus einer Ein-Bit-Version der Anordnung aus [Fig. 1](#) zeigen; und

[0018] [Fig. 4](#) eine Anordnung gemäß dem Stand der Technik zum Implementieren von PCI-Transaktionen auf einer PCI-zu-PCI-Brücke zeigt, die zwei PCI-Busse verbindet.

[0019] [Fig. 1](#) zeigt die erfindungsgemäße Anordnung **10** zum Implementieren von PCI-Transaktionen auf einer PCI-zu-PCI-Brücke **11**, die zwei PCI-Busse **12** und **13** verbindet. Diese Anordnung würde zwei Typen von PCI-Transaktionen handhaben, die Absendet-Speicher-Schreiben (PMW; posted memory write) und die Verzögert-Lesen-Anforderung (DRR; delayed read request). Eine PMW-Transaktion ist erforderlich, um frühere DRR-Transaktionen zu umgehen, um eine Blockierung zu vermeiden. Eine DRR-Transaktion darf jedoch keine PMW-Transaktionen umgehen, um das Ordnungsmodell durchzusetzen. Transaktionen, die an der PCI-Busbrücke **11** von dem PCI-Bus **12** ankommen, müssen auf solche Weise in die Warteschlange gestellt werden, dass diese Ordnungseinschränkungen beibehalten wer-

den. Somit verwendet dieser Lösungsansatz zwei Warteschlangen, eine Umgehungswarteschlange **14** ausschließlich für DRR-Transaktionen und eine gemeinsame Warteschlange **15**, in die PMW-Transaktionen geladen sind. Eine Steuerlogik (nicht gezeigt) würde bestimmen, welche Warteschlange mit einer bestimmten Transaktion beladen wird.

[0020] Es wird darauf hingewiesen, dass die zwei Transaktionen, PMW und DRR, ausschließlich beispielhaft sind. Der erfindungsgemäße Mechanismus kann mit jeder Anzahl von Transaktionstypen mit jedem Protokoll arbeiten. Es wird ferner darauf hingewiesen, dass der Ausdruck „Warteschlange“ eine FIFO-(first-in-first-out; Zuerst-Hinein-Zuerst-Hinaus-) Struktur impliziert.

[0021] [Fig. 1](#) zeigt eine Reihe aus Transaktionen, die an der Brücke **11** ankommen, T0, T1, T2, T3, T4, T5 und T6, in dieser Reihenfolge. T0, T3, T4 und T6 sind DRR-Transaktionen und somit umgehbar und werden in die Umgehungswarteschlange **14** geladen, wie gezeigt ist. T1, T2 und T5 sind PMW-Transaktionen und werden in die gemeinsame Warteschlange **15** geladen, wie gezeigt ist. Der Mechanismus verwendet Kennungen **16**, **17**, um eine Ordnung zwischen den unterschiedlichen Transaktionen in den unterschiedlichen Warteschlangen beizubehalten. Die Ordnung wird innerhalb jeder Warteschlange aufgrund des Eintrittsortes jeder Transaktion beibehalten. In der gemeinsamen Warteschlange **15** ist z. B. T1 am Kopf, während T2 und T5 sequentiell hinter T1 sind. Ähnliches gilt für die Umgehungswarteschlange **14**, in der T0, T3, T4 und T6 in ihren entsprechenden Reihenfolgen gespeichert sind. Somit werden Kennungen **16** und **17** verwendet, um die Ordnung zwischen den Warteschlangen beizubehalten und anzuzeigen, welcher Eintrag am Kopf jeder Warteschlange eine frühere Transaktion ist.

[0022] Wie in [Fig. 1](#) gezeigt ist, ist die Kennung der Eintrittsort der letzten Transaktion in die gemeinsame Warteschlange **15**. Wenn somit T0 in die Umgehungswarteschlange geladen wird, ist die gemeinsame Warteschlange leer und T0 ist mit 0 gekennzeichnet. Als nächstes werden T1 und T2 in die gemeinsame Warteschlange **15** geladen und mit 1 bzw. 2 gekennzeichnet. Wenn T3 in die Umgehungswarteschlange **14** kommt, ist der letzte Eintrag in der gemeinsamen Warteschlange T2 mit einer Kennung von 2 und somit wird T3 mit 3 gekennzeichnet. Auf ähnliche Weise, wenn T4 in die Umgebungswarteschlange **14** kommt, ist der letzte Eintrag in die gemeinsame Warteschlange immer noch T2 mit einer Kennung von 2 und somit wird T4 mit 2 gekennzeichnet. Als nächstes wird T5 in die gemeinsame Warteschlange geladen und mit 3 gekennzeichnet. Wenn somit T6 in die Umgebungswarteschlange **14** kommt, ist der letzte Eintrag in die gemeinsame Warteschlange T5 mit einer Kennung von 3 und somit wird T6 mit

3 gekennzeichnet.

[0023] Die Kennungsnummern zeigen die Anzahl von PMW-Transaktionen an, die ausgesendet werden müssen, bevor versucht werden kann, jegliche bestimmte DRR-Transaktion auszusenden. Zum Beispiel weist T0 eine Kennung von 0 auf, die angeht, dass keine PMW-Transaktion ausgesendet werden muss, bevor versucht werden kann, T0 auszusenden. Die PMW-Transaktionen könnten jedoch ausgesendet werden, bevor T0 versucht wird. T3 und T4 müssen warten, dass zwei PMW-Transaktionen ausgesendet werden, bevor sie ausgesendet werden, während T6 auf drei PMW-Transaktionen warten muss. Es wird darauf hingewiesen, dass alle PMW-Transaktionen entladen werden können, bevor jegliche der DRR-Transaktionen entladen werden, da Schreibvorgänge die Lesevorgänge umgehen können, aber Lesevorgänge die Schreibvorgänge nicht umgehen können. Somit ist die Flusssteuerungsregel, dass ein Eintrag aus der Lese- oder Umgehungswarteschlange ausgesendet wird, wenn seine Kennung niedriger ist als der Kennungswert des Schreibvorgangs, der am Kopf der Schreib- oder gemeinsamen Warteschlange ist, oder wenn die gemeinsame Warteschlange leer ist.

[0024] Daher kann T0 anfänglich aus der Lese-Warteschlange **14** entfernt werden, da seine Kennung geringer ist als der Kennungswert von 1 für die T1-Transaktion am Kopf der Schreib-Warteschlange **15**. Um jedoch den Punkt zu betonen, dass Lesevorgänge umgangen werden können, sei angenommen, dass T1 zuerst entladen wird. Es wird darauf hingewiesen, dass diese Regel sicherstellt, dass ein Lesevorgang immer den letzten Wert für einen bestimmten Eintrag wiedergewinnt, d. h. wenn Lesevorgänge Schreibvorgänge umgangen haben, dann liest der Lesevorgang einen alten Wert. Dann wird T2 als nächstes mit seinem Kennungswert von 2 entladen. Somit hat die Schreibwarteschlange nur einen Eintrag T5 mit einer Kennung von 3. Als nächstes wird T0 aus der Lese-Warteschlange entladen. Die Steuerlogik (nicht gezeigt) vergleicht die Kennung von T0 mit der Kennung der Transaktion am Kopf der Schreib-Warteschlange, die T5 mit einer Kennung von 3 ist. Da die Kennung von T0 geringer ist als die Kennung von T5, kann dann T0 entladen werden. T3 und T4 können ebenfalls entladen werden, da ihre Kennungen ebenfalls geringer sind als die Kennung von T5. Es wird darauf hingewiesen, dass die Steuerlogik ferner die Kennung der letzten bzw. neuesten Schreibtransaktion, die entladen werden soll, mit der Lesetransaktion vergleichen kann, wodurch somit die Steuerlogik bestimmt, ob die Lese-Kennung kleiner oder gleich der Kennung des entladenen Schreibvorgangs ist, wobei in diesem Fall der Lesevorgang entladen werden kann. T6 kann nicht entladen werden, bis T5 entladen ist, da die Kennung von T6 gleich der Kennung von T5 ist. Somit, nachdem T5 entladen ist,

gibt die Steuerlogik dann T6 frei. Es wird darauf hingewiesen, dass die Kennungen die Anzahl von Schreibtransaktionen anzeigen, die entladen werden müssen, bevor eine Lesetransaktionsentladung versucht werden kann.

[0025] Es wird darauf hingewiesen, dass die Kennungen von [Fig. 1](#) weiter zunehmen, da jede Schreibtransaktion, die herein kommt, einen inkrementell höheren Kennungswert erhält. Somit, um die Größe der Kennung einzuschränken, wird eine Ein-Bit-Anordnung verwendet. [Fig. 2A–Fig. 2G](#) zeigen das Laden einer Ein-Bit-Version **20** der Anordnung aus [Fig. 1](#). Die Sequenz aus Transaktionen ist dieselbe wie in [Fig. 1](#). Die Kennungen aus [Fig. 1](#) werden nun als Schlüssel und Verriegelungen bezeichnet. Das Schlüsselbit **21** ist der Schreib- oder gemeinsamen Warteschlange **15** zugeordnet, und das Verriegelungsbit **22** ist der Lese- oder Umgehungs-Warteschlange **14** zugeordnet. Somit weist jede Transaktion ein zugeordnetes Kennungsbit auf, entweder eine Verriegelung oder einen Schlüssel, abhängig davon, in welcher Warteschlange die Transaktion gespeichert ist. Ein verriegeltes Bit oder „1“ gekennzeichnet auf einer Lesetransaktion verhindert, dass die Lesetransaktion entladen wird, bis eine Schreibtransaktion mit einem Schlüsselbit entladen wurde. Wie z. B. in [Fig. 2G](#) gezeigt ist, weist T0 kein Verriegelungsbit auf, und somit kann T0 jederzeit entladen werden. T1 weist kein Schlüsselbit auf und somit sind T1 keine Lesetransaktionen sequentiell gefolgt. T2 weist ein Schlüsselbit auf und T3 weist ein Verriegelungsbit auf, da Lesen T3 Schreiben T2 folgte. Somit kann T3 nicht entladen werden, bis T2 entladen wurde. Es wird darauf hingewiesen, dass jedes Mal, wenn ein Lesenvorgang verriegelt ist, ein Schlüssel in einen entsprechenden Schreibvorgang eingefügt ist. Lesen T4 ist nicht verriegelt, da T4 Lesen T3 folgt, das durch T2 verriegelt ist, wodurch T4 nicht entladen werden kann, bis T2 entladen ist. T5 weist ein Schlüsselbit auf und T6 weist ein Verriegelungsbit auf, da Lesen T6 Schreiben T5 folgte. Somit kann T6 nicht entladen werden, bis T5 entladen wurde. Daher ist die Flusssteuerungsregel, dass jedes Mal, wenn ein Schalten von der Schreib- oder gemeinsamen Warteschlange zu der Lesen- oder Umgehungs-Warteschlange ausgeführt wird, die Transaktionen vor und nach dem Schalten verschlüsselt bzw. verriegelt werden. Es wird darauf hingewiesen, dass dabei davon ausgegangen wird, dass ein Eintrag in die Schreib- oder gemeinsame Warteschlange zur Zeit des Schaltens vorliegt. Ansonsten werden Nullen in die Etikettenwerte geladen.

[0026] In [Fig. 2A](#) wird eine Transaktion T0 in die Lesen-Warteschlange **14** geladen. Da die Schreib-Warteschlange leer ist und kein Schalten ausgeführt wurde, ist die Verriegelungsbitkennung 0. In [Fig. 2B](#) wird eine Transaktion T1 in die Schreib-Warteschlange **15** geladen. Da kein Schalten

von Schreiben zu Lesen ausgeführt wurde, ist die Schlüsselbitkennung 0. In [Fig. 2C](#) wird eine Transaktion T2 in die Schreiben-Warteschlange **15** geladen. Da kein Schalten ausgeführt wurde, ist die Schlüsselbitkennung 0. In [Fig. 2D](#) wird eine Transaktion T3 in die Lesen-Warteschlange **14** geladen. Da ein Schalten von Lesen zu Schreiben ausgeführt wurde, während sich ein Eintrag in der Schreiben-Warteschlange befindet, wird das Schlüsselbit von T2 zu 1 verändert, und das Verriegelungsbit von T3 wird mit 1 gekennzeichnet. Somit verhindert die Verriegelung an T3, dass T3 entladen wird, bevor T2 entladen wird. In [Fig. 2E](#) wird eine Transaktion T4 in die Lesen-Warteschlange **14** geladen. Da kein Schalten von der Schreib-Warteschlange in die Lesen-Warteschlange ausgeführt wurde, ist das Verriegelungsbit von T4 mit 0 gekennzeichnet. Es wird darauf hingewiesen, dass die Verriegelung an T3 verhindert, dass T4 entladen wird, bevor T2 entladen wird. In [Fig. 2F](#) wird eine Transaktion T5 in die Schreib-Warteschlange **15** geladen. Da kein Schalten ausgeführt wurde, ist das Schlüsselbit von T5 mit 0 gekennzeichnet. In [Fig. 2G](#) wird eine Transaktion T6 in die Lesen-Warteschlange **14** geladen. Da ein Schalten ausgeführt wurde, während ein Eintrag in der Schreib-Warteschlange vorhanden ist, wurde das Schlüsselbit von T5 auf 1 verändert und das Verriegelungsbit von T6 mit 1 gekennzeichnet. Somit verhindert die Verriegelung an T6, dass T6 entladen wird, bevor T5 entladen wird.

[0027] [Fig. 3A–Fig. 3N](#) zeigen das Laden einer Ein-Bit-Version **30** der Anordnung aus [Fig. 1](#), umfassen aber Entlade-Operationen vermischt mit den La-de-Operationen. Die Entladeentscheidungen werden durch die Steuerlogik (nicht gezeigt) ausgeführt und hängen von externen Faktoren ab, wie z. B. Warteschlangenraum an dem Zielort für die Transaktionen. Die Sequenz aus Transaktionen ist dieselbe wie in [Fig. 1](#) und 2. [Fig. 3A–Fig. 3N](#) umfassen ein Schlüsselregister **31**, das einen Zählerwert der Schlüssel beibehält, die mit den Schreibtransaktionen entladen wurden. Das Schlüsselregister wird inkrementiert, wenn Schreibtransaktionen mit Schlüsseln entladen werden, und wird dekrementiert, wenn Lesetransaktionen mit Verriegelungen entladen werden. Die Anordnung aus [Fig. 3](#) arbeitet gemäß den folgenden Regeln; dies bildet die Regeln, die bei jedem System verwendet werden können:

Regel 1: Ein Schlüssel wird freigegeben, wenn eine Transaktion in der gemeinsamen Warteschlange entladen wird, deren Schlüssel auf 1 gesetzt ist. Das Schlüsselregister oder der Zähler wird um 1 inkrementiert.

Regel 2: Eine Transaktion aus der Umgehungs-warteschlange wird als bereit zum Entladen betrachtet, wenn sie nicht verriegelt ist, d. h. Verriegelung = 0, oder wenn das Schlüsselregister größer als 0 ist.

Regel 3: Ein Eintrag aus der Umgehungs-warte-

schlange wird nur entladen, wenn er bereit ist, wie in Regel 2 definiert ist.

Regel 4: Wenn ein verriegelter Eintrag aus der Umgehungswarteschlange entladen wird, wird das Schlüsselregister um 1 dekrementiert.

Regel 5: Das Schlüsselregister wird während des Einschaltvorgangs auf 0 gesetzt.

[0028] Wie in [Fig. 3A](#) gezeigt ist, wird eine Transaktion T0 in die Lesen-Warteschlange **14** geladen. Da die Schreiben-Warteschlange leer ist und kein Schalten ausgeführt wurde, ist die Verriegelungsbekennung 0. Da keine Schlüssel oder Verriegelungen entladen wurden, bleibt das Schlüsselregister **31** auf 0, dem initialisierten Wert. In [Fig. 3B](#) wird eine Transaktion T1 in die Schreiben-Warteschlange **15** geladen. Da kein Schalten ausgeführt wurde, ist die Schlüsselbitkennung 0. In [Fig. 3C](#) wird eine Transaktion T2 in die Schreiben-Warteschlange **15** geladen. Da kein Schalten ausgeführt wurde, ist die Schlüsselbitkennung 0. In [Fig. 3D](#) wird die Transaktion T1 entladen. Da T1 nicht verschlüsselt war, bleibt das Schlüsselregister **31** auf 0. In [Fig. 3E](#) wird die Transaktion T3 in die Lesen-Warteschlange **14** geladen. Da ein Schalten ausgeführt wurde von dem Laden in die Schreiben-Warteschlange zu der Lesen-Warteschlange, während ein Eintrag in der Schreiben-Warteschlange vorliegt, wurde das Schlüsselbit von T2 zu 1 verändert, und das Verriegelungsbit von T3 mit 1 gekennzeichnet. Somit verhindert die Verriegelung an T3, dass T3 entladen wird, bevor T2 entladen wird.

[0029] In [Fig. 3F](#) wird eine Transaktion T4 in die Lesen-Warteschlange **14** geladen. Da kein Schalten von der Schreiben-Warteschlange in die Lesen-Warteschlange ausgeführt wurde, wird das Verriegelungsbit von T4 mit 0 gekennzeichnet. Es wird darauf hingewiesen, dass die Verriegelung an T3 verhindert, dass T4 entladen wird, bevor T2 entladen wird. In [Fig. 3G](#) wird Transaktion T2 entladen. Da T2 verschlüsselt war, wird das Schlüsselregister **31** auf 1 inkrementiert. In [Fig. 3H](#) wird die Transaktion T0 entladen. Da T0 nicht verriegelt war, wird das Schlüsselregister **31** bei 1 beibehalten. In [Fig. 3I](#) wird die Transaktion T3 entladen. Da T3 verriegelt war, wird das Schlüsselregister **31** um 1 auf 0 dekrementiert. In [Fig. 3J](#) wird die Transaktion T5 in die Schreiben-Warteschlange **15** geladen. Da kein Schalten ausgeführt wurde, wird das Schlüsselbit von T5 mit 0 gekennzeichnet. In [Fig. 3K](#) wird die Transaktion T5 entladen. Da T5 nicht verschlüsselt war, bleibt das Schlüsselregister **31** auf 0. In [Fig. 3L](#) wird die Transaktion T6 in die Lesen-Warteschlange **14** geladen. Es wird darauf hingewiesen, dass ein Schalten ausgeführt wurde, die Schreiben-Warteschlange jedoch leer ist, wodurch das Verriegelungsbit von T6 mit 0 gekennzeichnet wird. In den [Fig. 3M](#) und [Fig. 3N](#) werden die Transaktionen T4 bzw. T6 entladen.

[0030] Es wird darauf hingewiesen, dass die Se-

quenz, die in [Fig. 2](#) und [3](#) gezeigt ist, ausschließlich beispielhaft ist, da das Laden und Entladen der Warteschlangen in vielen unterschiedlichen Sequenzen bzw. Reihenfolgen auftreten kann, solange die Reihenfolgen- und Umgehungs-Regeln befolgt werden.

[0031] Es wird ferner darauf hingewiesen, dass der erfindungsgemäße Mechanismus mit einer einzelnen gemeinsamen Warteschlange und mehreren Umgehungswarteschlangen arbeiten kann. Jede Umgehungswarteschlange würde eine bestimmte Transaktion halten. Entsprechende Regeln dafür, welche Transaktionen welche Transaktionen umgehen können, müssten aufgeschrieben und in die Steuerlogik platziert werden. Eine Tabelle würde für jedes Paar aus Transaktionen Ti und Tj detailliert angeben, ob Ti Tj umgehen kann, ob Ti Tj nicht umgehen kann, oder ob es egal ist, ob Ti Tj umgehen kann oder nicht. Der Mechanismus würde ein separates Schlüsselbit in der gemeinsamen Warteschlange für jede der Umgehungswarteschlangen aufweisen. Somit würde die Schreiben-Warteschlange in [Fig. 2](#) und [3](#) mehrere Schlüsselbeträge aufweisen, einen für jede Lesen-Warteschlange. Jede Lesen-Warteschlange würde ein einzelnes Verriegelungsbit aufweisen. Somit steuert ein einzelner Eintrag in der Schreiben-Warteschlange das Entladen von mehreren gelesenen oder umgangenen Einträgen in der Lesen-Warteschlange.

[0032] Es wird ferner darauf hingewiesen, dass der erfindungsgemäße Mechanismus mit mehreren gemeinsamen Warteschlangen und einer Umgehungswarteschlange arbeiten kann. Der Mechanismus würde ein separates Verriegelungsbit in der Umgehungswarteschlange für jede der gemeinsamen Warteschlangen aufweisen. Somit würde jede der Schreiben-Warteschlangen in [Fig. 2](#) und [3](#) einen einzelnen Schlüsselbeträger aufweisen, während die Lesen-Warteschlange mehrere Verriegelungsbeträge aufweisen würde, einen für jede Schreiben-Warteschlange. Somit müsste jede umgangene Transaktion sicherstellen, dass Transaktionen aus beiden Schreiben-Warteschlangen entladen wurden, bevor die Lesetransaktion entladen wird.

[0033] Es wird ferner darauf hingewiesen, dass der erfindungsgemäße Mechanismus mit mehreren gemeinsamen Warteschlangen und mehreren Umgehungswarteschlangen arbeiten kann. Ein Paar aus Verriegelungs-Schlüssel-Warteschlangen wird für jede Umgehungswarteschlange benötigt, die einer gemeinsamen Warteschlange zugeordnet ist. Der erfindungsgemäße Mechanismus funktioniert auch bei einer hierarchischen Struktur, bei der eine gemeinsame Warteschlange mit einer oder mehreren Umgehungswarteschlangen eine weitere gemeinsame Warteschlange auf einer höheren Ebene speisen. In diesem Fall werden die gemeinsamen Warteschlangen zusammen mit den Umgehungswarteschlangen

in der höheren Ebene aus der Ausgabe der gemeinsamen Warteschlange und der Umgehungswarteschlangen in der niedrigeren Ebene gespeist.

[0034] Obwohl die vorliegende Erfindung und ihre Vorteile detailliert beschrieben wurden, sollte darauf hingewiesen werden, dass verschiedene Änderungen, Ersetzungen und Abänderungen hierin ausgeführt werden können, ohne von dem Schutzbereich der Erfindung abzuweichen, wie er in den beiliegenden Ansprüchen definiert ist.

Patentansprüche

1. Eine Busbrücke (**10**) zum Verbinden eines ersten Busses (**12**) mit einem zweiten Bus (**13**), wobei die Brücke die Übertragung einer Sequenz von Transaktionen von dem ersten Bus zu dem zweiten Bus ermöglicht, wobei die Sequenz von Transaktionen eine Mehrzahl von Transaktionstypen umfasst, gekennzeichnet durch:
 eine erste Warteschlange (**14**) zum Speichern zumindest einer Transaktion eines bestimmten Transaktionstyps, die durch zumindest einen anderen Transaktionstyp der Mehrzahl von Transaktionstypen umgangen werden kann; und
 eine zweite Warteschlange (**15**) zum Speichern zumindest einer Transaktion des zumindest einen anderen Transaktionstyps;
 einen ersten Satz von Kennungsregistern (**22**), wobei jedem Eintrag der ersten Warteschlange ein Kennungsregister zugeordnet ist; und
 einen zweiten Satz von Kennungsregistern (**21**), wobei jedem Eintrag der zweiten Warteschlange ein Kennungsregister zugeordnet ist; wobei die Busbrücke angeordnet ist zum:
 Laden jedes Kennungsregisters des zweiten Satzes (**21**) mit entweder einem Schlüsselbit oder einem Nichtschlüsselbit;
 Laden eines bestimmten Kennungsregisters des zweiten Satzes (**21**) für eine bestimmte Transaktion, die in die zweite Warteschlange geladen ist, mit dem Nichtschlüsselbit;
 Neuladen des bestimmten Kennungsregisters mit dem Schlüsselbit, falls eine nachfolgende Transaktion zu der bestimmten Transaktion in die erste Warteschlange (**14**) geladen wird, und die bestimmte Transaktion in der zweiten Warteschlange (**15**) vorliegt, wenn die nachfolgende Transaktion geladen wird;
 Laden jedes Kennungsregisters des ersten Satzes (**22**) mit entweder einem Verriegelungsbit oder einem Entriegelungsbit;
 Laden eines bestimmten Kennungsregisters des ersten Satzes für eine bestimmte Transaktion, die in die erste Warteschlange (**14**) geladen ist, mit dem Verriegelungsbit, falls eine bisherige Transaktion in die zweite Warteschlange (**15**) geladen wurde, die ein Kennungsregister (**21**) aufweist, das mit dem Schlüsselbit geladen ist, und die vorhergehende Transakti-

on in der zweiten Warteschlange vorliegt, wenn die bestimmte Transaktion in die erste Warteschlange geladen wird, und ansonsten Laden des bestimmten Kennungsregisters mit dem Entriegelungsbit; und Verhindern, dass eine bestimmte Transaktion, die in die erste Warteschlange (**14**) geladen ist und ein zugeordnetes Kennungsregister (**22**) aufweist, das mit dem Verriegelungsbit geladen ist, auf den zweiten Bus (**13**) entladen wird, bis eine weitere Transaktion, die in die zweite Warteschlange (**15**) geladen ist und ein zugeordnetes Kennungsregister (**21**) aufweist, das mit dem Schlüsselbit geladen ist, auf den zweiten Bus entladen wird; wodurch die zumindest eine Transaktion eines bestimmten Typs direkt von dem ersten Bus zu der ersten Warteschlange geleitet wird und Transaktionen von der ersten Warteschlange (**14**) und der zweiten Warteschlange (**15**) direkt zu dem zweiten Bus geleitet werden.

2. Die Busbrücke gemäß Anspruch 1, bei der: Transaktionen, die in der Busbrücke (**10**) gespeichert sind, gemäß Flusssteuerregeln auf den zweiten Bus (**13**) entladen werden.

3. Die Busbrücke gemäß Anspruch 2, bei der: die Flusssteuerregeln Umgehungsregeln umfassen, die definieren, welche Transaktionstypen der Mehrzahl von Transaktionstypen andere Transaktionstypen der Mehrzahl von Transaktionstypen umgehen können.

4. Die Busbrücke gemäß einem der vorhergehenden Ansprüche, die ferner folgendes Merkmal umfasst:
 ein Schlüsselregister (**31**), das angeordnet ist, um sowohl eine Anzahl von Transaktionen, die entladen wurden, und das Schlüsselbit (**21**) in ihrem zugeordneten Kennungsregister aufzuweisen, als auch eine Anzahl von Transaktionen, die entladen wurden und das Verriegelungsbit (**22**) in ihrem zugeordneten Kennungsregister aufzuweisen, zu verfolgen.

5. Die Busbrücke gemäß Anspruch 4, bei der die Busbrücke angeordnet ist zum:
 Inkrementieren des SchlüsselRegisters (**31**) für jede Transaktion, die entladen wurde und das Schlüsselbit (**21**) in ihrem zugeordneten Kennungsregister aufweist; und
 Dekrementieren des SchlüsselRegisters für jede Transaktion, die entladen wurde und das Verriegelungsbit (**22**) in ihrem zugeordneten Kennungsregister aufweist.

6. Ein Verfahren zum Betreiben einer Busbrücke (**10**) zum Verbinden eines ersten Busses (**12**) mit einem zweiten Bus (**13**), wobei die Brücke die Übertragung einer Sequenz von Transaktionen von dem ersten Bus zu dem zweiten Bus ermöglicht, wobei die Sequenz von Transaktionen eine Mehrzahl von

Transaktionstypen umfasst, wobei die Busbrücke folgende Merkmale umfasst:

eine erste Warteschlange (**14**) zum Speichern zumindest einer Transaktion eines bestimmten Transaktionstyps, die durch zumindest einen anderen Transaktionstyp der Mehrzahl von Transaktionstypen umgangen werden kann; und

eine zweite Warteschlange (**15**) zum Speichern zumindest einer Transaktion des zumindest einen anderen Transaktionstyps;

einen ersten Satz von Kennungsregistern (**22**), wobei jedem Eintrag der ersten Warteschlange ein Kennungsregister zugeordnet ist; und

einen zweiten Satz von Kennungsregistern (**21**), wobei jedem Eintrag der zweiten Warteschlange ein Kennungsregister zugeordnet ist;

wobei die Busbrücke angeordnet ist zum:

Laden jedes Kennungsregisters des zweiten Satzes (**21**) mit entweder einem Schlüsselbit oder einem Nichtschlüsselbit;

Laden eines bestimmten Kennungsregisters des zweiten Satzes (**21**) für eine bestimmte Transaktion, die in die zweite Warteschlange geladen ist, mit dem Nichtschlüsselbit;

Neuladen des bestimmten Kennungsregisters mit dem Schlüsselbit, falls eine nachfolgende Transaktion zu der bestimmten Transaktion in die erste Warteschlange (**14**) geladen wird, und die bestimmte Transaktion in der zweiten Warteschlange (**15**) vorliegt, wenn die nachfolgende Transaktion geladen wird;

Laden jedes Kennungsregisters des ersten Satzes (**22**) mit entweder einem Verriegelungsbit oder einem Entriegelungsbit;

Laden eines bestimmten Kennungsregisters des ersten Satzes für eine bestimmte Transaktion, die in die erste Warteschlange (**14**) geladen ist, mit dem Verriegelungsbit, falls eine bisherige Transaktion in die zweite Warteschlange (**15**) geladen wurde, die ein Kennungsregister (**21**) aufweist, das mit dem Schlüsselbit geladen ist, und die vorhergehende Transaktion in der zweiten Warteschlange vorliegt, wenn die bestimmte Transaktion in die erste Warteschlange geladen wird, und ansonsten Laden des bestimmten Kennungsregisters mit dem Entriegelungsbit; und Verhindern, dass eine bestimmte Transaktion, die in die erste Warteschlange (**14**) geladen ist und ein zugeordnetes Kennungsregister (**22**) aufweist, das mit dem Verriegelungsbit geladen ist, auf den zweiten Bus (**13**) entladen wird, bis eine weitere Transaktion, die in die zweite Warteschlange (**15**) geladen ist und ein zugeordnetes Kennungsregister (**21**) aufweist, das mit dem Schlüsselbit geladen ist, auf den zweiten Bus entladen wird;

wodurch die zumindest eine Transaktion eines bestimmten Typs direkt von dem ersten Bus zu der ersten Warteschlange geleitet wird und Transaktionen von der ersten Warteschlange (**14**) und der zweiten Warteschlange (**15**) direkt zu dem zweiten Bus geleitet werden.

7. Das Verfahren gemäß Anspruch 6, das ferner folgenden Schritt umfasst:

Entladen von Transaktionen, die in der Busbrücke (**10**) gespeichert sind, auf den zweiten Bus (**13**) gemäß Flusssteuerregeln.

8. Das Verfahren gemäß Anspruch 6 oder 7, das ferner folgende Schritte umfasst:

Betreiben eines SchlüsselRegisters (**31**), um sowohl eine Anzahl von Transaktionen, die entladen wurden und das Schlüsselbit (**21**) in ihrem zugeordneten Kennungsregister aufweisen, und eine Anzahl von Transaktionen, die entladen wurden und das Verriegelungsbit (**22**) in ihrem zugeordneten Kennungsregister aufweisen, zu verfolgen, durch:

Inkrementieren des SchlüsselRegisters (**31**) für jede Transaktion, die entladen wurde und das Schlüsselbit (**21**) in ihrem zugeordneten Kennungsregister aufweist; und

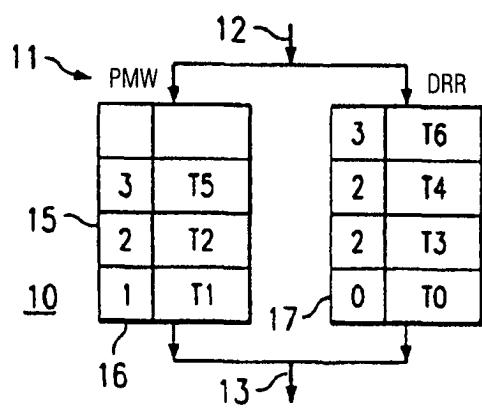
Dekrementieren des SchlüsselRegisters für jede Transaktion, die entladen wurde und das Verriegelungsbit (**22**) in ihrem zugeordneten Kennungsregister aufweist.

9. Ein Computerprogramm, das eine Computerprogrammcodeeinrichtung umfasst, zum Durchführen aller Schritte gemäß einem der Ansprüche 6, 7 oder 8, wenn das Programm auf einem Computer läuft.

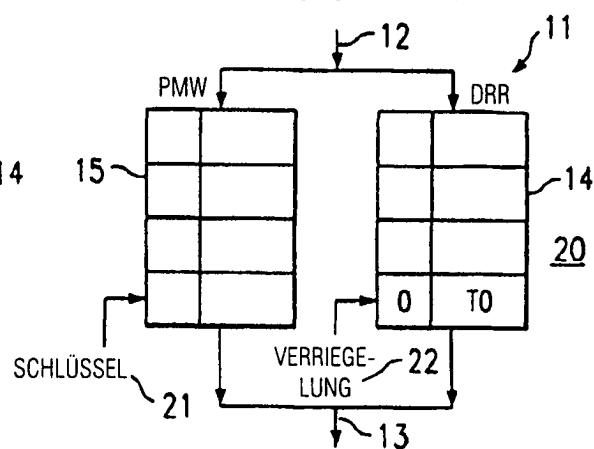
10. Ein Computerprogramm gemäß Anspruch 9, das auf einem computerlesbaren Medium ausgeführt ist.

Es folgen 4 Blatt Zeichnungen

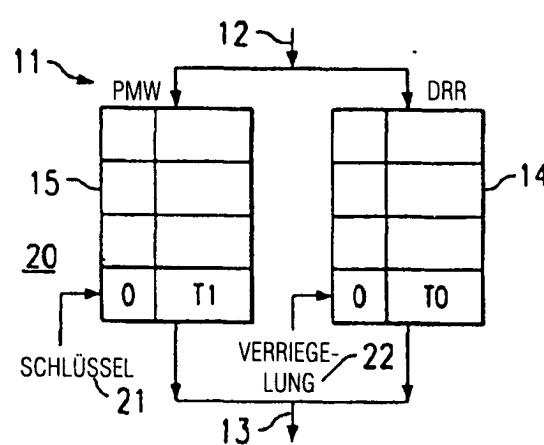
FIGUR 1



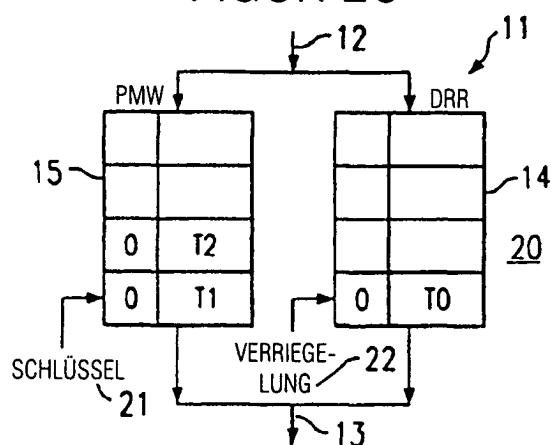
FIGUR 2A



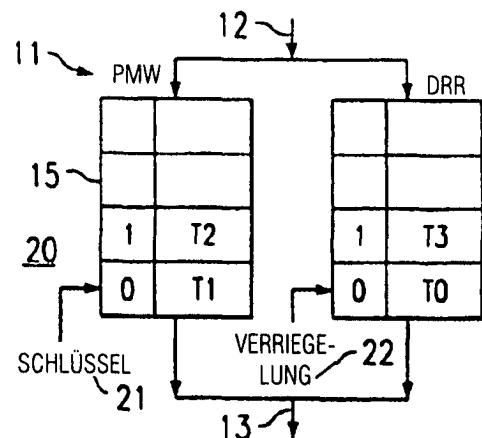
FIGUR 2B



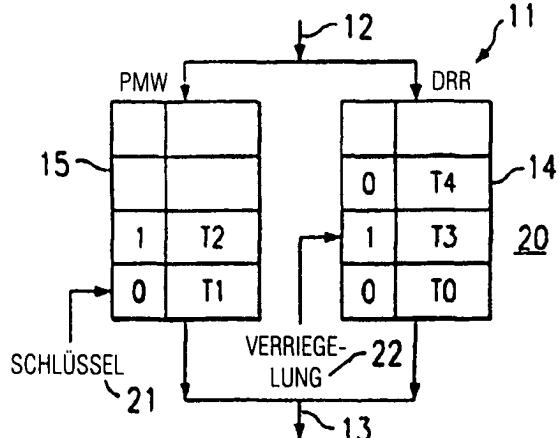
FIGUR 2C



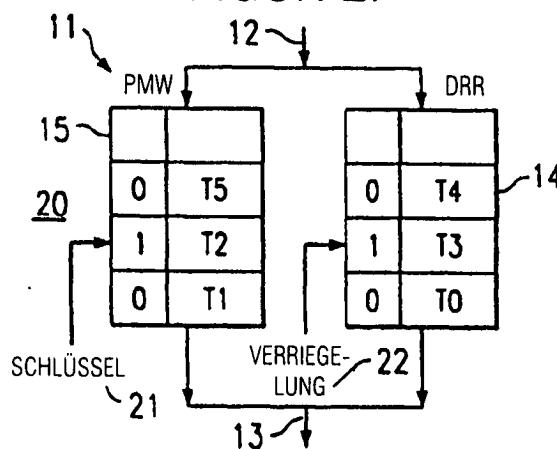
FIGUR 2D



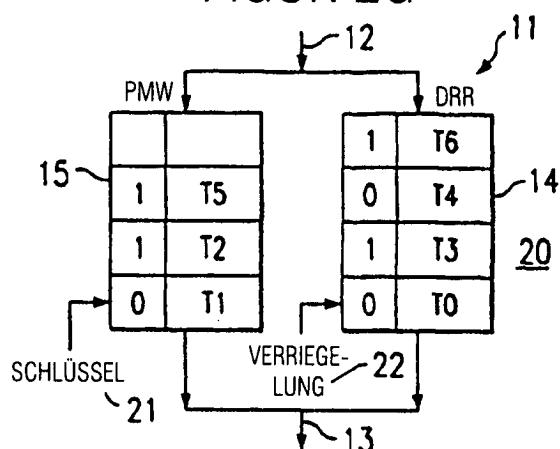
FIGUR 2E



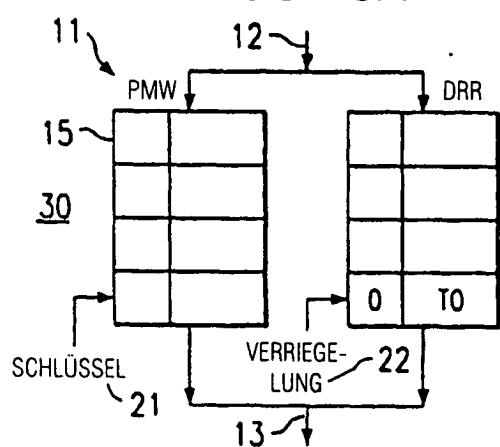
FIGUR 2F



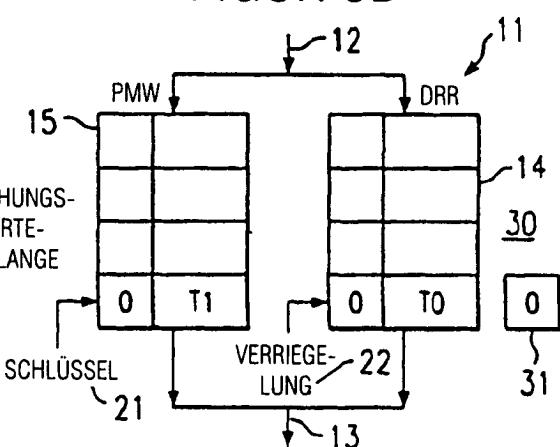
FIGUR 2G



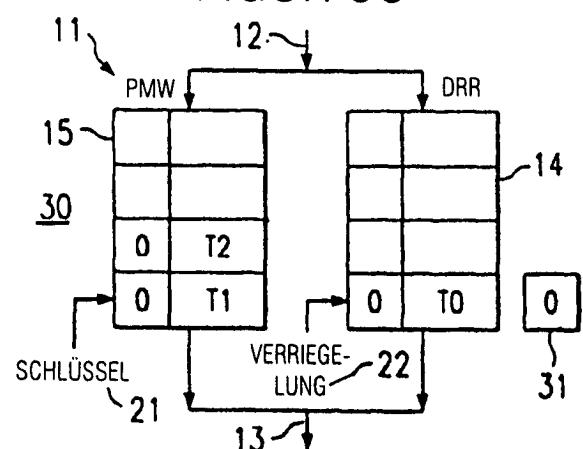
FIGUR 3A



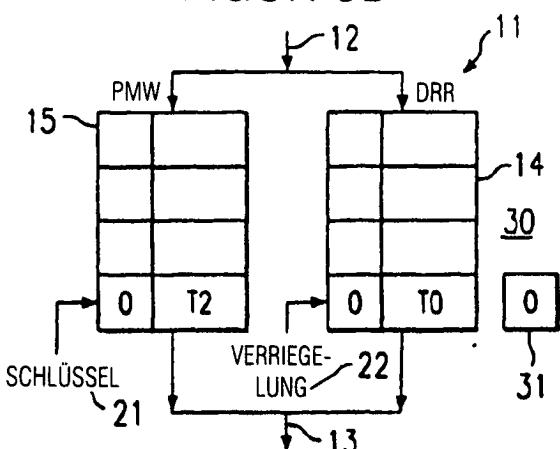
FIGUR 3B



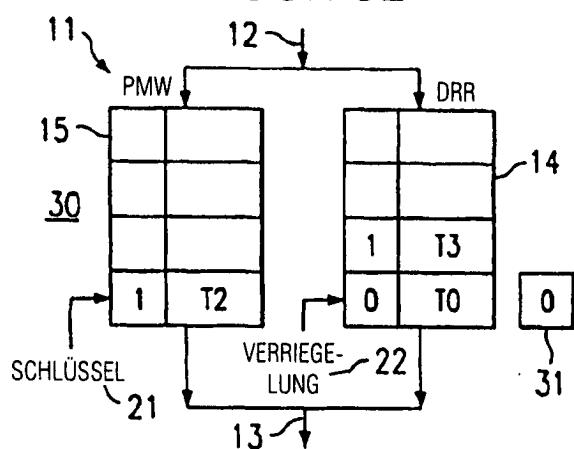
FIGUR 3C



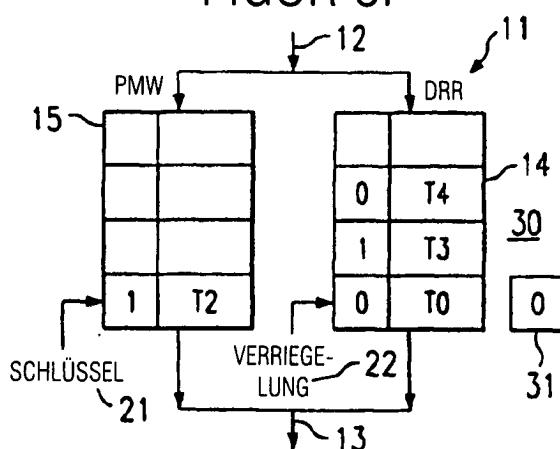
FIGUR 3D



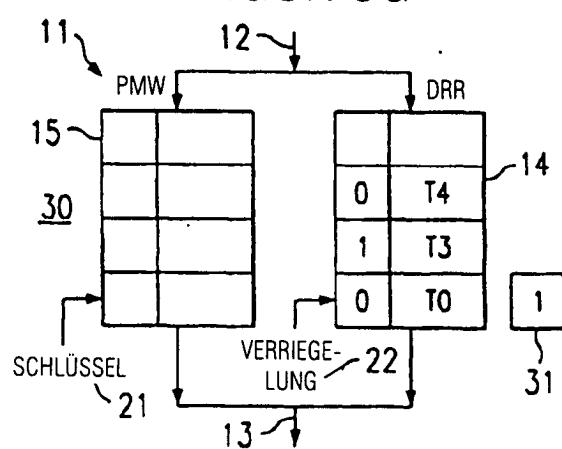
FIGUR 3E



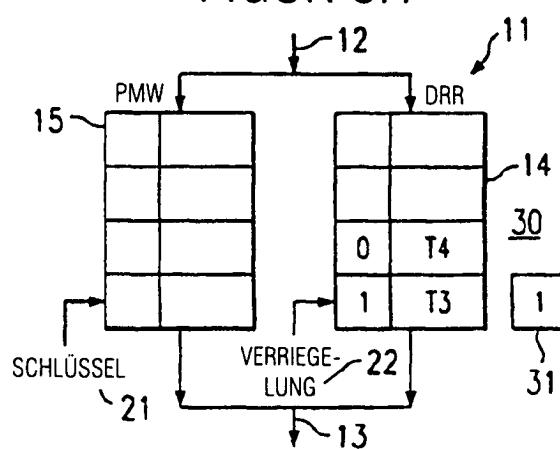
FIGUR 3F



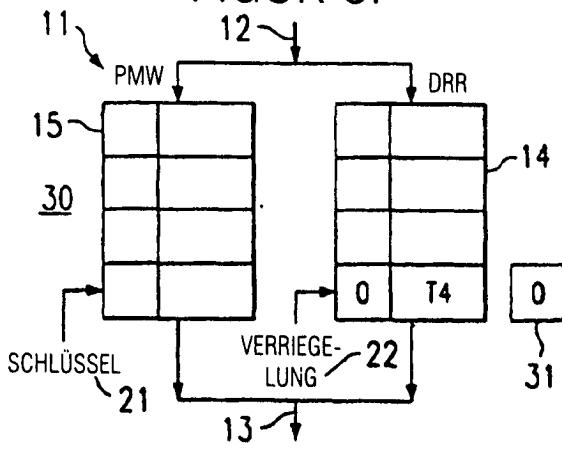
FIGUR 3G



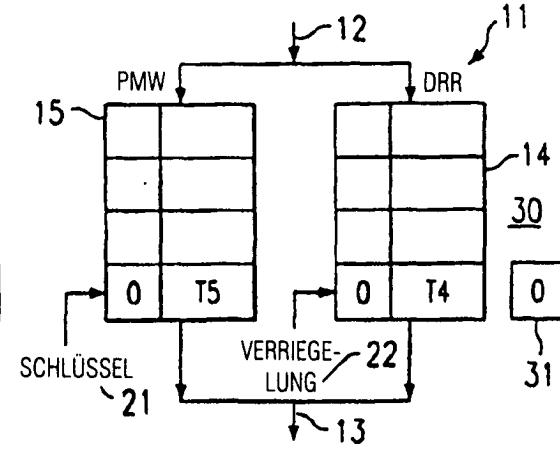
FIGUR 3H



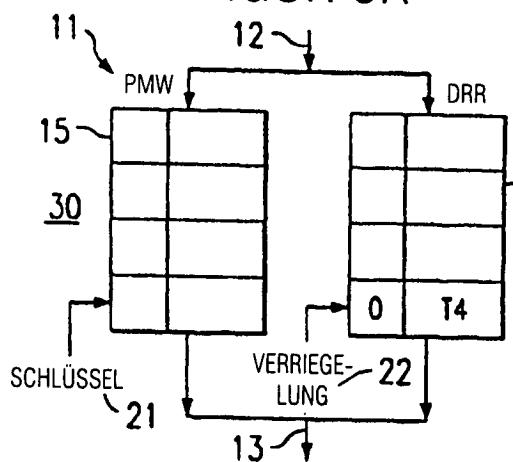
FIGUR 3I



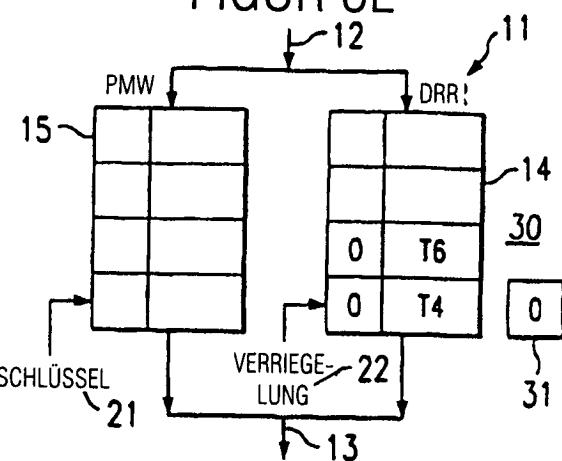
FIGUR 3J



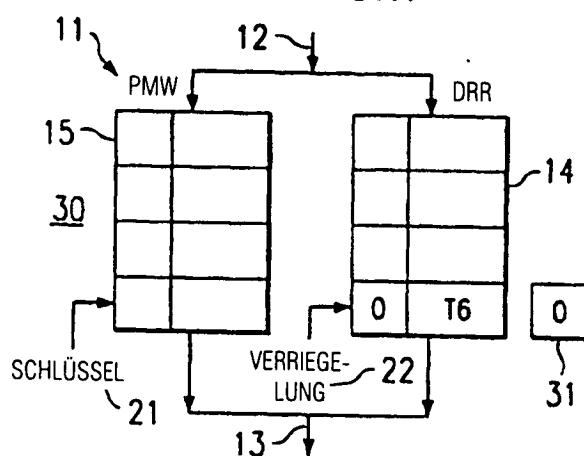
FIGUR 3K



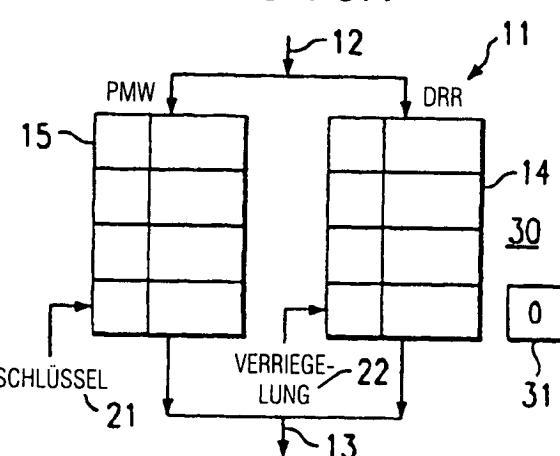
FIGUR 3L



FIGUR 3M

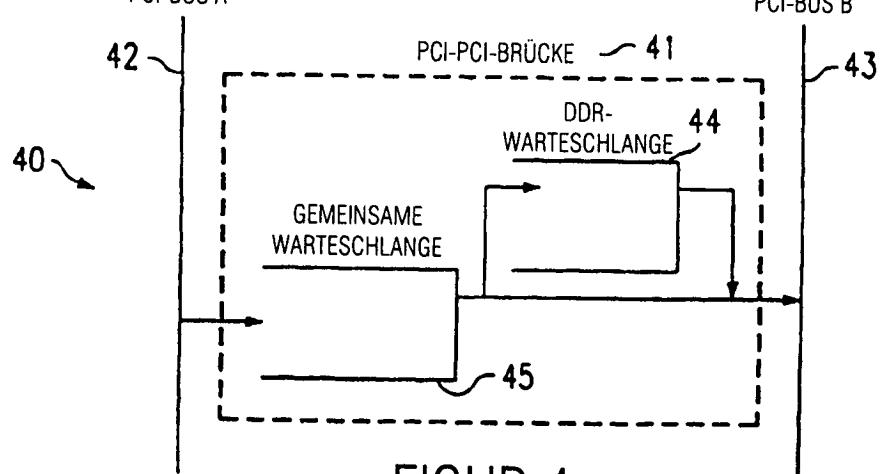


FIGUR 3N



PCI-BUS A

PCI-BUS B



FIGUR 4