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

(10) **DE 696 31 055 T2** 2004.07.08

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 843 854 B1**
(21) Deutsches Aktenzeichen: **696 31 055.4**
(86) PCT-Aktenzeichen: **PCT/US96/11974**
(96) Europäisches Aktenzeichen: **96 925 380.6**
(87) PCT-Veröffentlichungs-Nr.: **WO 97/006489**
(86) PCT-Anmeldetag: **18.07.1996**
(87) Veröffentlichungstag der PCT-Anmeldung: **20.02.1997**
(97) Erstveröffentlichung durch das EPA: **27.05.1998**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **10.12.2003**
(47) Veröffentlichungstag im Patentblatt: **08.07.2004**

(51) Int Cl.[7]: **G06F 13/16**

(30) Unionspriorität:
512613 **07.08.1995** **US**

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:
Applied Micro Circuits Corp., San Diego, Calif., US

(72) Erfinder:
ALEXANDER, Joffe, Palo Alto, US

(74) Vertreter:
Fiener, J., Pat.-Anw., 87719 Mindelheim

(54) Bezeichnung: **SPEICHERSCHNITTSTELLENEINHEIT, VERMITTLUNGSSYSTEM MIT GEMEINSAMEM SPEICHER UND VERFAHREN DAZU**

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

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

Beschreibung

HINTERGRUND DER ERFINDUNG

1. Bereich der Erfindung

[0001] Die Erfindung betrifft allgemein Speicherschnittstellensysteme und insbesondere ein Verfahren und eine Vorrichtung zum Bereitstellen und Gewähren von Zugang zu einem Gemeinschaftsspeicher durch mehrere Geräte für Anwendungen wie Multiprozessorsysteme und Kommunikationsschaltung.

2. Beschreibung der verwandten Technik

[0002] Gemeinschaftsspeicher werden benutzt, um das Leiten von Daten zwischen mehreren Prozessen zu erleichtern. Eine typische Gemeinschaftsspeicherimplementation beinhaltet die Verwendung von mehreren Ports. Jeder Port kann einem anderen externen Gerät Zugang zu dem Gemeinschaftsspeicher geben. Die verschiedenen Geräte können wiederum an der Steuerung oder Ausführung verschiedener Prozesse beteiligt sein, die die Daten übertragen.

[0003] Um jedem Port Zugang zu dem Gemeinschaftsspeicher zu geben, der durch Zugriffe anderer Ports auf den Gemeinschaftsspeicher nicht erheblich beeinflusst wird, wird der Pfad zu dem Gemeinschaftsspeicher gewöhnlich mit einer Bandbreite ausgelegt, die in der Nähe der Summe der Bandbreiten der einzelnen Ports liegt. Dadurch wird gewährleistet, dass die Datenübertragungskapazität des Pfades groß genug ist, damit kein Port große Verzögerungen beim Zugreifen auf den Gemeinschaftsspeicher erfährt, auch wenn vielleicht mehrere Ports Zugang zu dem Gemeinschaftsspeicher suchen. Dies wird im Allgemeinen mit einer von zwei Methoden oder einer Kombination davon erzielt. Zunächst kann die Zugriffszeit des Gemeinschaftsspeichers so ausgelegt werden, dass sie weitaus kürzer ist als die Datenübertragungszeiten für die einzelnen Ports. Zweitens kann die Pfadbreite zu dem Gemeinschaftsspeicher so ausgelegt werden, dass sie weitaus größer ist als die Pfadbreiten der individuellen Ports.

[0004] Der erste Ansatz besteht darin, einen Zeitschlitz für jeden Port zuzuweisen, innerhalb dessen Daten vom Gemeinschaftsspeicher gelesen oder auf diesen geschrieben werden können. Für jeden Port ist der zugewiesene Zeitschlitz kleiner als die tatsächliche Zeit, die zum Übertragen von Daten durch den Port benötigt wird. Somit werden die Daten während ihres Transfers zwischen einem Port und dem Gemeinschaftsspeicher vorübergehend gepuffert. Die Länge des Portzeitschlitzes ist gewöhnlich umgekehrt proportional zur Anzahl der Geräte, die sich den Zugang zum Gemeinschaftsspeicher teilen. Ein Gerät mit einem bestimmten Port kann Zugang zum Speicher nur während des zugewiesenen Zeitschlitzes für diesen Port erhalten. Daten werden zwischen Zeitschlitzen gepuffert. Dieser Ansatz hatte jedoch Nachteile. So erfordert er beispielsweise häufig, dass die Speicherzugriffszeit erheblich geringer ist als die Portdatentransferzeit. Eine solche Fehlübereinstimmung zwischen Datentransferraten für die Ports und dem Gemeinschaftsspeicher ist jedoch häufig unpraktisch.

[0005] Der zweite Ansatz beinhaltet ebenfalls die Zuweisung eines Zeitschlitzes zu jedem Port. So beinhalten beispielsweise Schreibvorgänge auf den Gemeinschaftsspeicher das vorübergehende Puffern mehrerer Datenwörter, die an einem jeweiligen Port eingehen, und das Übertragen aller dieser Wörter zu dem Speicher auf einem breiten Speicherpfad in einem einzigen Speicherzugriffszyklus während des für den Port zugewiesenen Zeitschlitzes. Umgekehrt beinhalten Lesevorgänge von dem Gemeinschaftsspeicher das Bereitstellen mehrerer Wörter auf dem breiten Pfad in einem einzigen Speicherzugriffszyklus während eines für einen jeweiligen Port designierten Zeitschlitzes, das vorübergehende Puffern der von dem Speicher gelesenen Wörter und das anschließende Übertragen der Wörter durch den Port. Dieser zweite Ansatz ist besonders gut für Burst-Modus-Systeme geeignet, in denen Datenwörter in Mehrwortbündeln durch jeweilige Ports geleitet werden. Ein komplettes Bündel kann in einem Speicherzugriffspuffer vorübergehend gespeichert und dann während eines einzelnen Speicherzugriffzyklus durch einen Pfad mit einer solchen großen Bandbreite zu dem Gemeinschaftsspeicher gelesen oder geschrieben werden. Durch Bereitstellen eines Pfades mit einer geeignet großen Bandbreite kann jeder Port aussehen gelassen werden, als hätte er exklusiven Zugang zu dem Gemeinschaftsspeicher, unbehindert durch Datentransfers durch andere Ports.

[0006] Das illustrative Blockdiagramm von **Fig.** 1 veranschaulicht eine frühere Implementation eines Multiport-Speichersystems, bei dem k Ports, jeweils mit einer Wortbreite m, sich einen Gemeinschaftsspeicher gleichermaßen teilen. Jedes Bündel enthält k Wörter. K Speicherzugriffspuffer können jeweils k m-Bit-Wörter speichern. Jeder Puffer ist über einen Pfad mit einer Leitungsbreite von k × m mit dem Gemeinschaftsspeicher verbunden. Der Gemeinschaftsspeicher hat eine Breite von k × m Bit.

[0007] Die illustrativen Zeichnungen von **Fig.** 2 zeigen ein Datenformat, das in einem typischen Multiport-Speichersystem wie dem von **Fig.** 1 benutzt wird. Während eines Transfers von Daten zu dem Gemeinschaftsspeicher wird ein k-Wort-Bündel mit m-Bit-Wörtern durch einen Port geleitet. Das gesamte Bündel wird in einem einzigen Speicherzugriffspuffer kurz gespeichert. Dann werden während eines vorgeschriebenen Zeitschlitzes alle k Wörter des Bündels gleichzeitig von dem Puffer übertragen und über den k × m Pfad auf den Gemeinschaftsspeicher geschrieben. Während einer Übertragung von Daten aus dem Gemeinschaftsspeicher hinaus werden k

Wörter von dem Gemeinschaftsspeicher während eines anderen vorgeschriebenen Zeitschlitzes gelesen und zu einem einzigen Speicherzugriffspuffer übertragen. Dann werden die gepufferten Daten durch den mit diesem Puffer assoziierten Port übertragen.

[0008] Der Port, der das Bündel ursprünglich eingegeben hat, kann sich von dem Port unterscheiden, der das Bündel ausgibt. Der Gemeinschaftsspeicher speichert das Bündel vorübergehend so, dass es vom Eingangs- zum Ausgangsport geleitet werden kann. Somit kann das System von **Fig.** 1 zum Leiten von Daten zwischen Ports verwendet werden.

[0009] Spezifischer ausgedrückt, es werden in einem Speicherschreibvorgang beispielsweise k m-Bit-Wörter, die durch einen jeweiligen Port empfangen wurden, von einem diesem Port zugewiesenen Speicherzugriffspuffer gepuffert. Danach werden während eines für diesen Speicherzugriffspuffer reservierten Zeitschlitzes alle in dem zugewiesenen Puffer gespeicherten k m-Bit-Wörter gleichzeitig über den gemeinsamen Pfad mit k × m Bit Breite auf den Gemeinschaftsspeicher geschrieben. Ebenso können alle anderen Puffer k m-Bit-Wörter für ihre eigenen assoziieren Ports speichern. Der gesamte Inhalt (alle k Wörter) jedes einzelnen Puffers kann während des für diesen Puffer reservierten individuellen Zeitschlitzes auf den Gemeinschaftsspeicher geschrieben werden. Ein Speicherlesevorgang ist analog, wobei die Schritte des Schreibvorgangs umgekehrt werden.

[0010] Ein Nachteil dieser früheren Implementation ist die große Zahl von Verdrahtungspins zwischen einem Puffer und dem Bus in einem großen Multiport-System. Das Blockdiagramm in **Fig.** 3 veranschaulicht ein weiteres konventionelles Multiport-Gemeinschaftsspeichersystem. Das verwendete Datenformat beinhaltet 16 Wortbündel von 72 Bit pro Wort (64 Bit Daten plus 8 Bit Parität). Der Gemeinschaftsspeicherbus hat eine Breite von 1152 Zeilen (16 Wörter × 72 Bit/Wort). Der Bus ist mit jedem der 16 Speicherzugriffspuffer verbunden. Jeder Puffer würde mehr als 1224 Datenpins benötigen, 1152 für die Verbindung mit dem Bus und 72 für die Verbindung mit dem Port. Leider würden die 1152 mit dem Bus verbundenen Datenpins jeweils eine hohe Ansteuerungsfähigkeit benötigen, um auf einem mit allen 16 Puffern und dem Gemeinschaftsspeicher verbundenen Bus zu arbeiten. **Fig.** 3 zeigt eine illustrative Buskapazität, die von den Pufferpins überwunden werden muss.

[0011] Die WO-A-95/05635 offenbart eine Speicherschnittstelleneinheit, die für den Einsatz in einem Multiport-Speichersystem angeordnet ist, die jedoch begrenzte Konnektivität mit einem Puffer davon hat.

[0012] Es besteht somit Bedarf an einer verbesserten Architektur zum Bereitstellen von Multiport-Zugriff auf einen Gemeinschaftsspeicher. Die Architektur soll weniger Pins für Speicherzugriffspuffer erfordern und soll keine hohe Ansteuerungsfähigkeit für Pufferpins erfordern. Die vorliegende Erfindung deckt diesen Bedarf.

[0013] Gemäß einem Aspekt der vorliegenden Erfindung wird eine Speicherschnittstelleneinheit für den Einsatz in einem Multiport-Speichersystem mit einem Direktzugriffsspeicher und mehreren Datenports bereitgestellt, wobei die Einheit Folgendes umfasst: eine Mehrzahl von Puffern, die jeweils eine Mehrzahl von Speicherelementen für die gleichzeitige Speicherung einer Mehrzahl von jeweiligen Datenbündelteilmengen beinhalten, die zwischen verschiedenen jeweiligen Datenports und dem genannten Direktzugriffsspeicher übertragen werden; und

eine Verbindungsschaltung, die eine Mehrzahl von Datenleitungen beinhaltet, die jeden jeweiligen Datenport mit einer Mehrzahl von jeweiligen Puffern verbinden, so dass verschiedene jeweilige Teilmengen eines beliebigen Datenbündels, das zwischen einem jeweiligen Datenport und dem genannten Direktzugriffsspeicher übertragen wird, zu oder von verschiedenen jeweiligen Puffern über verschiedene jeweilige Datenleitungen geleitet werden, die einen Port und einen Puffer miteinander verbinden;

wobei wenigstens eine einen Port und einen Puffer verbindende Datenleitung eine Breite von mehreren Bits hat.

[0014] Gemäß einem weiteren Aspekt der vorliegenden Erfindung wird ein Verfahren zum Übertragen von n-Wortdatenbündeln bereitgestellt, die m Bits pro Wort zwischen k Datenports und einem Direktzugriffsspeicher in einem Mehrportspeichersystem beinhalten, umfassend die folgenden Schritte:

Unterteilen von Datenbündeln, die durch jeweilige Datenports übertragen werden, in jeweilige Datenbündelteilmengen, in denen jede jeweilige Teilmenge mehr als n jeweilige Bits beinhaltet, wobei solche mehr als n jeweiligen Bits dieselben jeweiligen Bitpositionen in verschiedenen Datenwörtern eines jeweiligen Datenbündels belegen;

Bereitstellen von m Puffern, jeweils mit k Speicherelementen;

vorübergehendes Speichern verschiedener Datenbündelteilmengen eines jeweiligen Datenbündels auf seinem Weg zu oder von einem jeweiligen Datenport und dem Direktzugriffsspeicher in verschiedenen Speicherelementen in verschiedenen Puffern; und

Übertragen aller verschiedener Datenbündelteilmengen eines jeweiligen Datenbündels auf dem Weg zu oder von einem jeweiligen Datenport und dem Direktzugriffsspeicher parallel zwischen verschiedenen Speicherelementen in verschiedenen Puffern und dem Direktzugriffsspeicher.

[0015] **Fig.** 1 ist ein Blockdiagramn eines früheren Multiport-Gemeinschaftsspeichersystems;

[0016] **Fig.** 2 ist ein Datenformat, das in dem früheren System von **Fig.** 1 verwendet werden kann;

[0017] **Fig.** 3 ist ein Blockdiagramm eines weiteren früheren Multiport-Gemeinschaftsspeichersystems;

[0018] **Fig.** 4 ist ein verallgemeinertes Blockdiagramm eines ersten Multiport-Gemeinschaftsspeichersystems gemäß der vorliegenden Erfindung;

[0019] **Fig.** 5 zeigt ein Datenformat und einen Datenfluss, die in der Ausgestaltung von **Fig.** 4 verwendet werden;
[0020] **Fig.** 6 ist ein ausführlicheres Blockdiagramm von Speicherzugriffspuffern und Steuerlogik der Ausgestaltung von **Fig.** 4;
[0021] **Fig.** 7 ist ein Blockdiagramm eines repräsentativen Doppelregisterpaares der Speicherzugriffspuffer von **Fig.** 6;
[0022] **Fig.** 8 ist ein zeitliches Ablaufdiagramm, das den Betrieb der Steuerlogik und der Speicherzugriffspuffer von **Fig.** 6 erläutert;
[0023] **Fig.** 9 ist ein zeitliches Ablaufdiagramm, das den Betrieb der Doppelregisterpaare von **Fig.** 7 erläutert;
[0024] **Fig.** 10 ist ein Blockdiagramm eines Asynchronous Transfer Mode Schalters gemäß der vorliegenden Erfindung;
[0025] **Fig.** 11 ist ein Blockdiagramm eines Schaltsystems, das eine Speicherschnittstelleneinheit (MIU) und einen zugehörigen Steuerschaltkomplex sowie Register gemäß einer derzeit bevorzugten Ausgestaltung der Erfindung beinhaltet;
[0026] **Fig.** 12 ist ein vereinfachtes Blockdiagramm, das Datenleitungen und Steuerleitungen der Ausgestaltung von **Fig.** 11 illustriert;
[0027] **Fig.** 13 ist ein vereinfachtes Blockdiagramm, das eine mögliche Konfiguration des konfigurierbaren Schalters des Schaltsystems von **Fig.** 11 zeigt;
[0028] **Fig.** 14A ist ein Blockdiagramm, das Signalleitungen und Steuerleitungen des konfigurierbaren Schalters der Ausgestaltung von **Fig.** 11 zeigt;
[0029] **Fig.** 14B ist ein Logikdiagramm auf Gatterebene, das die Logikgatter im konfigurierbaren Schalter von **Fig.** 14A zeigt, die zum Erzeugen von c0 während eines Speicherschreibvorgangs für eine Zwei-MIU-Implementation verwendet werden;
[0030] **Fig.** 14C ist ein Logikdiagramm auf Gatterebene, das die Logikgatter im konfigurierbaren Schalter von **Fig.** 14A zeigt, die zum Erzeugen von d0 während eines Speicherlesevorgangs für eine Zwei-MIU-Implementation verwendet werden;
[0031] **Fig.** 15 ist ein illustratives Blockdiagramm, das Details eines Transferpuffers der Ausgestaltung von **Fig.** 11 zeigt;
[0032] **Fig.** 16 ist ein illustratives Blockdiagramm, das Details eines individuellen Speicherzugriffspuffers des Transferpuffers von **Fig.** 15 zeigt;
[0033] **Fig.** 17 ist ein Blockdiagramm eines Zwei-MIU-Systems gemäß einer vorliegenden Ausgestaltung der Erfindung;
[0034] **Fig.** 18 ist eine Seitenansicht von gestapelten Boxen, die das Zwei-MIU-System von **Fig.** 17 enthalten, die Port- und Adressverbindungen zwischen den beiden MIUs illustriert;
[0035] **Fig.** 19 ist eine Matrix, die Verbindungen zwischen konfigurierbaren Schaltern des Zwei-MIU-Systems von **Fig.** 17 darstellt;
[0036] **Fig.** 20 ist ein vereinfachtes Blockdiagramm des Zwei-MIU-Systems von **Fig.** 17, das den Ablauf von Bündelteilmengen während eines Speicherschreibvorgangs illustriert;
[0037] **Fig.** 21 ist ein zeitliches Ablaufdiagramm, das den zeitlichen Ablauf von Daten- und Steuersignalen während eines Speicherschreibvorgangs für das Zwei-MIU-System von **Fig.** 17 illustriert;
[0038] **Fig.** 22 ist ein vereinfachtes Blockdiagramm des Zwei-MIU-Systems von **Fig.** 17, das den Ablauf von Teilmengen während eines Speicherlesevorgangs illustriert;
[0039] **Fig.** 23 ist ein zeitliches Ablaufdiagramm, das den zeitlichen Ablauf von Daten- und Steuersignalen in Verbindung mit einem Lesevorgang für das Zwei-MIU-System von **Fig.** 17 illustriert;
[0040] **Fig.** 24 ist ein vereinfachtes Blockdiagramm eines Vier-MIU-Systems gemäß einer vorliegenden Ausgestaltung der Erfindung;
[0041] **Fig.** 25 ist eine Seitenansicht von gestapelten Boxen, die das Vier-MIU-System von **Fig.** 24 beinhalten, die die Port- und Adressverbindungen zwischen den MIUs illustriert;
[0042] **Fig.** 26 ist eine Matrix, die Verbindungen zwischen MIUs des Vier-MIU-Systems von **Fig.** 24 repräsentiert;
[0043] **Fig.** 27 ist ein vereinfachtes Blockdiagramm, das den Fluss von Bündelteilmengen innerhalb des Vier-MIU-Systems von **Fig.** 24 während eines Lesevorgangs illustriert;
[0044] **Fig.** 28 ist ein zeitliches Ablaufdiagramm, das den zeitlichen Ablauf von Daten- und Steuersignalen in Verbindung mit einem Lesevorgang für das Vier-MIU-System von **Fig.** 24 illustriert;
[0045] **Fig.** 29 ist ein vereinfachtes Blockdiagramm, das den Fluss von Bündelteilmengen innerhalb des Vier-MIU-Systems von **Fig.** 24 während eines Speicherschreibvorgangs illustriert; und
[0046] **Fig.** 30 ist ein zeitliches Ablaufdiagramm, das den zeitlichen Ablauf von Daten- und Steuersignalen in Verbindung mit einem Schreibvorgang für das Vier-MIU-System von **Fig.** 24 illustriert;
[0047] **Fig.** 31 illustriert eine selbstständige MIU-Applikation unter Verwendung der Ausgestaltung von **Fig.** 11;
[0048] **Fig.** 32 illustriert eine Vier-MIU-Applikation unter Verwendung der Ausgestaltung von **Fig.** 11;
[0049] **Fig.** 33 illustriert eine Ethernet-Schaltanwendung unter Verwendung der Ausgestaltung von **Fig.** 11;
[0050] **Fig.** 34 illustriert einen Ethernet-Schalter mit einer ATM-Uplink-Anwendung unter Verwendung der Ausgestaltung von **Fig.** 11;
[0051] **Fig.** 35 illustriert einen stapelbaren Zwei-MIU-Schalter/Router unter Verwendung der Ausgestaltung von **Fig.** 11;
[0052] **Fig.** 36 illustriert einen Vier-MIU-Fast-Ethernet-Schalter unter Verwendung der Ausgestaltung von **Fig.** 11;
[0053] **Fig.** 37 ist eine verbesserte Ausgestaltung des Schalters von **Fig.** 4, der mit Zeitmultiplexkopplung (TDM) arbeitet;

[0054] **Fig.** 38 ist ein Zeitdiagramm, das den Betrieb des Schalters von **Fig.** 37 illustriert; und

[0055] **Fig.** 39 ist eine illustrative Zeichnung einer Ausgestaltung des Schalters von **Fig.** 4, bei dem individuelle Byte-Teilmengen von jeweils mehreren Bits durch eine Schaltmatrix übertragen werden.

[0056] Die vorliegende Erfindung umfasst eine) neuartige(s) Verfahren und Vorrichtung zum Implementieren eines Multiport-Gemeinschaftsspeichersystems. Die nachfolgende Beschreibung soll eine Fachperson befähigen, die Erfindung durchzuführen und zu nutzen. Beschreibungen spezifischer Anwendungen dienen lediglich als Beispiele. Verschiedene Modifikationen an der bevorzugten Ausgestaltung werden für die Fachperson offensichtlich sein, und die hierin definierten allgemeinen Grundsätze können auch auf andere Ausgestaltungen und Applikationen angewendet werden, ohne vom Umfang der Erfindung gemäß Definition in den beiliegenden Ansprüchen abzuweichen. Somit soll die vorliegende Erfindung auf die gezeigte Ausgestaltung nicht begrenzt sein, sondern ihr soll der größtmögliche Umfang eingeräumt werden, der mit den hierin offenbarten Grundsätzen und Merkmalen im Einklang steht.

[0057] **Fig.** 4 zeigt ein Blockdiagramm eines ersten erfindungsgemäßen Multiport-Gemeinschaftsspeichersystems **18**. Das erste System **18** beinhaltet einen Satz von k Ports **20** jeweils zum Ein-/Ausgeben von m-Bit-Datenwörtern in n Wortbündeln, eine Verbindungsmatrixschaltung **22**, einen Satz von m Speicherzugriffspuffern **24** und einen Gemeinschaftsspeicher **26**. Ein Port dient als digitaler Zweiweg-Pfad, über den digitale Informationen zu und von dem externen Schaltkomplex wie z. B. ein Datenbus übertragen werden können. Portstrukturen sind der Fachperson hinlänglich bekannt und brauchen hier nicht beschrieben zu werden. Spezifische Regionen in dem Gemeinschaftsspeicher **26**, durch gestrichelte Linien angedeutet, sind für Teilmengen der in den Puffern **24** vorübergehend gespeicherten Daten reserviert.

[0058] Bei Speicherschreibvorgängen übertragen die Ports **20** binäre Daten von damit verbundenen einzelnen externen Geräten (nicht dargestellt). Diese Ports **20** senden die empfangenen binären Daten zur Verbindungsmatrix **22**, die die durch die Ports übertragenen Daten unter den Speicherzugriffspuffern **24** verteilt. In der vorliegenden Ausgestaltung erfolgt die Verteilung so, dass jeder Speicherzugriffspuffer **24** eine Teilmenge der von jedem der Ports übertragenen Daten empfängt. In der vorliegenden Ausgestaltung kann jeder Puffer **24** alle durch den Puffer von einem einzelnen Port empfangenen Daten parallel zum Speicher **26** übertragen. Darüber hinaus können alle von einem solchen individuellen Port empfangenen parallelen Datentransfers während eines einzelnen Speicherzugriffszyklus erfolgen.

[0059] Der Speicherlesevorgang ist zu den Schritten des Schreibvorgangs analog, aber umgekehrt. Das heißt, die Bits für n m-Bit-Wörter werden vom Gemeinschaftsspeicher übertragen und unter den Speicherzugriffspuffern **24** verteilt. Die Verbindungsmatrix **22** legt dann die verteilten Bits an einen einzelnen Port **20** an, durch den ein Bündel aus n m-Bit-Wörtern zu einem externen Gerät (nicht dargestellt) übertragen wird.

[0060] **Fig.** 5 zeigt ein Diagramm, das Details der Verbindungsmatrixschaltung **22**, das Datenformat und den Datenablauf illustriert, die derzeit im ersten System **18** verwendet werden. Jeder Speicherzugriffspuffer **24** hat die Aufgabe, eine spezifische Bitposition für jedes Datenwort zu speichern, das durch einen der Ports übertragen wird. Datenwörter können Daten und Paritätsinformationen enthalten. In einer derzeit bevorzugten Ausgestaltung ist die Gesamtzahl der Speicherzugriffspuffer **24** gleich der Gesamtzahl von Bits pro Wort (m), so dass jeder Puffer für eine einzelne Bitposition in jedem Wort verantwortlich sein kann. So speichert beispielsweise der Speicherzugriffspuffer **24-1** Bit **1** (B1) jedes Wortes, das durch einen der Ports **20-1** bis **20-k** übertragen wird; der Speicherzugriffspuffer **24-2** speichert Bit **2** (B2) jedes Wortes, das durch einen der Ports **20-1** bis **20-k** übertragen wurde, usw.

[0061] Wieder mit Bezug auf **Fig.** 4, jeder Speicherzugriffspuffer **24-1** bis **24-m** ist durch einen Satz von n Datenleitungen **28-1** bis **28-k** mit dem Gemeinschaftsspeicher **26** verbunden. Somit können die m Teilmengen, jede Teilmenge mit n Bits, von Bits unterschiedlicher Ordnung gleichzeitig zwischen dem Speicher **26** und den Puffern **24** übertragen werden. In der vorliegenden Ausgestaltung gibt es n Bits für jede der m Teilmengen; d. h. ein Bit für jedes der n Wörter in einem n-Wort-Bündel. Spezifischer ausgedrückt, wenn es n Wörter pro Bündel gibt und es gibt m Bits pro Wort, dann gibt es n Bit pro Bündel von Ordnung B1, n Bit pro Bündel von Ordnung B2, ..., und n Bit pro Bündel von Ordnung Bm. Alle n Bits einer vorgeschriebenen Ordnung können gleichzeitig über die n Leitungen zum Speicher **26** gesendet werden, die mit einem Puffer verbunden sind, der zum Speichern der Bits dieser Ordnung zugewiesen sind. So werden beispielsweise alle n B1-Bits auf die mit dem Puffer **24-1** verbundenen n Leitungen gesetzt. Dies bedeutet, dass die m Puffer n Wörter gleichzeitig zu oder von einem adressierten Speicherort übertragen können. Wie nachfolgend erläutert, hätten alle diese übertragenen Wörter ihren Ursprung an demselben externen Gerät oder wurden von diesem angepeilt.

[0062] Jeder Port **20** überträgt Daten in einem vorgeschriebenen Format zwischen der Verbindungsmatrix **22** und einem externen Schaltkomplex (nicht dargestellt). In diesem Dokument bedeutet ein "Port-Bündel" ein Datenbündel, das durch einen einzelnen Port in einer Richtung zu oder von dem externen Schaltkomplex übertragen wird. In einer in **Fig.** 5 gezeigten vorliegenden Ausgestaltung können m-Bit-Datenwörter von einem Port an die Verbindungsmatrix **22** für die Übertragung zu den Puffern

24 gesendet werden. Umgekehrt können m-Bit-Datenwörter durch eine Verbindungsmatrix **22** zwecks Transfer zum externen Schaltkomplex zu einem solchen Port gesendet werden. Das Datenwortformat beinhaltet jeweils Datenwörter-Bits B1 bis Bm, die in einer vorgeschriebenen Reihenfolge präsentiert werden. Die Verbindungsmatrix verteilt die Bits wie oben beschrieben.

[0063] Die Verbindungsmatrixschaltung **22** kann beispielsweise mit einer Leiterplatte (PCB), einer Drahtwicklung oder mit gelöteten Drähten implementiert werden. In der derzeit bevorzugten Ausgestaltung bildet die Verbindungsmatrix die Verbindungen zwischen Ports und Puffern. Wie nachfolgend erläutert, wird beispielsweise jedes über Port **20-1** übertragene B1 in jedem Wort durch die Matrix **22** mit einem Speicherelement von Puffer **24-1** in Verbindung mit Port **20-1** verbunden; jedes B1 in jedem Wort, das durch Port **20-2** übertragen wird, ist durch die Matrix **22** mit einem Speicherelement von Puffer **24-1** in Verbindung mit Port **20-2** verbunden; ... jedes B1 in jedem Wort, das durch Port **20-k** übertragen wird, ist durch die Matrix **22** mit einem Speicherelement von Puffer **24-1** in Verbindung mit Port **20-k** verbunden.

[0064] Die Speicherzugriffspuffer können mit standardmäßiger oder kundenspezifischer Logik implementiert werden. Die Zeichnungen der **Fig.** 6–9 illustrieren und erläutern den Betrieb der Speicherzugriffspuffer **24** und eines repräsentativen Doppelregisterpaares **34** von einem der Speicherzugriffspuffer **24**. Das illustrative Blockdiagramm von **Fig.** 6 zeigt die Speicherzugriffspuffer **24**, die Steuerlogik **29**, die seriellen Ein/Ausgangsleitungen **30**, die die Puffer **24** mit einer Verbindungsmatrix (nicht dargestellt) verbinden, und einen parallelen n-Bit-Ein-/Ausgangsbus **32**. Die Steuerlogik **29** steuert serielle E/A auf Leitungen **30** und parallele E/A auf Bus **32**. Der Betrieb der Steuerlogik wird mit Bezug auf das Zeitdiagramm von **Fig.** 8 erläutert.

[0065] **Fig.** 7 zeigt ein repräsentatives n-Bit-Doppelregisterpaar **34**. In einer vorliegenden Ausgestaltung der Erfindung hat jeder Speicherzugriffspuffer **24k** Doppelregisterpaare wie die, die in **Fig.** 7 gezeigt sind, für insgesamt **2k** Register pro Puffer **24**. Jedes Registerpaar in jedem Puffer ist mit einem der Ports assoziiert. Jedes Registerpaar hat die Aufgabe, alle Bits an einem vorgeschriebenen Ort (Ordnung) in jedem Datenwort zu speichern, das zu oder von dem Port in Verbindung mit diesem Registerpaar übertragen wird. Das heißt, jedes jeweilige Doppelregisterpaar eines jeweiligen Puffers speichert und überträgt alle Bits derselben Ordnung von jedem Datenwort, das durch einen jeweiligen Port übertragen wird.

[0066] So empfängt beispielsweise, mit Bezug auf **Fig.** 5, im Puffer **24-1** das Doppelregisterpaar R_{i1} das erste (in Folge) Bit B1 jedes durch Port P_i (nicht dargestellt) übertragene Datenwort und speichert es. Im Puffer **24-2** empfängt und speichert das Doppelregisterpaar R_{i2} das zweite (in Folge) Bit B2 jedes durch Port P_i übertragene Wort. In Puffer **24-m** empfängt und speichert das Doppelregisterpaar R_{im} das m_{te} (in Folge) Bit Bm jedes durch Port P_i übertragene Wort. Somit repräsentieren beispielsweise alle durch Port P_i übertragenen B1en eine Teilmenge der durch diesen Port übertragenen Daten. Ebenso repräsentieren alle B2en eine weitere Teilmenge, wie auch alle Bm zum Beispiel.

[0067] **Fig.** 7 zeigt Details eines Shift-In/Shift-Out-Registerpaares eines repräsentativen Puffers der Speicherzugriffspuffer. Das Doppelregisterpaar **34** beinhaltet ein Shift-in-Register **36** und ein Shift-out-Register **38**. Das Shift-in-Register **36** speichert zunächst parallele n Bits von Daten auf den Speicherbus **32**, die in den Gemeinschaftsspeicher **26** geschrieben werden sollen, und drückt sie dann parallel auf. Das Shift-out-Register liest parallel n Bits Daten von dem Speicherbus **32**, die aus dem Gemeinschaftsspeicher **26** "herausgelesen" wurden, und speichert sie.

[0068] Die im Shift-in-Register **36** gespeicherten Daten werden seriell von der Verbindungsmatrix **22** in das Shift-in-Register **36** verschoben und von dort parallel auf den Bus **32** aufgedrückt, um wie oben beschrieben auf den Speicher **26** geschrieben zu werden. Umgekehrt werden die im Shift-out-Register **38** gespeicherten Daten parallel vom Bus **32** in das Shift-out-Register gelesen, nachdem sie vom Speicher **26** gelesen wurden; von dort werden sie seriell vom Shift-out-Register auf die Verbindungsmatrix **22** verschoben. Der Ausgangsfreigabepuffer **40** steuert die serielle Übertragung von Daten vom Shift-out-Register **38** zur Verbindungsmatrix **22**.

[0069] **Fig.** 8 zeigt ein zeitliches Ablaufdiagramm für die Speicherzugriffspuffer **24-1** bis **24-m** der **Fig.** 4–6. Das Zeitdiagramm von **Fig.** 8 veranschaulicht den zeitlichen Ablauf für ein Zweiunddreißig-Port-System (k = 32), wobei jeder Port eine Bündelgröße von zweiunddreißig Wörtern (n = 32) hat. Somit ist n = k. In einer vorliegenden Ausgestaltung alterniert der Speicherzugriff zwischen **32** Speicherlesezyklen, gefolgt von **32** Speicherschreibzyklen.

[0070] Konzentrieren wir uns zunächst auf den zeitlichen Ablauf von Speicherschreibzugriffen. Während der Speicherdatenschreibperiode (Taktimpulse 33–64) kann ein anderes Zweiunddreißig-Wort-Bündel von Daten auf den Gemeinschaftsspeicher **26** für jeden Port **20** geschrieben werden. Jeder Port hat einen Taktzyklus, während dessen durch diesen Port übertragene Daten auf den Speicher **26** geschrieben werden können. Die Daten aufeinander folgender Ports werden in aufeinander folgenden Taktzyklen geschrieben, die die jeweiligen Schreib-"Zeitschlitze" für die jeweiligen Ports sind. In Vorbereitung für den ihm zugeordneten Schreibzeitschlitz beginnt jeder Port mit der Übertragung von Daten auf die Speicherzugriffspuffer zweiunddreißig Taktzyklen vor seinem Zeitschlitz, so dass genügend Zeit vorhanden ist, um alle zweiunddreißig Wörter eines von einem Port empfangenen Bündels auf die dem Port zugewiesenen m Pufferregister (die sich jeweils in einem ande-

ren Speicherzugriffspuffer befinden) vor der Ankunft des Zeitschlitzes des Ports zu übertragen. Somit erfolgen während der zweiunddreißig Taktzyklen vor dem "w1.C" auf der Speicherdaten-Zeitlinie zweiunddreißig serielle Bitübertragungen von Port **20-1** über die Verbindungsmatrix 32 zum Speicherzugriffspuffer **24-1**, auf der Datenzeile Port **20-1** "w1.1 ", "w1.2" ... "w1.32" genannt. Da jeder Port einen anderen Zeitschlitz hat, werden die Übertragungen von den Ports auf geeignete Weise gestaffelt, so dass das Bündel jedes Ports unmittelbar vor dem Zeitschlitz des Ports für die Übertragung auf den Gemeinschaftsspeicher bereit ist. Spezifischer ausgedrückt, beispielsweise während jedes Datenschreibzyklus von Port **20-1** gemäß **Fig.** 8 werden alle Bits in einem ersten Wort, das durch Port **20-1** übertragen wird, von der Verbindungsmatrix **22** unter den zugewiesenen Doppelportregisterpaaren der Speicherzugriffspuffer **24** verteilt. Man nehme beispielsweise an, es gebe zweiunddreißig Wörter pro Bündel (n = 32) und m Bits pro Wort, dann werden während des ersten Datenschreibzyklus von Port **20-1** (w1.1) Bit 1 (B1) von Wort 1 (W1), durch Port **20-1** übertragen, seriell auf ein vorgeschriebenes Registerpaar in Puffer **24-1** übertragen; B2 von W1 wird seriell auf ein vorgeschriebenes Registerpaar in Puffer **24-2** geschrieben; B3 von W1 wird seriell auf ein Registerpaar in Puffer **24-3** geschrieben, usw. Bm von W1 wird seriell auf ein Registerpaar in Puffer **24-m** geschrieben.

[0071] Ebenso wird beispielsweise während des 32. Datenschreibzyklus von Port **20-1** (w1.32) B1 von W32, durch Port **20-1** übertragen, seriell auf dasselbe Registerpaar von Puffer **24-1** wie B1 von W1 geschrieben, übertragen durch Port **20-1**. Ebenso wird beispielsweise während des dritten Datenschreibzyklus von Port **20-1** (w1.3) B6 von W3, übertragen durch Port **20-1**, seriell auf dasselbe Registerpaar im Speicherzugriffspuffer **24-6** (nicht dargestellt) geschrieben wie B6 von W1, übertragen durch Port **20-1**.

[0072] Somit werden während jedes der zweiunddreißig Datenschreibzyklen für Port **20-1** (w1.1 bis w1.32) Daten seriell in designierte Shift-in-Register in den jeweiligen Speicherzugriffspuffern **24-1** bis **24-m** verschoben. Während des Speicherdatenzyklus w1.C werden alle Daten, die seriell von Port **20-1** während der zweiunddreißig vorherigen seriellen Einschubzyklen (w1.1 bis w1.32) in die Shift-in-Register der Puffer **24-1** bis **24-m** verschoben wurden, parallel auf den Bus **32** aufgedrückt, so dass sie auf den Gemeinschaftsspeicher **26** geschrieben werden können.

[0073] Ebenso können während der Speicherleseperiode (Taktimpulse 1–32) zweiunddreißig Wörter Daten vom Gemeinschaftsspeicher für jeden Port gelesen werden. So können beispielsweise während eines einzelnen Eintakt-Speicherzugriffszyklus bei r1.C zweiunddreißig Wörter vom Speicher **26** für die nachfolgende Übertragung auf ein externes Gerät durch Port **20-1** gelesen werden. Somit ist r1.C der Lese-"Zeitschlitz" für Port **20-1**. Ebenso repräsentiert beispielsweise r31.C den Lesezeitschlitz für Port **20-31**.

[0074] Insbesondere empfängt beispielsweise im Zeitschlitz r1.C ein Shift-out-Register im Puffer **24-1** parallel alle B1en der zweiunddreißig Wörter, die parallel vom Speicher **26** gelesen wurden. Ebenso empfängt beispielsweise in Zeitschlitz r1.C ein Shift-out-Register im Puffer **24-18** (nicht dargestellt) parallel alle B18en der zweiunddreißig Wörter, die parallel vom Speicher **26** gelesen wurden.

[0075] Während der zweiunddreißig Datenlesezyklen von Port **20-1** (r1.1 bis r1.32) werden die während des Speicherdatenlesezyklus (r1.C) vom Speicher **26** gelesenen Wörter seriell über die Verbindungsmatrix **22** zu Port **20-1** übertragen. So werden beispielsweise während des Datenlesezyklus (r1.1) alle Bits für das erste Wort (W1), das durch Port **20-1** austransferiert werden soll, gleichzeitig seriell von m unterschiedlichen Shift-out-Registern über die Matrix **22** in m unterschiedliche Puffer **24** verschoben, um W1 an Port **20-1** anzulegen. Ebenso werden beispielsweise während des Datenlesezyklus (r1.31) alle Bits für das einunddreißigste Wort (W31), das durch Port **20-1** austransferiert werden soll, gleichzeitig seriell von m unterschiedlichen Shift-out-Registern in m unterschiedliche Puffer **24** über die Matrix **22** verschoben, um W31 an Port **20-1** anzulegen.

[0076] Wenn also an einem vorgeschriebenen Port empfangene Daten auf den Speicher **26** geschrieben werden sollen, dann verteilt die Verbindungsmatrix die Datenwortbits unter den Puffern **24-1** bis **24-m**, so dass jeder Puffer nur eine Teilmenge jedes zu speichernden Wortes speichert. In der bevorzugten Ausgestaltung speichert jeder Puffer nur Bits einer vorgeschriebenen Ordnung. Somit braucht jeder Puffer nur eine Teilmenge aller Daten auf den Bus **32** zu setzen, die gleichzeitig zum Speicher übertragen werden sollen.

[0077] Umgekehrt, wenn vom Speicher gelesene Daten auf einen vorgeschriebenen Port ausgegeben werden sollen, dann rekombiniert die Verbindungsmatrix die zwischen mehreren Puffern verteilten Bits so, dass sie als eine Reihe (oder ein Bündel) von kompletten Wörtern ausgegeben werden. Die gesamten durch einen Port auszutransferierenden Daten werden parallel von dem Speicher ausgegeben und in die Puffer eingegeben. Jeder Puffer empfängt aber nur eine Teilmenge dieser gesamten Daten. Die Verbindungsmatrix **22** rekombiniert die Teilmenge zu einem Bündel von Datenwörtern.

[0078] Somit verbindet jeder der m jeweiligen Puffer jeweils immer eine n-Bit-Teilmenge mit ihrem parallelen Bus **32**. Wie in **Fig.** 4 illustriert, gibt es jeweils immer nur m × n Verbindungen zwischen den m Puffern und dem Gemeinschaftsspeicher **26**. Daher erfordert jeder Puffer weniger Anschlüsse für die Verbindung mit dem Gemeinschaftsspeicher, was eine geringere kapazitive Belastung aufgrund der geringeren Zahl von Verbindungsanschlüssen ergibt.

[0079] Das illustrative zeitliche Ablaufdiagramm von **Fig.** 9 zeigt den Betrieb des Doppelregisterpaares von **Fig.** 7. Während des P.WRITE Impulses werden Daten, die während eines (rX.c) Speicherdatenzyklus aus dem Gemeinschaftsspeicher **26** gelesen wurden, parallel auf das Shift-out-Register **38** geschrieben. Während jedes SHIFT-OUT Impulses wird ein einzelnes Bit aus dem Shift-out-Register zur Verbindungsmatrix **22** für die Übertragung auf einen jeweiligen Port ausgeschoben. So entspricht beispielsweise jeder SHIFT-OUT Impuls für Puffer **24-1** einem r1.X Impuls in **Fig.** 8. Während jedes SHIFT-IN Impulses wird ein einzelnes Datenbit in das Shift-in-Register eingeschoben. So entspricht z. B. jeder SHIFT-IN Impuls für Puffer **24-1** einem w1.X Impuls in **Fig.** 8. Während des P.READ Impulses können alle Daten, die in das Shift-in-Register eingeschoben wurden, parallel gelesen werden, um während eines w1.C Zyklus auf den Gemeinschaftsspeicher **26** geschrieben zu werden.

[0080] In der vorliegenden Ausgestaltung kann der Gemeinschaftsspeicher **26** eine standardmäßige Direktzugriffsspeicherkonfiguration (RAM) mit einer Speicherbreite von (n × m) sein. Somit kann der Speicher (n × m) Bit Daten zu oder von den Speicherzugriffspuffern **24** gleichzeitig übertragen und alle (n × m) Bits an einem einzigen adressierten Speicherort speichern, wie in **Fig.** 5 dargestellt ist. In der derzeitigen Ausgestaltung wird nicht jedes Wort von einem Port als zusammenhängende Bits gespeichert, sondern die Wörter werden "verschachtelt", so dass alle Bits der ersten Ordnung = Ordnungsbits (B1en) nebeneinander gespeichert werden, gefolgt von allen Bits zweiter Ordnung (B2en), usw. In dieser Konfiguration können alle Bits, die die n Wörter zu oder von einem spezifischen Port **20** repräsentieren, leicht als eine Zeile des Gemeinschaftsspeichers **26** gespeichert werden.

[0081] In der vorliegenden Ausgestaltung werden Zeitschlitze vorzugewiesen. Zeitschlitze können jedoch auch auf der Basis von Prioritäten zugeteilt werden. Außerdem ist es effizient, da in der derzeitigen Ausgestaltung jeder der k Ports **20** verlangt, dass ein einzelner Taktzyklus-Zeitschlitz mit dem Gemeinschaftsspeicher **26** kommuniziert und jeder Port n Taktzyklen vor diesem Zeitschlitz benötigt, um ein gesamtes n-Wort-Datenbündel zu oder von den Speicherzugriffspuffern **24** zu übertragen, die Anzahl der Wörter pro Bündel (n) gleich der Zahl der Ports (k) zu machen, so dass ein glatter, zyklischer Prozess zum Übertragen von Daten ohne Engpässe oder Ruhezeiten entsteht. Die Beziehung zwischen der Zahl der Ports und der Wörter pro Bündel kann sich jedoch ändern, ohne von der Erfindung abzuweichen.

[0082] **Fig.** 10 ist ein Blockdiagranm eines 32-Port Asynchronous Transfer Mode (ATM) Schalters gemäß der vorliegenden Erfindung. ATM ist eine Nutzdaten-Multiplexiertechnik für den Informationstransfer mit Paketen fester Größe, Zellen genannt. In einer derzeitigen Implementation hat eine ATM-Zelle eine Länge von 53 Bytes und besteht aus einem 5-Byte-Kopf, der die Routing-Informationen führt, gefolgt von einem Informationsfeld von 48 Byte (Nutzdaten). Die Nutzdaten jeder ATM-Zelle, die in den Schalter eintreten, werden auf eine spezifische Stelle im Gemeinschaftsspeicher gesetzt. Da die ATM-Zellennutzdaten eine Breite von 48 Bytes (8 Bit pro Byte) haben, erhält ein Gemeinschaftsspeicher eine Breite von 384 Bit (48 × 8 = 384). Diese Implementation wurde so gewählt, dass sie einen 32-Bit-Bus von den Speicherzugriffspuffern zum Speicher hat, jeder Port mit einer Wortbreite von 12 Bit, und mit Portbündeln von zweiunddreißig Wörtern. Man beachte, dass 384 auch gleich (12 × 32) ist, d. h., eine Zelle wird in einem 32-Wort-Bündel von 12-Bit-Wörtern übertragen.

[0083] Der ATM-Schalter leitet jede empfangene Zelle auf einem Port zu einem Zielport gemäß Routing-Informationen, die in jeder solchen Zelle enthalten sind. Spezifischer ausgedrückt, eine Zelle wird von einem Port übertragen und in einem Gemeinschaftsspeicher gespeichert. Dann wird sie aus dem Gemeinschaftsspeicher abgerufen und zu einem in der Zelle angegebenen Zielport übertragen. Auf diese Weise können Daten zwischen Ports geschaltet werden. Der Betrieb des Steuerspeichers und des Schalter-Controllers wird von der Fachperson verstanden und ist nicht Bestandteil der Erfindung und braucht hierin nicht beschrieben zu werden.

[0084] Die illustrative Zeichnung von **Fig.** 11 zeigt ein Blockdiagramm eines Schaltsystems **98** mit einer Speicherschnittstelleneinheit (MIU) **100** gemäß einer derzeit bevorzugten Ausgestaltung der Erfindung. Die MIU **100** beinhaltet einen konfigurierbaren Schalter **104** und einen Datentransferpuffer **106**. Ein Busschnittstellenregister **102** ist für die Übertragung digitaler Informationen zu und von einem Burst-Modus-Bus **108** angeschlossen. Der Datentransferpuffer **106** hat die Aufgabe, Daten im Transit zwischen einem digitalen Speicher (nicht dargestellt) und dem konfigurierbaren Schalter **104** vorübergehend zu speichern. Digitale Daten werden über einen E/A-Treiber **110** zu und von dem Speicher übertragen. Ein Adressregister **112** empfängt digitale Adressinformationen vom Bus **108**. Von dem Register **112** empfangene Adressinformationen werden über einen weiteren E/A-Treiber **114** zum digitalen Speicher geleitet. Eine Steuereinheit **116** steuert den Betrieb der MIU **100** und des Speichers.

[0085] Der konfigurierbare Schalter **104** einer vorliegenden Ausgestaltung der Erfindung kann selektiv konfiguriert werden, um Bündelteilmengen zwischen dem Busschnittstellenregister **102** und dem Transferpuffer **106** oder den Master- (m_n) Verbindungen von Port p0, p1 oder p2 zu übertragen. Der Schalter **104** kann ferner so konfiguriert werden, dass er Bündelteilmengen zwischen dem Transferpuffer **106** und entweder dem Busschnittstellenregister **102** oder den Slave-Verbindungen (s_n) von Port p0, p1 oder p2 überträgt. In einer derzeit bevorzugten Ausgestaltung der Erfindung kann der Schalter **104** auf eine von drei

unterschiedlichen Weisen konfiguriert werden: selbstständig, 2 MIUs und 4 MIUs. In der selbstständigen Konfiguration werden Daten direkt durch den Schalter **104** von Leitungen b0 bis b3 zu Leitungen c0 bis c3 geleitet. Die Verbindungen im Zwei-MIU-Fall sind in **Fig.** 19 illustriert. Die Verbindungen im Vier-MIU-Fall sind in **Fig.** 26 illustriert.

[0086] Das Schaltsystem der vorliegenden Erfindung erlaubt somit die gemeinsame Nutzung von Speicherkapazität durch unterschiedliche Ports, wobei jeder Port mit einem anderen Burst-Modus-Bus verbunden ist. So erlaubt beispielsweise ein Burst-Modus-Bus wie PCI, SBUS oder GIO die Bereitstellung einer Adresse, die die auf dem Bus vorliegenden Daten begleitet. Darüber hinaus können einzelne MIUs miteinander "verkettet" oder "gestapelt" werden, um den Zugriff auf einen Gemeinschaftsspeicher durch Geräte zu koordinieren, die mit unterschiedlichen Burst-Modus-Bussen verbunden sind.

[0087] **Fig.** 12 ist ein illustratives Blockdiagramm, das die Datenleitungen und Steuerleitungen des Schaltsystems **98** von **Fig.** 11 ausführlicher zeigt. Das System **98** überträgt Signale zu und von einem Bus. In einer derzeitigen Ausgestaltung ist der Bus ein PCI-Bus. Die MIU kann jedoch auch mit anderen Typen von Burst-Modus-Bussen wie z. B.

[0088] SBUS, GIO usw. kommunizieren. Es können Daten-, Adress- und Lese/Schreib-Steuersignale an einen mit dem Schaltsystem **98** verbundenen digitalen Speicher angelegt werden. Das System **98** hat drei Ports, p0, p1 und p2, die beispielsweise zum Übertragen von Signalen zu anderen Schaltsystemen (nicht dargestellt) verwendet werden können. Jeder Port beinhaltet acht Master-Datenleitungen und eine Master-Anforderungsleitung. Jeder Port hat auch acht Slave-Datenleitungen und eine Slave-Anforderungsleitung. Daten werden zu und von dem in **Fig.** 11 gezeigten Transferpuffer **106** übertragen. Adressen kommen aus dem Adressregister **112**. L/S-Signale werden vom Controller **116** erzeugt. Die Portverbindungen bieten Zugang zum konfigurierbaren Schalter **104**.

[0089] Wieder mit Bezug auf **Fig.** 11 wird nun der Betrieb des neuartigen Schaltsystems **98** erläutert. Dazu wird zunächst auf einen Speicherschreibvorgang und dann auf einen Speicherlesevorgang eingegangen. Während eines Schreibvorgangs wird ein Bündel von digitalen Informationen vom Bus **108** auf das Busschnittstellenregister **102** übertragen. In einer derzeitigen Ausgestaltung hat der Bus **108** eine Breite von 32 Bit und das Bündel kann ein 32 Bit breites Adresswort gefolgt von vier 32 Bit breiten Datenwörtern enthalten. Das Bündel wird über die Multibit-Leitungen b0, b1, b2 und b3 zum konfigurierbaren Schalter **104** übertragen. Der Schalter kann so konfiguriert werden, dass er bestimmte Teilmengen des Bündels über eine oder mehrere Multibit-Leitungen c0, c1, c2 und c3 direkt zum Datentransferpuffer **106** überträgt. Der Schalter **104** kann auch so konfiguriert werden, dass er die übrigen Teilmengen des Bündels über eine oder mehrere der Master-Verbindungen m0, m1 und m2, auf die jeweils durch Port 0, Port 1 und Port 2 zugegriffen werden kann, zu einem oder mehreren anderen Schaltsystemen (nicht dargestellt) sendet, die mit anderen Speichern (nicht dargestellt) assoziiert sind. Die über die Master-Verbindung(en) zu anderen Schaltsystemen gesendeten Teilmengen werden in den digitalen Speichern gespeichert, die mit diesen anderen Systemen assoziiert sind. Die Speicherung von zu anderen Speichern übertragenen Teilmengen wird nachfolgend in Verbindung mit beispielhaften Zwei-MIU- und Vier-MIU-Systemen erörtert. Der Transferpuffer **106** speichert vorübergehend die zu ihm übertragenen Teilmengen und sendet sie dann über eine oder mehrere der Multibit-Leitungen d0, d1, d2 und d3 zum E/A-Treiber **110**, der sie zur Speicherung zu einem mit dem Schaltsystem **98** assoziierten digitalen Speicher sendet. Von dem Bus **108** empfangene digitale Adresssignale in Verbindung mit dem Bündel werden zum Designieren der Speicherorte verwendet, an denen die Teilmengen gespeichert werden sollen. In einer derzeitigen Ausgestaltung wird dieselbe Speicheradresse sowohl für den Speicher in Verbindung mit dem Schaltsystem **98** als auch für andere Speicher in Verbindung mit anderen Schaltsystemen verwendet. Somit können unterschiedliche Teilmengen des Bündels an derselben Adresse in unterschiedlichen Speichern gespeichert werden.

[0090] Zu Beginn eines Speicherlesevorgangs können die Teilmengen eines Bündels zwischen mehreren verschiedenen digitalen Speichern (in **Fig.** 11 nicht dargestellt) verteilt werden, auf die durch mehrere verschiedene Schaltsysteme (in **Fig.** 11 nicht dargestellt) zugegriffen werden kann. Gemäß **Fig.** 11 wird während eines Lesevorgangs ein Adresssignal auf den Bus gesetzt. Die Adresse wird durch das Adressregister **112** empfangen und zu jedem dieser verschiedenen Speicher gesendet. Es ist zu verstehen, dass **Fig.** 11 nur ein einziges Schaltsystem **98** zeigt, das Zugang zu einem einzelnen dieser Speicher bietet. Wie nachfolgend erläutert, können andere ähnliche Schaltsysteme angeschlossen sein, die Zugang zu anderen Speichern bieten. Das Adresssignal dient zum Zurückspeichern von verschiedenen Bündelteilmengen von den verschiedenen Speichern, in denen sie gespeichert sind. In dem Speicher gespeicherte Bündelteilmengen, auf die durch das Schaltsystem **98** zugegriffen wird, werden von diesem Speicher zum E/A-Treiber **110** gesendet, der solche Teilmengen über eine oder mehrere der Multibit-Leitungen d0, d1, d2 und d3 zur vorübergehenden Speicherung zum Transferpuffer **106** überträgt. Der Puffer **106** sendet solche Teilmengen wiederum zum konfigurierbaren Schalter **104** über eine oder mehrere der Multibit-Leitungen c0, c1, c2 und c3. Der konfigurierbare Schalter **104** kann so konfiguriert werden, dass er sie direkt zu einer oder mehreren der mit dem Busschnittstellenregister **102** verbundenen Multibit-Leitungen b0, b1, b2 und b3 sendet. Die Teil-

mengen, die nicht in dem Speicher gespeichert sind, auf die durch das Schaltsystem **98** zugegriffen wird, werden über eine oder mehrere der Multibit-Slave-Verbindungen s0, s1 und s2 jeweils von Port1, Port2 bzw. Port3 zum Schaltsystem **98** übertragen. Spezifischer ausgedrückt, diese in anderen Speichern gespeicherten Teilmengen werden zunächst aus diesen Speichern zurückgespeichert und dann zu einem oder mehreren der Slave-Verbindungen des Schaltsystems **98** übertragen. Die Rückspeicherung von Teilmengen aus diesen anderen Speichern wird nachfolgend in Verbindung mit beispielhaften Zwei-MIU- und Vier-MIU-Systemen erörtert. Der konfigurierbare Schalter **104** kann so konfiguriert werden, dass er Teilmengen an seinen Slave-Verbindungen mit einem oder mehreren der Multibit-Leitungen b0, b1, b2 und b3 zum Register **102** überträgt.

[0091] **Fig.** 13 ist ein vereinfachtes Blockdiagramm des Busschnittstellenregisters **102**, des konfigurierbaren Schalters **104** und des Transferpuffers **106** von **Fig.** 11, das eine mögliche Konfiguration eines Schalters **104** gemäß einer derzeitigen Ausgestaltung der Erfindung zeigt. Der Schalter **104** (mit gestrichelten Linien angedeutet) ist so konfiguriert, dass er Bündelteilmengen zwischen dem Puffer **106** und dem Register **102** über einen "direkten" Verbindungspfad **105** überträgt. Master-Verbindungen m0, ml und m2 erzeugen Pfade für die Übertragung von Bündelteilmengen zu und von dem Register **102**. Slave-Verbindungen s0, s1 und s2 erzeugen Pfade für die Übertragung von Bündelteilmengen zu und von dem Transferpuffer **106**.

[0092] Weiter mit Bezug auf **Fig.** 13, während eines Speicherschreibvorgangs wird z. B. ein Bündel vom Busschnittstellenregister **102** empfangen. Eine erste Bündelteilmenge wird über den direkten Pfad **105** zum Puffer **106** übertragen. Eine zweite Bündelteilmenge wird über den Master-Pfad m0 vom Register **102** weg übertragen. Eine dritte Bündelteilmenge wird über den Master-Pfad m1 von dem Register weg übertragen. Eine vierte Bündelteilmenge wird über den Master-Pfad m2 von dem Register weg übertragen. Somit kann die erste Bündelteilmenge in einem mit dem Puffer **106** verbundenen Speicher (nicht dargestellt) gespeichert werden, und die zweite, dritte und vierte Bündelteilmenge können an anderer Stelle in Speichern gespeichert werden, auf die über die Pfade m0, ml und m2 zugegriffen wird.

[0093] Umgekehrt wird während eines Speicherlesevorgangs beispielsweise eine erste Bündelteilmenge zunächst in einem mit dem Puffer **106** verbundenen Speicher (nicht dargestellt) gespeichert, und eine zweite, dritte und vierte Teilmenge werden an anderer Stelle in unterschiedlichen digitalen Speichern (nicht dargestellt) gespeichert. Der Transferpuffer **106** kann die erste Teilmenge vorübergehend speichern, wenn sie von dem mit diesem Puffer **106** verbundenen Speicher zurückgespeichert wird. Der Puffer überträgt dann die erste Teilmenge über den direkten Pfad **105** zu dem Register. Eine zweite Bündelteilmenge wird über den Slave-Pfad s0 empfangen. Eine dritte Bündelteilmenge wird über den Slave-Pfad s1 empfangen. Eine vierte Teilmenge wird über den Slave-Pfad s2 empfangen. Somit werden die Bündelteilmengen, die zwischen unterschiedlichen Speichern aufgeteilt sind, am Register **102** für die Übertragung zu einem Bus gesammelt.

[0094] Man beachte, dass die Konfiguration in **Fig.** 13 nur eine mögliche Konfiguration des Schalters **104** repräsentiert. So könnte der Schalter beispielsweise stattdessen so konfiguriert werden, dass er mehrere Direktpfade zwischen der Busschnittstellenlogik **116** und dem Puffer **106** bereitstellt. Die nachfolgende Erörterung in Verbindung mit einem Zwei-MIU-System **120** und einem Vier-MIU-System **106** erläutert einige beispielhafte alternative Schalterkonfigurationen.

[0095] **Fig.** 14A zeigt ein Blockdiagramm eines konfigurierbaren Schalters **104** einer derzeit bevorzugten Ausgestaltung, die viele der Signalleitungen des Schalters **104** illustriert. Der Schalter wird als Multiplexer auf eine Weise implementiert, die eine durchschnittliche Fachperson leicht verstehen wird. So ist **Fig.** 14B beispielsweise ein Logikdiagramm auf Gatterebene, das die interne Logik illustrier, die zum Erzeugen des c0-Ausgangs während eines "Lese"-Vorgangs durch eine nachfolgend beschriebene Zwei-MIU-Implementation verantwortlich ist. **Fig.** 14C ist ein Logikdiagramm auf Gatterebene, das die interne Logik illustrier, die zum Erzeugen des b0-Ausgangs während eines "Schreib"-Vorgangs durch eine nachfolgend beschriebene Zwei-MIU-Implementation verantwortlich ist. In der Implementation mit zwei MIUs kann der p0-Eingang beispielsweise die rechte MIU und p1 die linke MIU bezeichnen.

[0096] Die folgenden Gleichungen repräsentieren die vollständige Funktionalität des konfigurierbaren Schalters (Multiplexers) **104** während eines Schreibvorgangs für die Zwei-MIU-Implementation.

$$c0 = P1 \cdot s0 + P0 \cdot b0$$

$$c1 = P0 \cdot s0 + P1 \cdot b1$$

$$c2 = P1 \cdot s2 + P0 \cdot b2$$

$$c3 = P0 \cdot s2 + P1 \cdot b3$$

$$m0 = P1 \cdot b0 + P0 \cdot b1$$

$$m2 = P1 \cdot b2 + P0 \cdot b3$$

[0097] Die folgenden Gleichungen repräsentieren die vollständige Funktionalität des konfigurierbaren Schalters (Multiplexers) **104** während eines Lesevorgangs für die Zwei-MIU-Implementation.

$$b0 = P1 \cdot m0 + p0 \cdot c0$$

$$b1 = P0 \cdot m0 + P1 \cdot c1$$

b2 = P1·m2 + P0·c2

b3 = P0·m2 + p1·c3

s0 = P1·c0 + P0·c1

s2 = p1·c2 + P0·c3

[0098] Es ist klar, dass in der nachfolgend beschriebenen Vier-MIU-Implementation vier unterschiedliche Logikwerte P0, P1, P2 und P3 vorliegen, die zum Unterscheiden der vier unterschiedlichen MIUs verwendet werden können. Die Grundmultiplexiertechniken und der Logikschaltkomplex, die zum Umschalten von Signalen zwischen den vier MIUs verwendet werden, sind jedoch der Fachperson hinlänglich bekannt und brauchen hier nicht näher beschrieben zu werden.

[0099] Die illustrative Zeichnung von **Fig.** 15 zeigt zusätzliche Details des Transferpuffers **106** von **Fig.** 11. In einer derzeit bevorzugten Ausgestaltung beinhaltet der Puffer **106** vier Speicherzugriffspuffer **106-1**, **106-2**, **106-3** und **106-4**. Die Speicherzugriffspuffer empfangen jeweilige P_READ, P_WRITE, OE(0-3 und W(0-3) Steuersignale. Der konfigurierbare Schalter **104** greift auf Speicherzugriffspuffer zu, auf die über Byte-breite Ein-/Ausgabeleitungen c0, c1, c2 und c3 zugegriffen werden kann. In einer derzeitigen Implementation hat jede serielle Leitung eine Breite von einem Byte. Es sind jedoch auch Multibit-Fragmente einer Bündelteilmenge mit anderen Größen möglich. Auf den Speicher (nicht dargestellt) wird durch die Speicherzugriffspuffer über die Leitungen d0, d1, d2 und d3 über einen 32-Bit-Speicherbus **224** zugegriffen.

[0100] Der Speicherbus **224** überträgt digitale Informationen zwischen einzelnen Speicherzugriffspuffern und einem mit dem Puffer **106** verbundenen digitalen Speicher (nicht dargestellt). Während eines Speicherschreibvorgangs könnte beispielsweise eine Bündelteilmenge parallel (alle auf einmal) vom Speicherzugriffspuffer **106-1** über den Bus **224** zum Speicher übertragen werden. Während eines Speicherlesevorgangs könnte beispielsweise die Bündelteilmenge parallel (alle auf einmal) vom Speicher über den Bus **224** zum Speicherzugriffspuffer **106-1** übertragen werden.

[0101] Die illustrativen Zeichnungen von **Fig.** 16 zeigen Details eines beispielhaften Speicherzugriffspuffers **106-2** gemäß einer derzeitigen Ausgestaltung der Erfindung. Ein Shift-in-Register **226** empfängt eine Reihe von Byte-breiten Bündelteilmengenfragmenten auf den Leitungen c1, bis sich eine gesamte Bündelteilmenge angesammelt hat. Das Shift-in-Register kann die vollständige Bündelteilmenge vorübergehend speichern. Die gesamte Teilmenge wird dann parallel (alle Byte-breiten Fragmente auf einmal) auf den Speicherbus **224** aufgedrückt, so dass die Teilmenge parallel auf den digitalen Speicher geschrieben werden kann.

[0102] Bei Speicherlesevorgängen wird ein Shift-out-Register **228** verwendet. Eine vom Speicher gelesene ganze Bündelteilmenge wird parallel [lacuna] Speicherbus 224 zum Shift-out-Register **228** gesendet. Das Shift-out-Register kann die gesamte Bündelteilmenge vorübergehend speichern. Das Register kann dann eine Reihe von Byte-breiten Bündelteilmengenfragmenten auf die Leitungen c1 setzen, bis die gesamte Bündelteilmenge an dem konfigurierbaren Schalter **104** anliegt. Der Ausgangspuffer **230** steuert die Übertragung von Teilmengenfragmenten vom Shift-out-Register zu Leitungen c1.

[0103] Es ist zu verstehen, dass in einer derzeitigen Implementation der Erfindung jedes einzelne Schaltsystem eine Steuereinheit beinhaltet, die Steuersignale erzeugt, die den Betrieb des zugehörigen Transferpuffers steuern. Wenn mehrere Schaltsysteme zusammengeschaltet sind, wie z. B. in der Zwei-MIU-Konfiguration oder der Vier-MIU-Konfiguration, dann müssen die Controller der unterschiedlichen Schaltsysteme koordiniert werden, um eine gemeinsame Speichernutzung zu erzielen. Diese Koordination kann beispielsweise dadurch erzielt werden, dass jeder Controller einen Algorithmus abarbeitet, der zum Koordinieren des Zugriffs mehrerer Schaltsysteme auf einen Gemeinschaftsspeicher verwendet wird. Es können mehrere hinlänglich bekannte Algorithmen verwendet werden. So können beispielsweise Anforderungen für Speicherzugriff durch unterschiedliche Schaltsysteme gemäß einer festen Priorität zwischen Schaltern oder nach einem zyklischen Schema gehandhabt werden.

[0104] **Fig.** 17 ist ein illustratives Blockdiagramm eines Zwei-MIU-Systems **120** gemäß einer derzeitigen Ausgestaltung der Erfindung. Der Einfachheit halber sind Details wie die Steuerlogik und die Busschnittstellenlogik nicht dargestellt. Zwei MIUs **122** und **124** sind wie gezeigt zusammengeschaltet. Das heißt, die MIU-Schalter **123** und **125** sind so konfiguriert, dass m0 und s0 Verbindungen von MIU-Schalter **123** und MIU-Schalter **125** durch ihre jeweiligen Ports p0 miteinander verbunden sind. Ebenso sind die m2- und s2-Verbindungen von MIU-Schalter **123** und MIU-Schalter **125** durch ihre jeweiligen Ports p2 miteinander verbunden. Der Schalter **123** ist ferner so konfiguriert, dass er direkte Verbindungen **141** und **142** zwischen Busschnittstellenlogik **126** und Puffer **130** bereitstellt. Der Schalter **125** ist ferner so konfiguriert, dass er direkte Verbindungen **143** und **144** zwischen Busschnittstellenlogik **128** und Puffer **132** erzeugt. Der Puffer **130** ist zum Übertragen von Bündelteilmengen zu und von dem digitalen Speicher **134** geschaltet. Der Puffer **132** ist zum Übertragen von Bündelteilmengen zu und von dem digitalen Speicher **136** geschaltet.

[0105] **Fig.** 18 ist eine vereinfachte Seitenansicht von zwei gestapelten Boxen **156** und **158**, die gemeinsam das Zwei-MIU-System **120** aufnehmen. Die Box **156** beherbergt MIU **122**, Speicher **134** und Busschnittstellenlogik **126**. Box **158** beherbergt MIU **124**,

Speicher **136** sowie Busschnittstellenlogik **128**. Jede Box **156** und **158** hat drei Ports p0, p1 und p3. Jede Box hat auch einen gemeinschaftlichen Adressbus, über den sie Adressinformationen mit (einer) Nachbarboxen) gemeinsam nutzen kann. Die Ports p0 und die Ports p2 sowie der gemeinschaftliche Adressbus sind wie gezeigt verbunden.

[0106] **Fig.** 19 ist eine Matrix, die die Verbindungen zwischen den beiden MIUs **122** und **124** des Zwei-MIU-Systems von **Fig.** 17 repräsentiert. Eine "0" auf einer Leitungsschnittstelle zeigt eine Verbindung innerhalb des MIU-Schalters **23** an. Eine "1" an einer Leitungsschnittstelle zeigt eine Verbindung im MIU-Schalter **25** an. Der linke obere Viertelkreis repräsentiert Verbindungen zwischen Slave-Leitungen s0, s1 und s2 und Busschnittstellenlogikleitungen b0, b1, b2 und b3. Der rechte untere Viertelkreis repräsentiert Verbindungen zwischen Master-Leitungen m0, ml und m2 sowie MIU-Pufferleitungen c0, c1, c2 und c3. Der linke untere Viertelkreis repräsentiert Verbindungen zwischen Busschnittstellenlogikleitungen b0, b1, b2 und b3 sowie MIU-Pufferleitungen c0, c1, c2 und c3. Die Verbindungen für den MIU-Schalter **123** sind wie folgt: b0 direkt mit c0 verbunden; b1 mit m0 verbunden; b2 direkt mit c2 verbunden; b3 mit m2 verbunden; c1 mit s0 verbunden; und c3 mit s2 verbunden. Die Verbindungen für den MIU-Schalter **125** lauten wie folgt: b0 mit m0 verbunden; b1 direkt mit c1 verbunden; b2 mit m2 verbunden; b3 direkt mit c3 verbunden; c0 mit s0 verbunden; und c2 mit s2 verbunden.

[0107] Beim Betrieb werden Bündel zwischen den MIUs **122** und **124** und einem externen Bus (nicht dargestellt) über die Busschnittstellenlogikeinheiten **126** und **128** übertragen. Der Einfachheit halber sind die mit einem externen Bus verbundenen Adress- und Datenregister so dargestellt, dass sie zu Busschnittstellenlogikeinheiten zusammengefasst sind. Es ist ein gesamtes beispielhaftes Vier-Wort-Bündel **127** dargestellt, das durch die Logik **126** übertragen wird. Das Bündel **127** beinhaltet das Digitalwort, das als Multibit-Fragmente A00, A01, A02 und A03 enthaltend gekennzeichnet ist. In einer derzeitigen Ausgestaltung der Erfindung beinhaltet jedes Fragment mehrere Logikbits. Die Erfindung kann jedoch auch unter Verwendung von Einzelbit-Fragmenten als Alternative implementiert werden. Die Bündelteilmenge, die A00, A10, A20 und A30 beinhaltet, und die Bündelteilmenge, die A02, A12, A12 und A32 beinhaltet, werden zwischen der Busschnittstellenlogik **126** und dem Puffer **130** über Direktverbindungen **141** und **142** übertragen. Die Bündelteilmenge, die A01, A11, A21 und A31 beinhaltet, wird zwischen der Busschnittstellenlogik **126** und dem Puffer **132** über die Port-p0-Verbindungen der Schalter **123** und **125** übertragen. Die Bündelteilmenge, die A03, A13, A23 und A33 beinhaltet, wird zwischen der Busschnittstellenlogik **126** und dem Puffer **132** über die Port-p2-Verbindungen der Schalter **123** und **125** übertragen.

[0108] Ebenso ist ein gesamtes beispielhaftes Vier-Wort-Bündel **129** dargestellt, das durch Logik **128** übertragen wird. Das Bündel beinhaltet das digitale Wort, das als Multibit-Fragmente B00, B01, B02 und B03 enthaltend gekennzeichnet ist. Die Bündelteilmenge, die B00, B10, B20 und B30 enthält, sowie die Bündelteilmenge, die B02, B12, B22 und B32 enthält, werden zwischen der Busschnittstellenlogik **128** und dem Puffer **132** über direkte Verbindungen **143** und **145** übertragen. Die Bündelteilmenge, die B01, B11, B21 und B31 enthält, wird zwischen der Busschnittstellenlogik **128** und dem Puffer **130** über die Port-p0-Verbindungen der Schalter **123** und **125** übertragen. Die Bündelteilmenge, die B03, B13, B23 und B33 enthält, wird zwischen der Busschnittstellenlogik **128** und dem Puffer **130** über die Port-p2-Verbindungen der Schalter **123** und **125** übertragen.

[0109] Zwei Teilmengen von Bündel **127**, das durch die Busschnittstellenlogik **126** übertragen wird, und zwei Teilmengen von Bündel **129**, das durch die Busschnittstellenlogik **128** übertragen wird, können wie gezeigt in jedem der beiden Puffer **130** und **132** vorübergehend gespeichert werden. Ebenso werden die beiden Teilmengen von jedem der Bündel im digitalen Speicher **134** und zwei Teilmengen von jedem der beiden Bündel im digitalen Speicher **136** wie gezeigt gespeichert. Somit wird der Inhalt von jedem Bündel, das durch die Busschnittstellenlogikeinheit **126** oder **128** übertragen wird, zwischen zwei unterschiedlichen digitalen Speichern verteilt und vorübergehend dort gespeichert. In einer derzeitigen Ausgestaltung wird SRAM eingesetzt. Es können jedoch im Einklang mit der Erfindung auch andere Speicherarchitekturen wie z. B. DRAM verwendet werden.

[0110] **Fig.** 20 illustriert den Fluss von Bündelteilmengen durch das Zwei-MIU-System **120** von **Fig.** 17 während eines repräsentativen Speicherschreibvorgangs. Ein Fünf-Wort-Bündel **146** wird von der Busschnittstellenlogik (nicht dargestellt) empfangen. Das erste Wort des Bündels enthält Adressinformationen. Die nächsten vier Wörter enthalten digitale Informationen, die auf digitale Speicher **134** und **136** geschrieben werden sollen. Die Adressinformationen werden von der MIU **122** auf ein MIU-Adressregister **148** übertragen. Das MIU-Adressregister **148** legt die Bündeladresse an die Speicheradressregister **150** und **152** der digitalen Speicher **134** und **136** über die Verbindung **154** an. Die Bündelteilmenge d00, d10, d20 und d30 sowie die Bündelteilmenge d02, d12, d22 und d32 werden von der MIU **122** zum digitalen Speicher **134** zur Speicherung an einem Adressort übertragen, der vom Speicheradressregister **150** designiert wird. Die Bündelteilmenge d31, d21, d11 und d01 wird von der MIU **122** zur MIU **124** über ihre jeweiligen Port-p0-Verbindungen und dann durch MIU **124** zum digitalen Speicher **136** zur Speicherung an einem Speicherort übertragen, der vom Speicheradressregister **152** designiert wird. Ebenso wird die Bündelteilmenge d33, d23, d13 und d03 von der MIU **122** zur MIU **124** über deren jeweiligen Port-p2-Ver-

bindungen und dann von der MIU **124** zum digitalen Speicher **136** zur Speicherung an einem Speicherort übertragen, der vom Speicheradressregister **152** designiert wird. Jeder der von der MIU **122** zur MIU **124** übertragenen Bündelteilmengen steht ein Rq-Signal voran, das anzeigt, ob ein Lese- oder ein Schreibvorgang erfolgen soll.

[0111] **Fig.** 21 ist ein Zeitdiagramm, das den zeitlichen Ablauf von Daten- und Steuersignalen in Verbindung mit dem Speicherschreibvorgang von **Fig.** 17 für das Zwei-MIU-System **120** von **Fig.** 17 illustriert. Während der Taktzyklen 1–2 wird eine Adresse vom Bus empfangen. Während der Taktzyklen 2–3 wird ein Datenwort D0 vom Segmentbus empfangen, und die Rq-Signale werden über die p0- und p2-Verbindungen zur MIU **124** gesendet. Während der Taktzyklen 3–4 wird das Datenwort D1 vom Segmentbus empfangen, und Bündelteilmengenfragmente d01 und d03 werden jeweils über die Multibit-Master-Verbindungen von Port p0 bzw. p2 zur MIU **124** übertragen. Während der Taktzyklen 4–5 wird das Datenwort D2 vom Segmentbus empfangen, und Bündelteilmengenfragmente d11 und d13 werden jeweils über die Multibit-Master-Verbindungen von Port p0 bzw. p2 zur MIU **124** übertragen. Während der Taktzyklen 5–6 wird das Datenwort D3 vom Segmentbus empfangen, und Bündelteilmengenfragmente d21 und d23 werden jeweils über die Multibit-Master-Verbindungen von Port p0 bzw. p2 zur MIU **124** übertragen. Während der Taktzyklen 6–7 werden Bündelteilmengerfragmente d31 und d33 jeweils über die Multibit-Master-Verbindungen von Port p0 bzw. p2 zur MIU **124** übertragen. Ebenso wird während der Taktzyklen 5–7 eine Speicheradresse in Verbindung mit dem Bündel in den Registern **150** und **154** an jeden der digitalen Speicher **134** und **136** angelegt. Während der Taktzyklen 6–8 wird ein Speicherschreibfreigabesignal gleichzeitig durch zwei Steuereinheiten aufgedrückt, die die beiden MIUs steuern, und die gesamten in den Transferpuffern **130** und **132** gespeicherten Datenbündelteilmengen werden parallel auf die jeweiligen digitalen Speicher **134** und **136** geschrieben.

[0112] **Fig.** 22 illustriert den Fluss von Bündelteilmengen durch das Zwei-MIU-System **120** von **Fig.** 17 während eines repräsentativen Speicherlesevorgangs. Zunächst werden die vier Wörter eines Bündels über zwei unterschiedliche digitale Speicher **134** und **136** verteilt. Eine Speicheradresse **125** wird zum MIU-Adresspuffer **148** gesendet, der die Adresse über die Verbindung **154** zu den Speicheradressregistern **150** und **152** weiterleitet. Der digitale Speicher **134** sendet die Bündelteilmenge d00, d10, d20 und d30 zur MIU **122**, die sie direkt zur Busschnittstellenlogik **126** überträgt. Der digitale Speicher **134** sendet auch die Bündelteilmenge d02, d12, d22 und d32 zur MIU **122**, die sie direkt zur Busschnittstellenlogik (nicht dargestellt) überträgt. Der digitale Speicher **136** überträgt die Bündelteilmenge d01, d11, d21 und d31 zur MIU **124**, die sie über ihre Port-p0-Verbindungen zur MIU **122** überträgt. Die MIU **122** überträgt die Teilmenge dann zu ihrer Busschnittstellenlogik. Ebenso überträgt der digitale Speicher **136** die Bündelteilmenge d03, d13, d23 und d33 zur MIU **124**, die sie über ihre Port-p2-Verbindungen zur MIU **122** überträgt. Die MIU **122** überträgt die Teilmenge dann zu ihrer Busschnittstellenlogik. Somit wird das Bündel **146** von unterschiedlichen Speichern zurückgespeichert und an der Busschnittstellenlogik zur Verfügung gestellt.

[0113] **Fig.** 23 ist ein Zeitdiagramm, das den zeitlichen Ablauf von Daten- und Steuersignalen in Verbindung mit dem Speicherlesevorgang von **Fig.** 22 für das Zwei-MIU-System **120** von **Fig.** 17 illustriert. Während der Taktzyklen 1–2 wird ein Adresssignal vom Segmentbus empfangen. Während der Taktzyklen 2–3 werden von der Busschnittstellenlogik erzeugte Rq-Signale über die p0- und p2-Verbindungen der MIU **124** gesendet. Während der Taktzyklen 4–6 wird eine Speicheradresse in jedem der Speicheradressregister **150** und **152** zu den Speichern **134** und **136** gesendet. Während der Taktzyklen 5–7 wird ein Speicherausgangsfreigabesignal zu jedem der Speicher gesendet, und ganze Datenbündelteilmengen werden parallel aus jedem der beiden Speicher **134** und **136** gelesen und zur vorübergehenden Speicherung zu Transferpuffern **130** und **132** gesendet. Während der Taktzyklen 6–7 werden Fragmente d01 und d03 über die Slave-Verbindungen von Port p0 bzw. p2 zur MIU **122** übertragen. Während der Taktzyklen 7–8 werden Fragmente d11 und d13 über die Slave-Verbindungen von Port p0 bzw. p2 zur MIU 122 übertragen, und das Datenwort D0 wird auf den Segmentbus gesetzt. Während der Taktzyklen 8–9 werden Fragmente d21 und d23 über die Slave-Verbindungen von Port p0 bzw. p2 zur MIU **122** übertragen, und das Datenwort D 1 wird auf den Segmentbus gesetzt. Während der Taktzyklen 9–10 werden Fragmente d31 und d33 über die Slave-Verbindungen von Port p0 bzw. p2 zur MIU **122** übertragen, und das Datenwort D2 wird auf den Segmentbus gesetzt. Während der Taktzyklen 10–11 wird das Datenwort D3 auf den Segmentbus gesetzt.

[0114] **Fig.** 24 ist ein vereinfachtes Blockdiagramm eines Vier-MIU-Systems **160** gemäß einer derzeit bevorzugten Ausgestaltung der Erfindung. Details von Steuerlogik und Busschnittstellenlogik wurden weggelassen, um die nachfolgende Erläuterung zu vereinfachen. Das System **160** beinhaltet vier MIUs **162**, **164**, **166** und **168**. Jede MIU enthält eine konfigurierbare Matrix von elektrischen Verbindungsleitungen. Jede MIU beinhaltet drei Ports, p0, p1 und p2. Jede MIU beinhaltet auch eine Puffersektion **186**, **188**, **190** und **192**. Jede der vier MIUs ist mit einem anderen von vier digitalen Speichern **170**, **172**, **174** und **176** verbunden. Jede der vier MIUs ist auch mit einer anderen von vier Busschnittstellenlogikeinheiten **178**, **180**, **182** und **184** verbunden.

[0115] **Fig.** 25 ist eine Seitenansicht von vier gestapelten Boxen **194**, **196**, **198** und **200**, die gemeinsam

das Vier-MIU-System **160** von **Fig.** 23 beherbergen. Box **194** beherbergt MIU **162**, Speicher **170** und Busschnittstellenlogik **178**. Box **196** beherbergt MIU **164**, Speicher **172** und Busschnittstellenlogik **180**. Box **198** nimmt MIU **166**, Speicher **174** und Busschnittstellenlogik **182** auf. Box **200** beherbergt MIU **168**, Speicher **176** und Busschnittstellenlogik **184**. Jede Box hat drei Ports, p0, p1 und p2. Jede Box hat auch einen Adressbus, über den sie Adressinformationen mit einer Nachbarbox gemeinsam nutzen kann. Die Ports und die Adressbusse sind wie gezeigt geschaltet.

[0116] So sind beispielsweise die Port-p1-Verbindungen von MIU **162** in Box **194** und MIU **166** in Box **198** so geschaltet, dass sie Bündelteilmengen zwischen den beiden MIUs übertragen. Insbesondere ist beispielsweise die Port-p1-Master-Verbindung von MIU **162** mit der Port-p1-Slave-Verbindung von MIU **166** gekoppelt. Umgekehrt ist die Port-p1-Slave-Verbindung von MIU **162** mit der Port-p1-Master-Verbindung von MIU **166** gekoppelt. Beim Betrieb wird das Bündelsegment A02, A12, A22, A32 zwischen Busschnittstellenlogik **178** und digitalem Speicher **174** über die Port-p1-Master-Verbindung von MIU **162** und Port-p1-Slave-Verbindung von MIU **166** kommuniziert. Ebenfalls wird beim Betrieb das Bündelsegment C00, C10, C20, C30 zwischen Busschnittstellenlogik **182** und digitalem Speicher **170** über die Port-p1-Slave-Verbindung von MIU **162** und Port-p1-Master-Verbindung von MIU **166** kommuniziert.

[0117] **Fig.** 26 ist eine Matrix, die die Verbindungen zwischen den vier MIUs **162**, **164**, **166** und **168** des Vier-MIU-Systems **160** von **Fig.** 24 repräsentiert. Eine "0" an einem Linienschnittpunkt bedeutet eine Verbindung im MIU-Schalter **162**. Eine "1" an einem Linienschnittpunkt bedeutet eine Verbindung im MIU-Schalter **164**. Eine "2" an einem Linienschnittpunkt bedeutet eine Verbindung im MIU-Schalter **166**. Eine "3" an einem Linienschnittpunkt bedeutet eine Verbindung im MIU **168**. So lauten beispielsweise die Verbindungen für den MIU-Schalter **162** wie folgt: b0 direkt mit c0 verbunden; c1 mit s0 verbunden; c2 mit s1 verbunden; c3 mit s2 verbunden; b1 mit m0 verbunden; b2 mit m1 verbunden; und b3 mit m2 verbunden. Die Verbindungen für den MIU-Schalter **166** lauten beispielsweise wie folgt: b0 mit m2 verbunden; b1 mit ml verbunden; b2 mit m0 verbunden; b3 direkt mit c3 verbunden; c0 mit s2 verbunden; c1 mit s1 verbunden; und c2 mit s0 verbunden. Die Verbindungen der MIUs **164** und **168** gehen aus der Matrix von **Fig.** 22 und der obigen Erörterung hervor.

[0118] Der Betrieb des Vier-MIU-Systems **160** von **Fig.** 24 beim Übertragen von Bündelteilmengen wird verständlich, wenn man die Bündel betrachtet, die durch die Busschnittstellenlogik **178**, **180**, **182** und **184** übertragen werden; wenn man die Verteilung von Bündelteilmengen zwischen den Puffern **186**, **188**, **190** und **192** betrachtet; und wenn man die Speicherung von Bündelteilmengen in digitalen Speichern **170**, **172**, **174** und **176** betrachtet. Zum Beispiel wird, unter Bezugnahme auf den digitalen Speicher **172**, die Teilmenge A01, A11, A21, A31 durch die Busschnittstellenlogik **178** als Teil eines Vier-Wort-Bündels übertragen. Die Teilmenge B01, B11, B21, B31 wird durch die Busschnittstellenlogik **180** als Teil eines Vier-Wort-Bündels übertragen. Die Teilmenge C01, C11, C21, C31 wird durch die Busschnittstellenlogik **182** als Teil eines Vier-Wort-Bündels übertragen. Die Teilmenge D01, D11, D21, D31 wird durch die Busschnittstellenlogik **184** als Teil eines Vier-Wort-Bündels übertragen. Im Laufe einer Übertragung von einer beliebigen dieser vier Bündelteilmengen zwischen dem digitalen Speicher **172** und einer jeweiligen Busschnittstellenlogikeinheit wird die Teilmenge vorübergehend im Puffer **188** gespeichert.

[0119] **Fig.** 27 illustriert den Fluss von Bündelteilmengen durch das Vier-MIU-System **160** von **Fig.** 24 während eines repräsentativen Speicherschreibvorgangs. Ein Fünf-Wort-Bündel wird von der Busschnittstellenlogik in Verbindung mit der MIU **162** empfangen. Das erste Wort des Bündels enthält Adressinformationen. Die nächsten vier Wörter sind für eine Verteilung zwischen digitalen Speichern **170**, **172**, **174** und **176** vorgesehen. Die Adressinformationen werden von der MIU **162** zu einem MIU-Adressregister **204** übertragen. Das MIU-Adressregister **204** sendet die Adresse über den Bus **214** zu den Speicheradressregistern **206**, **208**, **210** und **212** der Digitalspeicher **172**, **174**, **176** und **178**. Die Bündelteilmenge d00, d10, d20 und d30 wird von der MIU **162** direkt zum Digitalspeicher **170** zur Speicherung an einem Adressort übertragen, der vom Speicheradressregister **206** designiert wird. Die Bündelteilmenge d31, d21, d11 und d01 wird von der MIU **162** über ihre jeweiligen Port-p0-Verbindungen zur MIU **164** und dann von der MIU **164** zum Digitalspeicher **172** zur Speicherung an einem Speicherort übertragen, der vom Speicheradressregister **208** designiert wird. Die Bündelteilmenge d02, d2, d22 und d32 wird von der MIU **162** über ihre jeweiligen Port-p1-Verbindungen zur MIU **166** und dann von der MIU **166** zum Digitalspeicher **174** zur Speicherung an einem Speicherort übertragen, der vom Speicheradressregister **210** designiert wird. Die Bündelteilmenge d33, d23, d13 und d03 wird von der MIU **162** über ihre jeweiligen Port-p2-Verbindungen zur MIU **168** und von der MIU **168** zum Digitalspeicher **176** zur Speicherung an einem Speicherort übertragen, der vom Speicheradressregister **212** designiert wird. Jeder der zwischen den MIUs übertragenen Bündelteilmengen steht ein Rq-Signal voran, das angibt, ob ein Speicherlese- oder -schreibvorgang erfolgen soll.

[0120] **Fig.** 28 ist ein Zeitdiagramm, das den zeitlichen Ablauf von Daten- und Steuersignalen in Verbindung mit dem Speicherschreibbetrieb von **Fig.** 27 für das Vier-MIU-System **160** von **Fig.** 24 illustriert. Während der Taktzyklen 1–2 wird eine Adresse vom Segmentbus empfangen. Während der Taktzyklen 2–3 wird das Datenwort D0 vom Segmentbus emp-

fangen, und die Rq-Signale werden über die Verbindungen p0, p1 und p2 zu den MIUs **164**, **166** und **168** gesendet. Während der Taktzyklen 3–4 wird das Datenwort D1 vom Segmentbus empfangen, und Bündelteilmengenfragmente d01, d02 und d03 werden über die Multibit-Master-Verbindungen von Port p0, p1 bzw. p2 jeweils zu den MIUs **164**, **166** und **168** übertragen. Während der Taktzyklen 4–5 wird das Datenwort D2 vom Segmentbus empfangen und Bündelteilmengenfragmente d11, d12 und d13 werden über die Multibit-Master-Verbindungen von Port p0, p1 bzw. p2 jeweils zu den MIUs **164**, **166** und **168** übertragen. Während der Taktzyklen 5–6 wird das Datenwort D3 vom Segmentbus empfangen, und Bündelteilmengenfragmente d21, d22 und d23 werden über die Multibit-Master-Verbindungen von Port p0, p1 bzw. p2 jeweils zu den MIUs **164**, **166** und **168** übertragen. Während der Taktzyklen 6–7 werden die Bündelteilmengenfragmente d31, d32 und d33 über die Multibit-Master-Verbindungen von Port p0, p1 bzw. p2 jeweils zu den MIUs **164**, **166** und **168** übertragen. Ebenso wird während der Taktzyklen 6–7 eine Speicheradresse in den Registern **206**, **208**, **210** und **212** zu jedem der digitalen Speicher **170**, **172**, **174** und **176** gesendet. Während der Taktzyklen 7–8 wird ein Speicherschreibfreigabesignal gleichzeitig von den Steuereinheiten in Verbindung mit jeder der MIUs aufgedrückt, und in den Transferpuffern **186**, **188**, **190** und **192** gespeicherte ganze Datenbündelteilmengen werden parallel auf die jeweiligen Digitalspeicher **170**, **172**, **174** und **176** geschrieben.

[0121] **Fig.** 29 illustriert den Fluss von Bündelteilmengen durch das Vier-MIU-System **160** von **Fig.** 24 während eines repräsentativen Speicherlesevorgangs. Zunächst werden die vier Wörter eines Bündels über vier unterschiedliche Digitalspeicher **170**, **172**, **174** und **176** verteilt. Eine Speicheradresse **216** wird zum MIU-Adresspuffer **204** gesendet, der die Adresse über die Verbindung **214** zu den Speicheradressregistern **206**, **208**, **210** und **212** leitet. Der Digitalspeicher **170** liefert die Bündelteilmenge d00, d10, d20 und d30 zur MIU **162** und überträgt sie direkt zu ihrer Busschnittstellenlogik (nicht dargestellt). Der Digitalspeicher **172** liefert die Bündelteilmenge d01, d11, d21 und d31 zur MIU **164**, die sie über Port-p0-Verbindungen zur MIU **162** überträgt. Die MIU 162 überträgt dann die Teilmenge zu ihrer Busschnittstellenlogik. Der Digitalspeicher **174** überträgt die Bündelteilmenge d02, d2, d22 und d32 zur MIU **166**, die sie über Port-p1-Verbindungen zur MIU **162** überträgt. Die MIU **162** überträgt dann die Teilmenge zu ihrer Busschnittstellenlogik. Der Digitalspeicher **176** überträgt die Bündelteilmenge d03, d13, d23 und d33 zur MIU **168**, die sie dann über Port-p2-Verbindungen zur MIU **162** überträgt. Die MIU **162** sendet dann die Teilmenge zu ihrer Busschnittstellenlogik. Somit wird das Bündel **202** von unterschiedlichen Speichern zurückgespeichert und an der Busschnittstellenlogik von MIU **162** zur Verfügung gestellt.

[0122] **Fig.** 30 ist ein Zeitdiagramm, das den zeitlichen Ablauf von Daten- und Steuersignalen in Verbindung mit dem Speicherlesebetrieb von **Fig.** 29 für das Vier-MIU-System **160** von **Fig.** 24 illustriert. Während der Taktzyklen 1–2 wird ein Adresssignal vom Segmentbus empfangen. In den Taktzyklen 2–3 werden von der Busschnittstellenlogik erzeugte Rq-Signale über die Verbindungen p0, p1 und p2 von MIU **162** gesendet. In den Taktzyklen 4–5 wird eine Speicheradresse in jedem der Speicheradressregister **206**, **208**, **210** und **212** zu den Speichern **170**, **172**, **174** und **176** gesendet. Während der Taktzyklen 5– 6 wird ein Speicherausgangsfreigabesignal zu jedem der Speicher gesendet, und ganze Datenbündelteilmengen werden parallel von jedem der vier Speicher **170**, **172**, **174** und **176** gelesen und zur vorübergehenden Speicherung zu Transferpuffern **186**, **188**, **190** und **192** gesendet. Während der Taktzyklen 6–7 werden Byte-breite Fragmente d01, d02 und d03 über die Slave-Verbindungen von Port p0, p1 bzw. p2 zur MIU **162** übertragen. Während der Taktzyklen 7–8 werden Fragmente d11, d12 und d13 über die Slave-Verbindungen von Port p0, p1 bzw. p2 zur MIU **162** übertragen, und das Datenwort D0 wird zum Segmentbus gesendet. Während der Taktzyklen 8–9 werden Fragmente d21, d22 und d23 über die Slave-Verbindungen von Port p0, p1 bzw p2 zur MIU **162** übertragen, und das Datenwort D1 wird zum Segmentbus gesendet. Während der Taktzyklen 9–10 werden Fragmente d31, d32 und d33 über die Slave-Verbindungen von Port p0, p1 bzw. p2 zur MIU **162** übertragen, und das Datenwort D2 wird zum Segmentbus gesendet. Während der Taktzyklen 10–11 wird das Datenwort D3 zum Segmentbus gesendet.

[0123] Die illustrative Zeichnung der **Fig.** 31–36 illustriert einige mögliche Anwendungen der vorliegenden Erfindung. In diesen Figuren wird das Schaltsystem **98** als Queue Management Unit ("QMU" = Warteschlangenmanagementeinheit) beim Implementieren von Gemeinschaftsspeicher bezeichnet. **Fig.** 31 zeigt die Verwendung eines Grundschalters in einer selbstständigen Konfiguration. **Fig.** 32 illustriert eine gestapelte Vier-MIU- (Vier-QMU) Konfiguration. **Fig.** 33 illustriert eine Ethernet-Schalterimplementation. **Fig.** 34 illustriert einen Ethernet-Schalter mit ATM-Uplink. **Fig.** 35 illustriert einen stapelbaren Zwei-MIU- (Zwei-QMU) Schalter/Router. **Fig.** 36 illustriert einen Vier-MIU- (Vier-QMU) Fast-Ethernet-Schalter mit Adressauflösung (ARL).

[0124] **Fig.** 37 zeigt eine verbesserte Vorrichtung, die zwei Gemeinschaftsspeicherschalter kombiniert, wie die von **Fig.** 4 oder **Fig.** 39, so dass ein blockierungsfreier größerer Schalter mit der zweifachen Bandbreite entsteht (z. B. Kombinieren von zwei 32 × 32 155-Mbps Port ATM Schaltern zum Konstruieren eines blockierungsfreien 64 × 64 155-Mbps Port ATM Schalters, zum Beispiel). Eine vorliegende Ausgestaltung hat einen 32 × 32 ATM-Schalter. Gemäß der Verbesserung arbeitet der 32 × 32 ATM-Schalter auf Zeitmultiplexweise (TDM). Mit Bezug auf den zeitli-

chen Ablauf des Diagramms von **Fig.** 38, der Schalter zykliert in 68 Taktperioden, unterteilt in 32 Takte Eingangsstufe und 32 Takte Ausgangsstufe (und eine Lücke von 4 Takten). Man nehme zwei 32 × 32 Schalter-Chipsets und zwei Gemeinschaftsspeicherbänke. Man versetze jeden 32 × 32 Chipset in die Lage, auf eine der beiden Bänke zuzugreifen. Man betreibe sie in einer Phasendifferenz, so dass, wenn ein Chipset in der Eingangsstufe ist, der andere in der Ausgangsstufe ist. Wenn jetzt der eine Chipset, der in der Ausgangsstufe ist, die Zelle abruft, die von der Bank übertragen werden soll, in der sich die Zelle befindet, dann wählt der andere Chipset, der in der Eingangsstufe ist, die andere Bank, um die eingehende Zelle darauf zu schreiben.

[0125] Die illustrative Zeichnung von **Fig.** 39 soll verdeutlichen, dass Matrixverbindungsleitungen zwischen einem Port und einem Puffer eine Breite von mehreren Bits für jedes Bündelsegment haben können. Somit können mehrere Bits pro Bündelsegment pro Taktzyklus übertragen werden.

[0126] Es wurden zwar hierin ausführlich besondere Ausgestaltungen der Erfindung beschrieben, aber es ist zu verstehen, dass verschiedene Modifikationen an der bevorzugten Ausgestaltung möglich sind, ohne vom Umfang der Erfindung gemäß Definition in den beiliegenden Ansprüchen abzuweichen. Somit soll die vorangehende Beschreibung die Erfindung nicht begrenzen, die in den beiliegenden Ansprüchen definiert ist.

Patentansprüche

1. Speicherschnittstelleneinheit für den Einsatz in einem Multiport-Speichersystem mit einem Direktzugriffsspeicher (RAM, 26) und mehreren Datenports (**20**), wobei die Einheit Folgendes umfasst: eine Mehrzahl von Puffern (**24**), die jeweils eine Mehrzahl von Speicherelementen für die gleichzeitige Speicherung einer Mehrzahl von jeweiligen Datenbündelteilmengen beinhalten, die zwischen verschiedenen jeweiligen Datenports und dem genannten Direktzugriffsspeicher übertragen werden; und
eine Verbindungsschaltung (**22**), die eine Mehrzahl von Datenleitungen beinhaltet, die jeden jeweiligen Datenport mit einer Mehrzahl von jeweiligen Puffern verbinden, so dass verschiedene jeweilige Teilmengen eines beliebigen Datenbündels, das zwischen einem jeweiligen Datenport und dem genannten Direktzugriffsspeicher übertragen wird, zu oder von verschiedenen jeweiligen Puffern über verschiedene jeweilige Datenleitungen geleitet werden, die einen Port und einen Puffer miteinander verbinden;
wobei wenigstens eine einen Port und einen Puffer verbindende Datenleitung eine Breite von mehreren Bits hat.

2. Speicherschnittstelleneinheit nach Anspruch 1, bei der jeweilige Speicherelemente einen Transferschaltkomplex beinhalten, der Datenbündelteilmengen zu und von der genannten Verbindungsschaltung überträgt, so dass jede Teilmenge in Gruppen von zwei oder mehr, aber weniger als alle Bits pro Taktzyklus übertragen wird.

3. Speicherschnittstelleneinheit nach Anspruch 1, bei der jeweilige Speicherelemente einen jeweiligen Shift-in/Parallel-Shift-out-Registerschaltkomplex zum Einschieben jeweiliger Datenbündelteilmengen von der genannten Verbindungsschaltung beinhaltet, so dass jede Datenbündelteilmenge in Gruppen von zwei oder mehr, aber weniger als alle Bits gleichzeitig eingeschoben wird, und so dass solche Datenbündelteilmengen parallel zu dem genannten Direktzugriffsspeicher ausgeschoben werden; und
wobei jeweilige Speicherelemente ferner einen jeweiligen Parallel-Shift-in/Shift-out-Registerschaltkomplex zum parallelen Einschieben jeweiliger Datenbündelteilmengen von dem genannten Direktzugriffsspeicher und zum Ausschieben solcher Datenbündelteilmengen zu der genannten Verbindungsschaltung beinhalten, so dass jede Datenbündelteilmenge in Gruppen von zwei oder mehr, aber weniger als alle Bits gleichzeitig ausgeschoben werden.

4. Speicherschnittstelleneinheit nach einem der vorherigen Ansprüche, ferner mit einem Steuerschaltkomplex, der Steuersignale erzeugt, die eine gleichzeitige parallele Übertragung aller Teilmengen eines Datenbündels zwischen dem genannten Direktzugriffsspeicher und mehreren verschiedenen Puffern steuern, die verschiedene Teilmengen eines solchen Datenbündels speichern.

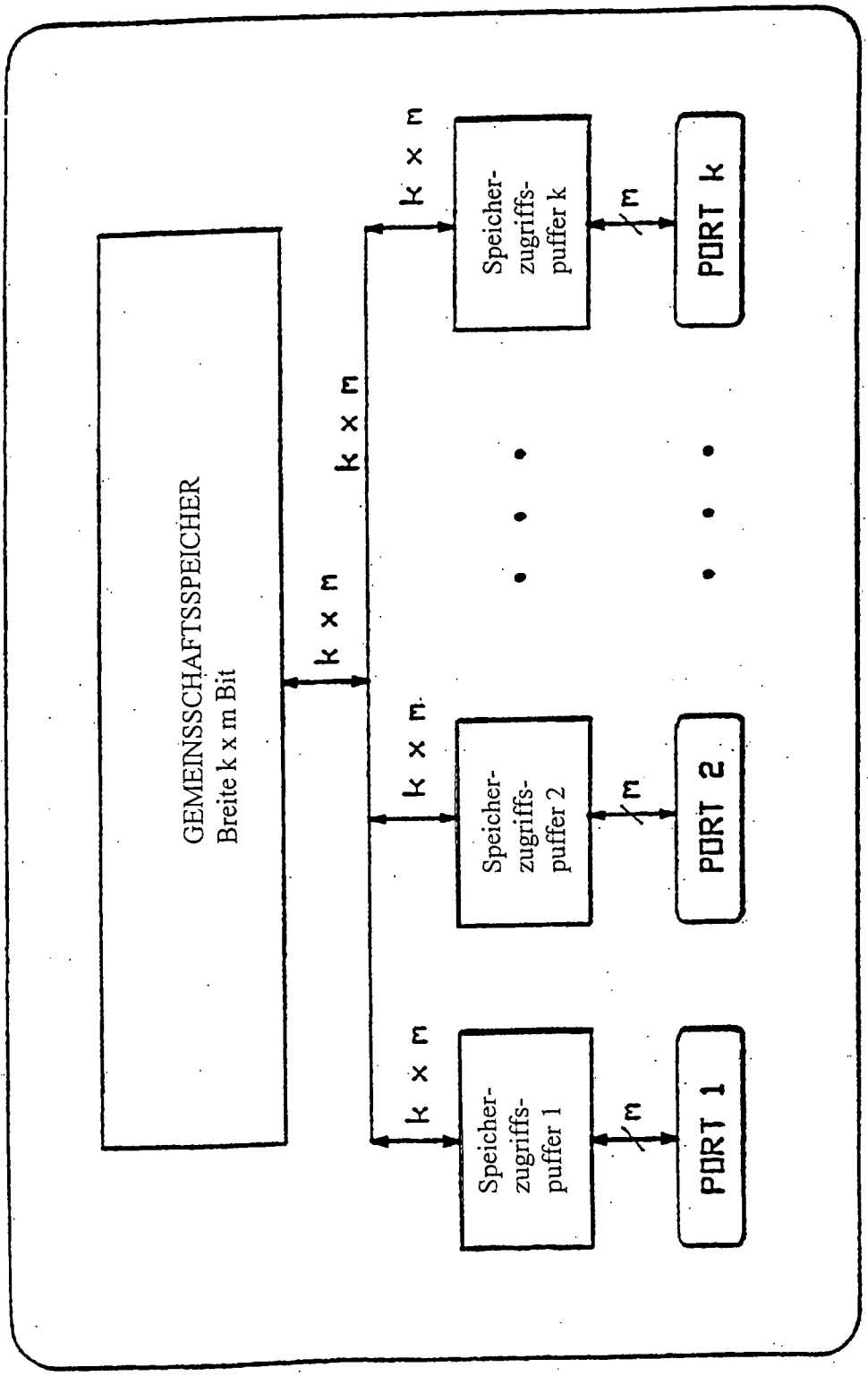
5. Speicherschnittstelleneinheit nach einem der vorherigen Ansprüche, umfassend k genannte Datenports und m genannte Puffer, wobei jeder Puffer k genannte Speicherelemente zum gleichzeitigen Speichern mehrerer jeweiliger Teilmengen von n-Wortdatenbündeln aufweist, die zwischen verschiedenen jeweiligen Datenports und dem genannten Direktzugriffsspeicher übertragen werden, wobei jeder Puffer ferner einen parallelen Datenpfadschaltkomplex hat, der zwischen einzelnen Speicherelementen und dem genannten Direktzugriffsspeicher geschaltet werden kann;
wobei jede jeweilige Teilmenge aus mehr als n jeweiligen Bits besteht, die dieselben jeweiligen Bitpositionen in verschiedenen jeweiligen Datenwörtern eines jeweiligen n-Wortdatenbündels belegen; und
wobei die Verbindungsschaltungsdatenleitungen jeden jeweiligen Datenport mit m verschiedenen Puffern verbinden, so dass die m verschiedenen jeweiligen Teilmengen eines beliebigen n-Wortdatenbündels, das zwischen einem jeweiligen Datenport und dem genannten Direktzugriffsspeicher übertragen wird, zu oder von m verschiedenen jeweiligen Puffern geleitet werden.

6. Verfahren zum Übertragen von n-Wortdatenbündeln, die m Bits pro Wort zwischen k Datenports (**20**) und einem Direktzugriffsspeicher (RAM, 26) in einem Mehrportspeichersystem beinhalten, umfassend die folgenden Schritte:
Unterteilen von Datenbündeln, die durch jeweilige Datenports übertragen werden, in jeweilige Datenbündelteilmengen, in denen jede jeweilige Teilmenge mehr als n jeweilige Bits beinhaltet, wobei solche mehr als n jeweiligen Bits dieselben jeweiligen Bitpositionen in verschiedenen Datenwörtern eines jeweiligen Datenbündels belegen;
Bereitstellen von m Puffern (**24**), jeweils mit k Speicherelementen;
vorübergehendes Speichern verschiedener Datenbündelteilmengen eines jeweiligen Datenbündels auf seinem Weg zu oder von einem jeweiligen Datenport und dem Direktzugriffsspeicher in verschiedenen Speicherelementen in verschiedenen Puffern; und
Übertragen aller verschiedener Datenbündelteilmengen eines jeweiligen Datenbündels auf dem Weg zu oder von einem jeweiligen Datenport und dem Direktzugriffsspeicher parallel zwischen verschiedenen Speicherelementen in verschiedenen Puffern und dem Direktzugriffsspeicher.

7. Verfahren nach Anspruch 6, ferner beinhaltend den Schritt des Übertragens mehrerer, aber weniger als aller Bits gleichzeitig, wobei jede jeweilige Datenbündelteilmenge eines jeweiligen Datenbündels, das zu oder von einem jeweiligen Datenport und dem RAM zwischen einem solchen jeweiligen Datenport und verschiedenen Speicherelementen in verschiedenen Puffern übertragen wird.

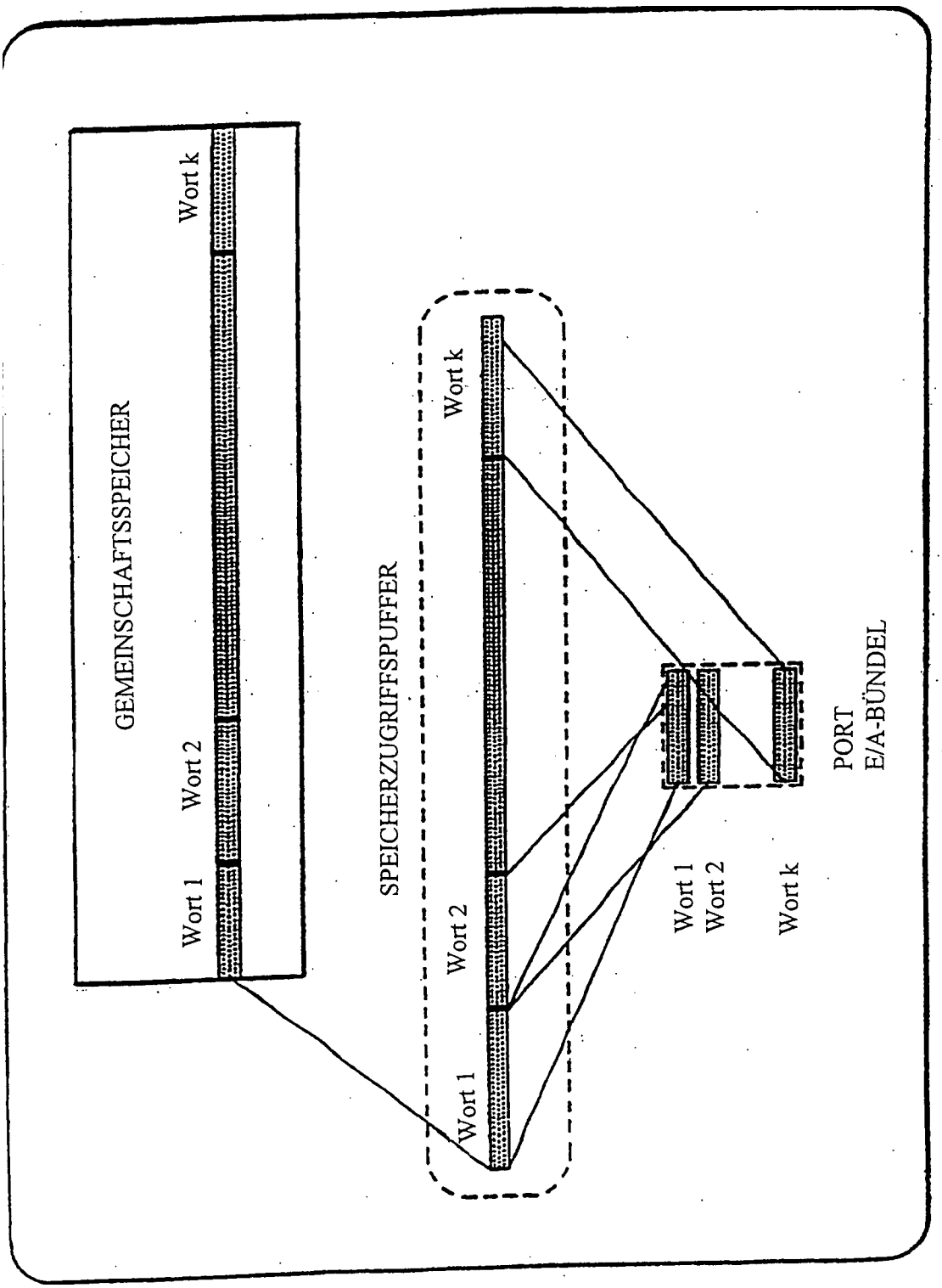
Es folgen 37 Blatt Zeichnungen

Anhängende Zeichnungen



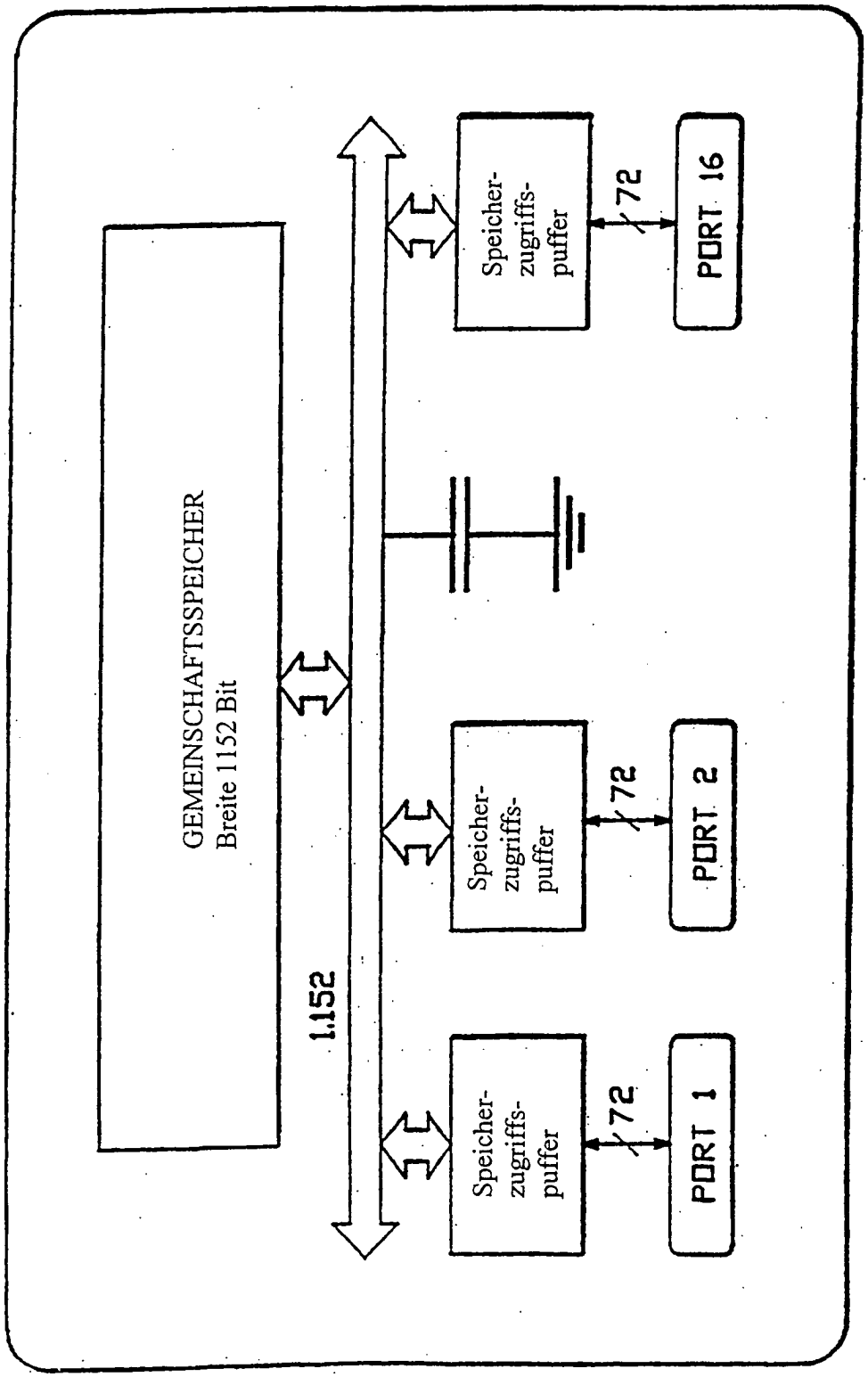
FIGUR 1

(STAND DER TECHNIK)



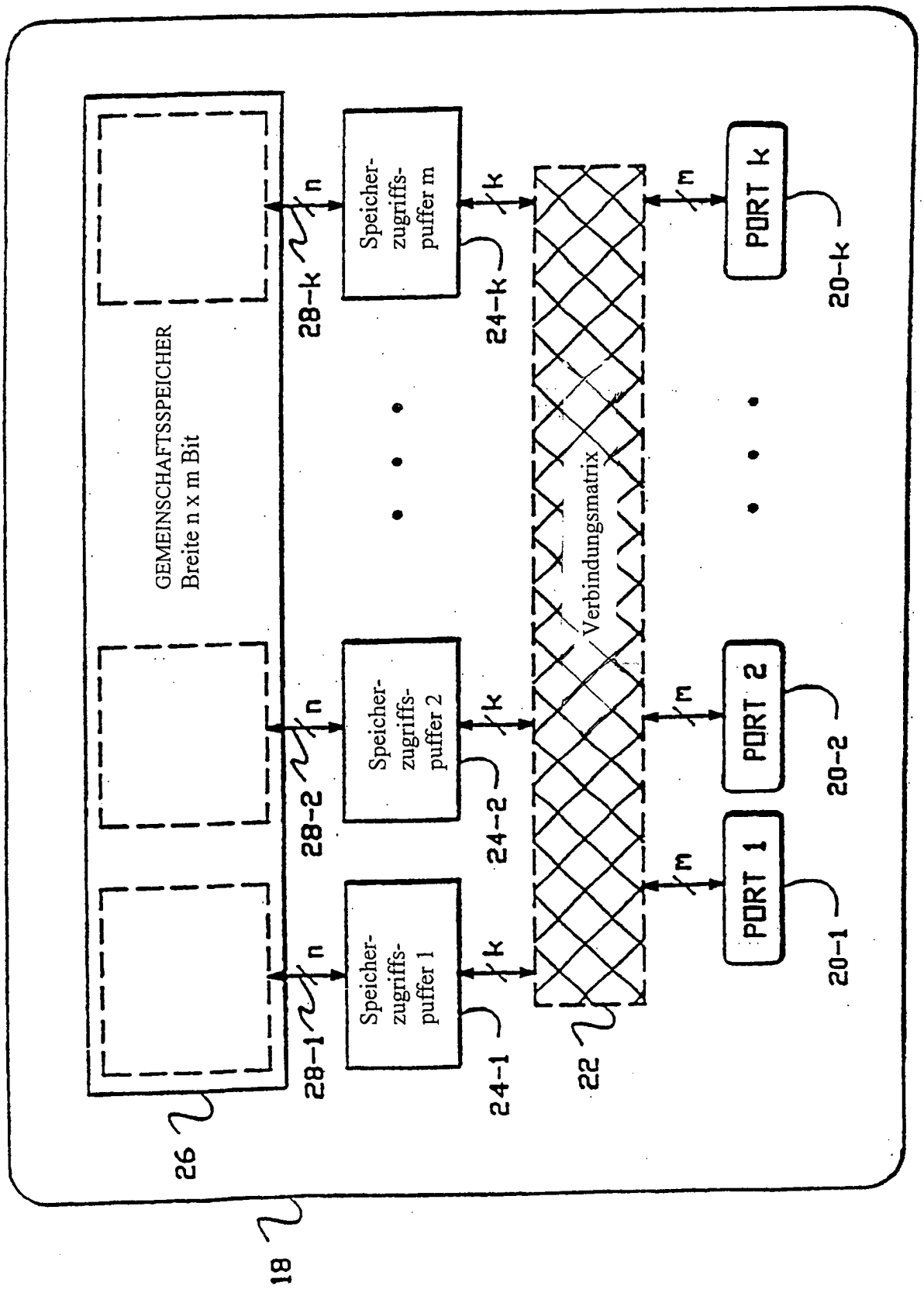
FIGUR 2

(STAND DER TECHNIK)

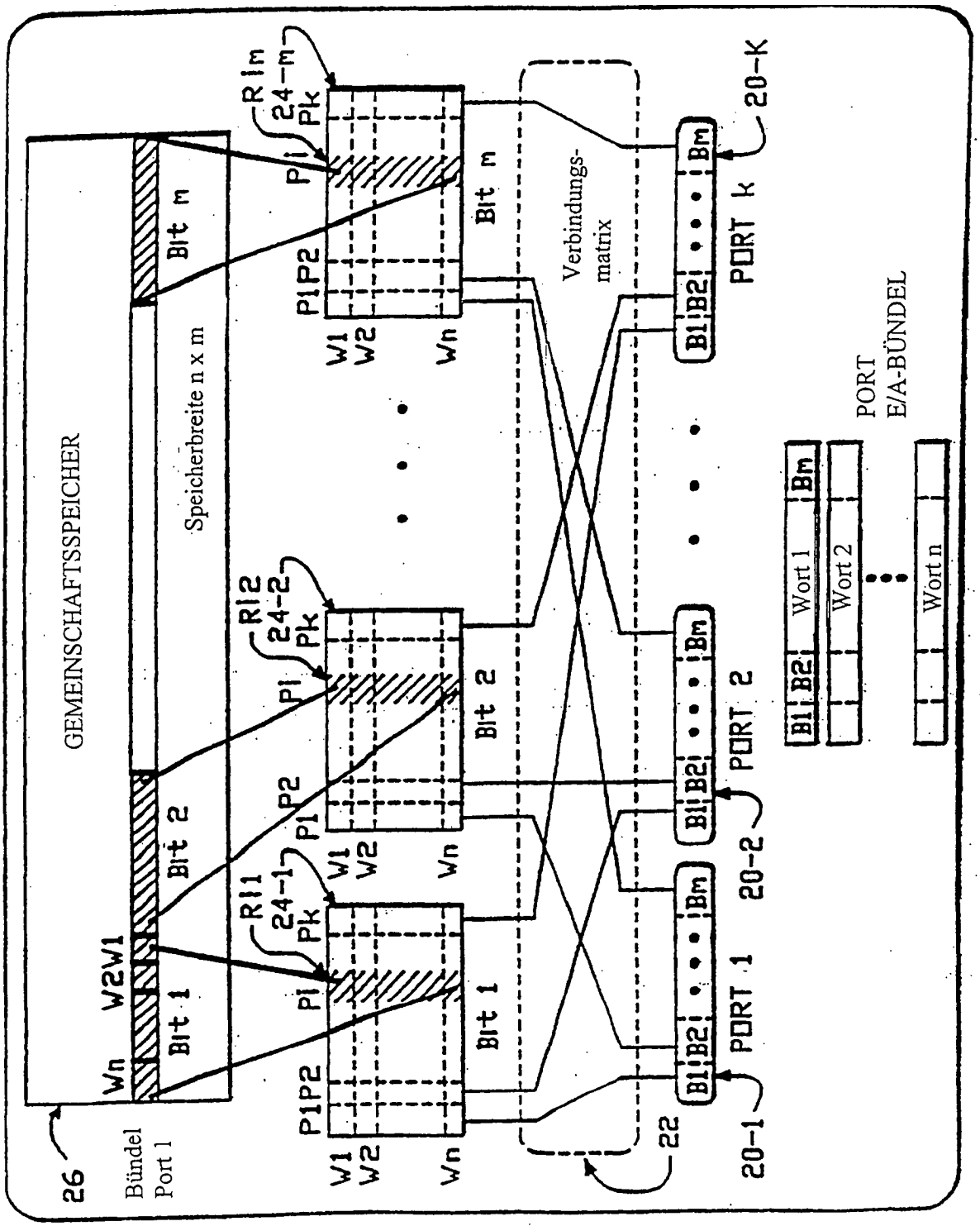


FIGUR 3

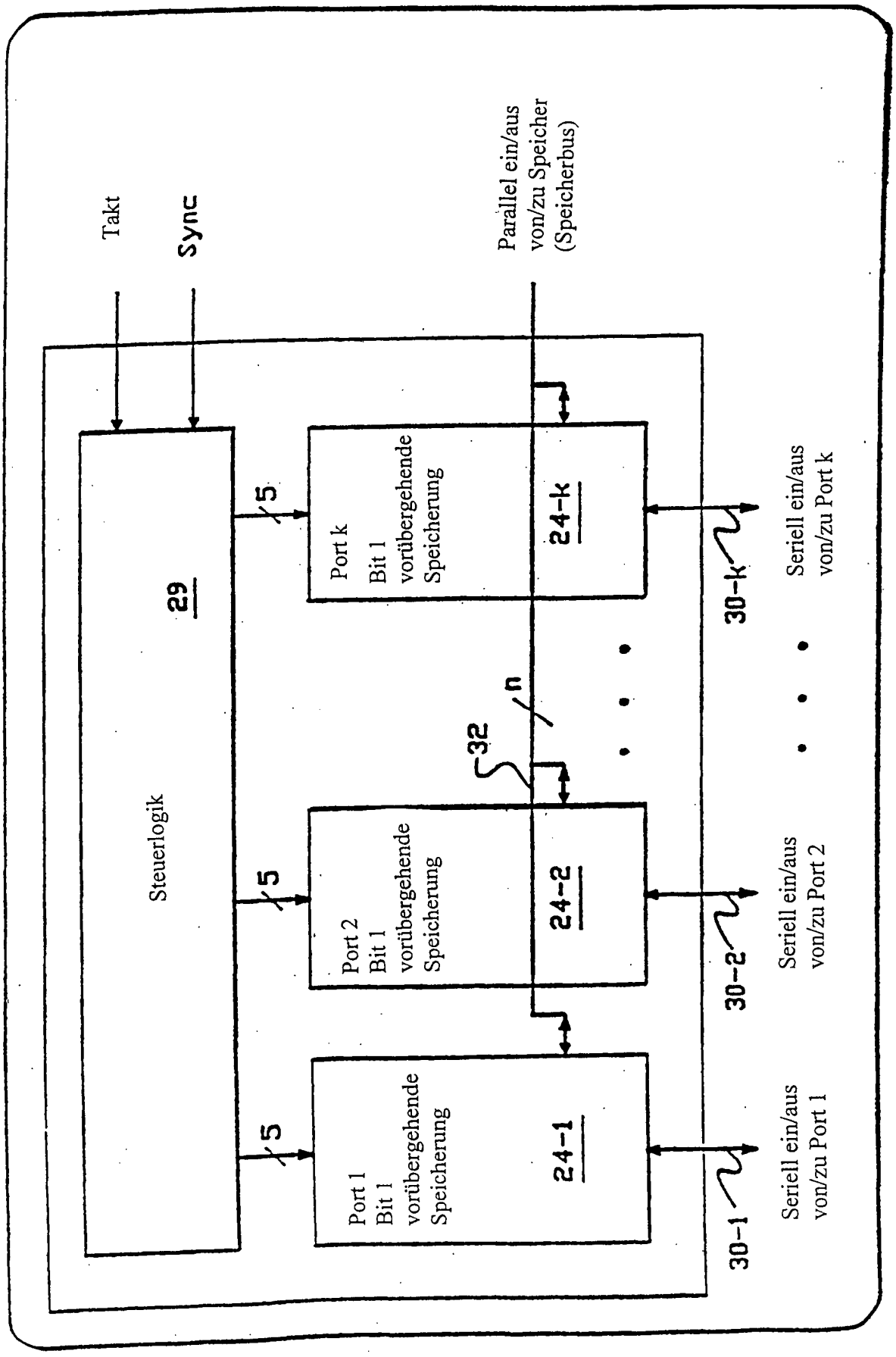
(STAND DER TECHNIK)


18
26
GEMEINSCHAFTSSPEICHER
Breite n x m Bit
28-1
28-2
28-k
n
Speicher-
zugriffs-
puffer 1
Speicher-
zugriffs-
puffer 2
Speicher-
zugriffs-
puffer m
k
24-1
24-2
24-k
22
Verbindungsmatrix
m
PORT 1
PORT 2
PORT k
20-1
20-2
20-k

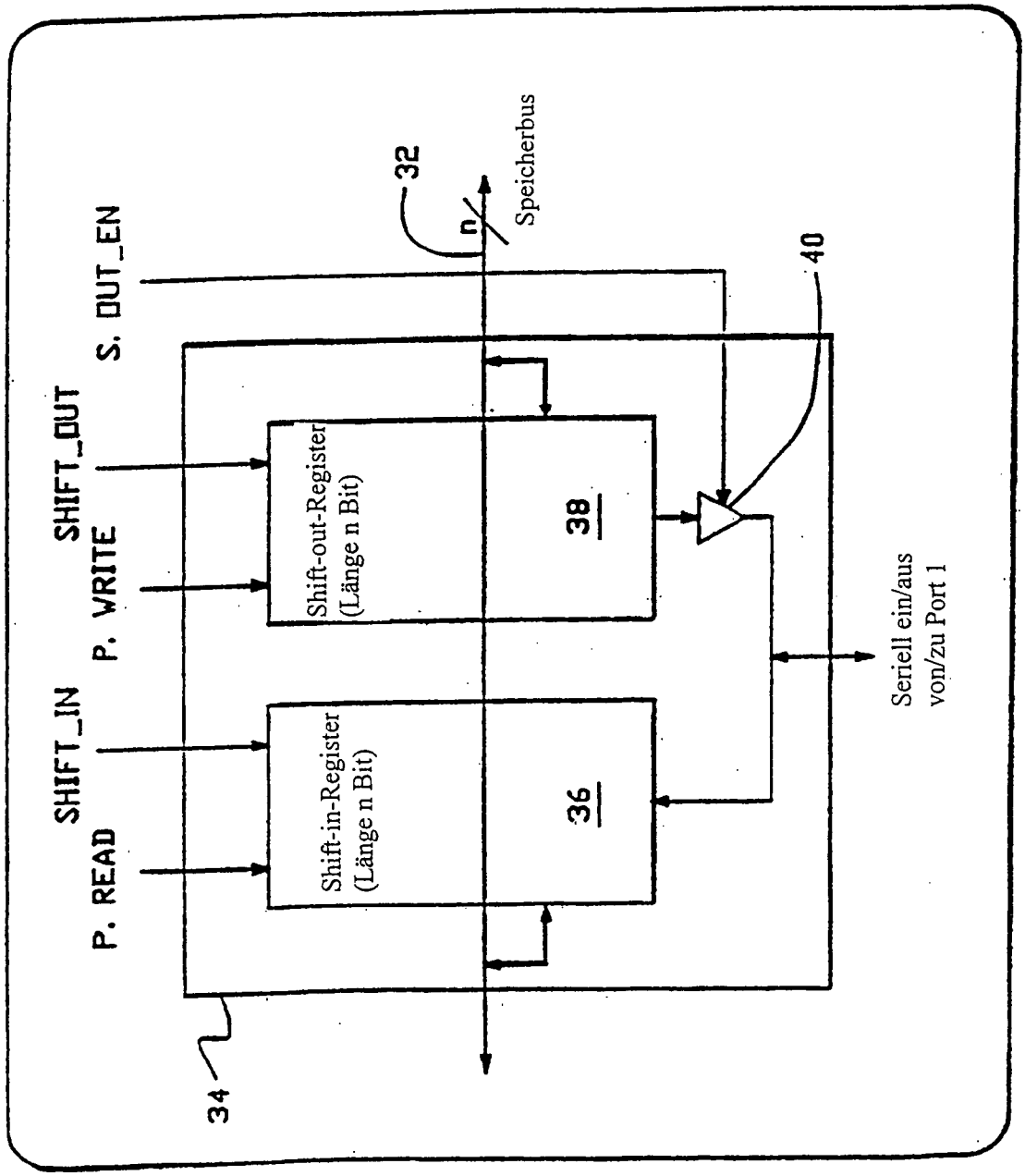
FIGUR 4


GEMEINSCHAFTSSPEICHER
Speicherbreite n x m
Bit m
Bit 2
Bit 1
W2W1
Wn
Bündel Port 1
26
R1m
24-m
R12
24-2
R11
24-1
Pk
Pi
P1P2
P1
P2
W1
W2
Wn
Verbindungs-matrix
22
PORT k
PORT 2
PORT 1
20-K
20-2
20-1
B1 B2 • • • Bm
PORT E/A-BÜNDEL
Wort 1
Wort 2
Wort n

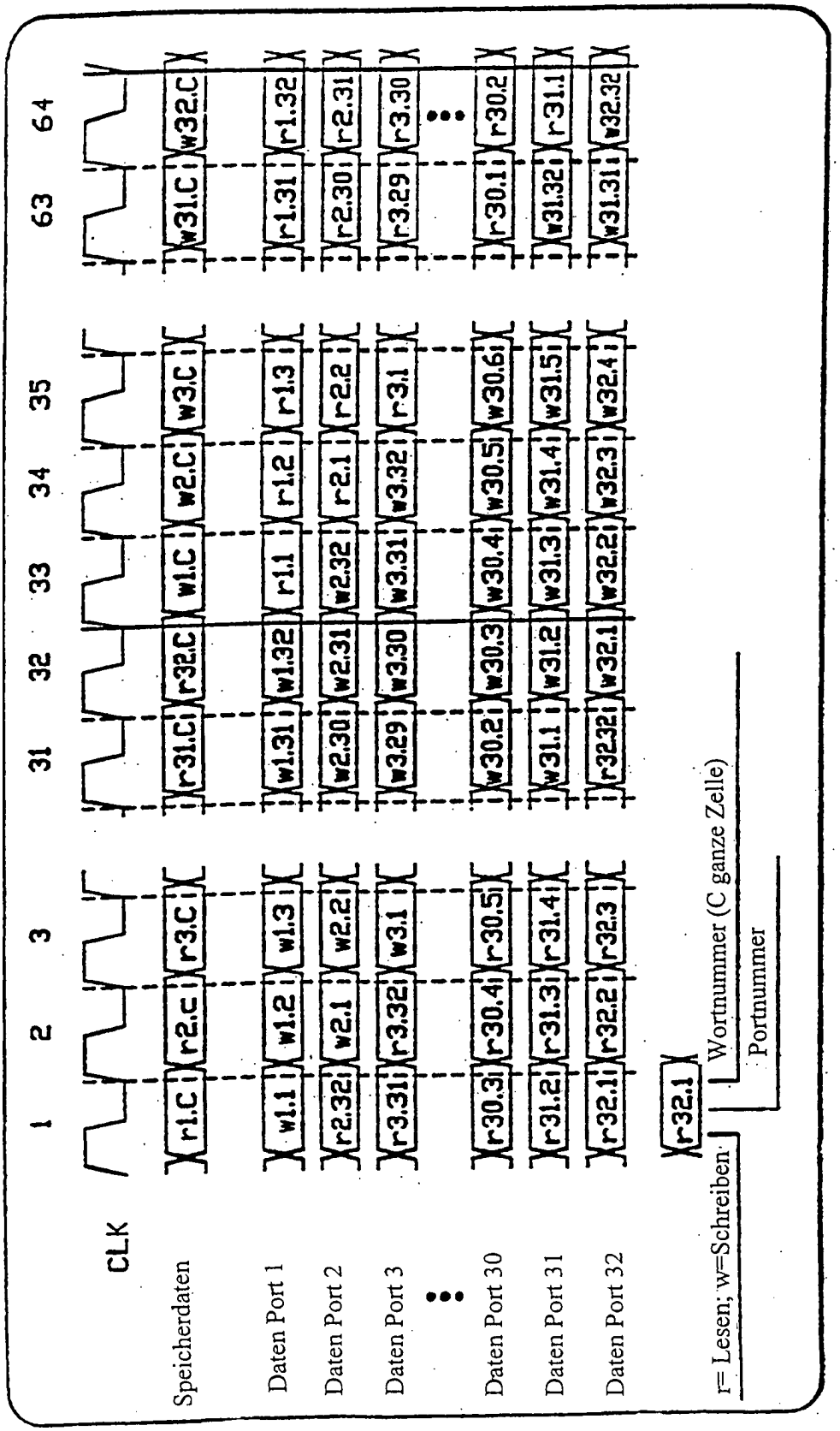
FIGUR 5


Takt
Sync
Parallel ein/aus von/zu Speicher (Speicherbus)
Steuerlogik
29
5
5
5
Port k Bit 1 vorübergehende Speicherung
Port 2 Bit 1 vorübergehende Speicherung
Port 1 Bit 1 vorübergehende Speicherung
24-k
24-2
24-1
32
n
30-k
30-2
30-1
Seriell ein/aus von/zu Port k
Seriell ein/aus von/zu Port 2
Seriell ein/aus von/zu Port 1

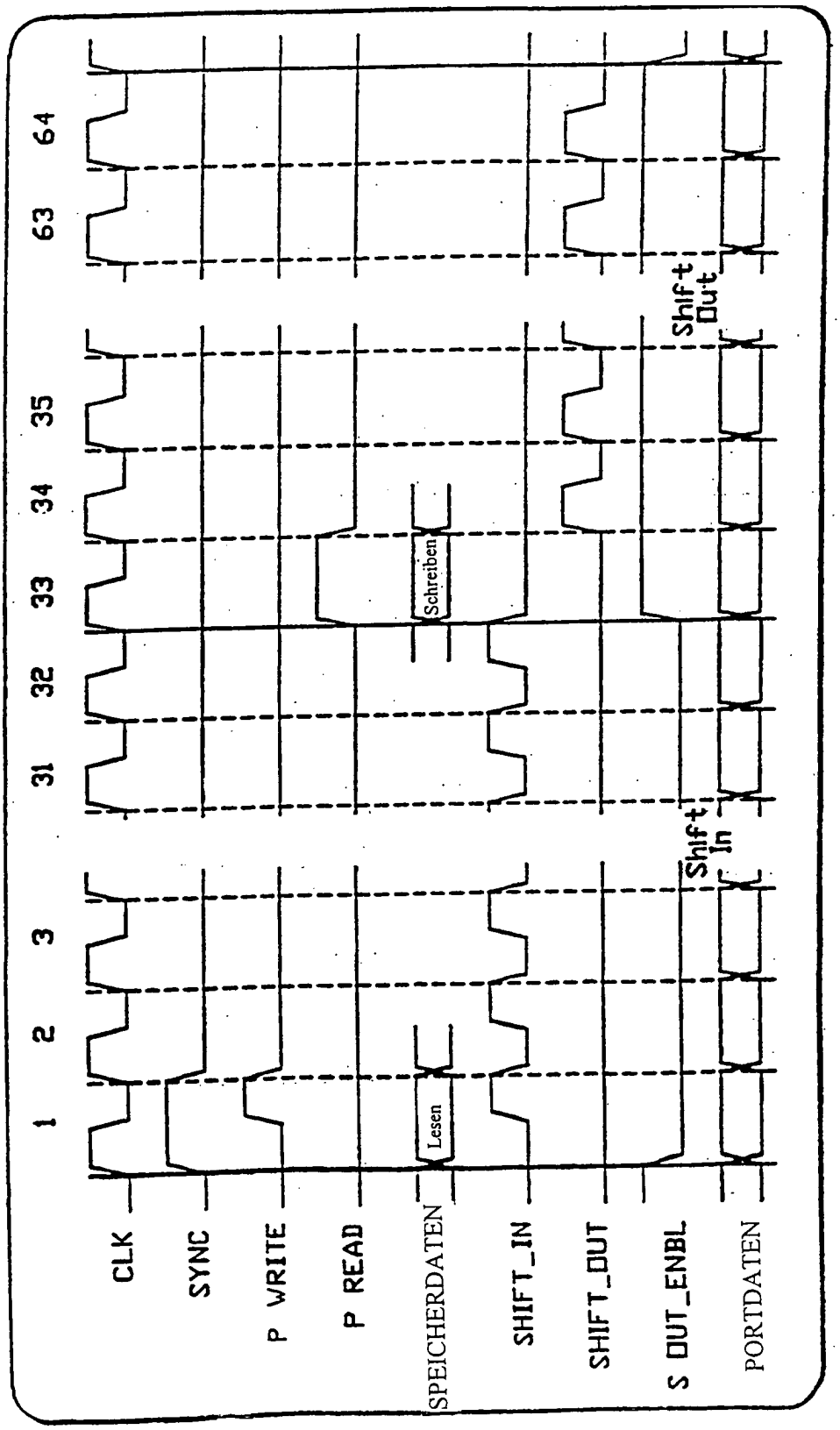
FIGUR 6


32
n
Speicherbus
S. OUT_EN
SHIFT_OUT
P. WRITE
SHIFT_IN
P. READ
Shift-out-Register
(Länge n Bit)
38
Shift-in-Register
(Länge n Bit)
36
40
Seriell ein/aus
von/zu Port 1
34

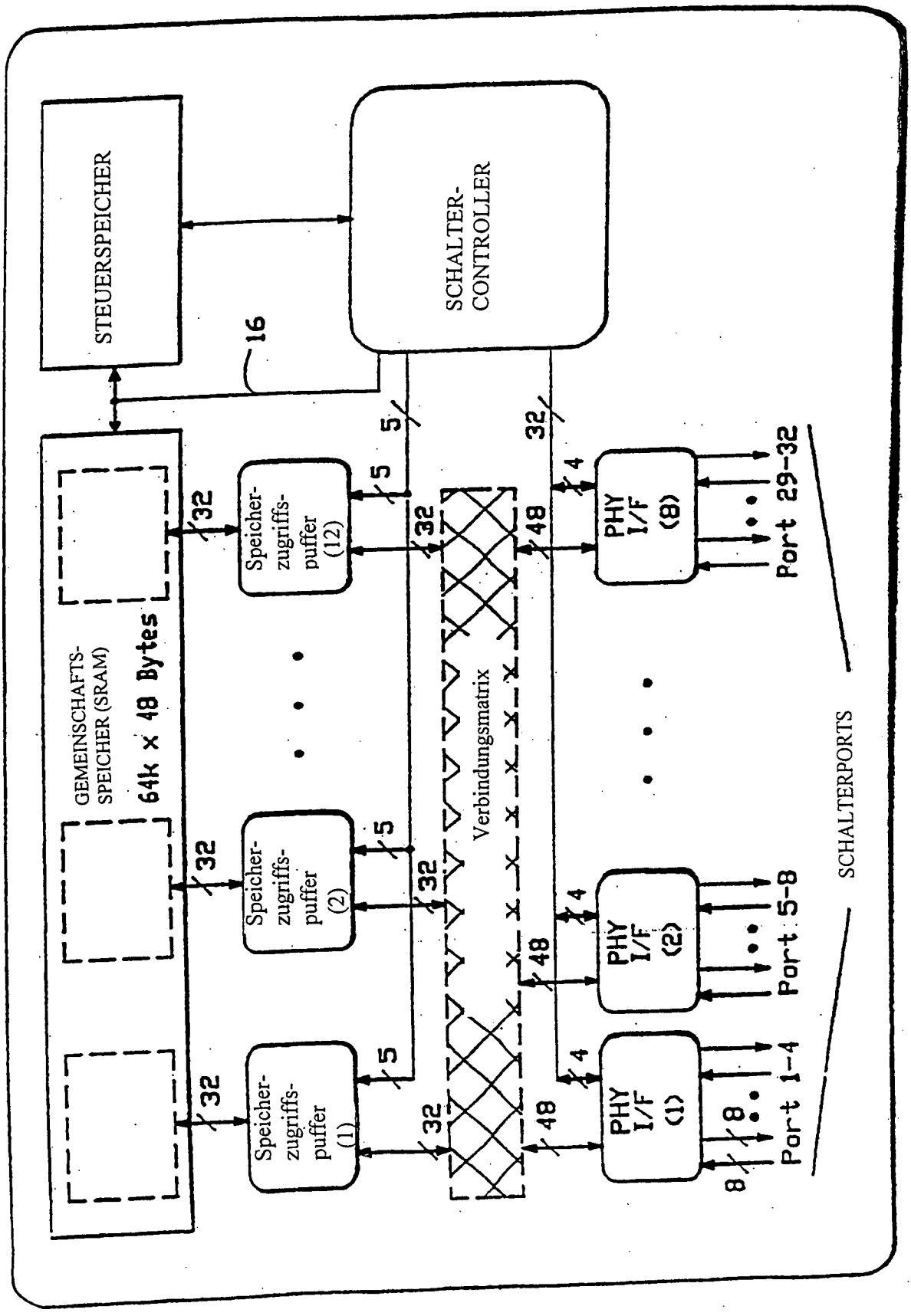
FIGUR 7


CLK
1
2
3
31
32
33
34
35
63
64
Speicherdaten
r1.C r2.C r3.C r31.C r32.C w1.C w2.C w3.C w31.C w32.C
Daten Port 1
w1.1 w1.2 w1.3 w1.31 w1.32 r1.1 r1.2 r1.3 r1.31 r1.32
Daten Port 2
r2.32 w2.1 w2.2 w2.30 w2.31 w2.32 r2.1 r2.2 r2.30 r2.31
Daten Port 3
r3.31 r3.32 w3.1 w3.29 w3.30 w3.31 w3.32 r3.1 r3.29 r3.30
Daten Port 30
r30.3 r30.4 r30.5 w30.2 w30.3 w30.4 w30.5 w30.6 r30.1 r30.2
Daten Port 31
r31.2 r31.3 r31.4 w31.1 w31.2 w31.3 w31.4 w31.5 w31.32 r31.1
Daten Port 32
r32.1 r32.2 r32.3 r32.32 w32.1 w32.2 w32.3 w32.4 w32.31 w32.32
r32.1
r= Lesen; w=Schreiben
Wortnummer (C ganze Zelle)
Portnummer

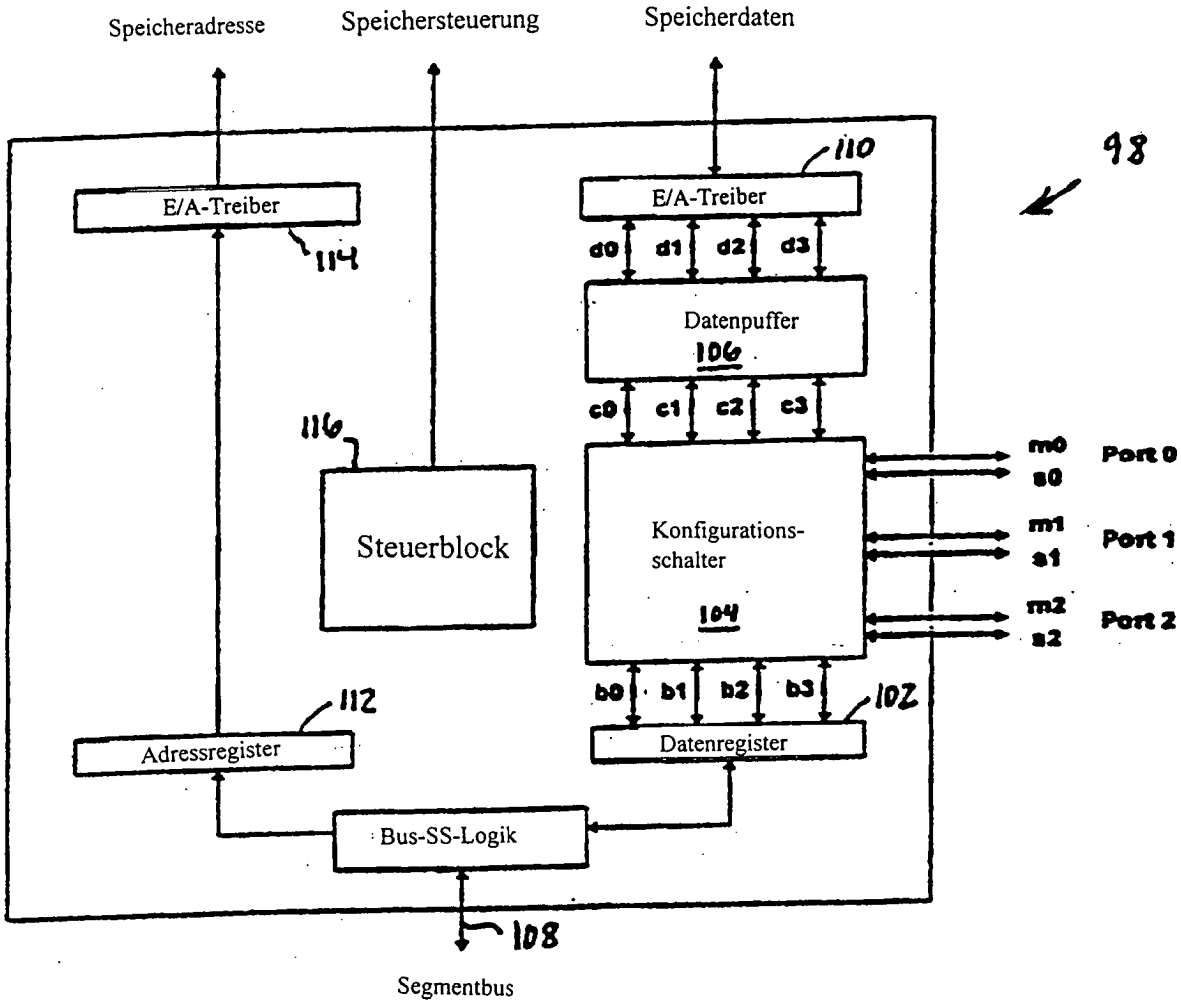
FIGUR 8


1
2
3
31
32
33
34
35
63
64
CLK
SYNC
P WRITE
P READ
SPEICHERDATEN
SHIFT_IN
SHIFT_OUT
S OUT_ENBL
PORTDATEN
Lesen
Schreiben
Shift In
Shift Out

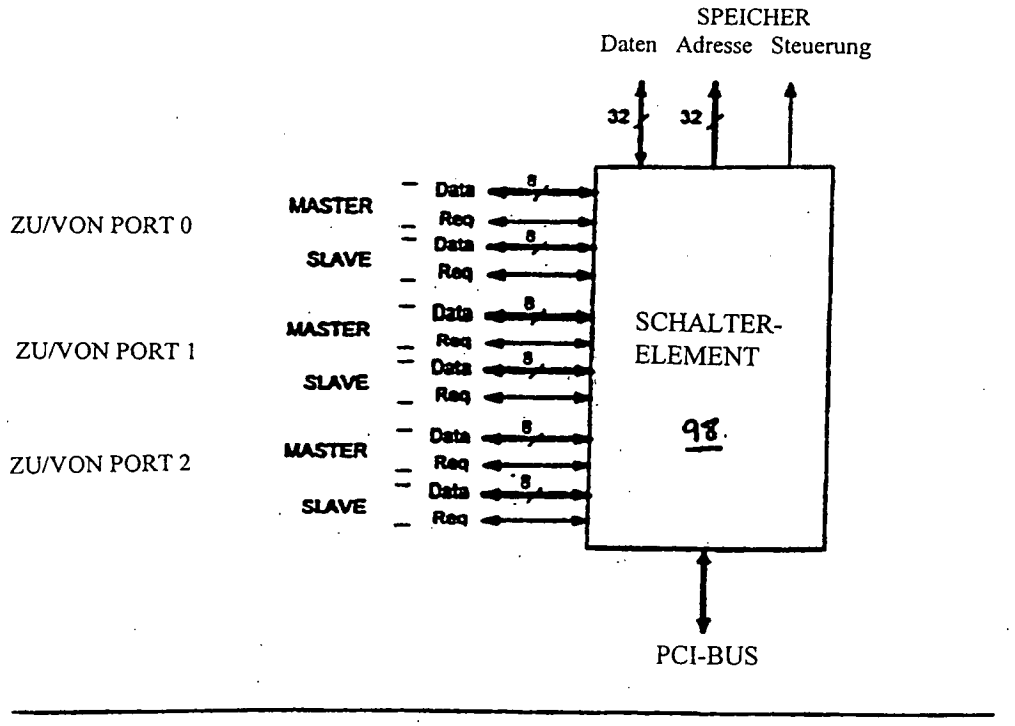
FIGUR 9


STEUERSPEICHER
SCHALTER-CONTROLLER
16
GEMEINSCHAFTS-SPEICHER (SRAM)
64k x 48 Bytes
Speicher-zugriffs-puffer (1)
Speicher-zugriffs-puffer (2)
Speicher-zugriffs-puffer (12)
Verbindungsmatrix
PHY I/F (1)
PHY I/F (2)
PHY I/F (8)
Port 1-4
Port 5-8
Port 29-32
SCHALTERPORTS
32
5
48
4
8

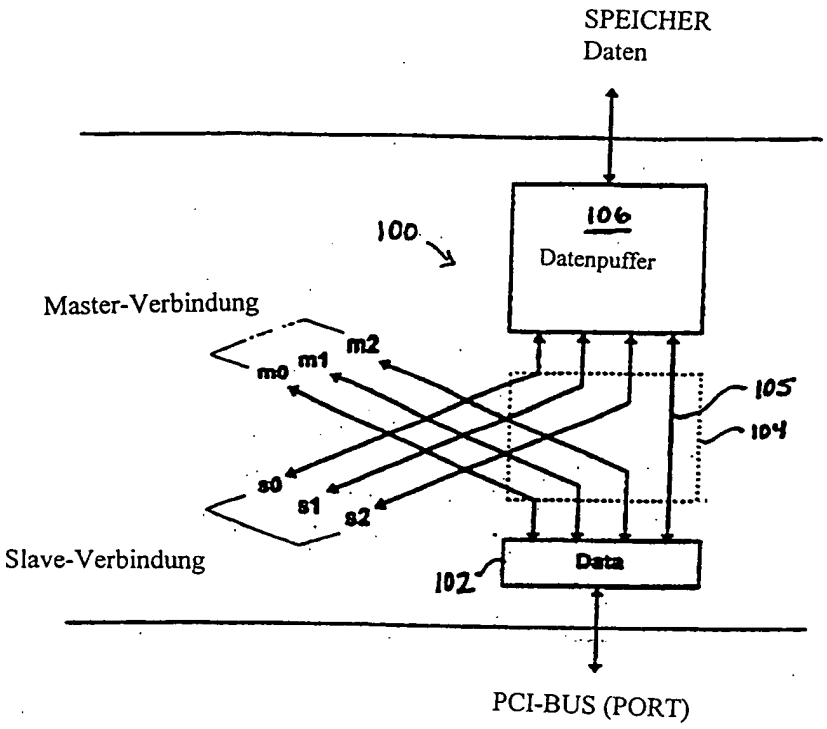
FIGUR 10


Speicheradresse
Speichersteuerung
Speicherdaten
98
110
E/A-Treiber
114
E/A-Treiber
d0
d1
d2
d3
Datenpuffer
106
c0
c1
c2
c3
116
Steuerblock
Konfigurations-
schalter
104
m0
s0
Port 0
m1
s1
Port 1
m2
s2
Port 2
b0
b1
b2
b3
102
112
Adressregister
Datenregister
Bus-SS-Logik
108
Segmentbus

Figur 11

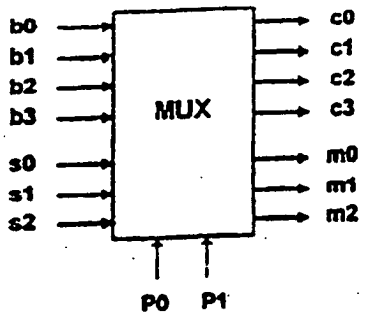

SPEICHER
Daten
Adresse
Steuerung
32
32
ZU/VON PORT 0
ZU/VON PORT 1
ZU/VON PORT 2
MASTER
SLAVE
MASTER
SLAVE
MASTER
SLAVE
Data
Req
Data
Req
Data
Req
Data
Req
Data
Req
Data
Req
8
8
8
8
8
8
SCHALTER-
ELEMENT
98
PCI-BUS

Figur 12

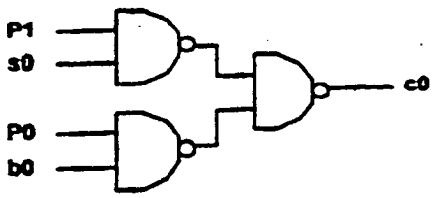

SPEICHER
Daten
106
Datenpuffer
100
Master-Verbindung
m2
m0
m1
105
104
s0
s1
s2
Slave-Verbindung
Data
102
PCI-BUS (PORT)

Figur 13

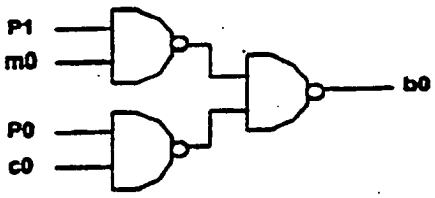
Speicherschreibvorgang



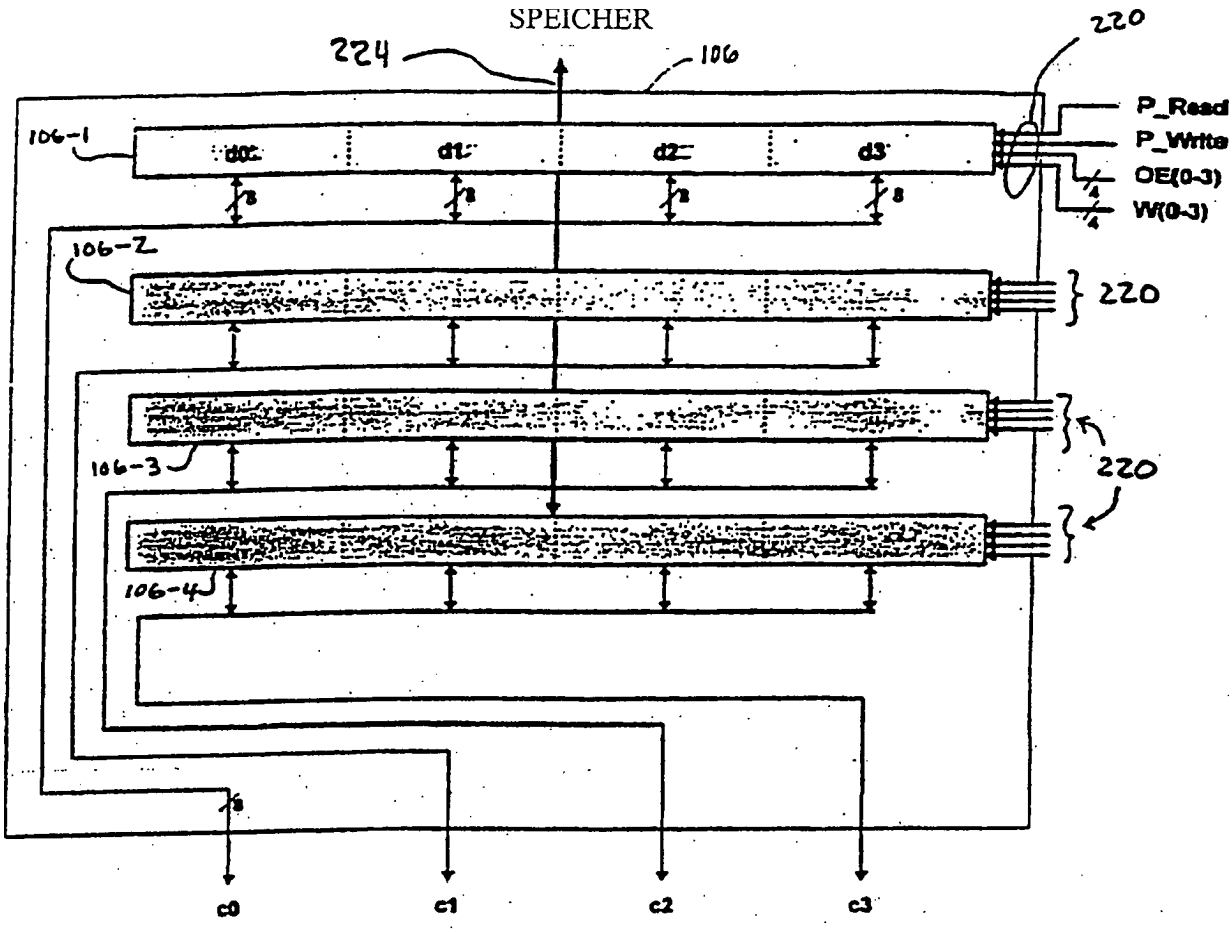
FIGUR 14A



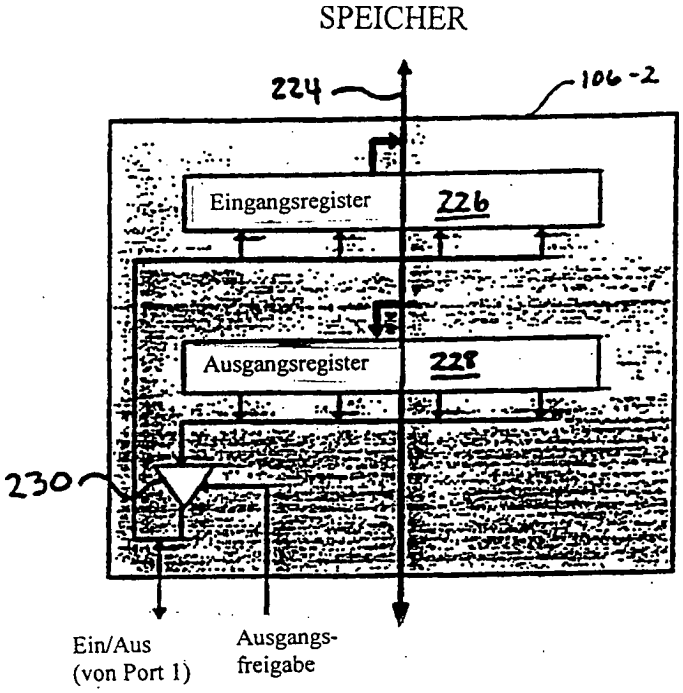
FIGUR 14B



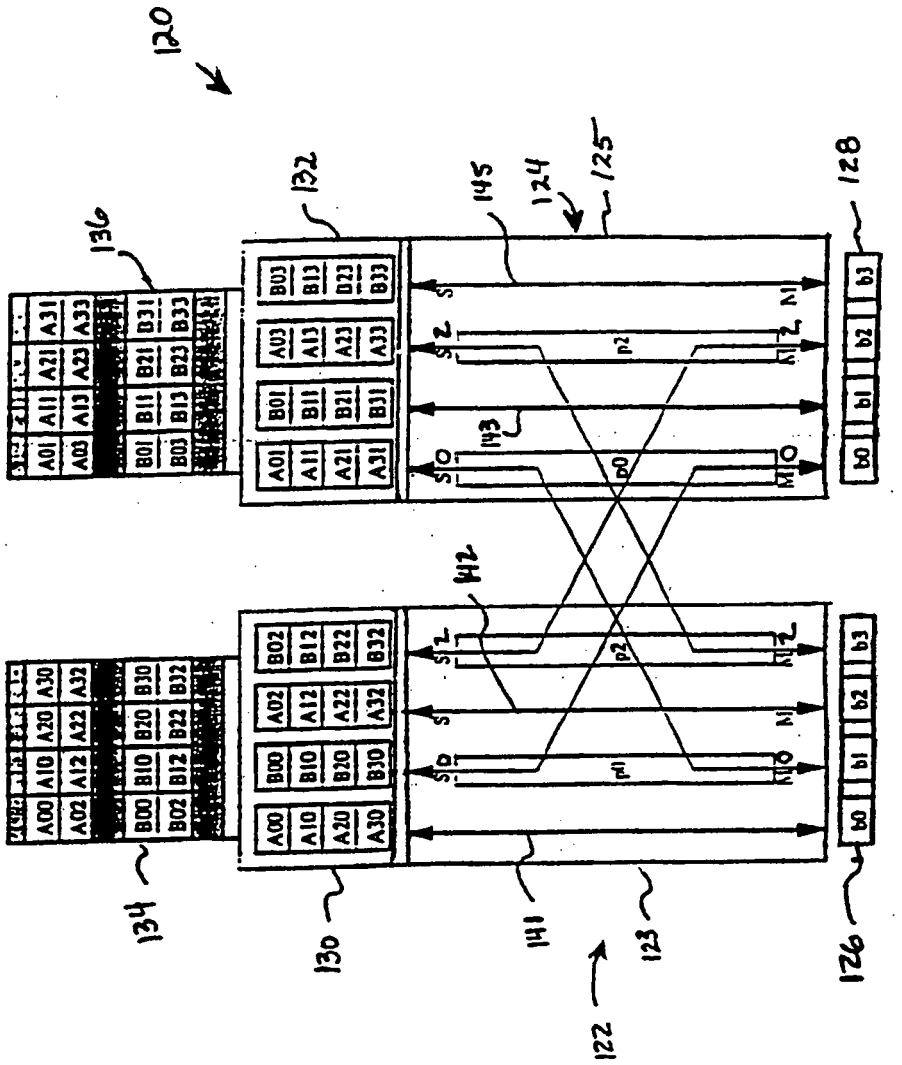
FIGUR 14C


SPEICHER
224
106
220
106-1
d0
d1
d2
d3
P_Read
P_Write
OE(0-3)
W(0-3)
106-2
220
106-3
220
106-4
c0
c1
c2
c3

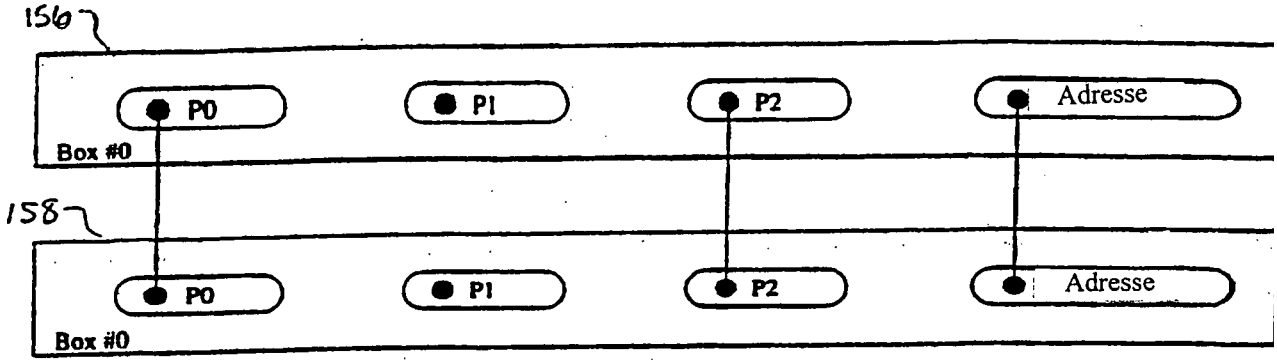
FIGUR 15


SPEICHER
224
106-2
Eingangsregister
226
Ausgangsregister
228
230
Ein/Aus
(von Port 1)
Ausgangs-
freigabe

FIGUR 16

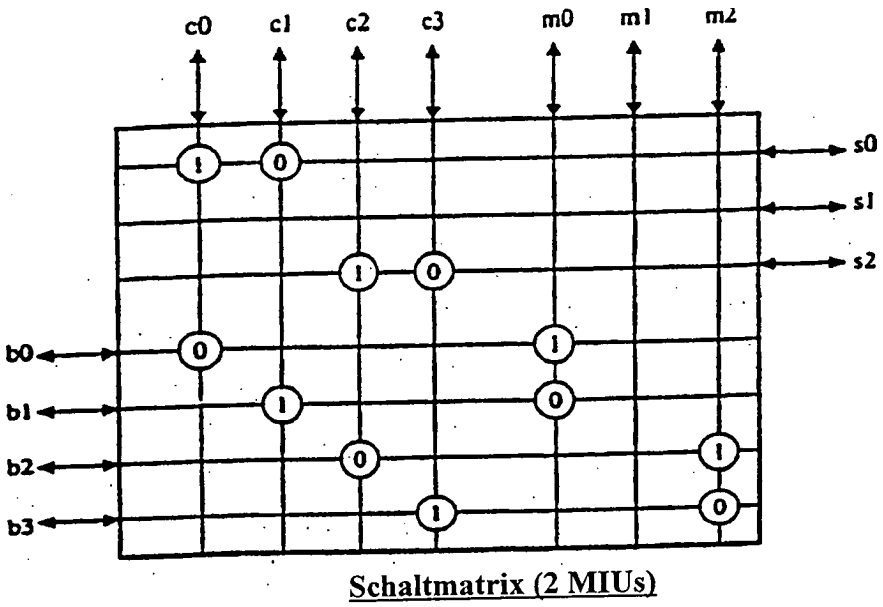

120
122
123
124
125
126
127
128
129
130
132
134
136
141
142
143
145
A00 A01 A02 A03
A10 A11 A12 A13
A20 A21 A22 A23
A30 A31 A32 A33
B00 B01 B02 B03
B10 B11 B12 B13
B20 B21 B22 B23
B30 B31 B32 B33
b0 b1 b2 b3
S0 S1 S2 S3
M0 M1 M2 M3
p0 p1 p2

FIGUR 17



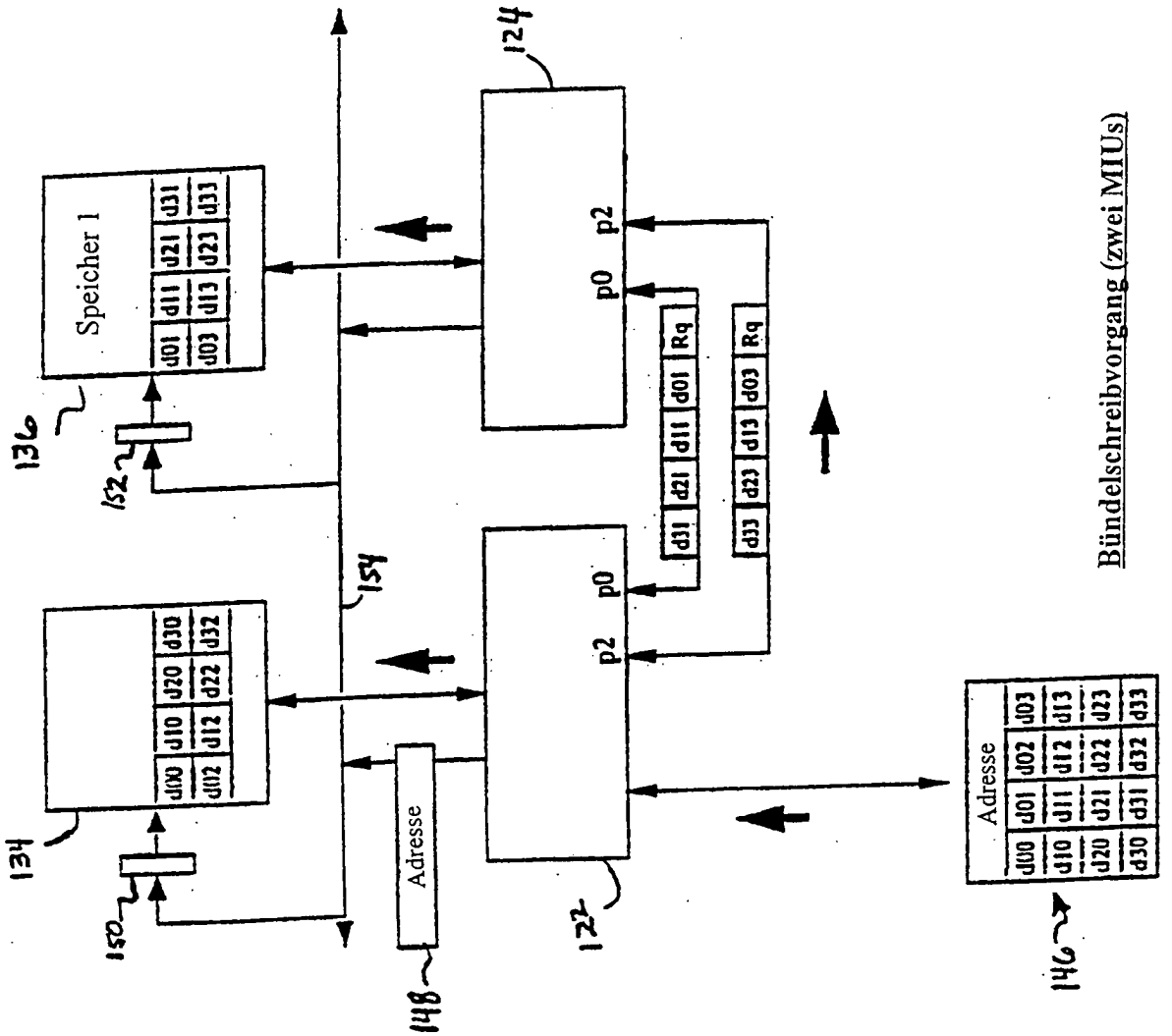
Verbindung von zwei gestapelten Boxen

FIGUR 18



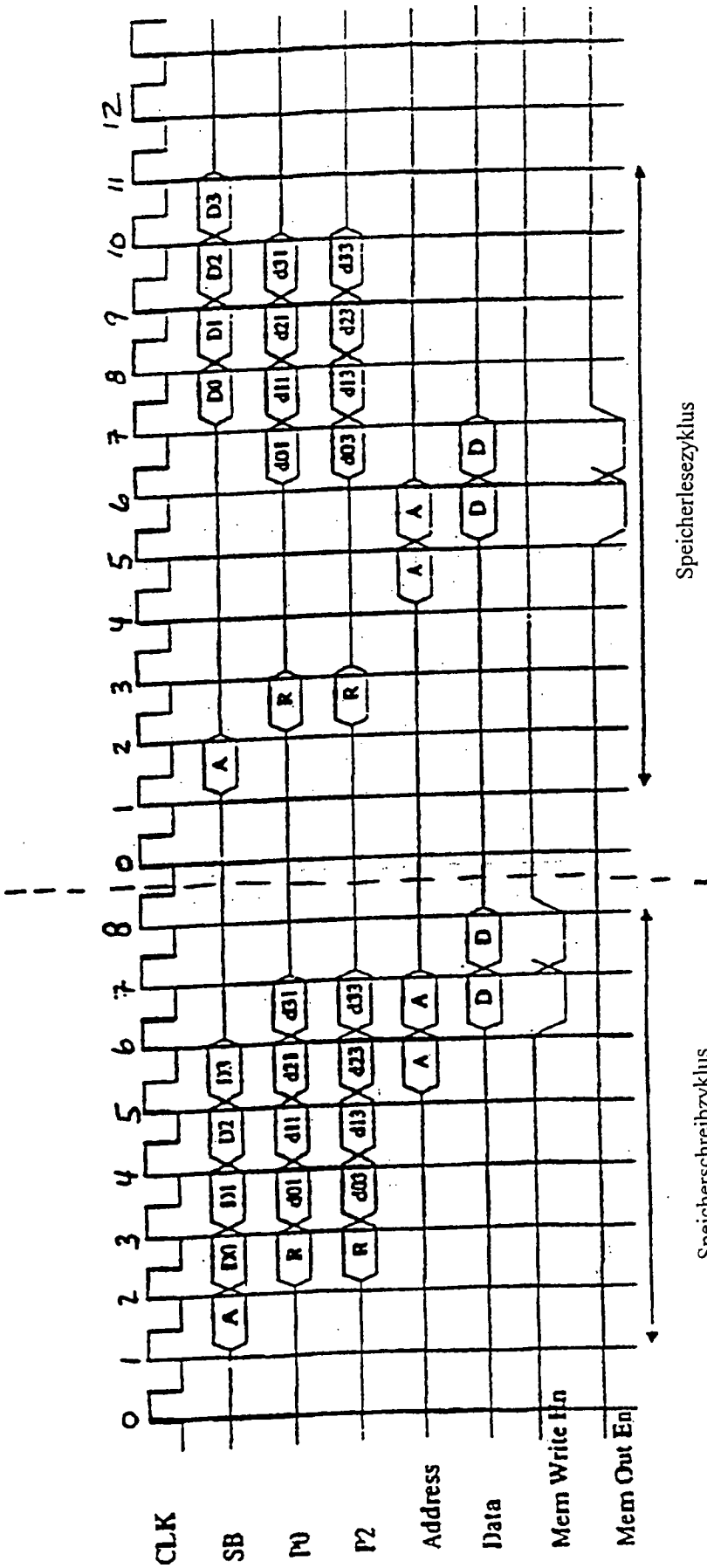
Schaltmatrix (2 MIUs)

FIGUR 19



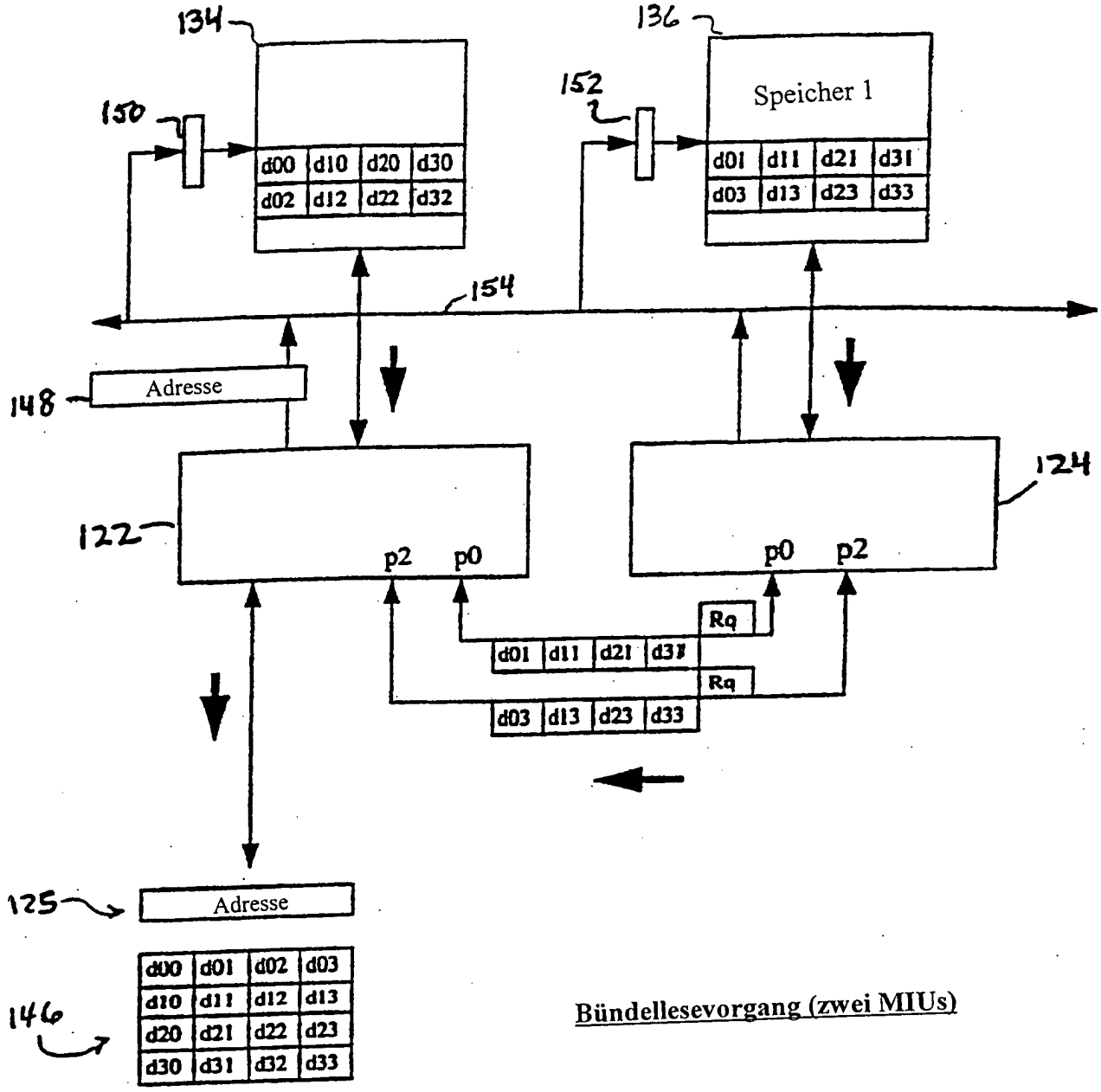
Bündelschreibvorgang (zwei MIUs)

FIGUR 2a



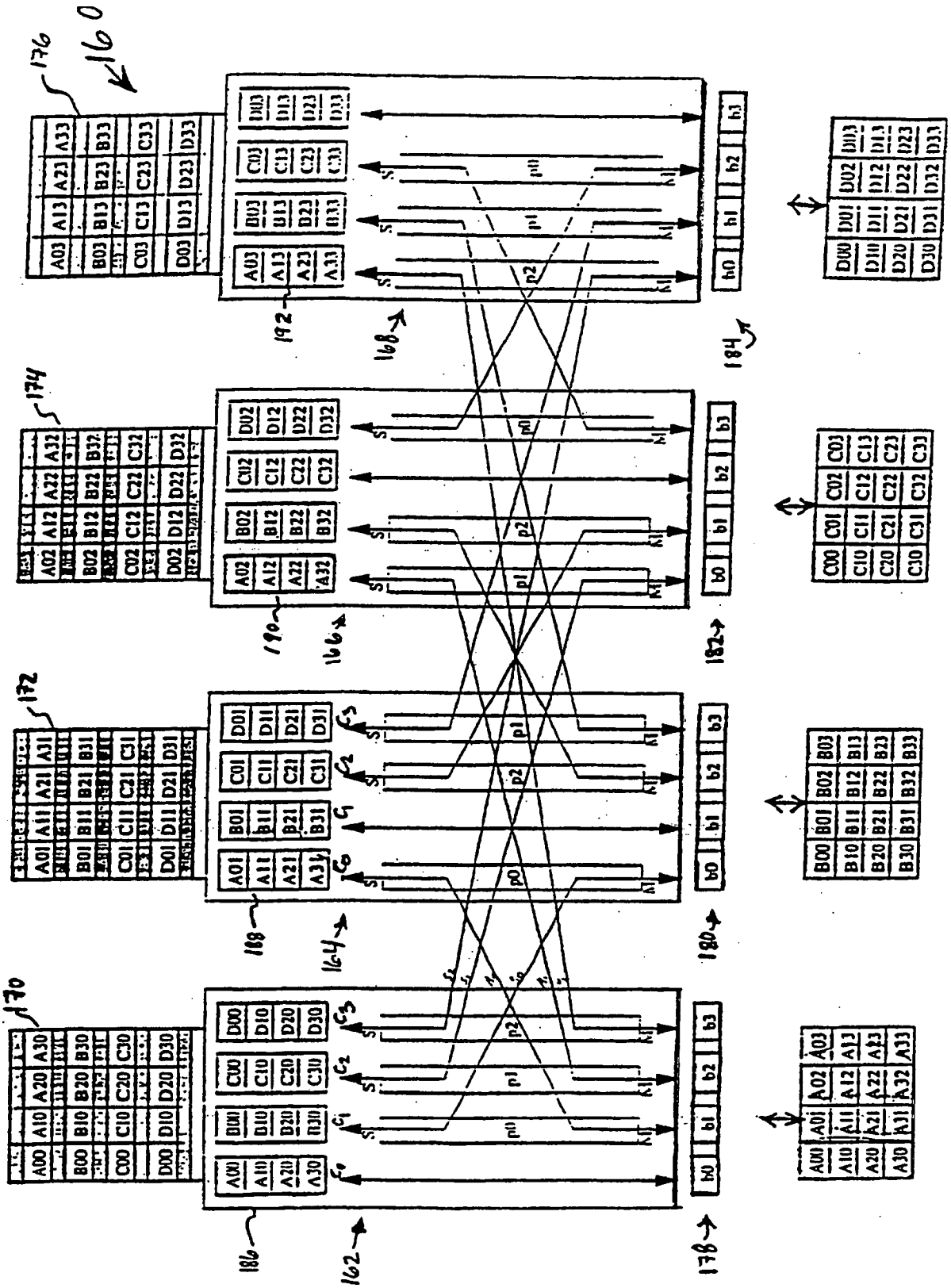
FIGUR 21

FIGUR 23

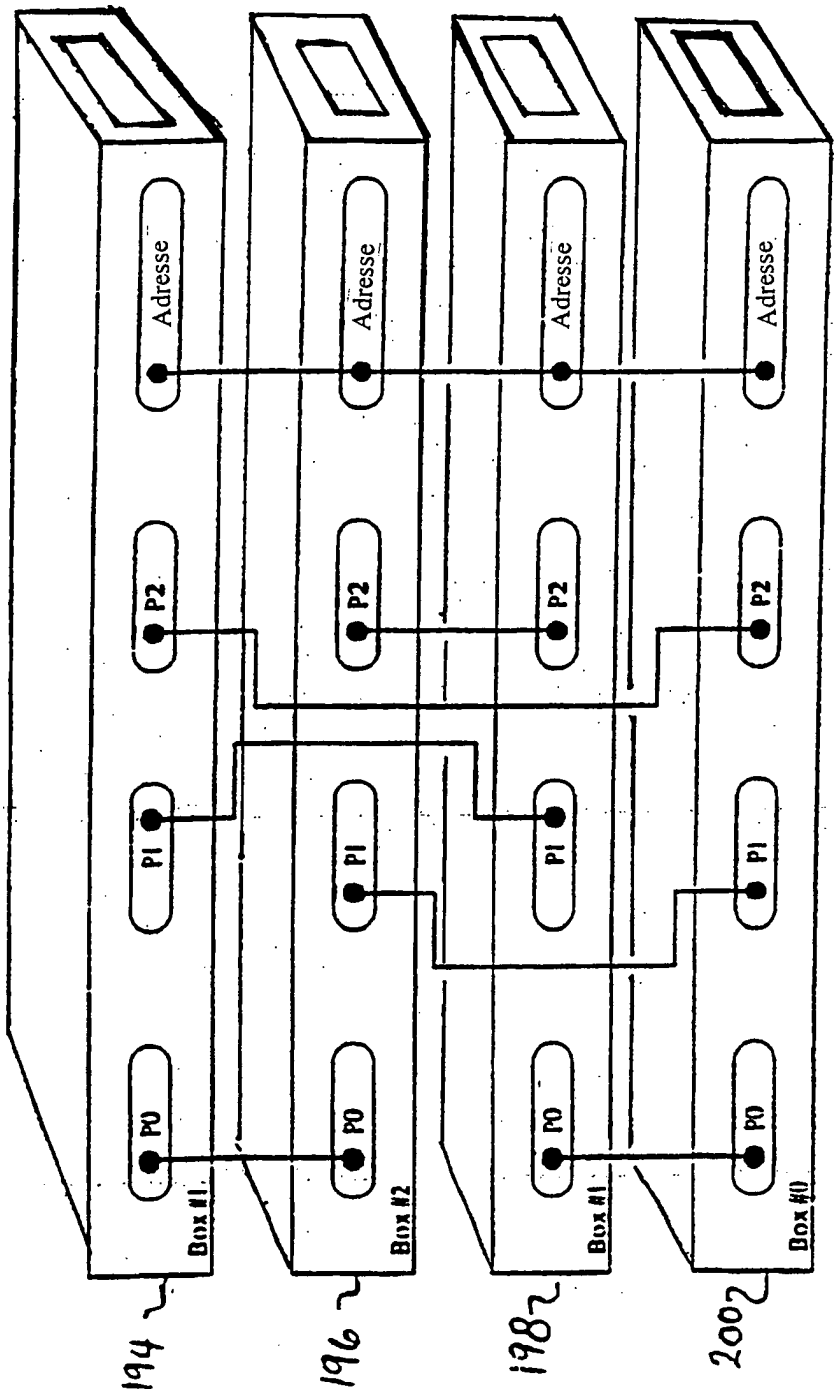


Bündellesevorgang (zwei MIUs)

FIGUR 22

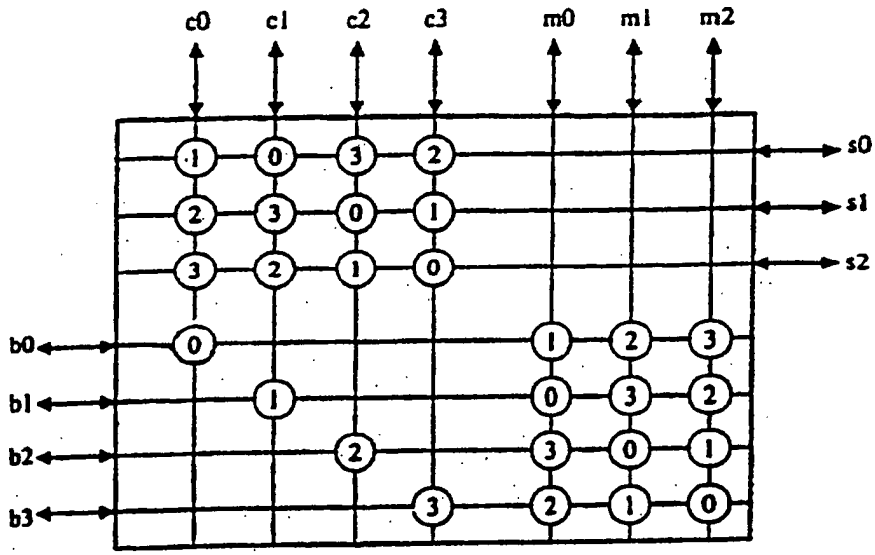


FIGUR 24



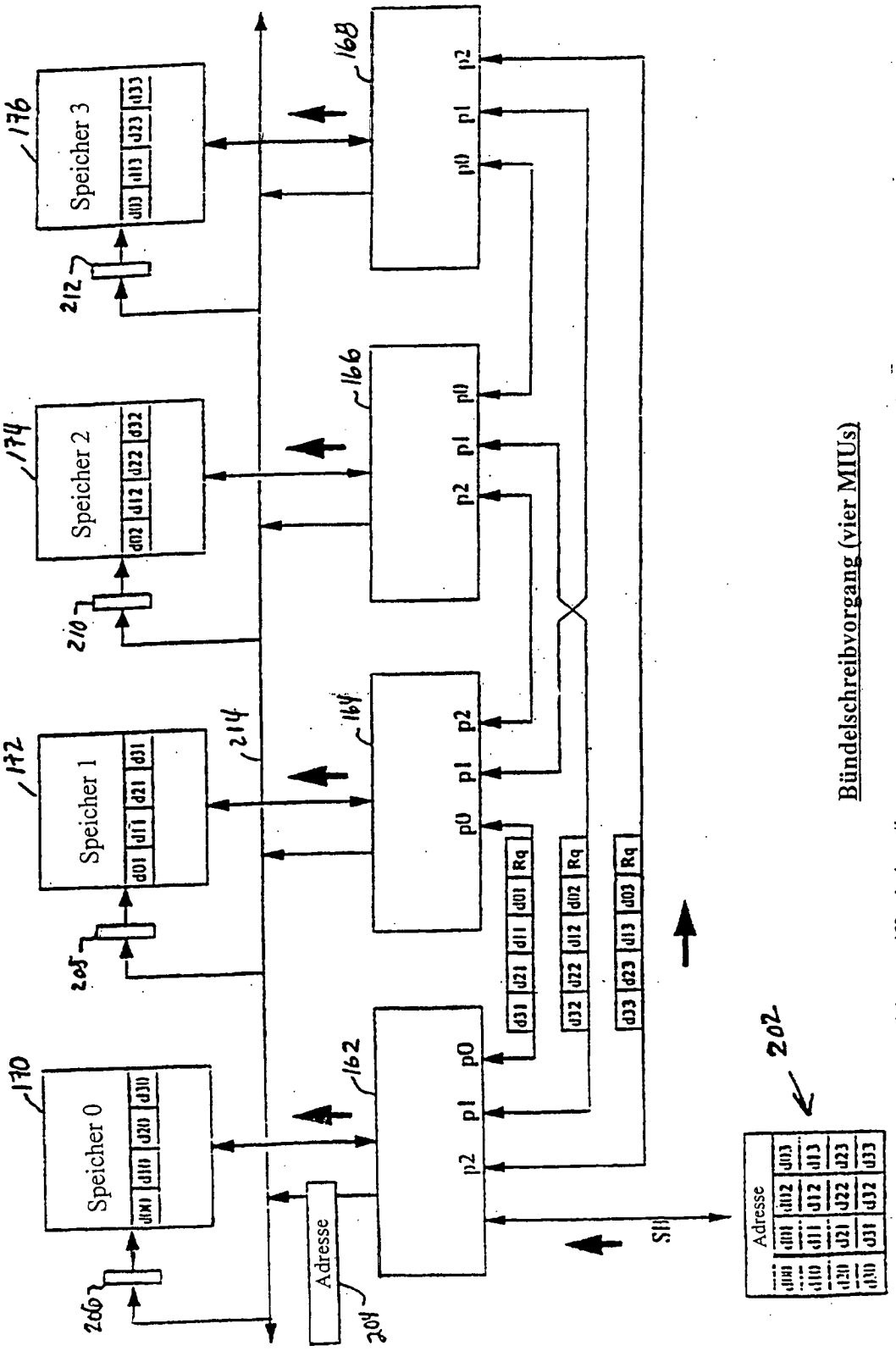
Verbindung von vier gestapelten Boxen

FIGUR 25



Schaltmatrix (4 MIUs)

FIGUR 26

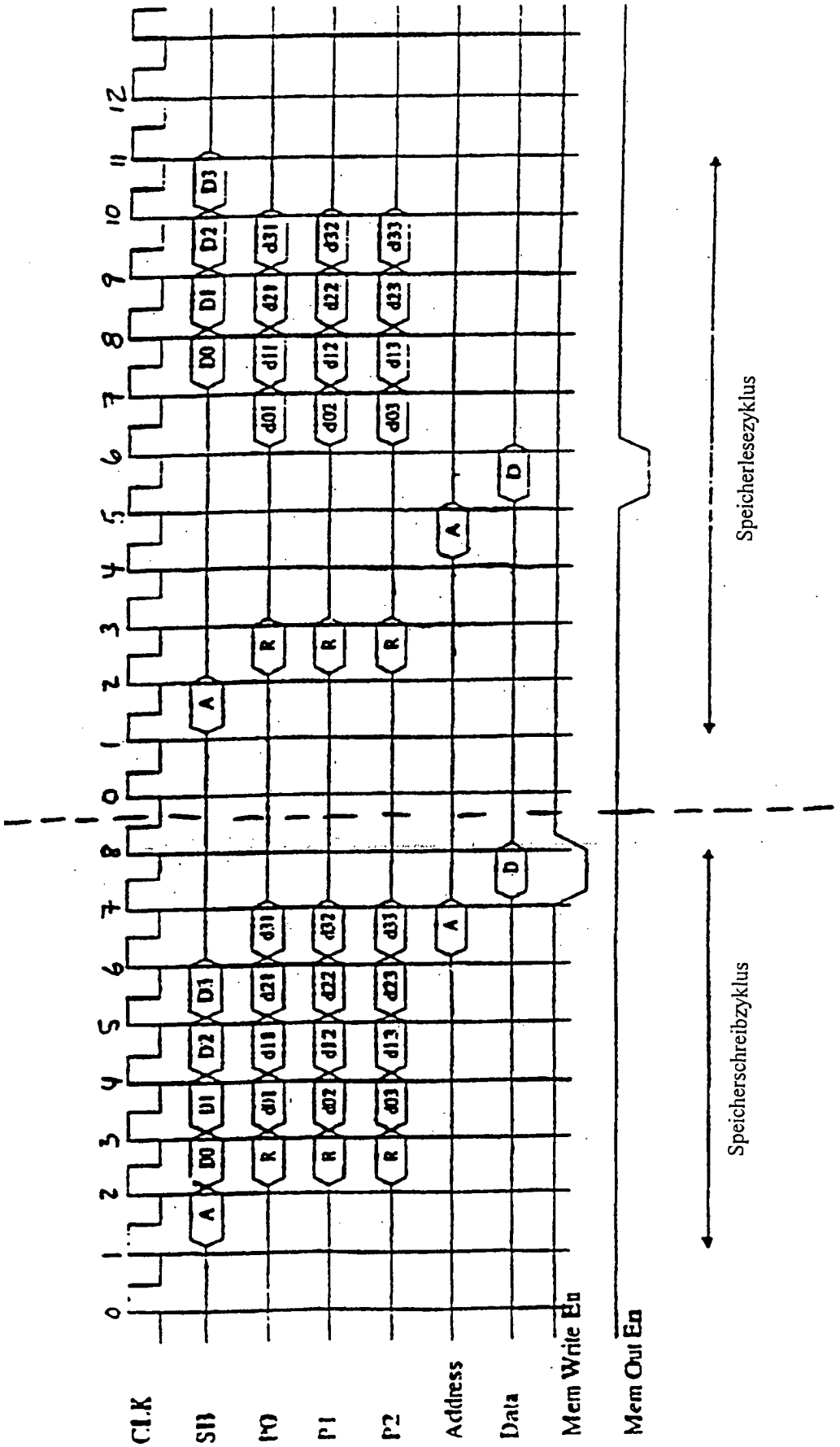


Bündelschreibvorgang (vier MIUs)

Hinweis: Graue Pfeile zeigen Datenbewegungsrichtung

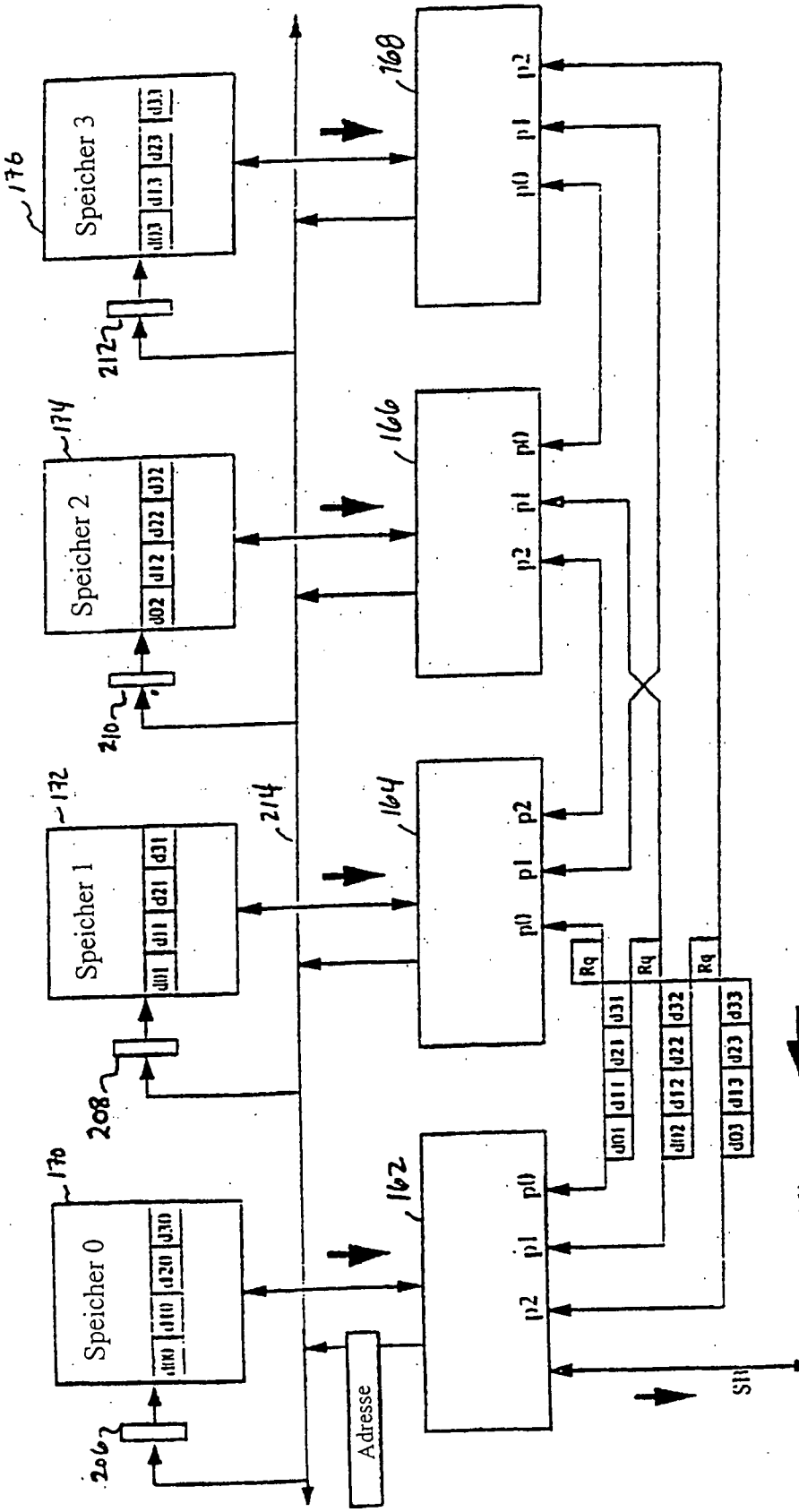
MAI = Speicherzugriffsschnittstelle
SB = Segmentbus

FIGUR 27

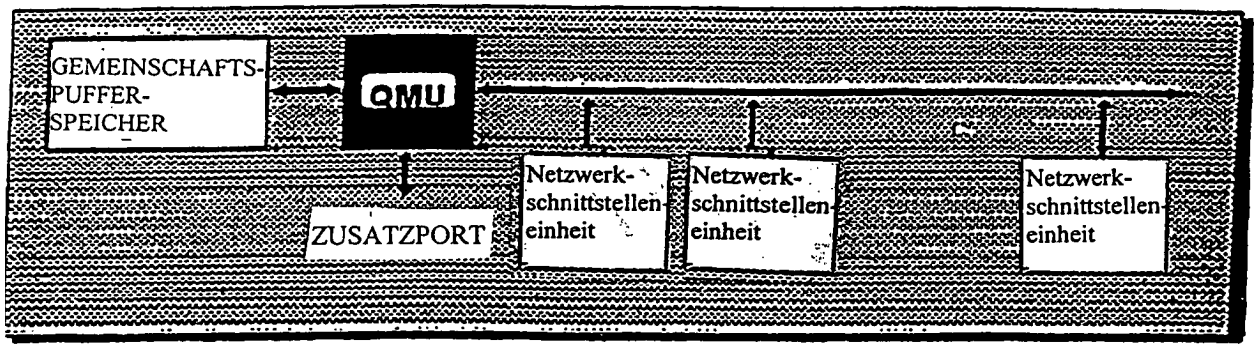


FIGUR 28

FIGUR 30

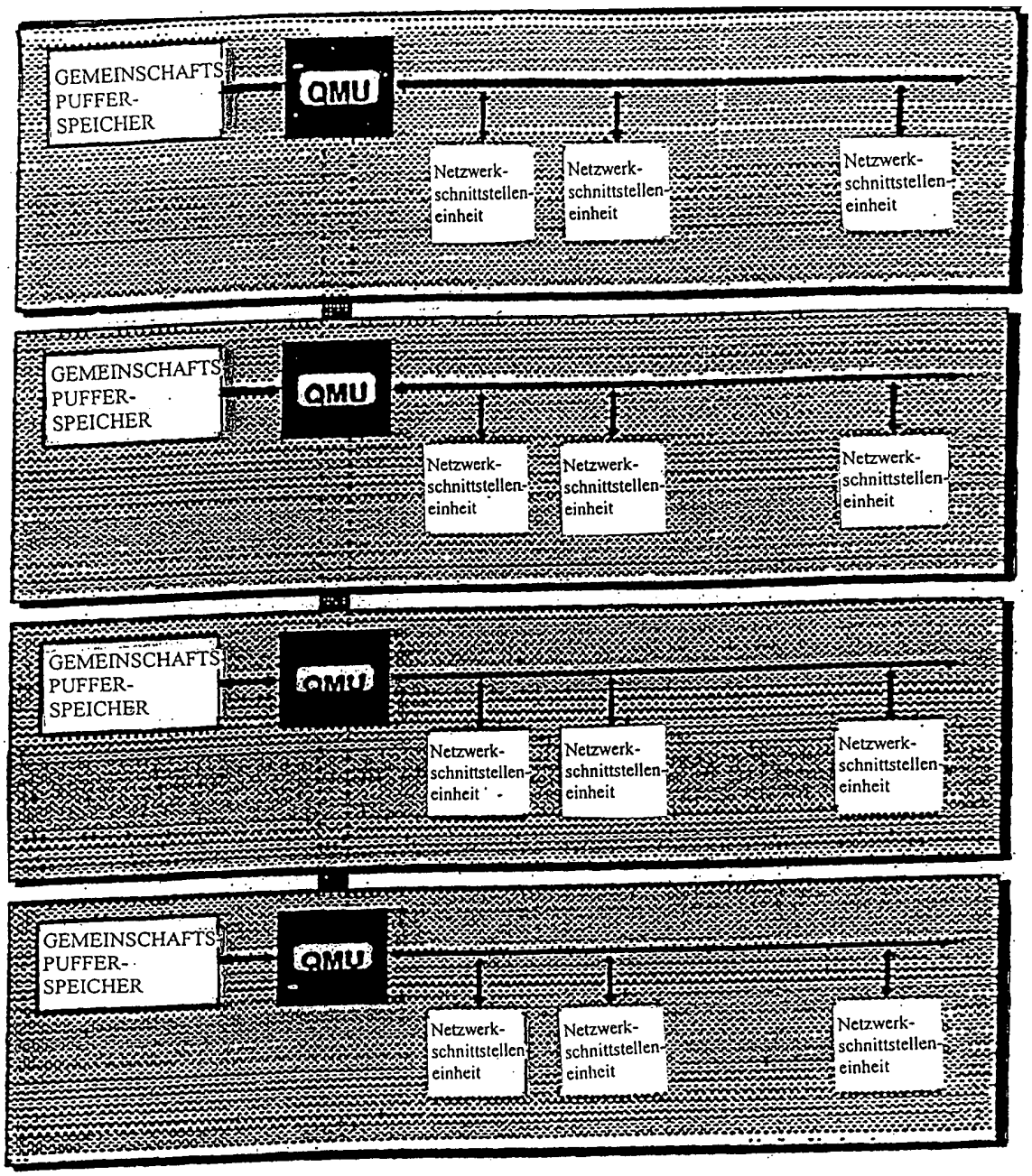

Speicher 0
Speicher 1
Speicher 2
Speicher 3
Adresse
Adresse
SB
MAI = Speicherzugriffsschnittstelle
SB = Segmentbus
Bündellesevorgang (vier MIUs)
Hinweis: Graue Pfeile zeigen Datenbewegungsrichtung

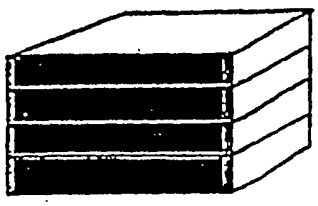
FIGUR 29



Schaltergrundeinheit

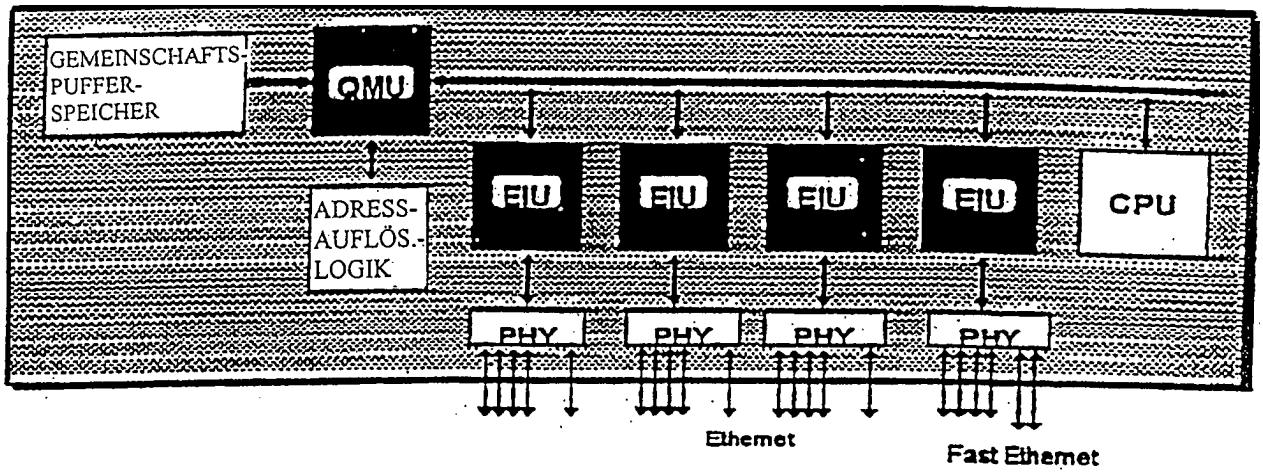
FIGUR 31





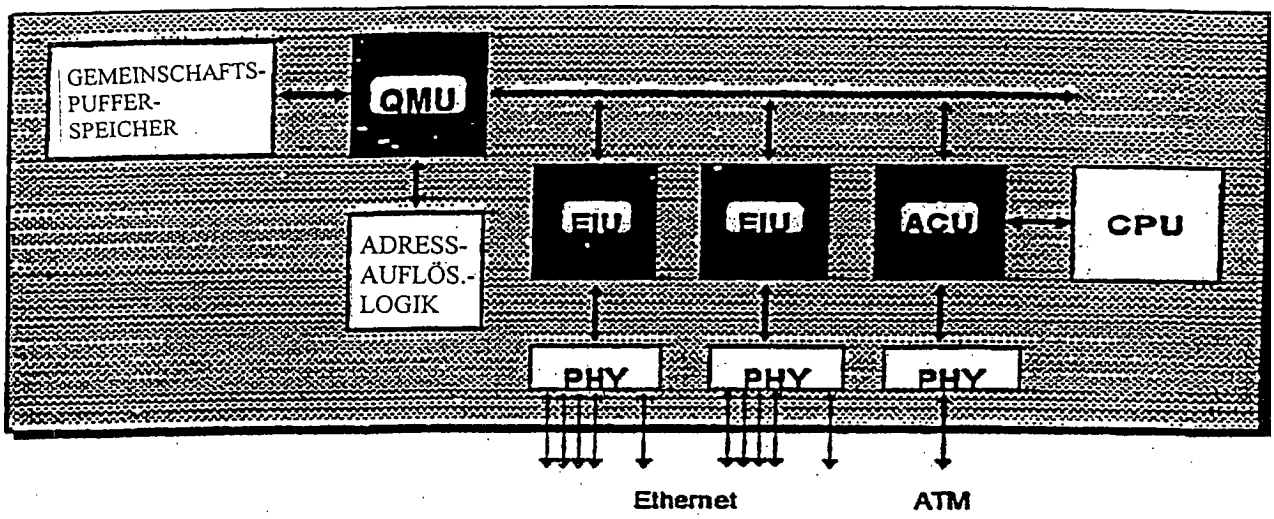
Maximale Schalterkonfiguration

FIGUR 32



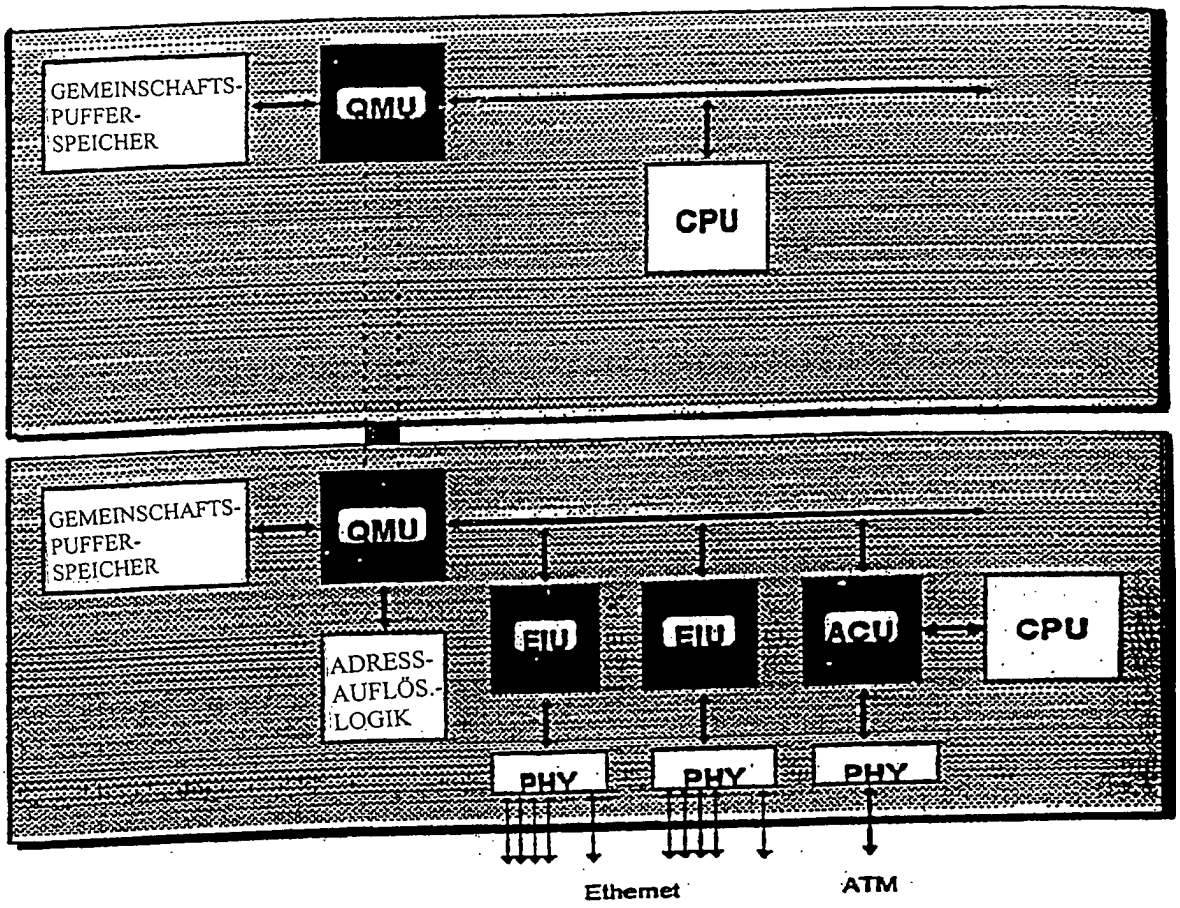
32-Port-Ethernet-Schalter mit dreißig 10-Mbps-Ports und zwei 100-Mbps-Ports

FIGUR 33



16-Port-Ethernet-Schalter mit ATM-Uplink

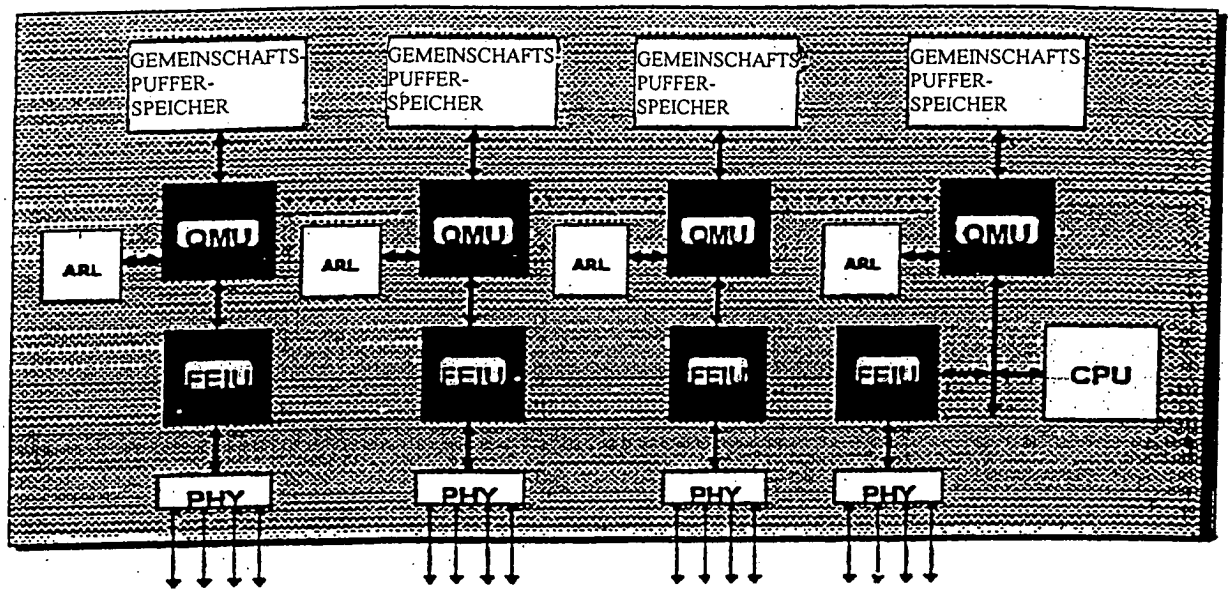
FIGUR 34





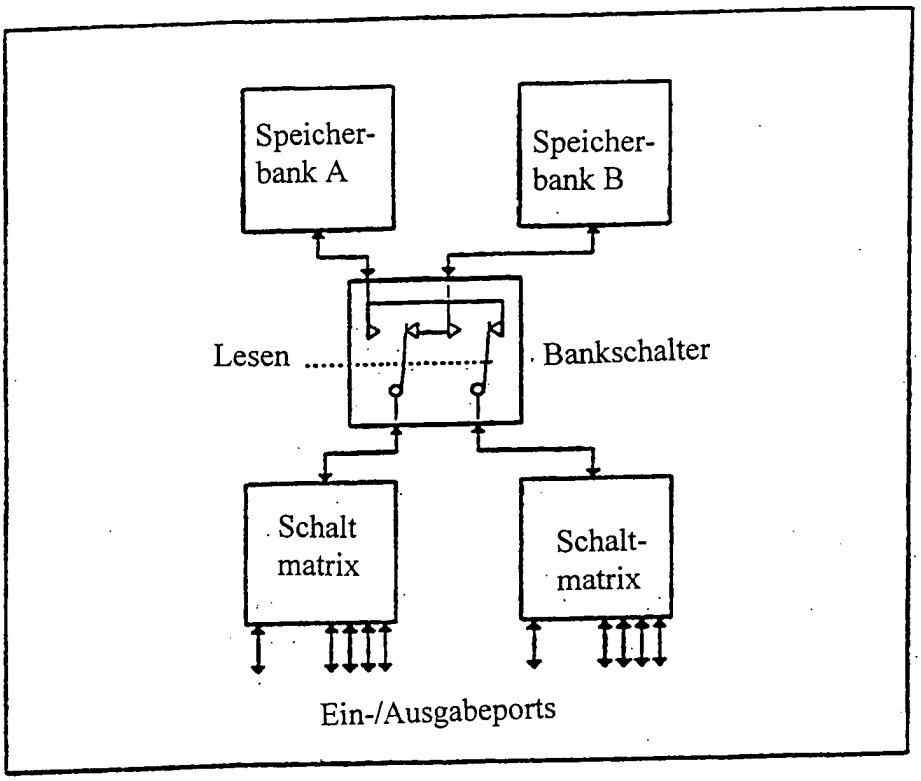
stapelbarer Schalter/Router

FIGUR 35

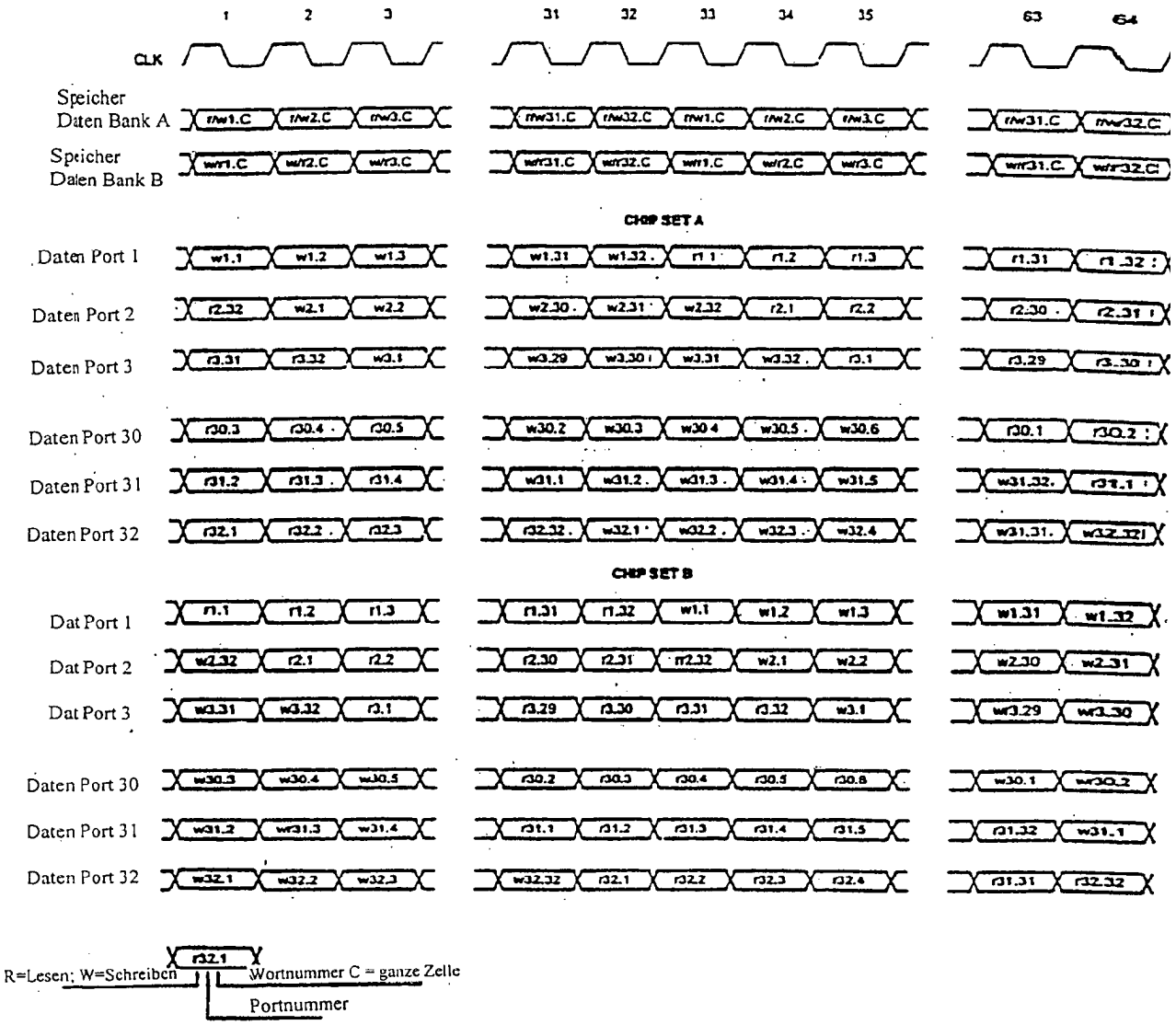


16-Port Fast Ethernet Schalter

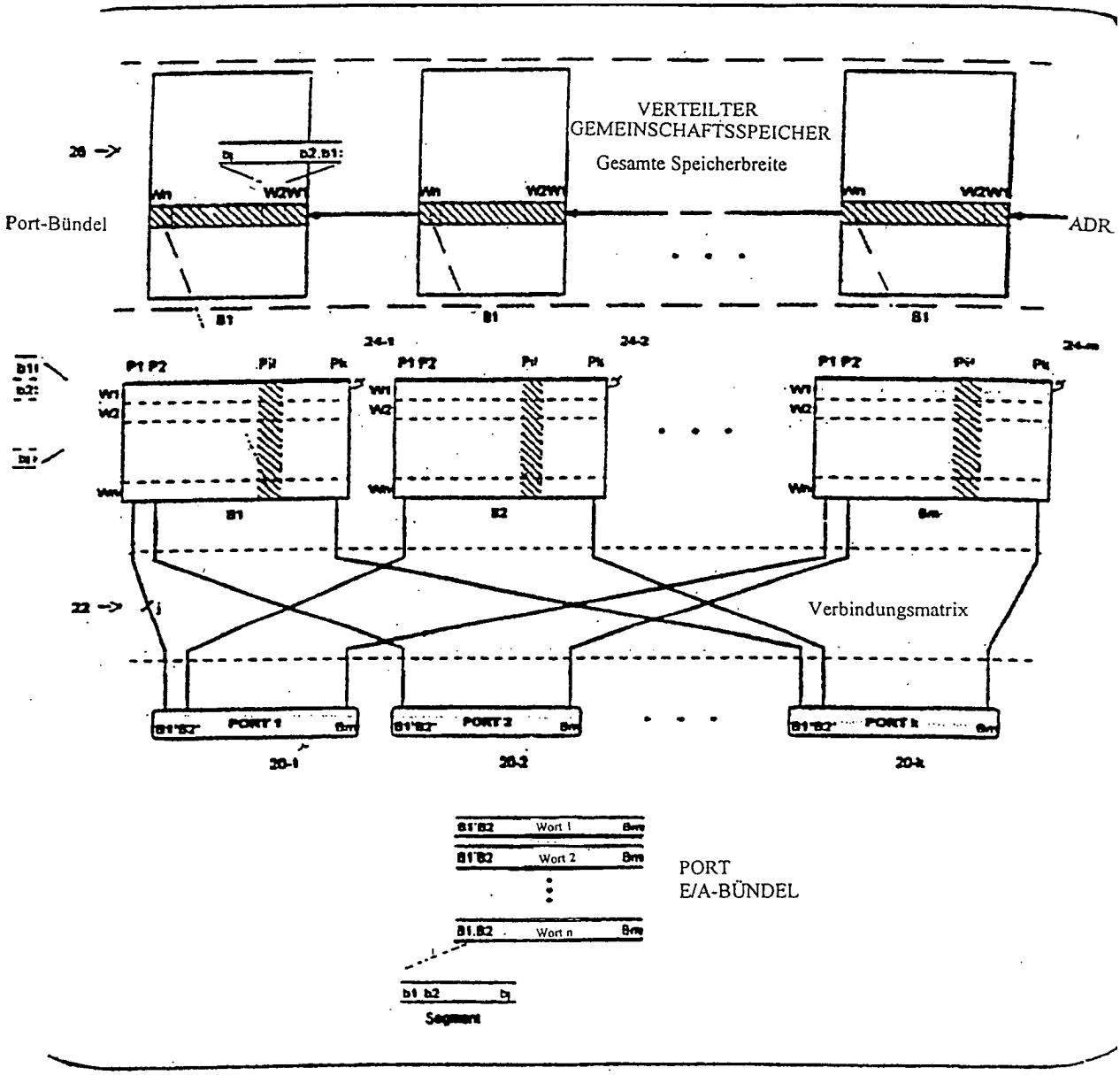
FIGUR 36


Speicher-
bank A
Speicher-
bank B
Lesen
Bankschalter
Schalt
matrix
Schalt-
matrix
Ein-/Ausgabeports

FIGUR 37


1
2
3
31
32
33
34
35
63
64
CLK
Speicher Daten Bank A
Speicher Daten Bank B
CHIP SET A
Daten Port 1
Daten Port 2
Daten Port 3
Daten Port 30
Daten Port 31
Daten Port 32
CHIP SET B
Dat Port 1
Dat Port 2
Dat Port 3
Daten Port 30
Daten Port 31
Daten Port 32
R=Lesen; W=Schreiben
Wortnummer C = ganze Zelle
Portnummer

FIGUR 38



Erweiterter Datenfluss

FIGUR 39