



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 691 33 550 T2** 2007.01.11

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 197 830 B1**

(21) Deutsches Aktenzeichen: **691 33 550.8**

(96) Europäisches Aktenzeichen: **02 000 378.6**

(96) Europäischer Anmeldetag: **16.04.1991**

(97) Erstveröffentlichung durch das EPA: **17.04.2002**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **04.10.2006**

(47) Veröffentlichungstag im Patentblatt: **11.01.2007**

(51) Int Cl.⁸: **G06F 1/04** (2006.01)

G11C 8/04 (2006.01)

G11C 11/401 (2006.01)

(30) Unionspriorität:

510898 18.04.1990 US

(73) Patentinhaber:

Rambus Inc., Los Altos, Calif., US

(74) Vertreter:

Eisenführ, Speiser & Partner, 28195 Bremen

(84) Benannte Vertragsstaaten:

DE, FR, GB, IT

(72) Erfinder:

Farmwald, Michael, Portola Valley, California

94028, US; Horowitz, Mark, Palo Alto, California

94306, US

(54) Bezeichnung: **Eingang/Ausgang einer integrierten Schaltung mit einer Hochleistungsbusschnittstelle**

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

Gebiet der Erfindung

[0001] Es wird eine Bus-Schnittstelle für integrierte Schaltkreis-Vorrichtungen für Computer- und Videosysteme beschrieben, die eine Hochgeschwindigkeitsübertragung von Blöcken von Daten, insbesondere zu und von Speichervorrichtungen, mit reduziertem Energieverbrauch und erhöhter System-Zuverlässigkeit gestattet. Ein neues Verfahren einer physikalischen Implementierung der Bus-Architektur wird ebenfalls beschrieben.

Hintergrund der Erfindung

[0002] Halbleiter-Computerspeicher wurden traditionell ausgebildet und aufgebaut, um eine Speichervorrichtung für jedes Bit oder kleine Gruppen von Bits jedes einzelnen Computer-Wortes zu verwenden, wobei die Wortgröße durch die Wahl des Computers bestimmt ist. Typische Wortgrößen reichen von 4 bis 64 Bits. Jede Speichervorrichtung ist normalerweise parallel an eine Reihe von Adressleitungen und an eine Reihe von Datenleitungen angeschlossen. Wenn der Computer aus einer bestimmten Speicherposition lesen oder darin schreiben möchte, wird eine Adresse auf die Adressleitungen gegeben und einige oder alle Speichervorrichtungen bzw. Speichergeräte werden unter Verwendung einer separaten Geräte-Auswahlleitung für jede benötigte Vorrichtung aktiviert. Eine oder mehrere Vorrichtungen können an jede Datenleitung angeschlossen sein, aber normalerweise ist nur eine geringe Anzahl von Datenleitungen an eine einzelne Speichervorrichtung angeschlossen. Daher ist die Datenleitung 0 an die Vorrichtung(en) 0 angeschlossen, Datenleitung 1 ist an die Vorrichtungen) 1 angeschlossen, und so weiter. Die Daten werden somit für jeden Speicher-Lese- oder -Schreib-Vorgang parallel angesprochen oder bereitgestellt. Damit das System einwandfrei arbeitet, muss jedes einzelne Speicherbit in jeder Speichervorrichtung zuverlässig und korrekt arbeiten.

[0003] Um das Konzept der vorliegenden Erfindung zu verstehen, ist es hilfreich, die Architektur konventioneller Speichervorrichtungen zu betrachten. In fast allen Arten von Speichervorrichtungen (einschließlich der am weitesten verbreitet genutzten dynamischen Wahlfrei-Zugriffs-Speicher (DRAM), Static RAM (SRAM) und Nur-Lese-Speicher (ROM)-Vorrichtungen) wird eine große Anzahl von Bits zu jeder Zeit parallel angesprochen, wenn das System einen Speicherzugriffs-Zyklus ausführt. Jedoch überschreitet nur ein geringer Prozentsatz der angesprochenen Bits, welche intern verfügbar sind, jedes Mal, wenn die Speichervorrichtung zyklisch angesprochen wird, die Vorrichtungs-Grenze zu der Außenwelt.

[0004] Gemäß [Fig. 1](#) weisen alle modernen DRAM-, SRAM- und ROM-Ausgestaltungen interne Architekturen mit Zeilen-(Wort-)Leitungen **5** und Spalten-(Bit-)Leitungen **6** auf, um den Speicherzellen zu gestatten, einen zweidimensionalen Bereich **1** aufzuspannen. Ein Datenbit wird an dem Schnittpunkt jeder Wort- und Bit-Leitung gespeichert. Wenn eine bestimmte Wortleitung freigegeben wird, werden alle entsprechenden Datenbits auf die Bitleitungen Obertragen. Einige bekannte DRAMs nutzen den Vorteil dieses Aufbaues, um die Anzahl von Pins zu verringern, die zum Senden der Adresse benötigt werden. Die Adresse einer gegebenen Speicherzelle ist in zwei Adressen aufgeteilt, Zeile und Spalte, von denen jede über einen Bus nur halb so breit wie die Speicherzellen-Adresse, die der Stand der Technik erfordert, gemultiplext werden kann.

Vergleich mit dem Stand der Technik

[0005] Bekannte Speichersysteme haben mit beschränktem Erfolg versucht, das Problem des Hochgeschwindigkeitszugriffs auf Speicher zu lösen. Das U.S.-Patent Nr. 3,821,715 (Hoff et al.) wurde für die Intel Corporation für den frühesten 4-Bit-Mikroprozessor erteilt. Das Patent beschreibt einen Bus, der eine einzelne zentrale Verarbeitungseinheit (CPU) mit mehreren RAMs und ROMs verbindet. Der Bus multiplext Adressen und Daten über einen 4-Bit breiten Bus und verwendet Punkt-zu-Punkt zum Auswählen bestimmter RAMs oder ROMs. Die Zugriffszeit ist festgelegt und nur ein einzelnes Verarbeitungselement ist erlaubt. Es gibt keinen Blockmodus-Typ des Betriebs, und als wichtigstes werden nicht sämtliche Schnittstellensignale zwischen den Vorrichtungen über einen Bus geführt (die ROM- und RAM-Steuerungsleitungen und die RAM-Auswahlleitungen sind Punkt-zu-Punkt).

[0006] In dem U.S.-Patent Nr. 4,315,308 (Jackson) ist ein eine einzelne CPU mit einer Bus-Schnittstelleneinheit verbindender Bus beschrieben. Die Erfindung verwendet gemultiplexte Adress-, Daten- und Steuerungs-Information über einen einzelnen 16-Bit breiten Bus. Blockmodus-Vorgänge sind mit der Länge des als Teil der Steuerungssequenz gesendeten Blockes definiert. Zusätzlich sind Vorgänge mit veränderlicher Zugriffszeit unter Verwendung eines "dehnbaren" Zyklussignals vorgesehen. Es gibt nicht mehrere Verarbeitungselemente und keine Fähigkeit für mehrere ausstehende Anforderungen, und nochmals nicht alle Schnitt-

stellensignale sind Bus-geführt.

[0007] In dem U.S.-Patent Nr. 4,449,207 (Kung et al.) ist ein DRAM beschrieben, welcher Adressen und Daten auf einem internen Bus multiplext. Die externe Schnittstelle zu diesem DRAM ist konventionell mit separaten Steuerungs-, Adress- und Daten-Verbindungen.

[0008] In den U.S.-Patenten Nr. 4,764,846 und 4,706,166 (Go) ist eine 3-D-Packungsanordnung gestapelter Die's mit Verbindungen entlang einer einzelnen Kante beschrieben. Solche Packungen sind wegen der erforderlichen Punkt-zu-Punkt-Verdrahtung zum Verbinden konventioneller Speichervorrichtungen mit Verarbeitungselementen schwer anzuwenden. Beide Patente beschreiben komplexe Schemata zum Lösen dieser Probleme. Es wird kein Versuch unternommen, das Problem durch Verändern der Schnittstelle zu lösen.

[0009] In dem U.S.-Patent Nr. 3,969,706 (Proebsting et al.) wird der nächstkommende Stand der Technik der DRAM-Schnittstelle beschrieben. Die Adresse ist wechselseitig gemultiplext und es sind separate Pins für Daten und Steuerung (RAS, CAS, WE, CS) vorgesehen. Die Anzahl von Pins nimmt mit der Größe des DRAM zu, und viele der Verbindungen müssen in einem solche DRAMs verwendenden Speichersystem Punkt-zu-Punkt ausgeführt werden.

[0010] In dem Stand der Technik sind viele Backplane-Busse beschrieben, aber nicht in der Kombination mit den Merkmalen der Erfindung. Viele Backplane-Busse multiplexen Adressen und Daten auf einem einzelnen Bus (z.B. dem NU-Bus). ELXSI und andere haben aufgeteilte Transaktions-Busse (U.S.-Patent Nr. 4,595,923 und 4,481,625 (Roberts)) implementiert. ELXSI hat ebenso einen Strom-Modus-ECL-Treiber mit relativ niedrigem Spannungs-Hub implementiert (etwa 1 V Hub). Adressraum-Register sind in den meisten Backplane-Bussen als eine Form des Blockmodus-Betriebs implementiert.

[0011] Fast alle modernen Backplane-Busse implementieren eine Art von Arbitrierungs-Schema, aber das in dieser Offenbarung verwendete Arbitrierungs-Schema unterscheidet sich von jedem von diesen. Die U.S.-Patent-Nummern 4,837,682 (Culler), 4,818,985 (Ikeda), 4,779,089 (Theus) und 4,745,548 (Blahut) beschreiben bekannte Schemata. Alle beziehen entweder log N Zusatzsignale (Theus, Blahut) mit ein, wobei N die Anzahl möglicher Bus-Anforderer ist, oder eine zusätzliche Verzögerung zum Erhalten der Steuerung des Busses (Ikeda, Culler). Keiner der in Patenten oder anderer Literatur beschriebenen Busse verwendet nur Busgeführte Verbindungen. Alle enthalten einige Punkt-zu-Punkt-Verbindungen in der Backplane. Keiner der anderen Aspekte dieser Offenbarung wie Leistungsverringerung durch Holen jedes Datenblockes von einer einzelnen Vorrichtung oder kompakte und kostengünstige 3-D-Packung werden auch auf Backplane-Busse angewendet.

[0012] Das in dieser Offenbarung verwendete Taktschema ist zuvor nicht verwendet worden und wäre infolge der durch die Verbinder-Stifte bewirkte Signalbeeinträchtigung in der Tat schwer zu implementieren. Das U.S.-Patent Nr. 4,247,817 (Heller) beschreibt ein zwei Taktleitungen verwendendes Taktschema, stützt sich aber auf sägezahnförmige Taktsignale gegenüber den in der vorliegenden Offenbarung verwendeten Signalen mit normaler Anstiegszeit.

[0013] In dem U.S. Patent Nr. 4,646,279 (Voss) ist ein Video-RAM beschrieben, welcher ein Parallel-Lade-, Seriell-Ausgabe-Schieberegister am Ausgang eines DRAM implementiert. Dies erlaubt allgemein eine deutlich verbesserte Bandbreite (und wurde auf 2, 4 und größere Breite von Ausgabepfaden erweitert). Der übrige Teil der Schnittstelle zu dem DRAM (RAS, CAS, gemultiplexte Adresse etc.) bleibt der gleiche wie bei konventionellen DRAMs.

[0014] Eine Aufgabe der vorliegenden Erfindung besteht darin, eine neue Bus-Schnittstelle in Halbleitervorrichtungen zu verwenden, um einen Hochgeschwindigkeitszugriff auf große Blöcke von Daten von einer einzigen Speichervorrichtung durch einen externen Benutzer der Daten, wie einen Mikroprozessor, in einer effizienten und Kosteneffektiven Weise zu unterstützen.

[0015] Eine weitere Aufgabe dieser Erfindung besteht darin, Vorrichtungen zur Verfügung zu stellen, insbesondere DRAMs, die zur Benutzung mit der Bus-Architektur der Erfindung geeignet ist.

[0016] D.K. Morgan, "The CVAX CMCTL – A CMOS Memory Controller Chip", Digital Technical Journal, Nr. 7, August 1988, Seiten 139 bis 143 bezieht sich auf eine Beschreibung eines Speicher-Controllerchips, welcher in Verbindung mit einem "CVAX-Bus" zu verwenden ist. Das Dokument beschreibt dass der CVAX-Speichercontroller über einen synchronen Bus mit einer CPU kommuniziert und über einen asynchronen Bus mit Speichermodulen, die mit dem Controller verbunden sind. Die Speichermodule, die mit dem Controller verbun-

den sind, enthalten Hunderte von Speichervorrichtungen. Die DRAMS auf den Speichermodulen werden nicht näher beschrieben.

[0017] Die "Intel Application Note AP-132", "Designing Memory Systems with the 8K × 8 IRAM", Juni 1982, Seiten 3-41 bis 3-45; Intel Preliminary Data Sheet "2186–8192 × 8 Bit Integrated RAM", September 1982, Seiten 3-281; und eine undatierte Seite ohne Seitenzahl, die sich auf die "2186 Familie" bezieht, offenbaren einen neuen Typ von RAM, der dafür bestimmt ist, die besten Eigenschaften von SRAM und DRAM zu verbinden. Diese Vorrichtung ist zielgerichtet zur Benutzung in Anwendungen mit kleinen Speicherefordernissen bestimmt. Die offenbarten IRAM-Vorrichtungen sind asynchrone Vorrichtungen (welche kein Taktsignal empfangen) die die Steuerschaltungen enthalten, die zur Steuerung der Refresh-Operationen des Speichers erforderlich sind. Systeme, die zu der Zeit der IRAM-Dokumente bekannt waren, enthielten solche Refresh-Steuer-schaltungen normalerweise in einem externen Speichercontroller

Zusammenfassung der Erfindung

[0018] Die vorliegende Erfindung bezieht sich auf eine synchrone DRAM-Halbleiterspeichervorrichtung wie in Anspruch 1 definiert.

[0019] In [Fig. 2](#) werden ein Standard-DRAM **13**, **14**, ROM (oder SRAM) **12**, Mikroprozessor CPU **11**, E/A-Vorrichtung, Disk-Controller oder andere Spezialvorrichtungen wie ein Hochgeschwindigkeitsschalter modifiziert, um eine vollständig Bus-basierte Schnittstelle statt der bekannten Kombination von Punkt-zu-Punkt- und Bus-basierten Verdrahtungen zu verwenden, welche bei konventionellen Versionen dieser Vorrichtungen verwendet werden. Der neue Bus beinhaltet Taktsignale, Stromversorgung und gemultiplexte Adress-, Daten- und Steuerungs-Signale. In einer bevorzugten Implementierung transportieren 8 Bus-Datenleitungen und eine AddressValid-Busleitung Adress-, Daten- und Steuerinformationen für Speicheradressen bis zu 40 Bits Breite. Ein Durchschnittsfachmann erkennt, dass 16 Bus-Datenleitungen oder andere Anzahlen von Bus-Datenleitungen verwendet werden können, um die Lehre dieser Erfindung zu implementieren. Der neue Bus wird verwendet, um Elemente wie Speicher-, Peripherie-, Schalt- und Verarbeitungseinheiten anzuschließen.

[0020] In dieser Erfindung empfangen DRAMs und andere Vorrichtungen Adress- und Steuerinformationen über den Bus und senden oder empfangen angeforderte Daten über den gleichen Bus. Jede Speichervorrichtung enthält nur eine einzelne Bus-Schnittstelle ohne zusätzliche Signal-Pins. Andere Vorrichtungen, die in dem System enthalten sein können, können an den Bus oder andere Nicht-Bus-Leitungen wie Eingabe/Ausgabe-Leitungen angeschlossen sein. Der Bus unterstützt große Datenblock-Übertragungen und aufgeteilte Transaktionen, um einem Benutzer zu erlauben, eine hohe Busnutzung zu verwirklichen. Diese Fähigkeit, schnell einen großen Datenblock für eine einzelne Vorrichtung zu einem Zeitpunkt zu lesen oder schreiben, ist ein wesentlicher Vorteil dieser Erfindung.

[0021] Die mit diesem Bus verbundenen DRAMs unterscheiden sich von konventionellen DRAMs auf mehrere Weisen. Register sind vorgesehen, welche Steuerungs-Informationen, Geräte-Identifikation, Geräte-Typ und andere für den Chip geeignete Informationen wie den Adressbereich für jeden unabhängigen Teil des Gerätes speichern können. Neue Bus-Schnittstellenschaltungen müssen hinzugefügt werden, und das Innere der bekannten DRAM-Geräte muss modifiziert werden, so dass sie Daten für den und von dem Bus mit der Spitzen-Datengeschwindigkeit des Busses bereitstellen und entgegennehmen können. Dies erfordert Änderungen in der Spalten-Zugriffsschaltung in dem DRAM, mit lediglich einer minimalen Erhöhung der Die-Größe. Eine Schaltung ist vorgesehen, um einen gering verzerrten internen Gerätetakt für Geräte auf dem Bus zu erzeugen, und andere Schaltungen stellen Demultiplex-Eingangs- und Multiplex-Ausgangs-Signale bereit.

[0022] Eine hohe Bus-Bandbreite wird verwirklicht durch Betreiben des Busses mit einer sehr hohen Taktfrequenz (mehrere hundert MHz). Diese hohe Taktfrequenz wird durch die erzwungene Umgebung des Busses ermöglicht. Die Busleitungen sind impedanzgesteuerte, doppelt abgeschlossene Leitungen. Bei einer Datenrate von 500 MHz ist die maximale Bus-Ausbreitungszeit geringer als 1 ns (die physikalische Buslänge beträgt etwa 10 cm). Aufgrund der verwendeten Packung kann zusätzlich der Abstand der Pins sehr nahe an dem Abstand der Anschlussflächen sein. Die aus den einzelnen Geräten resultierende Last auf dem Bus ist sehr klein. In einer bevorzugten Implementierung erlaubt dies allgemein Stiff-Kapazitäten von 1–2 pF und Induktivitäten von 0,5–2 nH. Jedes in [Fig. 3](#) gezeigte Gerät **15**, **16**, **17** hat nur an einer Seite Pins und diese Pins sind direkt mit dem Bus **18** verbunden. Ein Transceiver **19** kann enthalten sein, um mehrere Einheiten an einen Bus höherer Ordnung durch Pins **20** anzupassen.

[0023] Ein primäres Ergebnis der Architektur gemäß dieser Erfindung ist es, die Bandbreite des DRAM-Zu-

griffs zu erhöhen. Die Erfindung reduziert außerdem Herstellungs- und Produktionskosten, den Stromverbrauch und sie erhöht die Packungsdichte und Systemzuverlässigkeit.

Kurzbeschreibung der Zeichnungen

- [0024] [Fig. 1](#) ist eine Darstellung, welche eine Grund-2-D-Anordnung von Speichervorrichtungen darstellt;
- [0025] [Fig. 2](#) ist ein vereinfachtes Blockschaltbild, welches die Parallelverbindung sämtlicher Busleitungen und der seriellen Rücksetzleitung mit jedem Gerät in dem System darstellt;
- [0026] [Fig. 3](#) ist eine perspektivische Ansicht eines Systems, welches die 3-D-Packung von Halbleitervorrichtungen auf dem primären Bus darstellt;
- [0027] [Fig. 4](#) zeigt das Format eines Anforderungspaketes;
- [0028] [Fig. 5](#) zeigt das Format einer erneut versuchten Antwort von einem Slave;
- [0029] [Fig. 6](#) zeigt die Buszyklen, nachdem eine Anforderungspaket-Kollision auf dem Bus aufgetreten ist, und wie die Konfliktlösung gehandhabt wird;
- [0030] [Fig. 7](#) zeigt das Timing, wodurch Signale von zwei Geräten vorübergehend überlappen und den Bus gleichzeitig ansteuern können;
- [0031] [Fig. 8](#) zeigt die Verbindung und das Timing zwischen Bustakten und Geräten auf dem Bus;
- [0032] [Fig. 9](#) ist eine perspektivische Ansicht, die zeigt, wie Transceiver verwendet werden können, um eine Anzahl von Buseinheiten an einen Transceiver-Bus anzuschließen;
- [0033] [Fig. 10](#) ist ein vereinfachtes Blockschaltbild einer Eingabe/Ausgabe-Schaltung, welche zum Verbinden von Geräten mit dem Bus verwendet wird;
- [0034] [Fig. 11](#) ist ein vereinfachtes Schaltbild eines als Buseingangs-Empfänger verwendeten, getakteten Le-severstärkers;
- [0035] [Fig. 12](#) ist ein Blockschaltbild, das zeigt, wie der interne Gerätetakt aus zwei Bustaktsignalen unter Verwendung eines Satzes anpassbarer Verzögerungsleitungen erzeugt wird;
- [0036] [Fig. 13](#) ist ein Zeitdiagramm, das die Beziehung von Signalen in dem Blockschaltbild in [Fig. 12](#) zeigt;
- [0037] [Fig. 14](#) ist ein Zeitdiagramm einer bevorzugten Einrichtung zum Implementieren des Rücksetzvorgangs;
- [0038] [Fig. 15](#) ist eine Darstellung der allgemeinen Organisation eines in 8 Teilbereiche aufgeteilten 4 Mbit-DRAM.

Detaillierte Beschreibung

[0039] Die vorliegende Erfindung ist ausgebildet, eine synchrone DRAM Halbleitervorrichtung zur Verwendung mit einem gemultiplexten Hochgeschwindigkeits-Bus zur Kommunikation zwischen Verarbeitungsvorrichtungen und Speichervorrichtungen bereitzustellen. Der Bus kann ebenfalls verwendet werden, um Verarbeitungsgeräte und andere Geräte wie E/A-Schnittstellen oder Disk-Controller mit oder ohne Speichervorrichtungen auf dem Bus zu verbinden. Der Bus besteht aus einer relativ kleinen Anzahl von Leitungen, welche parallel an jede Vorrichtung auf dem Bus angeschlossen sind. Der Bus transportiert im Wesentlichen sämtliche Adress-, Daten- und Steuerinformationen, welche von den Vorrichtungen zur Kommunikation mit anderen Vorrichtungen auf dem Bus benötigt werden. In vielen die vorliegende Erfindung verwendenden Systemen transportiert der Bus fast jedes Signal zwischen jeder Vorrichtung in dem gesamten System. Es besteht kein Bedarf nach separaten Vorrichtungs-Auswahlleitungen, da die Vorrichtungs-Auswahlinformation für jede Vorrichtung auf dem Bus über den Bus transportiert wird. Es besteht kein Bedarf für separate Adress- und Datenleitungen, da Adress- und Dateninformationen über die gleichen Leitungen gesendet werden können. Bei der Verwendung der hier beschriebenen Organisation können sehr große Adressen (40 Bits in der bevorzugten Implemen-

tierung) und große Datenblöcke (1024 Bytes) über eine geringe Anzahl von Busleitungen (8 plus eine Steuerungsleitung in der bevorzugten Implementierung) gesendet werden.

[0040] Im Wesentlichen können sämtliche der von einem Computersystem benötigten Signale über den Bus gesendet werden. Ein Durchschnittsfachmann erkennt, dass bestimmte Geräte wie CPUs an andere Signalleitungen und möglicherweise an unabhängige Busse, z.B. einen Bus zu einem unabhängigen Cache-Speicher, zusätzlich zu dem Bus dieses Ausführungsbeispiels angeschlossen sein können. Bestimmte Geräte, z.B. Cross-Point-Schalter, können an mehrere unabhängige Busse dieses Ausführungsbeispiels angeschlossen werden. In der bevorzugten Implementierung sind Speichergeräte vorgesehen, die keine anderen Verbindungen als die hierin beschriebenen Busverbindungen aufweisen, und CPUs sind vorgesehen, die den Bus dieses Ausführungsbeispiels als die grundlegende, wenn nicht exklusive Verbindung zum Speicher und zu anderen Geräten auf dem Bus verwenden.

[0041] Alle modernen DRAM-, SRAM- und ROM-Ausbildungen haben interne Architekturen mit Zeilen- (Wort-) und Spalten- (Bit-) Leitungen, um effizient einen 2-D-Bereich abzudecken. In [Fig. 1](#) wird ein Datenbit an dem Schnittpunkt zwischen jeder Wortleitung **5** und jeder Bitleitung **6** gespeichert. Wenn eine bestimmte Wortleitung freigegeben wird, werden sämtliche, entsprechende Datenbits auf die Bitleitungen übertragen. Diese Daten, etwa 4000 Bits zugleich bei einem 4 MBit DRAM, werden dann in Spalten-Leseverstärker **3** geladen und durch die E/A-Schaltungen zur Verwendung bereitgehalten.

[0042] In dem hier dargestellten Ausführungsbeispiel werden die Daten von den Leseverstärkern mit 32 Bits zugleich auf einem internen Gerätebus freigegeben, der mit etwa 125 MHz läuft. Dieser interne Gerätebus bewegt die Daten zu der Peripherie der Geräte, wo die Daten in einer 8 Bit breiten externen Bus-Schnittstelle gemultiplext werden, die mit etwa 500 MHz arbeitet.

[0043] Die Bus-Architektur gemäß diesem Ausführungsbeispiel verbindet Master- oder Bus-Controller-Vorrichtungen, die CPUs, Direkt-Speicher-Zugriffs-Geräte (DMAs) oder Fließkomma-Einheiten (FPUs), und Slave-Geräte wie DRAM-, SRAM- oder ROM-Speichergeräte. Ein Slave-Gerät reagiert auf Steuerungssignale; ein Master sendet Steuerungssignale. Der Durchschnittsfachmann erkennt, dass sich einige Geräte zu verschiedenen Zeiten als Master und Slave verhalten können, abhängig von dem Betriebsmodus und dem Status des Systems. Zum Beispiel hat ein Speichergerät typisch nur Slave-Funktionen, während ein DMA-Controller, Disk-Controller oder eine CPU Slave- und Master-Funktionen beinhalten kann. Viele andere Halbleitergeräte einschließlich E/A-Geräte, Disk-Controller oder anderen Spezialgeräte wie Hochgeschwindigkeitsschalter können zur Verwendung mit dem Bus gemäß dem vorliegenden Ausführungsbeispiel modifiziert werden.

[0044] Jedes Halbleitergerät enthält einen Satz interner Register, der vorzugsweise ein Geräte-Identifizierungs- (Geräte-ID-)Register, ein Gerätetyp-Deskriptor-Register, ein Steuerungs-Register und andere Register mit anderen für den Gerätetyp relevanten Informationen aufweist. In einer bevorzugten Ausführungsform enthalten die an den Bus angeschlossenen Halbleitergeräte Register, welche die innerhalb des Gerätes enthaltene Speicheradresse bestimmen, und Zugriffszeit-Register, welche einen Satz von einer oder mehreren Verzögerungszeiten speichern, zu welcher das Gerät verfügbar sein kann oder soll, um Daten zu senden oder zu empfangen.

[0045] Die meisten dieser Register können modifiziert und bevorzugt als Teil einer Initialisierungssequenz gesetzt werden, die auftritt, wenn das System eingeschaltet oder zurückgesetzt wird. Während der Initialisierungssequenz wird jedem Gerät auf dem Bus eine einzelne Geräte-ID-Nummer zugeordnet, welche in dem Geräte-ID-Register gespeichert wird. Ein Bus-Master kann dann diese Geräte-ID-Nummern verwenden, um geeignete Register in anderen Geräten, einschließlich Zugriffszeit-Registern, Steuerungsregistern und Speicherregistern, anzusprechen und zu setzen, um das System zu konfigurieren. Jeder Slave kann eines oder mehrere Zugriffszeit-Register (vier in einer bevorzugten Ausführungsform) aufweisen. In einer bevorzugten Ausführungsform wird ein Zugriffszeit-Register in jedem Slave permanent oder semi-permanent mit einem festen Wert programmiert, um bestimmte Steuerungsfunktionen zu unterstützen. Eine bevorzugte Implementierung einer Initialisierungssequenz wird unten detaillierter beschrieben.

[0046] Sämtliche zwischen Master-Geräten und Slave-Geräten gesendeten Informationen werden über den externen Bus gesendet, welcher zum Beispiel 8 Bits breit sein kann. Dies wird verwirklicht durch Definieren eines Protokolls, wobei ein Master-Gerät, wie ein Mikroprozessor, eine ausschließliche Steuerung des externen Busses übernimmt (d.h., der Bus-Master wird) und eine Bus-Transaktion durch Senden eines Anforderungspaketes (eine Folge von Bytes mit Adress- und Steuerungs-Informationen) zu einem oder mehreren Slave-Geräten auf dem Bus auslöst. Eine Adresse kann gemäß den Lehren dieses Ausführungsbeispiels aus 16

bis 40 oder mehr Bits bestehen. Jeder Slave auf dem Bus muss das Anforderungspaket dekodieren, um zu erkennen, ob der Slave auf das Paket antworten soll. Der Slave, an den das Paket gerichtet ist, muss dann eine interne Verarbeitung beginnen, die benötigt wird, um die geforderte Bus-Transaktion in der geforderten Zeit auszuführen. Der anfordernde Master kann ebenfalls bestimmte interne Verarbeitungen übermitteln müssen, bevor die Bus-Transaktion beginnt. Nach einer festgelegten Zugriffszeit antworten die Slaves durch Zurücksenden von einem oder mehreren Bytes (8 Bits) von Daten oder durch Speichern der von dem Bus verfügbar gemachten Informationen. Mehr als eine Zugriffszeit kann bereitgestellt werden, um unterschiedliche Arten von Antworten zu erlauben, die zu unterschiedlichen Zeiten auftreten.

[0047] Ein Anforderungspaket und der entsprechende Bus-Zugriff werden durch eine ausgewählte Anzahl von Bus-Zyklen separiert und erlauben dem Bus, in den dazwischen liegenden Bus-Zyklen durch die gleichen oder andere Master für zusätzliche Anforderungen oder kurze Bus-Zugriffe verwendet zu werden. Somit sind mehrere unabhängige Zugriffe erlaubt und erlauben eine maximale Nutzung des Busses zur Übertragung von Daten kurzer Blöcke. Übertragungen von Daten langer Blöcke nutzen den Bus effizient, auch ohne Überlappen, da der Überhang infolge von Bus-Adress-Steuerungs- und Zugriffs-Zeiten, verglichen mit der Gesamtzeit zum Anfordern und Übertragen des Blockes, gering ist.

Geräte-Adress-Zuordnung

[0048] Ein weiterer Einzelaspekt dieser Erfindung ist es, dass jedes Speichergerät ein vollständiges, unabhängiges Speicher-Subsystem mit der gesamten Funktionalität einer bekannten Speicherplatine in einem konventionellen Backplane-Bus-Computersystem ist. Einzelne Speichergeräte können einen einzelnen Speicherabschnitt enthalten oder können weiter in mehr als einen diskreten Speicherabschnitt unterteilt sein. Speichergeräte beinhalten bevorzugt Speicheradress-Register für jeden diskreten Speicherabschnitt. Ein fehlerhaftes Speichergerät (oder auch ein Teilabschnitt eines Gerätes), kann "ausgeblendet" werden, mit lediglich dem Verlust eines kleinen Bruchteils des Speichers, um im wesentlichen die vollen Systemmöglichkeiten beizubehalten. Ein Ausblenden fehlerhafter Geräte kann auf zwei Weisen verwirklicht werden, die beide mit dieser Erfindung kompatibel sind.

[0049] Das bevorzugte Verfahren verwendet Adressregister in jedem Speichergerät (oder jedem unabhängigen, diskreten Abschnitt davon) zum Speichern von Informationen, welche den Bereich der Busadressen bestimmen, auf welche dieses Speichergerät reagiert. Dies ist vergleichbar mit bekannten Schemata, welche bei Speicherplatinen in konventionellen Backplane-Bus-Systemen verwendet werden. Die Adressregister können einen einzelnen Zeiger beinhalten, der gewöhnlich auf einen Block bekannter Größe zeigt, einen Zeiger und einen festen oder variablen Blockgrößenwert oder zwei Zeiger, von denen einer auf den Anfang und einer auf das Ende (oder die "Spitze" und den "Boden") jedes Speicherblockes zeigt. Durch geeignete Einstellungen der Adressregister kann eine Reihe funktionaler Speichergeräte oder diskreter Speicherabschnitte veranlasst werden, auf einen fortlaufenden Bereich von Adressen zu reagieren, welche den Systemzugriff auf einen fortlaufenden, guten Speicherblock geben, beschränkt vielmehr durch die Anzahl mit dem Bus verbundener, guter Geräte. Ein Speicherblock in einem ersten Speichergerät oder Speicherabschnitt kann einem bestimmten Bereich von Adressen zugeordnet werden, dann kann ein Speicherblock in einem nächsten Speichergerät oder Speicherabschnitt Adressen zugeordnet werden, welche mit einer um eins höheren (oder abhängig von dem Speicheraufbau, niedrigeren) Adresse als die letzte Adresse des vorherigen Blockes beginnt.

[0050] Vorzugsweise umfassen Geräte zur Verwendung in dieser Ausführungsform eine gerätetypische Registerinformationen, welche die Art des Chips und wieviel Speicher in welcher Konfiguration in diesem Gerät verfügbar ist, festlegen. Ein Master kann einen geeigneten Speichertest wie Lesen und Schreiben jeder Speicherzelle in einer oder mehreren ausgewählten Reihenfolgen ausführen, um eine einwandfreie Funktion jedes ansprechbaren, diskreten Teiles des Speichers (teilweise basierend auf Informationen wie der Geräte-ID-Nummer und dem Gerätetyp) und Schreibadresswerte (bis zu 40 Bits in der bevorzugten Ausführungsform, 10^{12} Bytes) zu prüfen, bevorzugt fortlaufend in Geräte-Adressraum-Registern. Nicht funktionierende oder schadhafte Speicherabschnitte können einem besonderen Adresswert zugeordnet werden, welchen das System interpretieren kann, um die Verwendung dieses Speichers zu vermeiden.

[0051] Der zweite Ansatz belastet den System-Master oder die Master mit der Belastung der Vermeidung der schlechten Geräte. CPUs und DMA-Controller haben normalerweise eine Art von Translations-Seitenblick-Puffern (Translation look-aside buffer TLBs), welche der physikalischen (Bus-)Adresse virtuell zugeordnet sind. Mit einer relativ einfachen Software können die TLBs programmiert werden, um nur arbeitenden Speicher zu verwenden (Datenstrukturen, welche funktionierenden Speicher beschreiben, werden leicht erzeugt). Für Master, welche keine TLBs enthalten (z.B. einen Videoanzeigengenerator) kann ein kleiner, einfacher RAM verwen-

det werden, um einen zusammenhängenden Bereich von Adressen auf den Adressen der funktionierenden Speichergeräte abzubilden.

[0052] Ein anderes Schema arbeitet so und erlaubt einem System, einen signifikanten Prozentsatz nicht funktionierender Geräte aufzuweisen und immer noch weiter mit dem verbleibenden Speicher zu arbeiten. Dies bedeutet, dass gemäß diesem Ausführungsbeispiel aufgebaute Systeme eine deutlich verbesserte Zuverlässigkeit gegenüber vorhandenen Systemen aufweisen, einschließlich der Fähigkeit, Systeme mit fast keinen Teilbild-Fehlern aufzubauen.

Bus

[0053] Die bevorzugte Bus-Architektur gemäß diesem Ausführungsbeispiel umfasst 11 Signale: BusData [0:7]; AddrValid; Clk1 und Clk2; plus einem Eingangreferenzpegel und an jedes Gerät parallel angeschlossene Spannungs- und Masseleitungen. Signale werden auf dem Bus während konventioneller Bus-Zyklen angesteuert. Die Notation "Signal[i:j]" bezieht sich auf einen bestimmten Bereich von Signalen oder Leitungen, zum Beispiel bedeutet BusData[0:7] BusData0 BusData1, ..., BusData7. Die Bus-Leitungen für die BusData[0:7]-Signale bilden einen Bytebreiten, gemultiplexten Daten/Adress/Steuerungs-Bus. AddrValid wird verwendet, um anzugeben, wann der Bus eine gültige Adressanforderung hält, und weist einen Slave an, die Bus-Daten als eine Adresse zu dekodieren, und, wenn die Adresse in dem Slave enthalten ist, die anhängige Anforderung zu handhaben. Die zwei Takte stellen zusammen einen synchronisierten Hochgeschwindigkeitstakt für alle Geräte auf dem Bus bereit. Zusätzlich zu den Bus-geführten Signalen ist eine weitere Leitung (ResetIn, ResetOut) vorhanden, welche jedes Gerät in Reihe zur Verwendung während der Initialisierung verbindet, um jedem Gerät in dem System eine einzelne Geräte-ID-Nummer (unten detailliert beschrieben) zuzuordnen.

[0054] Um die außerordentlich hohe Datenrate dieses externen Busses, bezogen auf die Gate-Verzögerungen der internen Logik, zu unterstützen, werden die Bus-Zyklen in Paaren geradzahlig/ungeradzahlig Zyklen gruppiert. Es ist anzumerken, dass alle mit einem Bus verbundenen Geräte bevorzugt die gleiche geradzahlig/ungeradzahlig Bezeichnung von Bus-Zyklen verwenden sollen und bevorzugt mit Betrieb in geradzahlig Zyklen beginnen sollen. Dies wird durch das Takt-Schema erzwungen.

Protokoll und Bus-Betrieb

[0055] Die Busse verwenden für Bus-Transaktionen ein relativ einfaches, synchrones Block-orientiertes Protokoll für aufgeteilte Transaktionen. Eines der Ziele des Systems ist es, die Intelligenz in den Mastern zusammengefasst zu halten, um somit die Slaves so einfach wie möglich zu halten (da typisch viel mehr Slaves als Master vorhanden sind). Um die Komplexität der Slaves zu verringern, soll ein Slave bevorzugt auf eine Anforderung in einer festgelegten Zeit antworten, die ausreichend ist, um dem Slave zu gestatten, eine Geräte-interne Phase einschließlich einer internen Aktion zu beginnen oder möglicherweise zu beenden, welche der nachfolgenden Bus-Zugriffsphase vorausgehen muss. Die Zeit für diese Bus-Zugriffsphase ist allen Geräten auf dem Bus bekannt – jeder Master ist dafür verantwortlich, sicherzustellen, dass der Bus frei ist, wenn der Bus-Zugriff beginnt. Somit müssen die Slaves niemals befürchten, um den Bus zu konkurrieren. Dieser Ansatz beseitigt die Konkurrenz in einem Einzel-Master-System und macht ebenfalls die Slave-Bus-Schnittstelle einfacher.

[0056] In einer bevorzugten Implementierung sendet ein Master zum Initialisieren einer Bus-Übertragung über den Bus ein Anforderungspaket, eine aufeinanderfolgende Reihe von Bytes mit Adressen und Steuerinformationen. Es wird bevorzugt, ein Anforderungspaket mit einer geradzahlig Anzahl von Bytes zu verwenden und ebenfalls bevorzugt, jedes Paket in einem geradzahlig Bus-Zyklus zu beginnen.

[0057] Eine Geräteauswahl-Funktion wird unter Verwendung der Bus-Datenleitungen gehandhabt. AddrValid wird angesteuert, welches alle Slaves anweist, die Anforderungspaketadresse zu dekodieren, zu bestimmen, ob sie die angeforderte Adresse enthalten, und wenn dies der Fall ist, in einer Datenblockübertragung die Daten für den Master (in dem Fall einer Leseanforderung) bereitzustellen oder Daten von dem Master entgegenzunehmen (in dem Fall einer Schreibanforderung). Ein Master kann ebenfalls ein bestimmtes Gerät durch Senden einer Geräte-ID-Nummer in einem Anforderungspaket auswählen. In einer bevorzugten Implementierung wird eine bestimmte Geräte-ID-Nummer gewählt, um anzuzeigen, dass das Paket von allen Geräten auf dem Bus interpretiert werden soll. Dieses erlaubt einem Master, eine Mitteilung zu übermitteln, z.B. zum Setzen eines ausgewählten Steuerungsregisters sämtlicher Geräte mit dem gleichen Wert.

[0058] Die Datenblockübertragung tritt zu einer in der Anforderungspaket-Steuerinformation festgelegten

späteren Zeit auf, bevorzugt beginnend in einem geradzahligem Zyklus. Ein Gerät beginnt eine Datenblockübertragung fast sofort mit einer Geräte-internen Phase, wenn das Gerät bestimmte Funktionen initialisiert, wie Einstellen der Speicheradressierung, bevor die Bus-Zugriffsphase beginnt. Die Zeit, nach welcher ein Datenblock an die Bus-Leitungen angelegt wird, wird aus den in den Slave-Zugriffszeit-Registern gespeicherten Werten ausgewählt. Das Timing von Daten zum Lesen und Schreiben ist bevorzugt das gleiche; der einzige Unterschied ist, welches Gerät den Bus ansteuert. Zum Lesen steuert der Slave den Bus an, und der Master speichert die Werte von dem Bus zwischen. Zum Schreiben steuert der Master den Bus an, und der ausgewählte Slave speichert die Werte von dem Bus zwischen.

[0059] In einer in [Fig. 4](#) gezeigten, bevorzugten Implementierung enthält ein Anforderungspaket **22** 6 Datenbytes -- 4,5 Adressbytes und 1,5 Steuerungsbytes. Jedes Anforderungspaket verwendet alle neun Bits der gemultiplexten Daten/Adress-Leitungen (AddrValid **23** + BusData[0:7] **24**) für alle sechs Bytes des Anforderungspaketes. Eine Einstellung **23** AddrValid = 1 in einem sonst nicht verwendeten, geradzahligem Zyklus zeigt den Beginn eines Anforderungspaketes (Steuerinformation) an. In einem gültigen Anforderungspaket muss AddrValid **27** in dem letzten Byte 0 sein. Ein Festlegen dieses Signals in dem letzten Byte macht das Anforderungspaket ungültig. Dieses wird für die Kollisionserfassung und Arbitrierungslogik (unten beschrieben) verwendet. Die Bytes 25–26 enthalten die ersten 35 Adress-Bits, Adresse [0:35]. Das letzte Byte enthält AddrValid **27** (der Ungültigkeits-Schalter) und **28**, die verbleibenden Adress-Bits, Adresse[36:39] und Blocksize[0:3] (Steuerinformation).

[0060] Das erste Byte enthält zwei 4-Bit-Felder mit Steuerinformationen, Access-Type[0:3], einen Op-Code (Operations-Code), welcher zum Beispiel die Art des Zugriffs festlegt, und Master[0:3], eine für den das Paket sendenden Master reservierte Position zum Enthalten seiner Master-ID-Nummer. Nur die Master-Nummern 1 bis 15 sind zulässig – Master-Nummer 0 ist für besondere Systembefehle reserviert. Jedes Paket mit Master[0:3] = 0 ist ein ungültiges oder besonderes Paket und wird entsprechend behandelt.

[0061] Das AccessType-Feld gibt an, ob der angeforderte Vorgang Lesen oder Schreiben ist, und die Art des Zugriffs, zum Beispiel, ob er auf die Steuerungsregister oder Teile des Gerätes, wie Speicher, erfolgt. In einer vorbestimmten Implementierung ist AccessType[0] ein Lese/Schreib-Schalter: wenn er 1 ist, ruft der Vorgang ein Lesen von dem Slave auf (den Slave zum Lesen des angeforderten Speicherblockes und Anlegen des Speicherinhalts an den Bus); wenn er eine 0 ist, ruft der Ablauf ein Schreiben in den Slave auf (den Slave zum Lesen von Daten von dem Bus und Schreiben in den Speicher). AccessType[1:3] stellt bis zu 8 unterschiedliche Zugriffstypen für einen Slave bereit. AccessType[1:2] bezeichnet bevorzugt das Timing der Antwort, welche in einem Zugriffszeit-Register, Access-RegN, gespeichert wird. Die Auswahl des Zugriffszeit-Registers kann durch einen bestimmten Op-Code direkt gewählt werden, der das Register auswählt, oder indirekt durch eine Slave-Antwort auf ausgewählte Op-Codes mit vorausgewählten Zugriffszeiten (siehe Tabelle unten). Das verbleibende Bit AccessType[3] kann verwendet werden, um zusätzliche Informationen über die Anforderung zu den Slaves zu senden.

[0062] Eine besondere Art von Zugriff ist der Steuerungsregister-Zugriff, welcher eine Adressierung eines ausgewählten Registers in einem ausgewählten Slave beinhaltet. In der bevorzugten Implementierung gibt AccessType [1:3] gleich Null eine Steuerungsregister-Anforderung an, und das Adressfeld des Paketes bezeichnet das gewünschte Steuerungsregister. Die zwei signifikantesten Bytes können zum Beispiel die Geräte-ID-Nummer sein (die angeben, welcher Slave angesprochen wird), und die drei am wenigsten signifikanten Bytes können eine Registeradresse spezifizieren und können ebenfalls in das Steuerungsregister zu ladende Daten darstellen oder beinhalten. Steuerungsregister-Zugriffe werden verwendet, um die Zugriffszeit-Register zu initialisieren, und somit wird es bevorzugt, eine feste Antwortzeit zu verwenden, welche vorprogrammierbar oder auch fest verdrahtbar ist, zum Beispiel der Wert in AccessRegO, bevorzugt 8 Zyklen. Ein Steuerungsregister-Zugriff kann ebenfalls verwendet werden, um andere Register einschließlich Adress-Registern zu initialisieren oder modifizieren.

[0063] Das Verfahren gemäß dieser Ausführungsform stellt eine Zugriffsmodussteuerung insbesondere für die DRAMs bereit. Ein solcher Zugriffsmodus bestimmt, ob der Zugriff ein Seitenmodus oder ein normaler RAS-Zugriff ist. Im normalen Modus (bei konventionellen DRAMs und gemäß dieser Erfindung) wurden die DRAM-Spalten-Leseverstärker oder Zwischenspeicher auf einen Wert zwischen logisch 0 und 1 voraufgeladen. Dieses Voraufladen erlaubt einen Zugriff auf eine Zeile in dem RAM, zu beginnen, sobald die Zugriffsanforderung entweder für Eingaben (Schreiben) oder Ausgaben (Lesen) empfangen ist und erlaubt den Spalten-Leseverstärkern, die Daten schnell zu erfassen. In dem Seitenmodus (sowohl konventionell als auch gemäß dieser Erfindung) hält der DRAM die Daten in den Spalten-Leseverstärkern oder Zwischenspeichern von dem vorherigen Lese- oder Schreib-Vorgang. Wenn eine nachfolgende Anforderung auf Zugriffsdaten an die

gleiche Zeile gerichtet ist, muss das DRAM nicht auf die zu erfassenden Daten warten (sie wurden bereits erfasst), und die Zugriffszeit für diese Daten ist deutlich kürzer als die normale Zugriffszeit. Der Seitenmodus gestattet allgemein einen erheblich schnelleren Zugriff auf Daten, aber auf einen kleineren Datenblock (gleich der Anzahl der Leseverstärker). Wenn die angeforderten Daten jedoch nicht in der ausgewählten Zeile sind, ist die Zugriffszeit länger als die normale Zugriffszeit, da die Anforderung warten muss, bis der RAM voraufgeladen ist, bevor der Normalmodus-Zugriff beginnen kann. Zwei Zugriffszeit-Register in jedem DRAM enthalten bevorzugt die für die Normal- und Seiten-Modus-Zugriffe verwendeten Zugriffszeiten.

[0064] Der Zugriffsmodus bestimmt ebenfalls, ob der DRAM die Leseverstärker voraufladen oder die Inhalte der Leseverstärker für einen nachfolgenden Seitenmodus-Zugriff speichern soll. Typische Einstellungen sind "Voraufladen nach normalem Zugriff" und "Speichern nach Seitenmodus-Zugriff", aber "Voraufladen nach Seitenmodus-Zugriff" oder "Speichern nach Normalzugriff" sind erlaubte, auswählbare Betriebsmodi. Der DRAM kann ebenfalls eingestellt werden, um die Leseverstärker voraufzuladen, wenn sie für einen ausgewählten Zeitabschnitt nicht angesprochen werden.

[0065] Im Seitenmodus können in den DRAM-Leseverstärkern gespeicherte Daten innerhalb deutlich kürzerer Zeit angesprochen werden als es erfordert, Daten im Normalmodus auszulesen (10–20 ns gegenüber 40–100 ns). Diese Daten können für lange Zeitabschnitte verfügbar gehalten werden. Wenn diese Leseverstärker (und damit Bit-Leitungen) jedoch nach einem Zugriff nicht voraufgeladen werden, leidet ein nachfolgender Zugriff auf ein anderes Speicherwort (Zeile) unter einer Voraufladezeit-Verzögerung von etwa 40–100 ns, da die Leseverstärker vor dem Zwischenspeichern eines neuen Wertes aufgeladen werden müssen.

[0066] Die Inhalte der Leseverstärker können somit gehalten und als ein Cache genutzt werden, und erlauben einen schnelleren, wiederholten Zugriff auf kleine Datenblöcke. DRAM-basierte Seitenmodus-Caches wurden im Stand der Technik unter Verwendung konventioneller DRAM-Organisationen versucht, aber sie sind nicht sehr wirksam, da mehrere Chips pro Computerwort benötigt werden. Solch ein konventioneller Seitenmodus-Cache enthält viele Bits (zum Beispiel 32 Chips \times 4kBit), hat aber nur sehr wenige unabhängige Speichereinträge. Mit anderen Worten halten die Leseverstärker zu jedem gegebenen Zeitpunkt nur einige unterschiedliche Blöcke oder Speicher-"Positionen" (einen einzelnen Block aus 4k Wörtern in dem obigen Beispiel). Simulationen haben gezeigt, dass oberhalb von 100 Blöcken hohe Trefferhäufigkeiten verwirklicht werden müssen (> 90% der Anforderungen finden die angeforderten Daten bereits in dem Cache-Speicher), ungeachtet der Größe jedes Blockes. Siehe zum Beispiel Anant Agarwal, et al., "An Analytic Cache Model", ACM Transactions on Computer Systems, Band 7(2), Seiten 184–215 (Mai 1989).

[0067] Die Organisation von Speicher gemäß dem vorliegenden Ausführungsbeispiel erlaubt jedem DRAM, einen oder mehrere (4 für 4MBit DRAMs) separat adressierte und unabhängige Datenblöcke zu halten. Ein Personal Computer oder ein Arbeitsplatz mit 100 solcher DRAMs (d.h. 400 Blöcken oder Positionen) kann außerordentlich hohe, leicht wiederholbare Trefferraten (98–99% im Durchschnitt) verwirklichen, verglichen mit niedrigeren (50–80%) breit variierenden Trefferraten unter Verwendung von in konventioneller Weise organisierten DRAMs. Infolge des der fehlenden Voraufladung zugeordneten Zeitverlustes bei einem "Fehlen" des Seitenmodus-Cache wurde weiterhin der konventionelle DRAM-basierte Seitenmodus-Cache allgemein als weniger gut arbeitend als gar kein Cache gefunden.

[0068] Für DRAM-Slave-Zugriff werden die Zugriffstypen bevorzugt auf folgende Weise verwendet:

<u>AccessType[1:3]</u>	<u>Verwendung</u>	<u>AccessTime</u>
0	Steuerungsregisterzugriff	Fest, 8[AccessReg0]
1	Frei	Fest, 8[AccessReg0]
2-3	Frei	AccessReg1
4-5	Seitenmodus-DRAM-Zugriff	AccessReg2
6-7	Normaler DRAM-Zugriff	AccessReg3

[0069] Durchschnittsfachleute erkennen, dass eine Reihe verfügbarer Bits als Schalter zum Steuern dieser Zugriffsmodi vorgesehen sein kann. Zum Beispiel:

AccessType[2] = Seitenmodus/Normal-Schalter

AccessType[3] = Vorauflade/Datensicherungs-Schalter

[0070] BlockSize[0:3] gibt die Größe der Datenblockübertragung an. Wenn BlockSize[0] 0 ist, sind die verblei-

benden Bits die binäre Darstellung der Blockgröße (0-7). Wenn BlockSize[0] 1 ist, geben die verbleibenden Bits die Blockgröße als eine binäre Potenz von 2 von 8 bis 1024 an. Ein Block der Länge Null kann als ein besonderer Befehl interpretiert werden, zum Beispiel zum Auffrischen eines DRAM ohne Daten zurückzugeben, oder zum Wechseln des DRAM von dem Seitenmodus in einen normalen Zugriffsmodus oder umgekehrt.

BlockSize[0:2]	Anzahl von Bytes im Block
0-7	0-7 entsprechend
8	8
9	16
10	32
11	64
12	128
13	256
14	512
15	1024

[0071] Durchschnittsfachleute erkennen, dass andere Blockgrößen-Kodierungsschemata oder Werte verwendbar sind.

[0072] In den meisten Fällen antwortet ein Slave zu dem ausgewählten Zugriffszeitpunkt durch Lesen oder Schreiben von Daten von oder auf den Bus über die Bus-Leitungen BusData[0:7] und AddrValid wird logisch 0. In einer bevorzugten Ausführungsform betrifft im wesentlichen jeder Speicherzugriff noch ein einzelnes Speichergerät, d.h., ein einzelner Block wird gelesen aus einem einzelnen oder geschrieben in ein einzelnes Speichergerät.

Wiederholungs-Format

[0073] In einigen Fällen kann ein Slave außer Stande sein, auf eine Anforderung korrekt zu antworten, z.B. für ein Lesen oder Schreiben. In solch einer Situation soll der Slave eine Fehlermitteilung zurückgeben, die manchmal bezeichnet wird als N(o)ACK(nowledge) oder Wiederholungs-Mitteilung. Die Wiederholungs-Mitteilung kann Informationen über die eine Wiederholung erfordernden Bedingungen beinhalten, aber dies erhöht die Systemanforderungen für Schaltungen in Slave und Master. Eine einfache Mitteilung, die nur anzeigt, dass ein Fehler aufgetreten ist, gestattet einen weniger komplexen Slave und der Master kann eine beliebige Aktion unternehmen, die benötigt wird, um den Auslöser des Fehlers zu verstehen und zu korrigieren.

[0074] Zum Beispiel kann ein Slave unter bestimmten Bedingungen nicht in der Lage sein, die angeforderten Daten zu liefern. Während eines Seitenmodus-Zugriffs muss der ausgewählte DRAM im Seitenmodus sein und die angeforderte Adresse muss mit der Adresse der in den Leseverstärkern oder Zwischenspeichern gehaltenen Daten übereinstimmen. Jeder DRAM kann diese Übereinstimmung während eines Seitenmodus-Zugriffs prüfen. Wenn keine Übereinstimmung festgestellt wird, beginnt der DRAM mit dem Voraufladen und gibt eine Wiederholungs-Mitteilung zu dem Master während des ersten Zyklus' des Datenblockes zurück (der Rest des zurückgegebenen Blockes wird ignoriert). Der Master muss dann die Voraufladezeit abwarten (welche eingestellt ist, um dem Typ des in Frage stehenden Slave zu entsprechen, gespeichert in einem besonderen Register, Pre-ChargeReg), und dann die Anforderung als einen normalen DRAM-Zugriff erneut senden (AccessType = 6 oder 7).

[0075] Gemäß einem bevorzugten Ausführungsbeispiel signalisiert ein Slave eine Wiederholung durch Anlegen von AddrValid als wahr, zu dem Zeitpunkt, zu welchem dem Slave vorgegeben wird, das Lesen oder Schreiben von Daten zu beginnen. Ein Master, welcher zu dem Slave schreiben will, muss AddrValid während des Schreibens überwachen und eine Korrekturaktion ausführen, wenn er eine Wiederholungs-Mitteilung erfasst. [Fig. 5](#) zeigt das Format einer Wiederholungs-Mitteilung **28**, welches für Lese-Anforderungen nützlich ist, bestehend aus 23 AddrValid = 1 mit Master[0:3] = 0 in dem ersten (geradzahligen) Zyklus. Es ist anzumerken, dass AddrValid für Datenblockübertragungen normalerweise 0 ist, und dass kein Master 0 (nur 1 bis 15 sind zugelassen) vorhanden ist. Alle DRAMs und Master können ein solches Paket als ein ungültiges angefordertes Paket leicht erkennen, und daher eine Wiederholungs-Mitteilung. Bei dieser Art von Bus-Transaktion können sämtliche Felder mit Ausnahme des Masters[0:3] und AddrValid **23** als Informationsfelder verwendet werden, obwohl bei der Implementierung beschrieben ist, dass die Inhalte unbestimmt sind. Durchschnittsfachleute erkennen, dass ein weiteres Verfahren zum Zuweisen einer Wiederholungs-Mitteilung ist, eine DataInvalid-Leitung und ein Signal zu dem Bus hinzuzufügen. Dieses Signal kann im Falle von NACK gesetzt werden.

Bus-Arbitrierung

[0076] In dem Fall eines einzelnen Masters ergeben sich per Definition keine Arbitrierungsprobleme. Der Master sendet Anforderungspakete und verfolgt die Perioden, wenn der Bus als Antwort auf dieses Paket belegt ist. Der Master kann mehrere Anforderungen einplanen, so dass sich die entsprechenden Datenblock-Übertragungen nicht überlappen.

[0077] Die Bus-Architektur gemäß diesem Ausführungsbeispiel ist ebenfalls nützlich bei Konfigurationen mit mehreren Mastern. Wenn zwei oder mehr Master auf dem gleichen Bus vorhanden sind, muss jeder Master alle anhängigen Transaktionen verfolgen, und daher weiß jeder Master, wann er ein Anforderungspaket senden und auf die entsprechende Datenblock-Übertragung zugreifen kann. Es ergeben sich jedoch Situationen, in welchen zwei oder mehr Master ein Anforderungspaket zu etwa der gleichen Zeit senden und eine Mehrzahl von Anforderungen erfasst werden muss, welche dann durch eine Art von Bus-Arbitrierung sortiert werden.

[0078] Es gibt viele Wege für jeden Master, zu verfolgen, wann der Bus belegt ist und sein wird. Ein einfaches Verfahren ist für jeden Master, eine Bus-Belegt-Datenstruktur zu unterstützen, zum Beispiel durch Unterstützen von zwei Zeigern, einer zum Anzeigen des frühesten Punktes in der Zukunft, wenn der Bus belegt sein wird, und der andere zum Anzeigen des frühesten Punktes in der Zukunft, wann der Bus frei sein wird, d.h. das Ende der spätesten, anhängigen Datenblockübertragung. Unter Verwendung dieser Information kann jeder Master bestimmen, ob und wann ausreichend Zeit vorhanden ist, um ein Anforderungspaket (wie oben unter Protokoll beschrieben) zu senden, bevor der Bus durch eine andere Datenblockübertragung belegt wird, und ob die entsprechende Datenblockübertragung eine Wechselwirkung mit anhängigen Bus-Transaktionen bewirken wird. Somit muss jeder Master jedes Anforderungspaket lesen und seine Bus-Belegt-Datenstruktur aktualisieren, um Informationen darüber beizubehalten, wann der Bus frei ist und sein wird.

[0079] Bei zwei oder mehr Mastern auf dem Bus werden Master gelegentlich unabhängige Anforderungspakete während des gleichen Bus-Zyklus' senden. Diese Mehrfach-Anforderungen kollidieren, da jeder dieser Master den Bus gleichzeitig mit unterschiedlichen Informationen ansteuert, resultierend in vermengten Anforderungsinformationen und keiner gewünschten Datenblockübertragung. In einer bevorzugten Ausführungsform steuert jedes Gerät auf dem Bus, das versucht, eine logische 1 auf einer BusData- oder AddrValid-Leitung zu schreiben, diese Leitung mit einem Strom an, der ausreichend ist, um eine Spannung größer oder gleich dem High-Logikwert für das System zu unterstützen. Geräte steuern nicht Leitungen an, die eine logische 0 aufweisen; diese Leitungen werden einfach bei einer Spannung entsprechend einem Low-Logikwert gehalten. Jeder Master prüft die Spannung auf wenigstens einigen, bevorzugt allen BusData- und AddrValid-Leitungen, so dass der Master eine logische "1" erfassen kann, wo der erwartete Pegel auf einer Leitung "0" ist, die er während eines gegebenen Bus-Zyklus' nicht ansteuert, aber ein anderer Master ansteuert.

[0080] Ein weiterer Weg zum Erfassen von Kollisionen ist es, eine oder mehrere Bus-Leitungen zur Kollisionsanzeige auszuwählen. Jeder eine Anforderung sendende Master steuert die Leitung oder Leitungen an und überwacht die ausgewählten Leitungen auf mehr als den normalen Ansteuerungsstrom (oder einen logischen Wert von ">1"), der Anforderungen von mehr als einem Master angibt. Durchschnittsfachleute erkennen, dass dies mit einem Protokoll implementierbar ist, das BusData- und AddrValid-Leitungen betrifft, oder unter Verwendung einer zusätzlichen Bus-Leitung implementierbar ist.

[0081] In der bevorzugten Ausführungsform erfasst jeder Master Kollisionen durch Überwachen von Leitungen, welche er nicht ansteuert, um zu erkennen, ob ein anderer Master diese Leitungen ansteuert. In [Fig. 4](#) beinhaltet das erste Byte des Anforderungspaketes die Nummer jedes Masters, der versucht, den Bus zu nutzen (Master[0:3]). Wenn zwei Master zum gleichen Zeitpunkt beginnend Paketanforderungen senden, werden die Master-Nummern logisch "oder"- zusammen mit wenigstens diesen Mastern verknüpft, und somit kann einer oder beide Master durch Überwachen der Daten auf dem Bus und Vergleichen mit dem, was er sendet, eine Kollision erfassen. Wenn zum Beispiel Anforderungen durch die Master Nummer 2 (0010) und 5 (0101) kollidieren, wird der Bus mit dem Wert Master[0:3] = 7 (0010 + 0101 = 0111) angesteuert. Der Master Nummer 5 erfasst, dass das Signal Master[2] = 1 ist und Master 2 erfasst, dass Master[1] und Master[3] = 1 sind, beiden Mastern mitteilend, dass eine Kollision aufgetreten ist. Ein weiteres Beispiel ist Master 2 und 11, für welche der Bus angesteuert wird mit dem Wert Master [0:3] = 11 (0010 + 1011 = 1011), und obwohl Master 11 diese Kollision nicht leicht erfassen kann, kann dies Master 2. Wenn eine Kollision erfasst wird, steuert jeder eine Kollision erfassende Master den Wert von AddrValid **27** in Byte 5 des Anforderungspaketes **22** auf 1, welches von allen Mastern erfasst wird, einschließlich Master 11 in dem zweiten Beispiel oben, und erzwingt einen Bus-Arbitrierungszyklus, wie unten beschrieben.

[0082] Eine weitere Kollisionsbedingung kann entstehen, wenn Master A ein Anforderungspaket in Zyklus 0 sendet und Master B versucht, ein Anforderungspaket beginnend in Zyklus 2 des ersten Anforderungspaketes zu senden, und überlappt dabei das erste Anforderungspaket. Dies tritt von Zeit zu Zeit auf, da der Bus mit hohen Geschwindigkeiten arbeitet, und somit kann die Logik in einem zweitauslösenden Master nicht schnell genug sein, um eine durch einen ersten Master in Zyklus 0 ausgelöste Anforderung zu erfassen und schnell genug durch Verzögern seiner eigenen Anforderung zu reagieren. Der Master B teilt schließlich mit, dass er nicht versucht hat, ein Anforderungspaket zu senden (und demnach fast sicher die Adresse, die Master A versucht zu senden, zerstört), und, wie in dem Beispiel einer gleichzeitigen Kollision oben eine 1 auf AddrValid während Byte 5 des ersten Anforderungspaketes **27** ansteuert, um eine Arbitrierung zu erzwingen. Die Logik in der bevorzugten Implementierung ist schnell genug, dass ein Master ein Anforderungspaket durch einen anderen Master durch Zyklus 3 des ersten Anforderungspaketes erfasst, so dass wahrscheinlich kein Master versucht, ein potentiell kollidierendes Anforderungspaket später als Zyklus 2 zu senden.

[0083] Slave-Geräte müssen eine Kollision nicht direkt erfassen, aber sie müssen warten, etwas nicht Wiederherstellbares zu tun, bis das letzte Byte (Byte 5) gelesen ist, um sicherzustellen, dass das Paket gültig ist. Ein Anforderungspaket mit Master[0:3] gleich 0 (ein Wiederholungssignal) wird ignoriert und bewirkt nicht eine Kollision. Die nachfolgenden Bytes eines solchen Paketes werden ignoriert.

[0084] Um eine Arbitrierung nach einer Kollision zu beginnen, warten die Master eine vorausgewählte Anzahl von Zyklen nach dem abgebrochenen Anforderungspaket (4 Zyklen in einer bevorzugten Implementierung) und verwenden dann den nächsten freien Zyklus, um den Bus zu unterscheiden (um über den Bus zu entscheiden) (den nächsten verfügbaren geradzahligen Zyklus in der bevorzugten Implementierung). Jeder kollidierende Master signalisiert allen anderen kollidierenden Mastern, dass er versucht, ein Anforderungspaket zu senden, eine Priorität wird jedem der kollidierenden Master zugeordnet und dann wird jedem Master erlaubt, seine Anforderung in der Reihenfolge der Priorität auszuführen.

[0085] [Fig. 6](#) stellt eine bevorzugte Weise der Implementierung dieser Arbitrierung dar. Jeder kollidierende Master signalisiert seine Absicht, ein Anforderungspaket zu senden, durch Ansteuern einer einzelnen BusData-Leitung während eines einzelnen Bus-Zyklus' entsprechend seiner zugeordneten Master-Nummer (1–15 in dem vorliegenden Beispiel). Während eines Zwei-Byte-Arbitrierungszyklus' **29** wird Byte 0 den Anforderungen 1–7 von den Mastern 1–7 zugeordnet (Bit 0 wird nicht verwendet), und Byte 1 wird den Anforderungen 8–15 von den Mastern 8–15 zugeordnet. Wenigstens ein Gerät und bevorzugt jeder kollidierende Master liest die Werte auf dem Bus während der Arbitrierungszyklen zum Bestimmen und Speichern, welcher Master den Bus nutzen möchte. Durchschnittsfachleute erkennen, dass ein einzelnes Byte für Arbitrierungsanforderungen belegt werden können, wenn das System mehr Bus-Leitungen als Master beinhaltet. Mehr als 15 Master können durch Belegung zusätzlicher Bus-Zyklen aufgenommen werden.

[0086] Ein festes Prioritätsschema (welches bevorzugt die Master-Nummern verwendet und die niedrigsten Nummern zuerst auswählt) wird dann verwendet, um zu priorisieren, dann die Anforderungen in einer Bus-Arbitrierungs-Warteschlange zu sequenzieren, welche durch wenigstens ein Gerät unterstützt wird. Diese Anforderungen werden durch jeden Master in der Bus-belegt-Datenstruktur in der Warteschlange eingereicht und es werden keine weiteren Anforderungen zugelassen, bis die Bus-Arbitrierungs-Warteschlange gelöscht ist. Durchschnittsfachleute erkennen, dass andere Prioritäts-Schemata verwendbar sind, einschließlich der Zuordnung einer Priorität entsprechend der physikalischen Position jedes Masters.

System-Konfiguration/Zurücksetzen

[0087] In dem Bus-basierten System gemäß diesem Ausführungsbeispiel ist ein Mechanismus vorgesehen, um jedem Gerät auf dem Bus einen eindeutigen Geräte-Identifizierer (Geräte-ID) nach dem Einschalten oder unter anderen Bedingungen, wie durch das System gewünscht oder benötigt, zu geben. Ein Master kann dann diese Geräte-ID verwenden, um ein bestimmtes Gerät anzusprechen, insbesondere, um Register des bestimmten Gerätes zu setzen oder zu modifizieren, einschließlich der Steuerungs- und Adress-Register. In der bevorzugten Ausführungsform ist ein Master zugewiesen, um den gesamten System-Konfigurationsvorgang auszuführen. Der Master stellt eine Reihe von eindeutigen Geräte-ID-Nummern für jedes an das Bus-System angeschlossene einzelne Gerät bereit. In der bevorzugten Ausführungsform enthält jedes an den Bus angeschlossene Gerät ein besonderes Gerätetyp-Register, welches den Typ des Gerätes angibt, zum Beispiel CPU, 4 MBit Speicher, 64 MBit Speicher oder Disk-Controller. Der Konfigurationsmaster soll jedes Gerät prüfen, den Gerätetyp bestimmen und geeignete Steuerungsregister setzen, einschließlich der Zugriffszeit-Register. Der Konfigurationsmaster soll jedes Speichergerät prüfen und alle geeigneten Speicher-Adress-Register setzen.

[0088] Ein Mittel zum Setzen eindeutiger Geräte-ID-Nummern ist, jedes Gerät eine Geräte-ID der Reihe nach auswählen zu lassen und den Wert in einem internen Geräte-ID-Register zu speichern. Ein Master kann zum Beispiel sequentielle Geräte-ID-Nummern durch Schieberegister in jedem einer Reihe von Geräten leiten, oder einen Token von Gerät zu Gerät reichen, wodurch das Gerät mit dem Token Geräte-ID-Informationen von einer weiteren Leitung oder Leitungen einliest. In einer bevorzugten Ausführungsform werden Geräte-ID-Nummern Geräten entsprechend ihrer physikalischen Beziehung, zum Beispiel ihrer Reihenfolge entlang des Busses, zugeordnet.

[0089] In einer bevorzugten Ausführungsform wird die Geräte-ID-Einstellung unter Verwendung von zwei Pins an jedem Gerät, ResetIn und ResetOut, verwirklicht. Diese Pins handhaben normale Logiksignale und werden nur während der Geräte-ID-Konfiguration verwendet. An jeder ansteigenden Flanke des Taktes kopiert jedes Gerät ResetIn (ein Eingang) in ein vierstufiges Rücksetz-Schieberegister. Der Ausgang des Rücksetz-Schieberegisters ist mit ResetOut verbunden, welcher wiederum mit ResetIn für das nächste der Reihe nach angeschlossene Gerät verbunden ist. Im Wesentlichen alle Geräte auf dem Bus sind dadurch miteinander verkettet. Ein erstes Rücksetzsignal bewirkt, während zum Beispiel ResetIn an einem Gerät logisch 1 ist, oder wenn ein ausgewähltes Bit des Rücksetz-Schieberegisters von Null zu nicht-Null übergeht, dass das Gerät hart zurückgesetzt wird, zum Beispiel durch Löschen aller internen Register und Zurücksetzen aller Zustands-Maschinen. Ein zweites Rücksetzsignal, zum Beispiel die abfallende Flanke von ResetIn, zusammengefasst mit veränderbaren Werten auf dem externen Bus, bewirkt, dass das Gerät die Inhalte des externen Busses in dem internen Geräte-ID-Register (Gerät[0:7]) zwischenspeichert.

[0090] Um alle Geräte auf einem Bus zurückzusetzen, setzt ein Master die ResetIn-Leitung des ersten Geräts ausreichend lange auf "1", um sicherzustellen, dass alle Geräte auf dem Bus zurückgesetzt wurden (4 Zyklen mal der Anzahl der Geräte -- es ist anzumerken, dass die maximale Anzahl von Geräten in der bevorzugten Bus-Konfiguration **256** ist (8 Bits), so dass 1024 Zyklen stets ausreichend Zeit zum Zurücksetzen sämtlicher Geräte darstellt). Dann fällt ResetIn auf "0" ab und die BusData-Leitungen werden mit der ersten, gefolgt durch die anschließenden Geräte-ID-Nummern angesteuert, die sich alle vier Taktimpulse ändert. Nachfolgende Geräte setzen diese Geräte-ID-Nummern in den entsprechenden Geräte-ID-Registern, wenn sich die abfallende Flanke von ResetIn durch die Schieberegister der verketteten Geräte ausbreitet. [Fig. 14](#) zeigt ResetIn an einem ersten Gerät auf "low" gehend, während ein Master eine erste Geräte-ID an die Bus-Datenleitungen BusData[0:3] anlegt. Das erste Gerät speichert dann diese erste Geräte-ID. Nach vier Taktzyklen ändert der Master BusData[0:3] zu der nächsten Geräte-ID-Nummer und ResetOut geht an dem ersten Gerät auf "low", welcher ResetIn für das nächste verkettete Gerät auf "low" zieht und dem nächsten Gerät erlaubt, die nächste Geräte-ID-Nummer von den Bus-Data[0:3] zu speichern. In der bevorzugten Ausführungsform ist ein Master der Geräte-ID 0 zugeordnet, und es ist die Verantwortlichkeit des Masters, die ResetIn-Leitung zu steuern und aufeinanderfolgende Geräte-ID-Nummern auf dem Bus zu geeigneten Zeiten anzulegen. In der bevorzugten Ausführungsform wartet jedes Gerät zwei Taktzyklen, nachdem ResetIn auf "low" geht, bevor es eine Geräte-ID-Nummer von den Bus-Data[0:3] speichert.

[0091] Durchschnittsfachleute erkennen, dass längere Geräte-ID-Nummern zu den Geräten verteilt werden können, indem jedes Gerät mehrere Bytes von dem Bus liest und die Werte in dem Geräte-ID-Register zwischenspeichert. Durchschnittsfachleute erkennen ebenfalls, dass alternative Wege zum Erhalten der Geräte-ID-Nummern für einzelne Geräte vorhanden sind. Zu einem bestimmten Zeitpunkt kann jedes Gerät angewiesen werden, den gegenwärtigen Rücksetz-Schieberegisterwert in dem Geräte-ID-Register zwischenspeichern.

[0092] Der Konfigurations-Master sollte eine Zugriffszeit in jedem Zugriffszeit-Register in jedem Slave für eine Periode wählen und setzen, die ausreichend lang ist; um zuzulassen, dass der Slave einen tatsächlichen, gewünschten Speicherzugriff ausführt. Für einen normalen DRAM-Zugriff muss diese Zeit zum Beispiel länger sein als die Zeilenadressimpuls-(RAS, row address stroke)-Zugriffszeit. Wenn die Bedingung nicht erfüllt wird, kann der Slave nicht die korrekten Daten liefern. Der in einem Slave-Zugriffszeit-Register gespeicherte Wert entspricht bevorzugt der Hälfte der Anzahl der Buszyklen, welche das Slave-Gerät warten soll, bevor es den Bus als Antwort auf eine Anfrage verwendet. Ein Zugriffszeitwert von "1" gibt somit an, dass der Slave nicht auf den Bus zugreifen sollte, bis wenigstens zwei Zyklen nach dem letzten Byte des Anforderungspaketes empfangen wurden. Der Wert von AccessReg0ist bevorzugt bei 8 (Zyklen) fixiert, um einen Zugriff auf Steuerungsregister zu unterstützen.

[0093] Die Bus-Architektur gemäß diesem Ausführungsbeispiel kann mehr als ein Master-Gerät beinhalten. Die Rücksetz- oder Initialisierungs-Sequenz sollte ebenfalls eine Bestimmung beinhalten, ob mehrere Master auf dem Bus vorhanden sind, und wenn dies der Fall ist, eindeutige Master-ID-Nummern jedem zuzuordnen.

Durchschnittsfachleute erkennen, dass es viele Wege gibt, dieses auszuführen. Der Master kann zum Beispiel jedes Gerät abfragen, um zu bestimmen, welche Art von Gerät es ist, zum Beispiel durch Lesen eines besonderen Registers und dann für jedes Master-Gerät die nächste verfügbare Master-ID-Nummer in ein besonderes Register zu schreiben.

ECC

[0094] Fehler-Erfassungs- und -Korrektur-("ECC")-Verfahren sind im Stand der Technik bekannt und können in diesem System implementiert werden. ECC-Informationen werden normalerweise für einen Datenblock zu dem Zeitpunkt errechnet, in dem der Datenblock zuerst in den Speicher geschrieben wird. Der Datenblock hat gewöhnlich eine integrale binäre Größe, z.B. 256 Bits, und die ECC-Information verwendet signifikant weniger Bits. Ein potentielles Problem ergibt sich dadurch, dass jeder binäre Datenblock in bekannten Schemata normalerweise mit angefügten ECC-Bits gespeichert wird, was zu einer Blockgröße führt, die nicht eine integrale binäre Potenz ist.

[0095] In einer bevorzugten Ausführungsform wird ECC-Information getrennt von den entsprechenden Daten gespeichert, welche dann in Blöcken mit integraler Binärgröße speicherbar sind. ECC-Informationen und entsprechende Daten können zum Beispiel in getrennten DRAM-Geräten gespeichert werden. Daten können ohne ECC unter Verwendung eines einzelnen Anforderungspaketes ausgelesen werden, aber zum Schreiben oder Lesen Fehler-korrigierter Daten werden zwei Anforderungspakete benötigt, eines für die Daten und ein zweites für die entsprechende ECC-Information. Die ECC-Information kann nicht immer permanent gespeichert werden, und in einigen Situationen kann die ECC-Information verfügbar gemacht werden, ohne ein Anforderungspaket zu senden oder ohne eine Busdaten-Blockübertragung.

[0096] In einer bevorzugten Ausführungsform kann eine Standard-Datenblockgröße zur Verwendung mit ECC ausgewählt werden, und das ECC-Verfahren bestimmt die benötigte Anzahl von Bits von Informationen in einem entsprechenden ECC-Block. RAMs mit ECC-Informationen können programmiert werden, um eine Zugriffszeit zu speichern, die gleich ist zu: (1) der Zugriffszeit des normalen RAM (mit Daten) plus der Zeit zum Zugriff auf einen Standard-Datenblock (für korrigierte Daten) minus der Zeit zum Senden eines Anforderungspaketes (6 Bytes); oder (2) der Zugriffszeit eines normalen RAM minus der Zeit zum Zugriff auf einen Standard-ECC-Block minus der Zeit zum Senden eines Anforderungspaketes. Um einen Datenblock und den entsprechenden ECC-Block zu lesen, gibt der Master einfach eine Anforderung für die Daten aus, unmittelbar gefolgt durch eine Anforderung für den ECC-Block. Der ECC-RAM wartet die ausgewählte Zugriffszeit und legt dann seine Daten an den Bus direkt nachdem (im Fall (1) oben) der Daten-RAM das Ausgeben des Datenblockes beendet hat. Durchschnittsfachleute erkennen, dass die in dem Fall (2) oben beschriebene Zugriffszeit verwendet werden kann, um ECC-Daten anzusteuern, bevor die Daten an die Busleitungen angelegt werden, und erkennt, dass ein Schreiben von Daten analog mit dem zum Lesen beschriebenen Verfahren ausgeführt werden kann. Durchschnittsfachleute erkennen ebenfalls die Anpassungen, die in der Bus-belegt-Struktur und den Anforderungs-Paket-Arbitrierungsverfahren gemäß diesem Ausführungsbeispiel vorgenommen werden müssen, um diese gepaarten ECC-Anforderungen aufzunehmen.

[0097] Da das System vollkommen flexibel ist, kann der System-Designer die Größe der Datenblöcke und die Anzahl der die Speichergeräte gemäß diesem Ausführungsbeispiel verwendenden ECC-Bits wählen. Es ist anzumerken, dass der Datenstrom auf dem Bus auf verschiedene Weise interpretierbar ist. Die Sequenz kann zum Beispiel 2^n Datenbytes, gefolgt von 2^m ECC-Bytes sein (oder umgekehrt), oder die Sequenz kann 2^k Iterationen von 8 Datenbytes plus 1 ECC Byte sein. Andere Informationen, wie durch ein Verzeichnis-basiertes Cache-Kohärenz-Schema verwendete Information kann ebenfalls auf diese Weise verwaltet werden. Siehe zum Beispiel Anant Agarwal et al., "Scaleable Directory Schemes for Cache Consistency", 15th International Symposium on Computer Architecture, Juni 1988, Seite 280–289. Durchschnittsfachleute erkennen alternative Verfahren der Implementierung von ECC-Schemata.

Low-Power-3D-Packaging

[0098] Ein weiteres Hauptvorteil einer bevorzugten Ausführungsform dieser Erfindung ist, dass sie den Speichersystem-Energieverbrauch deutlich verringert. Fast die gesamte in bekannten DRAM verbrauchte Energie wird bei der Ausführung eines Zeilenzugriffs verbraucht. Bei Verwendung eines Einzelzeilen-Zugriffs in einem einzelnen RAM zum Liefern sämtlicher Bits für eine Blockanfrage (verglichen mit einem Reihenzugriff in jedem der mehreren RAMs in konventionellen Speichersystemen) kann die Leistung pro Bit sehr klein gemacht werden. Da die von Speichergeräten gemäß diesem Ausführungsbeispiel verbrauchte Leistung signifikant verringert ist, können die Geräte potentiell sehr viel enger zueinander angeordnet werden, als bei konventionellen

Ausbildungen.

[0099] Die Bus-Architektur gemäß diesem Ausführungsbeispiel ermöglicht eine innovative 3D-Packaging-Technologie. Durch Verwenden eines schmalen, gemultiplexten (zeitgeteilten) Busses kann die Pin-Zählung für ein beliebig großes Speichergerät recht klein gehalten werden – in der Größenordnung von 20 Pins. Weiterhin kann diese Pin-Anzahl von einer Generation von DRAM-Dichte zu der nächsten konstant gehalten werden. Der geringe Leistungsverbrauch erlaubt, dass jedes Package kleiner wird, mit geringeren Pin-Abständen (Abstand zwischen den IC-Pins). Gegenwärtige Surface-Mount-Technologie unterstützt Pin-Abstände von nur 20 mils, alle äußeren Geräteanschlüsse können an einem einzelnen Rand des Speichergerätes implementiert sein. In dieser Ausführungsform nützliche Halbleiter-Dies haben bevorzugt Verbindungen oder Anschlussflächen entlang eines Randes des Dies, welche dann verdrahtet oder anders mit den Package-Pins mit Drähten gleicher Länge verbunden werden können. Diese Geometrie erlaubt ebenfalls sehr kurze Leiter, bevorzugt mit einer wirksamen Leiterlänge von weniger als 4 mm. Weiterhin verwendet diese Ausführungsform nur Bus-geführte Zwischenverbindungen, d.h., jede Anschlussfläche an jedem Gerät ist durch den Bus mit der entsprechenden Anschlussfläche jedes anderen Gerätes verbunden.

[0100] Die Verwendung einer geringen Pin-Anzahl und eines Rand-angeschlossenen Busses gestattet eine einfache 3D-Package, wodurch die Geräte gestapelt werden und der Bus entlang eines einzelnen Randes des Stapels angeschlossen ist. Die Tatsache, dass sämtliche Signale Bus-geführt sind, ist wichtig für die Implementierung einer einfachen 3D-Struktur. Ohne dies wird die Komplexität der "Backplane" zu schwierig, um sie kostengünstig mit der gegenwärtigen Technologie herzustellen. Die einzelnen Geräte in einem Stapel gemäß diesem Ausführungsbeispiel können sehr eng gepackt werden wegen der durch das gesamte Speichersystem verbrauchten, geringen Leistung, welche erlaubt, dass die Geräte Seite-an-Seite (Bumper to Bumper) oder Oberseite auf Unterseite gestapelt werden. Konventionelle Kunststoff-Spritzguss-Small-Outline (SO)-Packages können mit einem Abstand von etwa 2,5 mm (100 mils) verwendet werden, die ultimative Grenze ist die Geräte-Die-Dicke, welche eine Größenordnung kleiner ist, 0,2–0,5 mm unter Verwendung gegenwärtiger Wafer-Technologie.

Elektrische Bus-Beschreibung

[0101] Durch Verwendung von Geräten mit sehr geringem Leistungsverbrauch und enger physikalischer Packung kann der Bus sehr kurz ausgeführt werden, was wiederum kurze Ausbreitungszeiten und hohe Datengeschwindigkeiten erlaubt. Der Bus gemäß einem bevorzugten Ausführungsbeispiel besteht aus einem Satz Widerstandsabgeschlossener Sendeleitungen mit gesteuerter Impedanz, welcher mit einer Frequenz von bis zu 500 MHz (2 ns-Zyklen) arbeiten kann. Die Merkmale der Sendeleitungen werden stark beeinflusst durch die durch die DRAMs (oder andere Slaves) bewirkte Last, welche auf dem Bus angebracht sind. Diese Geräte fügen unerwünschte Kapazitäten zu den Leitungen hinzu, welche die Impedanz der Leitungen senken und die Sendegeschwindigkeit verringern. In der belasteten Umgebung ist die Bus-Impedanz etwa in dem Bereich von 25 Ohm und die Ausbreitungsgeschwindigkeit etwa $c/4$ (c = Lichtgeschwindigkeit) oder 7,5 cm/ns. Um mit einer 2 ns Datenfrequenz zu arbeiten, sollte die Übergangszeit auf dem Bus bevorzugt unter 1 ns gehalten werden, um 1 ns für die Einstellung und die Haltezeit des Eingangsempfängers (unten beschrieben) plus Takt-Begründung übrig zu lassen. Somit müssen die Bus-Leitungen relativ kurz gehalten werden, unter etwa 8 cm für maximale Leistungsfähigkeit. Systeme mit geringerer Leistungsfähigkeit können deutlich längere Leitungen aufweisen, z.B. kann ein 4 ns-Bus 24 cm-Leitungen aufweisen (3 ns Übergangszeit, 1 ns Einstell- und Haltezeit).

[0102] In der bevorzugten Ausführungsform verwendet der Bus Stromquellen-Treiber. Jeder Ausgang muss in der Lage sein, 50 mA aufzunehmen, welche einen Ausgangs-Hub von 500 mV oder mehr bereitstellen. In der bevorzugten Ausführungsform ist der Bus aktiv "low". Der unbestimmte Zustand (der High-Wert) wird bevorzugt als logisch Null betrachtet, und der bestimmte Wert (Low-Zustand) ist daher eine logische 1. Durchschnittsfachleute erkennen, dass das Verfahren gemäß dem vorliegenden Ausführungsbeispiel auch durch Verwendung der entgegengesetzten logischen Beziehung der Spannung implementierbar ist. Der Wert des unbestimmten Zustands wird durch die Spannung an den Abschlusswiderständen gesetzt und sollte hoch genug sein, um den Ausgängen zu erlauben, als Stromquellen zu wirken, während er so gering wie möglich ist, um den Leistungsverbrauch zu verringern. Diese Beschränkungen führen zu einer Abschluss-Spannung von etwa 2 V oberhalb von Masse in der bevorzugten Implementierung. Stromquellen-Treiber bewirken, dass die Ausgangsspannung proportional zu der Summe der den Bus steuernden Quellen ist.

[0103] In Fig. 7 können, obwohl kein stabiler Zustand vorhanden ist, in welchem zwei Geräte den Bus gleichzeitig ansteuern können, Bedingungen infolge einer Ausbreitungsverzögerung auf den Drähten entstehen, bei

welcher ein Gerät, A **41**, beginnen kann, seinen Teil auf dem Bus **44** anzusteuern, während der Bus noch durch ein anderes Gerät, B **42** (setzt bereits eine logische 1 auf dem Bus) angesteuert wird. In einem Strom-Treiber verwendenden System ist der Wert an den Punkten **44** und **45** logisch 1, wenn B **42** den Bus ansteuert (vor dem Zeitpunkt **46**). Wenn B **42** zum Zeitpunkt **46** abschaltet, genau dann, wenn A **41** anschaltet, bewirkt die zusätzliche Ansteuerung durch das Gerät A **41**, dass die Spannung an dem Ausgang **44** von A **41** kurz unter den normalen Wert abfällt. Die Spannung kehrt zu ihrem normalen Wert zum Zeitpunkt **47** zurück, wenn die Wirkung der Abschaltung des Gerätes B **42** entfällt. Die Spannung an dem Punkt **45** wird logisch 0, wenn das Gerät B **42** abgeschaltet wird, und fällt zum Zeitpunkt **47**, wenn die Wirkung des Einschaltens des Gerätes A **41** entfällt. Da die durch den Strom von dem Gerät A **41** angesteuerte, logische 1 ungeachtet des vorherigen Wertes auf dem Bus ausgebreitet wird, wird der Wert auf dem Bus garantiert nach einer Zeit der Übergangsverzögerung (t_r) eingestellt, d.h., die Zeit, die ein Signal benötigt, um sich von einem Ende des Busses zu dem anderen auszubreiten. Wenn ein Spannungstreiber verwendet wird (wie bei einer verdrahteten ECL-OR-Verknüpfung) verhindert eine logische 1 auf dem Bus (von dem vorher angesteuerten Gerät B **42**) den durch das Gerät A **41** ausgegebenen Übergang, der bei dem am weitesten entfernten Teil des Systems, z.B. dem Gerät **43**, entfällt, bis die Abschalt-Wellenform von dem Gerät B **42** das Gerät A **41** erreicht, plus eine Zeit der Übergangsverzögerung, und gibt eine Einstellzeit im ungünstigsten Fall des zweifachen der Zeit der Übergangsverzögerung.

Taktung

[0104] Exaktes Takten eines Hochgeschwindigkeitsbusses ohne Einführen eines Fehlers infolge von Ausbreitungsverzögerungen kann implementiert werden, indem dieses Gerät zwei Bus-Taktsignale überwacht und dann intern einen Gerätetakt ableitet, den wahren Systemtakt. Die Bus-Taktinformation kann auf einer oder zwei Leitungen gesendet werden, um einen Mechanismus für jedes Bus-geführte Gerät bereitzustellen, um einen internen Gerätetakt mit Null Verzerrung relativ zu allen anderen Gerätetakten zu erzeugen. In **Fig. 8** sendet in der bevorzugten Implementierung ein Bus-Taktgenerator **50** an einem Ende des Busses ein frühes Bus-Taktsignal in einer Richtung entlang des Busses aus, zum Beispiel auf Leitung **53** von links nach rechts, zu dem entfernten Ende des Busses. Das gleiche Taktsignal wird dann durch die in einer zweiten Leitung **54** gezeigte, direkte Verbindung geführt, und kehrt als ein spätes Bus-Taktsignal entlang des Busses von dem fernen Ende zu dem Ursprung zurück und verläuft von rechts nach links. Eine einzelne Bus-Taktleitung kann verwendet werden, wenn sie an dem entfernten Ende des Busses unabgeschlossen bleibt, um dem frühen Bus-Taktsignal zu erlauben, entlang der gleichen Leitung als ein spätes Bus-Taktsignal zurück reflektiert zu werden.

[0105] **Fig. 8b** zeigt, wie jedes Gerät **51**, **52** jedes der zwei Bus-Taktsignale zu einem unterschiedlichen Zeitpunkt empfängt (wegen der Ausbreitungsverzögerung entlang der Drähte), mit konstantem Mittelpunkt in der Zeit zwischen den zwei Bus-Takten entlang des Busses. Bei jedem Gerät **51**, **52** ist die ansteigende Flanke **55** von Takt1 **53** gefolgt von der ansteigenden Flanke **56** von Takt2 **54**. Ebenso folgt der abfallenden Flanke **57** von Takt1 **53** die abfallende Flanke **58** von Takt2 **54**. Diese Wellenform-Beziehung wird bei allen anderen Geräten entlang des Busses beobachtet. Geräte, die näher an dem Taktgenerator sind, haben einen größeren Abstand zwischen Takt1 und Takt2, verglichen mit weiter von dem Generator entfernten Geräten, wegen der für jeden Taktimpuls erforderlichen längeren Zeit, um den Bus entlang und zurück entlang Leitung **54** zu laufen, aber der zeitliche Mittelpunkt **59**, **60** zwischen den entsprechenden ansteigenden oder abfallenden Flanken ist fest, da für jedes gegebene Gerät die Länge jeder Taktleitung zwischen dem fernen Ende des Busses und dem Gerät gleich ist. Jedes Gerät muss zwei Bus-Takte abtasten und seinen eigenen inneren Gerätetakt am Mittelpunkt der zwei erzeugen.

[0106] Taktverteilungsprobleme können weiterhin verringert werden durch Verwenden eines Bus-Taktes und einer Gerätetaktfrequenz gleich der Buszyklus-Datenfrequenz, dividiert durch zwei, d.h. die Bus-Taktperiode ist das zweifache der Bus-Zyklusperiode. Somit verwendet ein 500 MHz-Bus bevorzugt eine 250 MHz-Taktfrequenz. Die Verringerung der Frequenz bietet zwei Vorteile. Zuerst weisen alle Signale auf dem Bus die gleichen Worst-Case-Datenfrequenzen auf – Daten können sich auf einem 500 MHz-Bus nur alle 2 ns ändern. Zweitens macht das Takten mit der halben Buszyklus-Datenfrequenz die Bezeichnung der ungeradzahigen und geradzahigen Bus-Zyklen einfach, zum Beispiel durch Bestimmen geradzahiger Zyklen als diejenigen, wenn der interne Gerätetakt 0 ist, und ungeradzahiger Zyklen, wenn der interne Gerätetakt 1 ist.

Mehrere Busse

[0107] Die oben beschriebene Beschränkung der Buslänge beschränkt die Gesamtanzahl von Geräten, die auf einem einzelnen Bus plazierbar sind. Bei der Verwendung von 2,5 mm Abstand zwischen Geräten kann

ein einzelner 8 cm-Bus 32 Geräte halten. Durchschnittsfachleute erkennen bestimmte Anwendungen der vorliegenden Ausführungsform, bei welcher die gesamte Datenfrequenz auf dem Bus adäquat ist, aber Speicher- oder Verarbeitungsanforderungen eine deutlich größere Anzahl von Geräten (viel mehr als 32) erfordern. Größere Systeme können unter Verwendung der Lehren dieses Ausführungsbeispiels leicht aufgebaut werden, indem eine oder mehrere als primäre Bus-Einheiten bezeichnete Speicher-Subsysteme verwendet werden, von denen jede aus zwei oder mehr Geräten besteht, normalerweise 32 oder nahe an dem durch die Bus-Gestaltungsanforderungen zugelassenen Maximum, angeschlossen an einen Transceiver.

[0108] Entsprechend [Fig. 9](#) kann jede primäre Buseinheit auf einer einzelnen Schaltkreisplatine **66** angebracht sein, die manchmal als Speicherstift bezeichnet wird. Jeder Transceiver **19** ist wiederum an einen Transceiver-Bus **65** angeschlossen, elektrisch oder in anderer Hinsicht vergleichbar oder identisch mit dem oben ausgiebig beschriebenen Primär-Bus **18**. In einer bevorzugten Implementierung sind alle Master auf dem Transceiver-Bus angeordnet, so dass keine Transceiver-Verzögerungen zwischen Mastern und sämtlichen Speichergeräten auf den Primär-Buseinheiten auftreten, so dass alle Speicherzugriffe eine äquivalente Transceiver-Verzögerung erfahren, aber Durchschnittsfachleute erkennen, wie Systeme zu implementieren sind, welche Master auf mehr als einer Buseinheit und Speichergeräte auf dem Transceiver-Bus ebenso wie auf primären Buseinheiten aufweisen. Allgemein kann jede Lehre dieses Ausführungsbeispiels, welche sich auf eine Speichervorrichtung bezieht, unter Verwendung eines Transceivers und einem oder mehreren Speichergeräten an einer angeschlossenen Primär-Buseinheit praktisch ausgeführt werden. Andere Geräte, welche generisch als periphere Geräte bezeichnet werden, einschließlich Disk-Controllern, Video-Controllern oder E/A-Geräte können ebenfalls entweder an den Transceiver-Bus oder eine Primär-Buseinheit angeschlossen sein, wie gewünscht. Durchschnittsfachleute erkennen, wie eine einzelne Primär-Buseinheit oder mehrere Primär-Buseinheiten, wie sie bei einem Transceiver-Bus in bestimmten System-Ausbildungen benötigt werden, verwendet werden.

[0109] Die Transceiver haben eine wirklich einfache Funktion. Sie erfassen angeforderte Pakete auf dem Transceiver-Bus und senden sie zu ihrer primären Buseinheit. Wenn die Anforderungspaketaufrufe zum Schreiben auf ein Gerät auf einer Primär-Buseinheit eines Transceivers sind, verfolgt dieser Transceiver die Zugriffszeit und die Blockgröße und leitet alle Daten von dem Transceiver-Bus während dieser Zeit zu der primären Buseinheit weiter. Die Transceiver überwachen ebenfalls ihre primäre Buseinheit, leiten alle Daten weiter, die dort für den Transceiver-Bus auftreten. Die Hochgeschwindigkeit der Busse bedeutet, dass die Transceiver parallel angeordnet werden müssen, und erfordert zusätzlich einen oder zwei Zyklen Verzögerung für Daten, um den Transceiver in jeder Richtung zu durchlaufen. In Mastern auf dem Transceiver-Bus gespeicherte Zugriffszeiten müssen erhöht werden, um die Transceiver-Verzögerung zu berücksichtigen, in Slaves auf einer primären Buseinheit gespeicherte Zugriffszeiten sollen nicht modifiziert werden.

[0110] Durchschnittsfachleute erkennen, dass ein weiter entwickelter Transceiver Sendungen zu und von primären Buseinheiten steuern kann. Eine zusätzliche Steuerungsleitung TrncvrRW kann zu allen Geräten auf dem Transceiver-Bus Busgeführt werden, und verwendet die Leitung in Verbindung mit der AddrValid-Leitung, um allen Geräten auf dem Transceiver-Bus anzuzeigen, dass die Information auf den Datenleitungen: 1) ein Anforderungspaket, 2) gültige Daten für einen Slave, 3) gültige Daten von einem Slave oder 4) ungültige Daten (oder leerlaufender Bus) sind. Eine Benutzung dieser zusätzlichen Steuerungsleitung beseitigt den Bedarf für die Transceiver, zu verfolgen, wann Daten von ihrem Primär-Bus zu dem Transceiver-Bus weitergeleitet werden müssen – sämtliche Transceiver senden alle Daten von ihrem Primär-Bus zu dem Transceiver-Bus, wenn das Steuerungssignal die Bedingung 2) oben angibt. In einer bevorzugten Implementierung ist, wenn AddrValid und TrncvrRW beide "low" sind, keine Busaktivität vorhanden, und die Transceiver bleiben in einem Leerlaufzustand. Eine ein Anforderungspaket sendende Steuerung setzt AddrValid high, um für alle Geräte auf dem Transceiver-Bus anzuzeigen, dass ein Anforderungspaket gesendet ist, welches jeder Transceiver zu seiner primären Buseinheit weiterleiten soll. Jede zu einem Slave zu schreiben versuchende Steuerung soll AddrValid und TrncvrRW high ansteuern, um anzuzeigen, dass gültige Daten für einen Slave auf den Datenleitungen vorhanden sind. Jedes Transceiver-Gerät sendet dann sämtliche Daten von den Transceiver-Busleitungen zu jeder primären Buseinheit. Jede Steuerung, die erwartet, Informationen von einem Slave zu empfangen, soll ebenfalls die TrncvrRW-Leitung high ansteuern, aber nicht AddrValid ansteuern, um dadurch für jeden Transceiver anzuzeigen, von jedem Slave auf seinem primären Lokalbus kommende Daten zu dem Transceiver-Bus zu senden. Ein noch weiter entwickelter Transceiver erkennt Signale, die an ihre primäre Buseinheit adressiert sind oder von ihr kommen, und sendet Signale nur zu geforderten Zeiten.

[0111] Ein Beispiel der physikalischen Anordnung der Transceiver ist in [Fig. 9](#) gezeigt. Ein wesentliches Merkmal dieser physikalischen Anordnung ist es, den Bus jedes Transceivers **19** mit dem Original-Bus von DRAMs oder anderen Geräten **15**, **16**, **17** auf der primären Buseinheit **66** zu integrieren. Die Transceiver **19**

haben Pins an zwei Seiten und werden bevorzugt flach auf der primären Buseinheit angebracht, mit einem ersten mit dem primären Bus **18** verbundenen Satz von Pins. Ein zweiter Satz Transceiver-Pins **20**, bevorzugt orthogonal zu dem ersten Satz von Pins, ist ausgerichtet, um dem Transceiver **19** zu erlauben, an dem Transceiver-Bus **65** in im wesentlichen der gleichen Weise angebracht zu werden, wie die DRAMs an der primären Buseinheit angebracht werden. Der Transceiver-Bus kann allgemein planar und in einer anderen Ebene, bevorzugt orthogonal zu der Ebene jeder primären Buseinheit sein. Der Transceiver-Bus kann ebenfalls allgemein rund sein, mit primären Buseinheiten, die senkrecht und tangential an den Transceiver-Bus angebracht sind.

[0112] Eine Verwendung dieses Zwei-Ebenen-Schemas erlaubt es, leicht ein System aufzubauen, das über 500 Slaves (16 Busse mit jeweils 32 DRAMs) enthält.

[0113] Durchschnittsfachleute können das oben beschriebene Geräte-ID-Schema modifizieren, um mehr als 256 Geräte aufzunehmen, zum Beispiel durch Verwendung einer längeren Geräte-ID oder Verwendung zusätzlicher Register zum Halten einiger der Geräte-ID. Dieses Schema kann noch in einer dritten Dimension erweitert werden, um einen Transceiver-Bus zweiter Ordnung zu schaffen, der mehrere Transceiver-Busse durch Ausrichten der Transceiver-Buseinheiten parallel zu- und übereinander verbindet, und der entsprechende Signalleitungen durch einen geeigneten Transceiver zum Bus führt. Die Verwendung eines solchen Transceiver-Busses zweiter Ordnung erlaubt das Anschließen vieler Tausender Slave-Geräte an das, was effektiv ein einzelner Bus ist.

Geräte-Schnittstelle

[0114] Die Geräte-Schnittstelle zu dem Hochgeschwindigkeits-Bus kann in drei Hauptteile aufgeteilt werden. Der erste Teil ist die elektrische Schnittstelle. Dieser Teil beinhaltet die Eingangsempfänger, Bustreiber und die Takterzeugungsschaltung. Der zweite Teil enthält die Adress-Vergleichsschaltung und Timing-Register. Dieser Teil nimmt das eingegebene Anforderungspaket und bestimmt, ob die Anforderung für dieses Gerät ist, und wenn dies der Fall ist, beginnt der interne Zugriff und liefert die Daten zur richtigen Zeit zu den Pins. Der letzte Teil, insbesondere für Speichergeräte wie DRAMs, ist der DRAM-Spalten-Zugriffspfad. Dieser Teil benötigt eine Bereitstellung von Bandbreite in und aus den DRAM-Leseverstärkern, die größer als die in konventionellen DRAMs vorgesehene Bandbreite ist. Die Implementierung der elektrischen Schnittstelle und der DRAM-Spalten-Zugriffspfad werden noch detaillierter in den folgenden Abschnitten beschrieben. Durchschnittsfachleute erkennen, wie bekannte Adress-Vergleichsschaltungen und bekannte Registerschaltungen zu modifizieren sind.

Elektrische Schnittstelle – Eingabe/Ausgabe-Schaltung

[0115] Ein Blockschaltbild der bevorzugten Eingabe/Ausgabe-Schaltung für Adress/Daten/Steuerungs-Leitungen ist in [Fig. 10](#) gezeigt. Diese Schaltung ist insbesondere gut geeignet für die Verwendung in DRAM-Geräten, kann aber durch einen Durchschnittsfachmann zur Verwendung in anderen an den Bus gemäß diesem Ausführungsbeispiel angeschlossenen Geräten verwendet oder modifiziert werden. Sie besteht aus einem Satz von Eingangsempfängern **71**, **72** und Ausgabe-Treibern **76**, welche an die Eingabe/Ausgabe-Leitung **69** und Anschlussfläche **75** angeschlossen sind und Schaltungen zum Verwenden des internen Taktes **73** und des internen Takt-Komplements **74** zum Ansteuern der Eingabe-Schnittstelle. Die getakteten Eingangsempfänger nutzen den Vorteil der synchronen Natur des Busses. Um die Leistungsanforderungen für Geräte-Eingabe-Empfänger weiter zu verringern, ist jeder Geräte-Pin und somit jede Busleitung an zwei getaktete Empfänger angeschlossen, einen zum Abtasten der geradzahigen Zyklus-Eingaben und den anderen zum Abtasten der ungeradzahigen Zyklus-Eingaben. Durch dieses De-multiplexen des Eingangs **70** an dem Pin wird jedem getakteten Verstärker ein voller 2 ns-Zyklus zum Verstärken des Niedrigspannungs-Hub-Signals auf dem Bus zu einem vollwertigen CMOS-Logiksignal gegeben. Durchschnittsfachleute erkennen, dass zusätzlich getaktete Eingangsempfänger verwendet werden können. Zum Beispiel können vier Eingangsempfänger an jeden Geräte-Pin angeschlossen sein und durch einen modifizierten internen Gerätetakt getaktet werden, um sequentielle Bits von dem Bus zu internen Geräteschaltungen zu übertragen und noch höhere externe Busgeschwindigkeiten oder noch längere Einstellzeiten zum Verstärken des Niedrigspannungs-Hub-Signales des Busses zu einem vollwertigen CMOS-Logiksignal zu erlauben.

[0116] Die Ausgabe-Treiber sind wirklich einfach und bestehen aus einem einzelnen NMOS-Pulldown-Transistor **76**. Dieser Transistor ist so bemessen, dass er unter ungünstigsten Bedingungen immer noch die von dem Bus benötigten 50 mA aufnehmen kann. Bei einer 0,8 Micron CMOS-Technologie ist der Transistor etwa 200 Micron lang. Die gesamte Busleistung kann verbessert werden durch Verwenden von Rückkopplungstech-

niken zum Steuern des Ausgabe-Transistorstromes, so dass der Strom durch das Gerät ungefähr 50 mA unter allen Betriebsbedingungen ist, obwohl dies für einen einwandfreien Busbetrieb nicht absolut erforderlich ist. Ein Beispiel von einem aus vielen, Durchschnittsfachleuten bekannten Verfahren bei der Verwendung von Rückkopplungstechniken zum Steuern eines Stromes ist beschrieben in Hans Schumacher, et al., "CMOS Subnanosecond True-ECL Output Buffer", J. Solid State Circuits, Band 25 (1), Seiten 150–154 (Feb. 1990). Ein Steuern dieses Stromes verbessert die Leistungsfähigkeit und verringert den Energieverbrauch. Dieser Ausgabetreiber, welcher mit 500 MHz betrieben werden kann, kann wiederum durch einen geeigneten Multiplexer mit zwei oder mehr (bevorzugt vier) Eingängen gesteuert werden, welche an die weitere interne Chip-Schaltung angeschlossen sind, von denen alle gemäß dem bekannten Stand der Technik ausgebildet sind.

[0117] Die Eingangsempfänger jedes Slave müssen in der Lage sein, während jedes Zyklus' zu arbeiten, um zu bestimmen, ob das Signal auf dem Bus ein gültiges Anforderungspaket ist. Diese Anforderung führt zu einer Anzahl von Beschränkungen für die Eingabeschaltung. Zusätzlich zu der Anforderung kleiner Erfassungs- und Auflösungs-Verzögerungen dürfen die Schaltungen geringe oder keine Gleichspannung, geringe Wechselspannung aufnehmen und einen sehr kleinen Strom zurück in die Eingabe- oder Referenz-Leitungen abgeben. Der in [Fig. 11](#) gezeigte, getaktete Standard-DRAM-Leseverstärker erfüllt all diese Anforderungen mit Ausnahme des Bedarfs nach geringen Eingangsströmen. Wenn dieser Leseverstärker vom Lesen zum Abtasten übergeht, wird die Kapazität der internen Knoten **83** und **84** in [Fig. 11](#) durch die Referenz-Leitung **68** und den Eingang **69** entladen. Dieser bestimmte Strom ist gering, aber die Summe solcher Ströme von sämtlichen Eingängen auf die Referenz-Leitungen, summiert über alle Geräte, kann beträchtlich groß sein.

[0118] Die Tatsache, dass das Vorzeichen des Stromes von den vorher empfangenen Daten abhängt, macht die Sache ungünstig. Ein Weg zum Lösen dieses Problems ist es, die Abtastperiode in zwei Phasen aufzuteilen. Während der ersten Phase werden die Eingänge gegen eine gepufferte Version des Referenzpegels (welcher ein Offset aufweisen kann) kurzgeschlossen. Während der zweiten Phase werden die Eingänge an die wahren Eingänge angeschlossen. Dieses Schema entfernt den Eingangsstrom nicht vollständig, da der Eingang die Knoten **83** und **84** von dem Referenzwert zu dem gegenwärtigen Eingangswert aufladen muss, aber er verringert die erforderliche Gesamtladung um etwa einen Faktor von 10 (es erfordert lediglich eine 0,25V Änderung statt einer 2,5V Änderung). Durchschnittsfachleute erkennen, dass viele andere Verfahren verwendet werden können, um einen getakteten Verstärker anzugeben, der mit sehr niedrigen Eingangsströmen arbeitet.

[0119] Ein wesentlicher Teil der Eingabe/Ausgabe-Schaltung erzeugt einen internen Gerätetakt basierend auf frühen und späten Bus-Takten. Ein Steuern der Takt-Verzerrung (der Differenz im Takt-Timing zwischen Geräten) ist in einem mit 2 ns-Zyklen laufenden Gerät wichtig, und somit wird der interne Gerätetakt erzeugt, so dass der Eingangs-Abtaster und der Ausgangs-Treiber zeitlich so nahe wie möglich an der Mitte zwischen den zwei Bus-Takten arbeitet.

[0120] Ein Blockschaltbild der internen Gerätetakt-Erzeugungsschaltung ist in [Fig. 12](#) gezeigt, und das entsprechende Zeitdiagramm in [Fig. 13](#). Die Grundidee hinter dieser Schaltung ist relativ einfach. Ein Gleichspannungsverstärker **102** wird verwendet, um den Bus-Takt mit geringem Hub in ein CMOS-Signal mit vollem Hub umzuwandeln. Dieses Signal wird dann in eine veränderliche Verzögerungsleitung **103** eingespeist. Der Ausgang der Verzögerungsleitung **103** versorgt drei zusätzliche Verzögerungsleitungen: **104** mit einer festen Verzögerung; **105** mit der gleichen festen Verzögerung plus einer zweiten veränderlichen Verzögerung; und **106** mit der gleichen festen Verzögerung plus einer Hälfte der zweiten veränderlichen Verzögerung. Die Ausgänge **107**, **108** der Verzögerungsleitungen **104** und **105** steuern getaktete Eingangsempfänger **101** und **111** an, die mit frühen und späten Bus-Takteingängen **100** und **110** verbunden sind. Diese Eingangsempfänger **101** und **111** haben den gleichen Aufbau wie die oben beschriebenen und in [Fig. 11](#) gezeigten Empfänger. Veränderliche Verzögerungsleitungen **103** und **105** werden über Rückkopplungsleitungen **116**, **115** angepasst, so dass Eingangsempfänger **101** und **111** die Bus-Takte genau in deren Übergängen abtasten. Verzögerungsleitungen **103** und **105** werden so angepasst, dass die fallende Flanke **120** des Ausgangs **107** der fallenden Flanke **121** des frühen Bus-Taktes, Takt1 **53**, um einen Zeitbetrag **128** vorausgeht, der gleich der Verzögerung in dem Eingangsabtaster **101** ist. Die Verzögerungsleitung **108** wird auf die gleiche Weise angepasst, so dass die fallende Flanke **122** der fallenden Flanke **123** des späten Bus-Taktes, Takt2 **54**, um die Verzögerung **128** in dem Eingangsabtaster **111** vorausgeht.

[0121] Da die Ausgänge **107** und **108** mit den zwei Bus-Takten synchronisiert sind und der Ausgang **73** der letzten Verzögerungsleitung **106** mittig zwischen den Ausgängen **107** und **108** ist, d.h., der Ausgang **73** folgt dem Ausgang **107** mit dem gleichen Zeitbetrag **129**, den der Ausgang **73** dem Ausgang **108** vorausgeht, stellt

der Ausgang **73** einen internen Gerätetakt mittig zwischen den Bus-Takten bereit. Die fallende Flanke **124** des internen Gerätetaktes **73** geht der Zeit der tatsächlichen Eingabe-Abtastung **125** um eine Abtaster-Verzögerung voraus. Es ist anzumerken, dass diese Schaltungsanordnung die Verzögerung in im wesentlichen allen Geräte-Eingangsempfängern **71** und **72** ([Fig. 10](#)) automatisch ausgleicht, da die Ausgänge **107** und **108** angepasst werden, so dass die Bus-Takte durch Eingangsempfänger **101** und **111** genau bei den Bus-Takt-Übergängen abgetastet werden.

[0122] In der bevorzugten Ausführungsform werden zwei Sätze dieser Verzögerungsleitungen verwendet, einer zum Erzeugen des wahren Wertes des internen Gerätetaktes **73**, und der andere zum Erzeugen des Komplements **74**, ohne eine Invertierer-Verzögerung hinzuzufügen. Diese duale Schaltung erlaubt die Erzeugung echter komplementärer Takte mit extrem geringer Verzerrung. Der komplementäre, interne Gerätetakt wird verwendet, um die "geradzahligem" Eingangsempfänger zu takten, um zum Zeitpunkt **127** abzutasten, während der wahre interne Gerätetakt verwendet wird, um die "ungeradzahligem" Eingangsempfänger zu takten, um zum Zeitpunkt **125** abzutasten. Die wahren und komplementären internen Gerätetakte werden ebenfalls verwendet, um auszuwählen, welche Daten an die Ausgabe-Treiber angelegt werden. Die Gate-Verzögerung zwischen dem internen Gerätetakt und Ausgabeschaltungen, welche den Bus ansteuern, ist geringfügig größer als die entsprechende Verzögerung für die Eingangsschaltungen, was bedeutet, dass die neuen Daten stets an den Bus angelegt werden, kurz nachdem die alten Daten abgetastet wurden.

DRAM-Spalten-Zugriffsmodifikation

[0123] Ein Blockschaltbild eines konventionellen 4 MBit-DRAM **130** ist in [Fig. 15](#) gezeigt. Die DRAM-Speichermatrix ist in eine Anzahl von Teilmatrizen **150–157**, zum Beispiel 8, aufgeteilt. Jede Teilmatrix ist in Matrizen **148, 149** aus Speicherzellen aufgeteilt. Eine Zellenadressauswahl wird durch Dekodierer **146** ausgeführt. Ein Spaltendekodierer **147A, 147B** einschließlich Spalten-Leseverstärkern an jeder Seite des Dekodierers durchläuft den Kern jeder Teilmatrix. Diese Spalten-Leseverstärker können eingestellt werden, um den zuletzt gespeicherten Wert vorauf-zuladen oder zwischenzuspeichern, wie detailliert oben beschrieben. Interne E/A-Leitungen verbinden jeden Satz von Leseverstärkern, wie durch entsprechende, vorgeschaltete Spalten-Dekodierer, mit einer Eingabe- und Ausgabeschaltung, die direkt mit den Geräte-Pins verbunden ist. Diese internen E/A-Leitungen werden verwendet, um die Daten von den ausgewählten Bitleitungen zu den Daten-Pins (einige der Pins **131–145**) zu führen, oder Daten von den Pins zu entnehmen und die ausgewählten Bit-Leitungen zu schreiben. Solch ein durch bekannte Beschränkungen organisierter Spaltenzugriffspfad weist keine ausreichende Bandbreite für eine Schnittstelle mit einem Hochgeschwindigkeits-Bus auf. Das Verfahren gemäß diesem Ausführungsbeispiel erfordert nicht die Änderung des für den Spaltenzugriff verwendeten gesamten Verfahrens, ändert aber Implementierungs-Einzelheiten. Viele dieser Einzelheiten wurden selektiv in bestimmten schnellen Speichergeräten implementiert, aber niemals in Verbindung mit der Bus-Architektur gemäß diesem Ausführungsbeispiel.

[0124] Ein Betreiben der internen E/A-Leitungen auf konventionelle Weise mit hohen Bus-Zyklusfrequenzen ist nicht möglich. Bei dem bevorzugten Verfahren werden mehrere (bevorzugt 4) Bytes während jedes Zyklus' gelesen oder geschrieben und der Spalten-Zugriffspfad ist modifiziert, um mit einer niedrigeren Frequenz zu arbeiten (die Inverse der Anzahl der pro Zyklus angesprochenen Bytes, bevorzugt 1/4 der Bus-Zyklusfrequenz). Drei verschiedene Techniken werden verwendet, um die erforderlichen zusätzlichen internen E/A-Leitungen bereitzustellen und Daten zu Speicherzellen mit dieser Frequenz zu liefern. Zuerst wird die Anzahl von E/A-Bit-Leitungen in jeder Teilmatrix, welche den Spaltendekodierer **147** durchläuft, erhöht, z. B. auf 16, acht für jede der zwei Spalten der Spalten-Leseverstärker, und der Spaltendekodierer wählt einen Satz von Spalten aus der "oberen" Hälfte **148** der Teilmatrix **150** und einen Satz von Spalten aus der "unteren" Hälfte **149** während jedes Zyklus', wobei der Spaltendekodierer einen Spalten-Leseverstärker pro E/A-Bit-Leitung auswählt. Zweitens wird jede Spalten-E/A-Leitung in zwei Hälften aufgeteilt, welche Daten unabhängig über getrennte interne E/A-Leitungen von der linken Hälfte **147A** und der rechten Hälfte **147B** jeder Teilmatrix (teilt jede Teilmatrix in Quadranten) transportiert, und der Spaltendekodierer wählt Leseverstärker von jeder rechten und linken Hälfte der Teilmatrix aus, um die Anzahl von in jedem Zyklus verfügbaren Bits zu verdoppeln. Somit schaltet jede Spalten-Dekodiererauswahl n Spalten-Leseverstärker an, wobei n gleich vier (obere linke und rechte, untere linke und rechte Quadranten) mal die Anzahl von E/A-Leitungen in dem Bus zu jedem Teilmatrix-Quadranten (8 Leitungen jeweils $\times 4 = 32$ Leitungen in der bevorzugten Implementierung) ist. Schließlich werden während jedes RAS-Zyklus' zwei unterschiedliche Teilmatrizen, z.B. **157** und **153**, angesprochen. Dies verdoppelt erneut die verfügbare Anzahl von Daten enthaltenden E/A-Leitungen. Zusammengefasst erhöhen diese Änderungen die interne E/A-Bandbreite um wenigstens einen Faktor 8. Vier interne Busse werden verwendet, um diese internen E/A-Leitungen zu führen. Ein Erhöhen der Anzahl von E/A-Leitungen und Aufteilen derselben in der Mitte verringert deutlich die Kapazität jeder internen E/A-Leitung, was wiederum die

Spaltenzugriffszeit verringert und die Spaltenzugriffsbandbreite noch weiter erhöht.

[0125] Die Mehrzahl der oben beschriebenen Eingangsempfänger mit vorgeschaltetem Gate erlaubt eine Hochgeschwindigkeitseingabe von den Geräte-Pins auf den internen E/A-Leitungen und direkt in den Speicher. Der oben beschriebene, gemultiplexte Ausgabetreiber wird verwendet, um mit dem durch die Verwendung dieser Techniken verfügbaren Datenfluss Schritt zu haften. Steuerungseinrichtungen sind vorgesehen, um auszuwählen, ob Information an den Geräte-Pins als eine Adresse behandelt werden soll, und daher zu dekodieren ist, oder als Eingabe- oder Ausgabedaten, die anzulegen sind an die oder zu lesen sind von den internen E/A-Leitungen.

[0126] Jede Teilmatrix kann 32 Bits pro Zyklus ansprechen, 16 Bits von der linken Teilmatrix und 16 von der rechten Teilmatrix. Mit 8 E/A-Leitungen pro Leseverstärker-Spalte und Ansprechen von zwei Teilmatrizen gleichzeitig kann der DRAM 64 Bits pro Zyklus bereitstellen. Diese zusätzliche E/A-Bandbreite wird nicht zum Lesen benötigt (und wird wahrscheinlich nicht verwendet), kann aber zum Schreiben benötigt werden. Die Verfügbarkeit von Schreib-Bandbreite ist ein schwierigeres Problem als Lese-Bandbreite, da Überschreiben eines Wertes in einem Leseverstärker ein langsamer Vorgang sein kann, abhängig davon, wie der Leseverstärker an die Bit-Leitung angeschlossen ist. Der zusätzliche Satz interner E/A-Leitungen stellt einen Bandbreiten-Streifen für Schreibvorgänge bereit.

Patentansprüche

1. Synchron dynamische Halbleiter-Direktzugriffsspeichervorrichtung (DRAM), welche ein Speicherfeld mit einer Mehrzahl dynamischer Speicherzellen beinhaltet, wobei die Speichervorrichtung umfasst: Kopplungsmittel, welche ausgestaltet sind, um das DRAM an einen externen Bus, welcher Teil einer Halbleiter-Busarchitektur ist, zu koppeln, wobei die Halbleiter-Busarchitektur eine Mehrzahl von Halbleitervorrichtungen beinhaltet, welche parallel an den externen Bus gekoppelt sind, wobei der externe Bus eine Mehrzahl von Busleitungen beinhaltet, um im Wesentlichen alle Adress-, Daten-, und Steuerinformationen zu führen, die von dem DRAM zur Kommunikation mit im Wesentlichen allen anderen Halbleitervorrichtungen, die an den externen Bus gekoppelt sind, benötigt werden, wobei die Kopplungsmittel ausgestaltet sind, um gemultiplexte Adressen zu empfangen, eine Taktempfängerschaltung, um ein externes Taktsignal zu empfangen, ein programmierbares Zugriffszeit-Register zum Speichern eines Wertes, der eine Anzahl von Taktzyklen eines externen Taktsignals (**53, 54**) repräsentiert, die verstreichen sollen und nach denen das DRAM auf einen Operations-Code antwortet, wobei der externe Bus auf das programmierbare Zugriffszeit-Register über die Kopplungsmittel zugreifen kann, und Daten zu dem programmierbaren Zugriffszeit-Register über den externen Bus übertragen werden, um den Wert in dem programmierbaren Zugriffszeit-Register zu setzen, Leseverstärker, welche mit dem Speicherfeld gekoppelt sind, um Daten aus den dynamischen Speicherzellen auszulesen, eine Mehrzahl von Eingangsempfängern, um den Operations-Code synchron in Bezug auf einen ersten Pegelwechsel des externen Taktsignals abzutasten, wobei der Operations-Code die Speichervorrichtung dazu anweist: einen Lesevorgang auszuführen, so dass die Speichervorrichtung die Daten, die von den Leseverstärkern gelesen wurden, ausgibt, und die Leseverstärker automatisch voraufzuladen, nachdem die Daten gelesen sind, und eine Mehrzahl von Ausgabe-Treibern (**76**) zum Ausgeben der Daten auf den externen Bus (**18, 65**) als Antwort auf den Operations-Code, wobei die Ausgabe-Treiber (**76**) die Daten, nachdem die Anzahl von Taktzyklen des externen Taktsignals abgelaufen ist, auf den externen Bus (**18, 65**) und synchron in Bezug auf das externe Taktsignal (**53, 54**) ausgeben, so dass der Empfang des Operations-Codes und die korrespondierende Antwort um die Anzahl von Taktzyklen getrennt sind, wie sie durch den in dem programmierbaren Zugriffszeit-Register gespeicherten Wert gewählt wurde, wobei jeder Ausgabe-Treiber aus der Mehrzahl von Ausgabe-Treibern (**76**) die Daten auf den externen Bus (**18, 65**) mit einer Buszyklus-Datenrate ausgibt, die das Zweifache der Rate des externen Taktsignals ist.

2. DRAM nach Anspruch 1, wobei die Mehrzahl von Eingangsempfängern einen ersten Eingangsempfänger beinhaltet, um ein Bit des Operations-Codes abzutasten, wobei das Bit anzeigt, dass die Leseverstärker automatisch voraufzuladen sind, nachdem die Daten gelesen wurden.

3. DRAM nach einem der vorstehenden Ansprüche, wobei die Mehrzahl von Eingangsempfängern Eingangsempfänger beinhaltet, um Blockgrößeninformation synchron mit dem externen Taktsignal abzutasten, wobei die Blockgrößeninformation eine Datenmenge repräsentiert, die von der Speichervorrichtung als Ant-

wort auf den Operations-Code auszugeben ist.

4. DRAM nach einem der vorstehenden Ansprüche, ferner umfassend: einen internen Takterzeugungsschaltkreis der Vorrichtung, der mit der Taktempfängerschaltung gekoppelt ist, um ein internes Taktsignal zu erzeugen, welches verwendet wird, um Daten als Antwort auf den Operations-Code auszugeben, wobei der interne Takterzeugungsschaltkreis der Vorrichtung eine zeitliche Beziehung zwischen den auszugebenden Daten und dem externen Taktsignal überwacht und steuert.

5. DRAM nach Anspruch 4, wobei der interne Takterzeugungsschaltkreis der Vorrichtung vorgesehen ist, um ein komplementäres internes Taktsignal zu erzeugen, wobei das interne Taktsignal und das komplementäre interne Taktsignal verwendet werden, um zu selektieren, welche Daten der Mehrzahl von Ausgabe-Treibern zugeführt werden.

6. DRAM nach Anspruch 5, wobei die Mehrzahl von Ausgabe-Treibern (**76**) einen ersten Abschnitt der Daten als Antwort auf eine steigende Flanke eines Pegelwechsels des internen Taktsignals ausgibt.

7. DRAM nach einem der Ansprüche 4, 5 oder 6, wobei der interne Takterzeugungsschaltkreis der Vorrichtung eine Rückkopplungsleitung verwendet, um eine variable Verzögerungsleitung beim Erzeugen des internen Taktsignals einzustellen.

Es folgen 11 Blatt Zeichnungen

Anhängende Zeichnungen

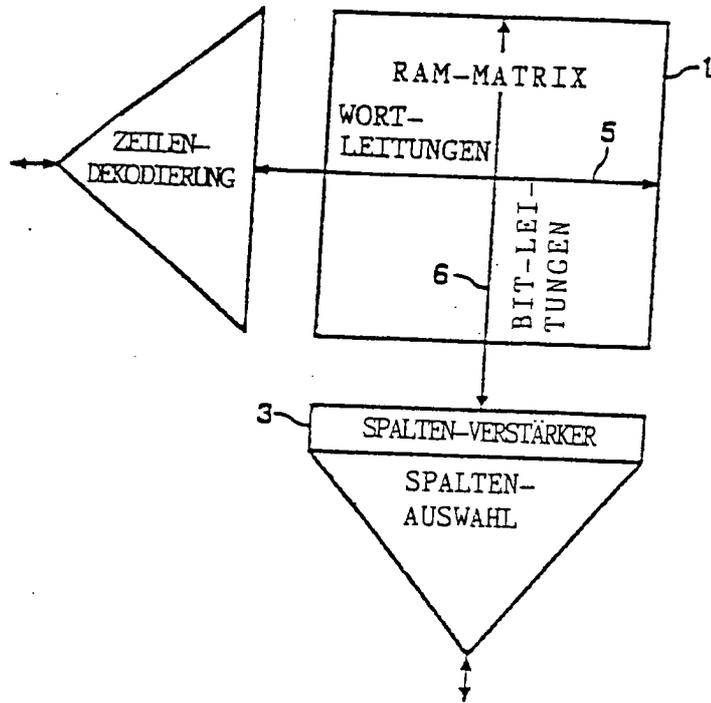


FIG. 1

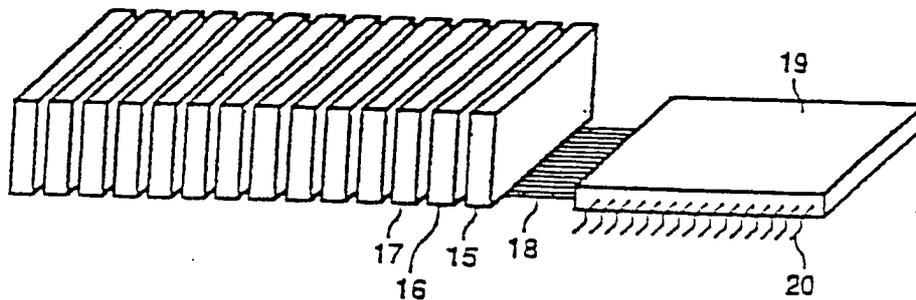


FIG. 3

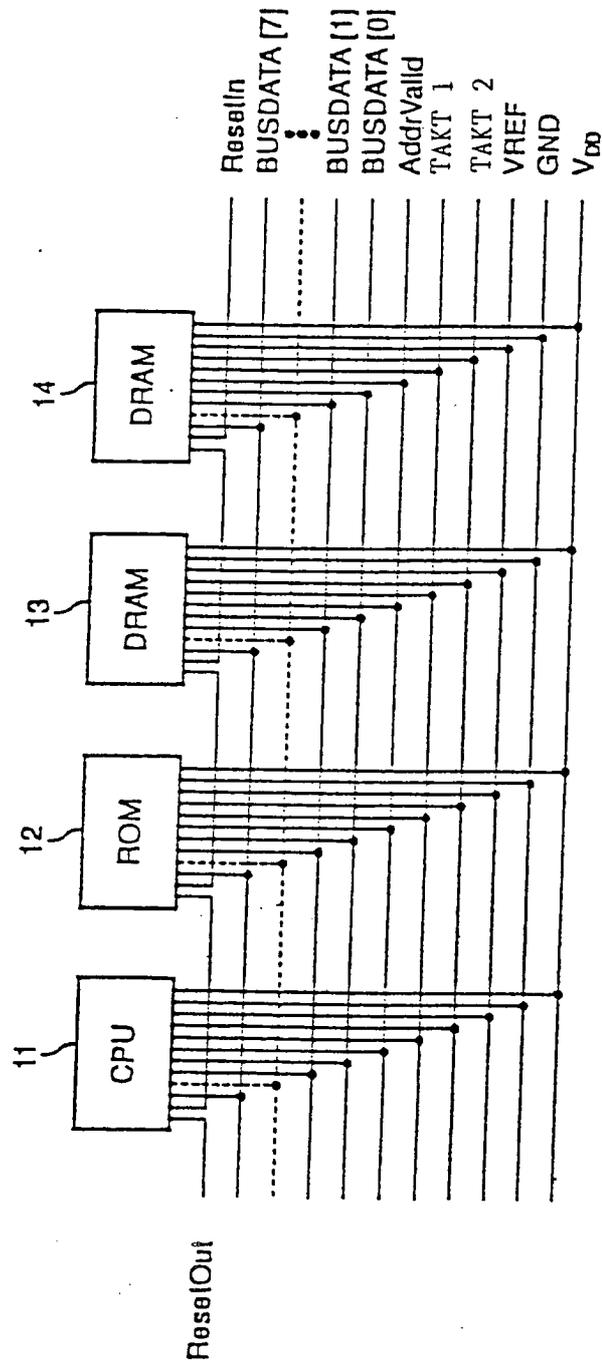


FIG. 2

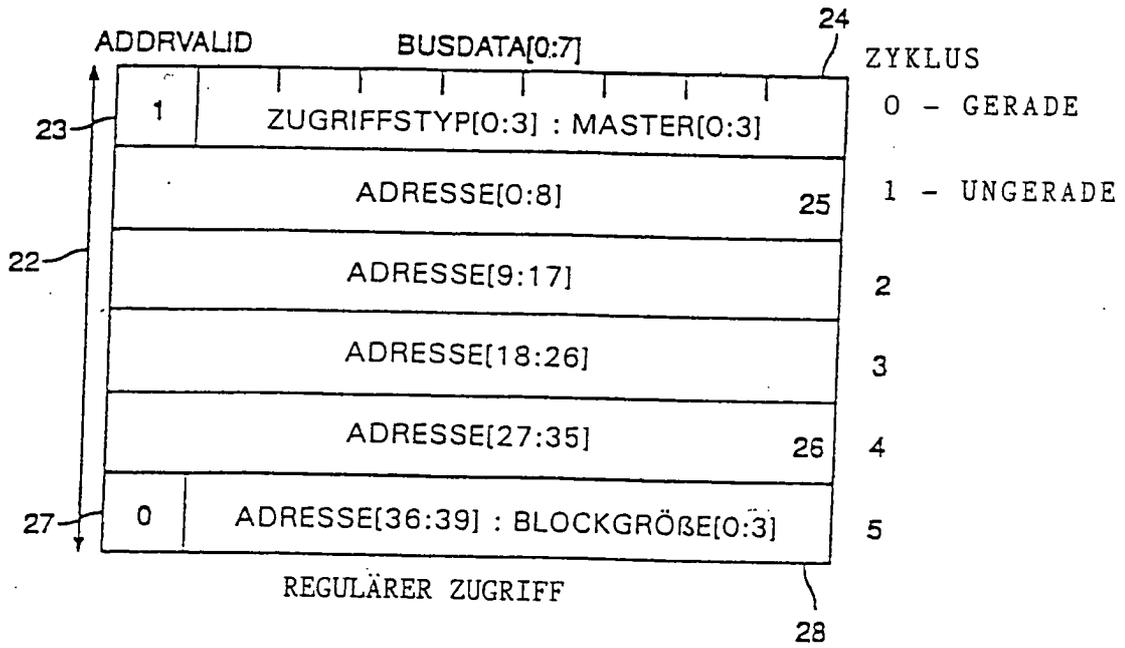


FIG. 4

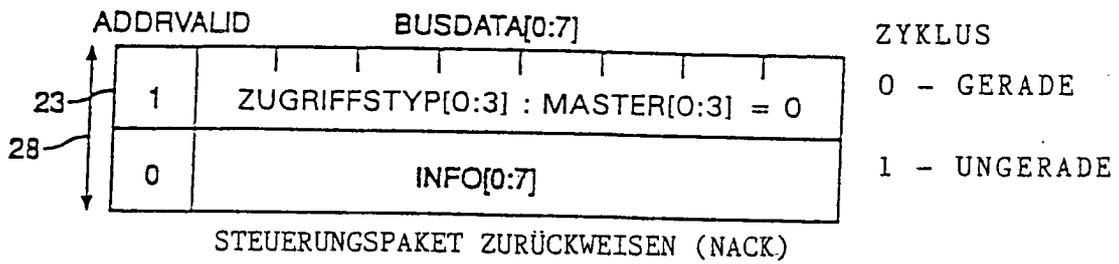


FIG. 5

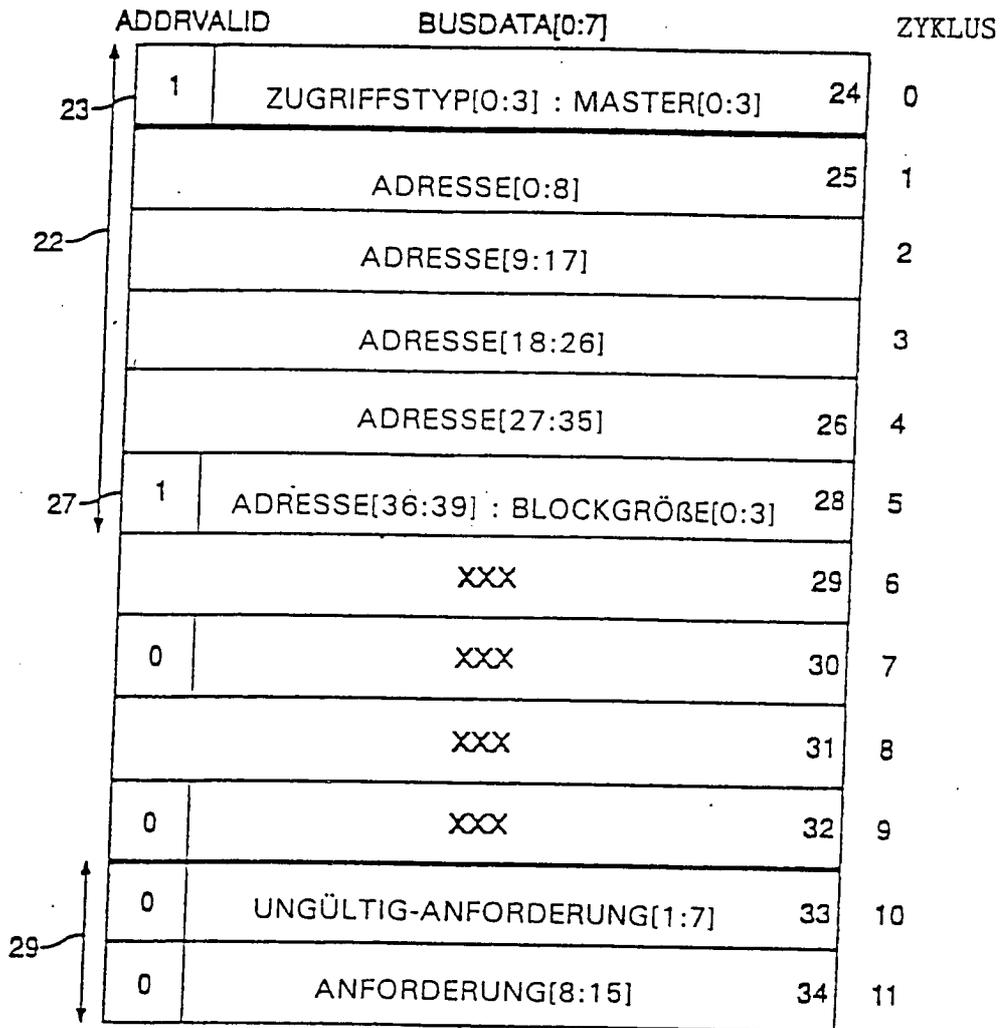


FIG. 6

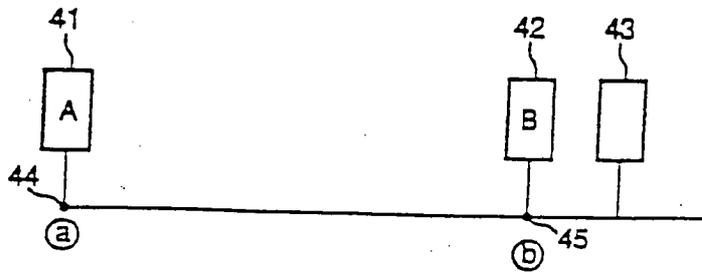


FIG. 7a

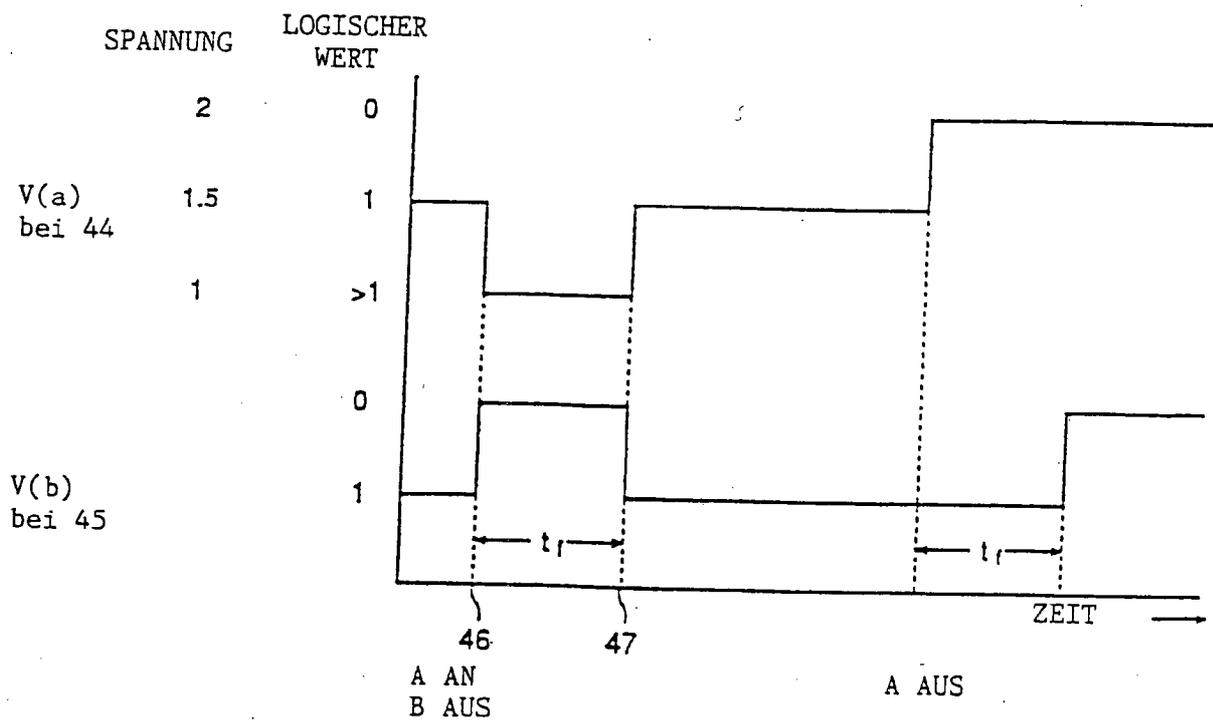


FIG. 7b

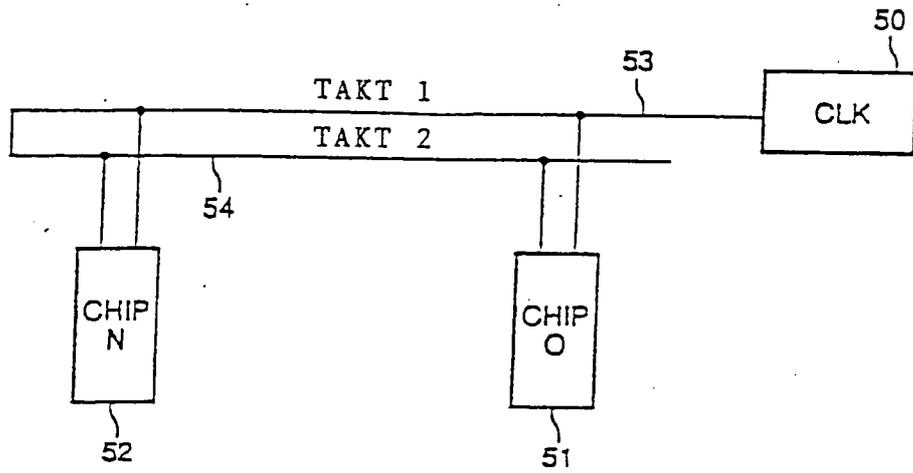


FIG. 8a

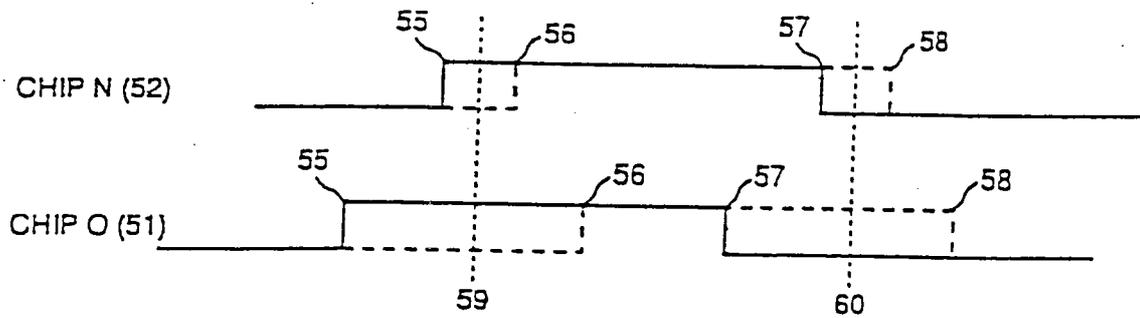


FIG. 8b

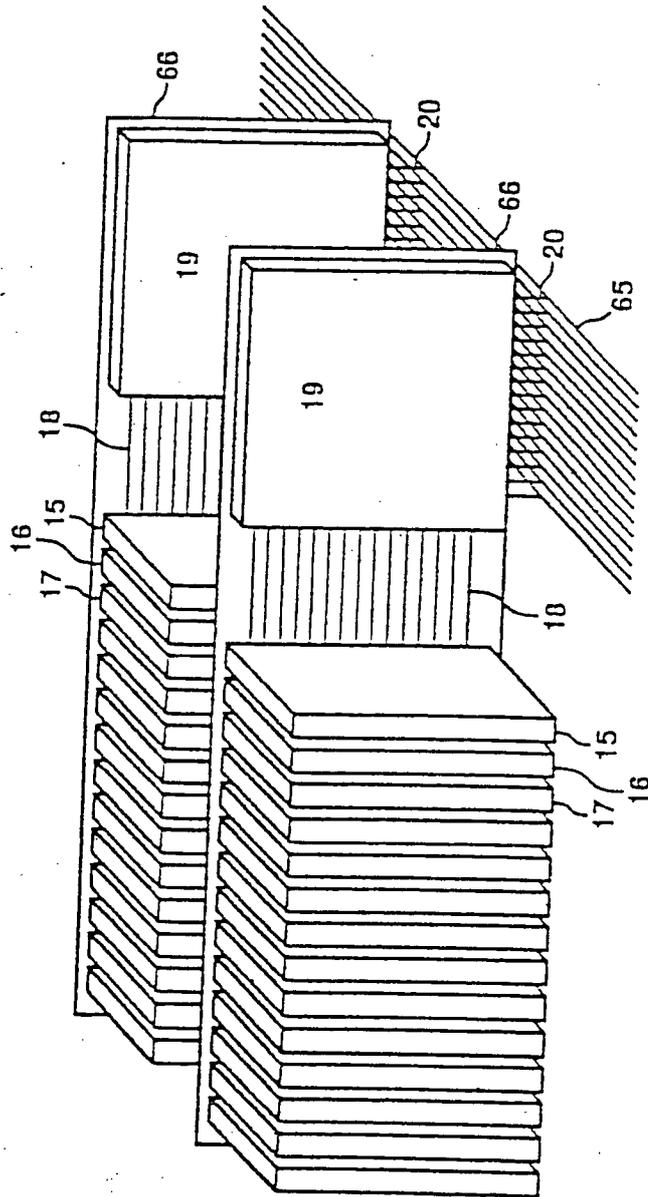


FIG. 9

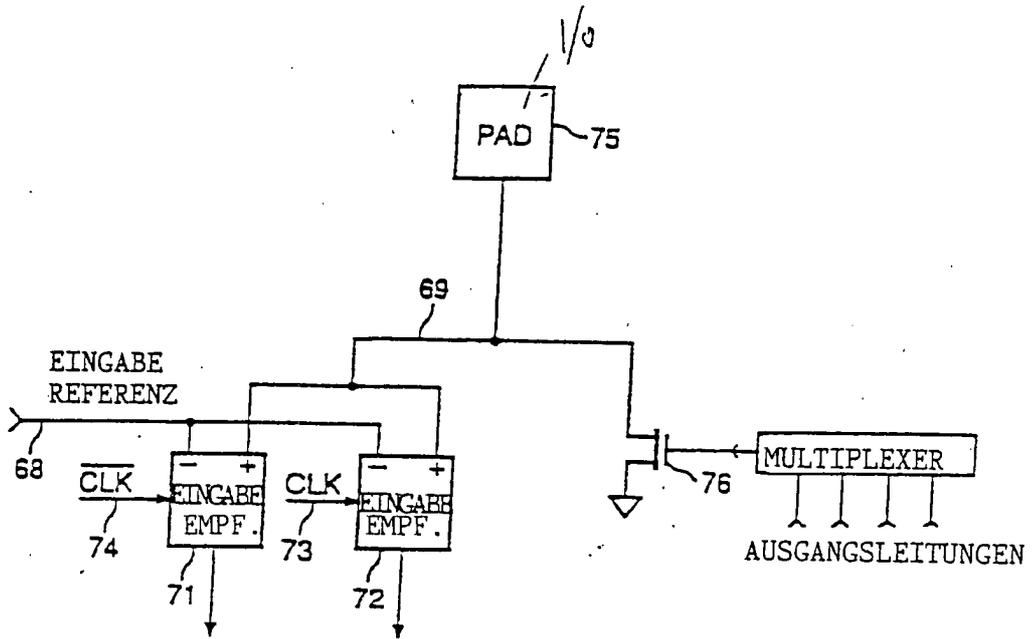


FIG. 10

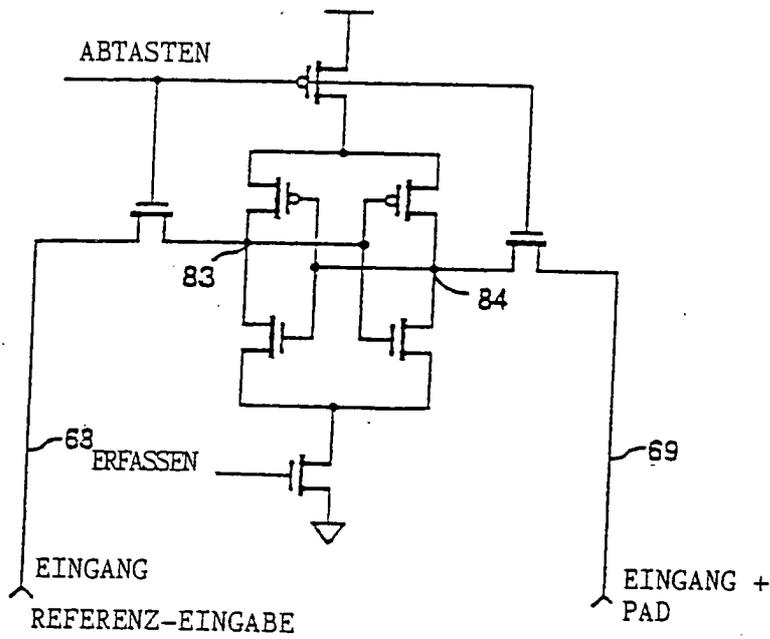


FIG. 11

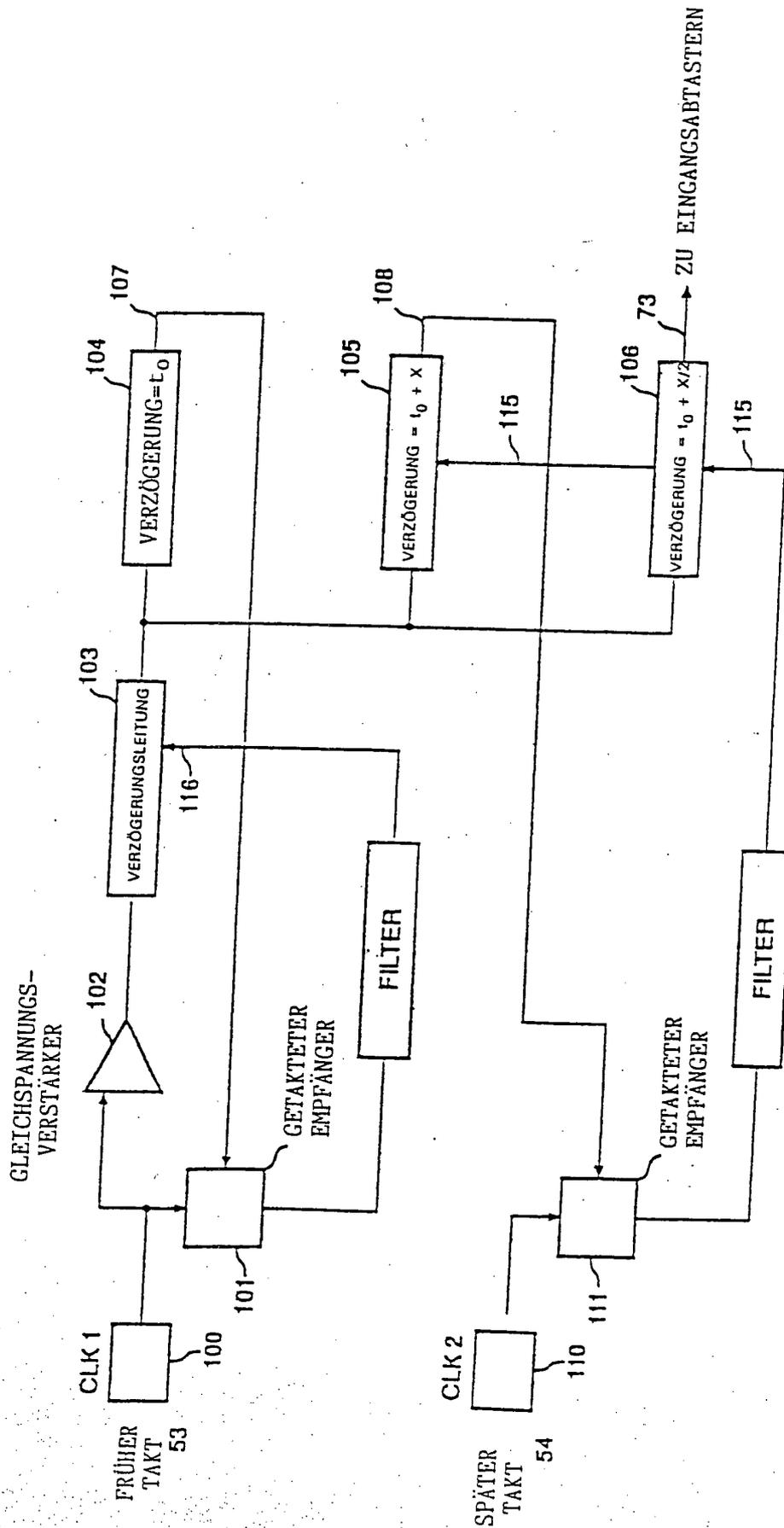


FIG. 12

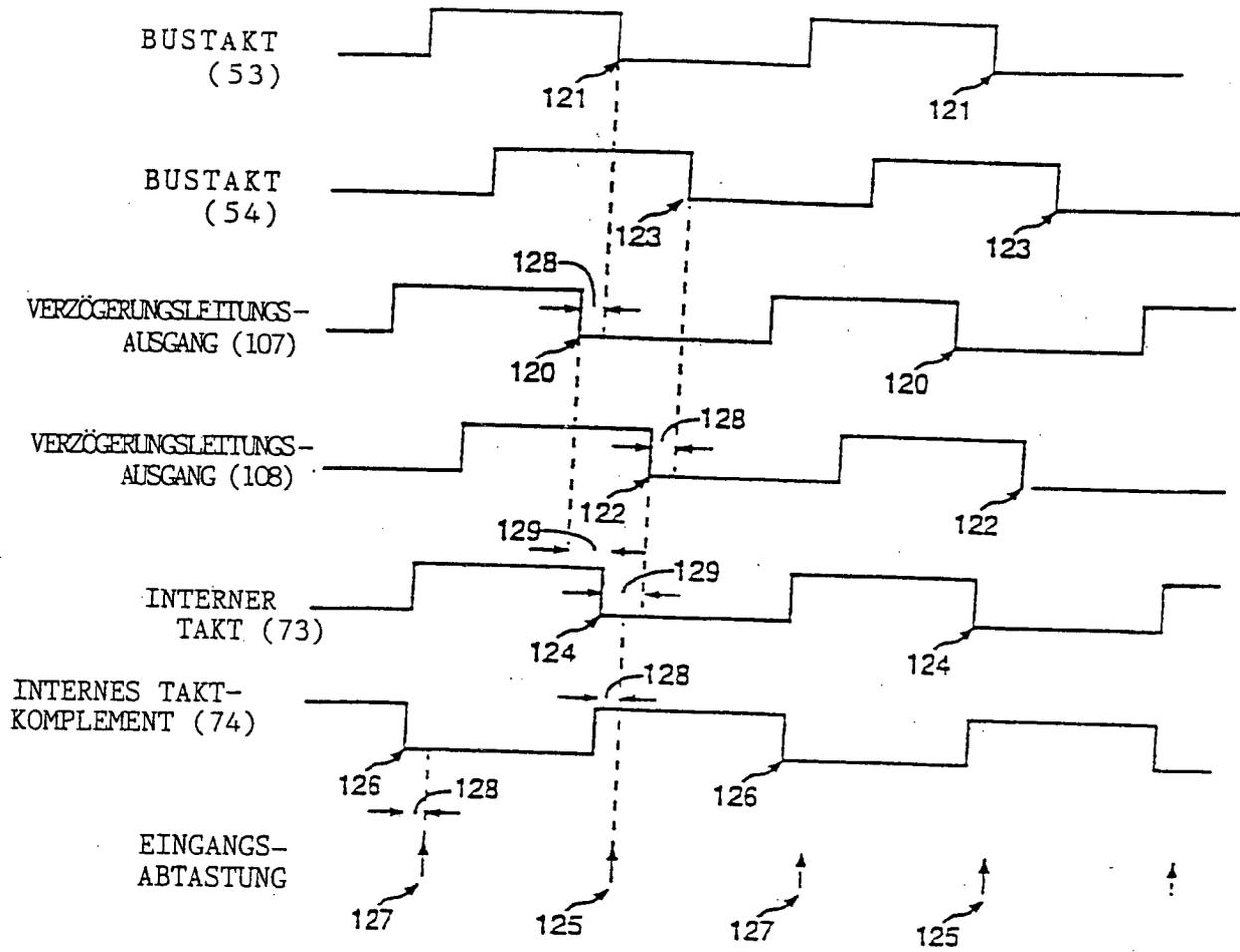


FIG. 13

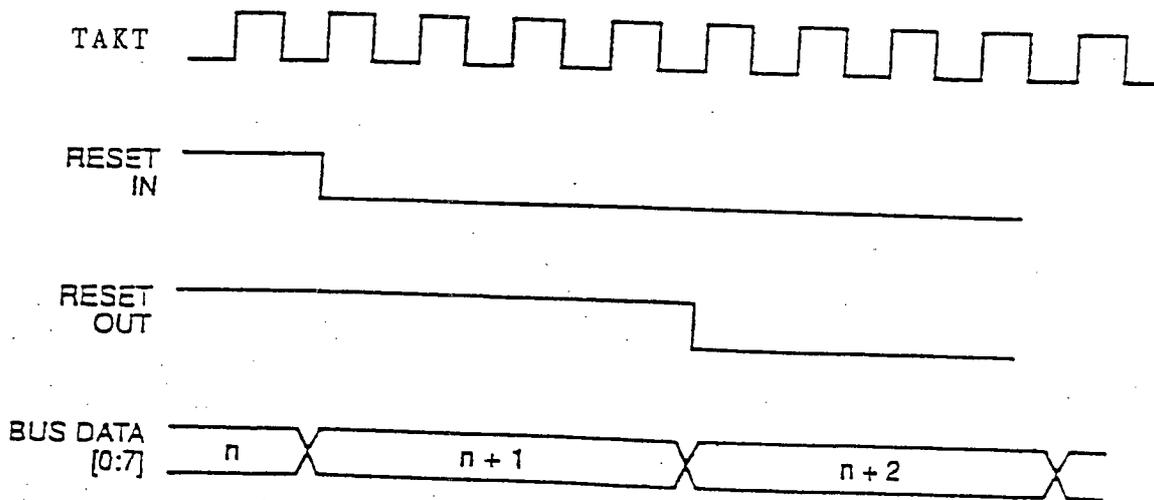


FIG. 14

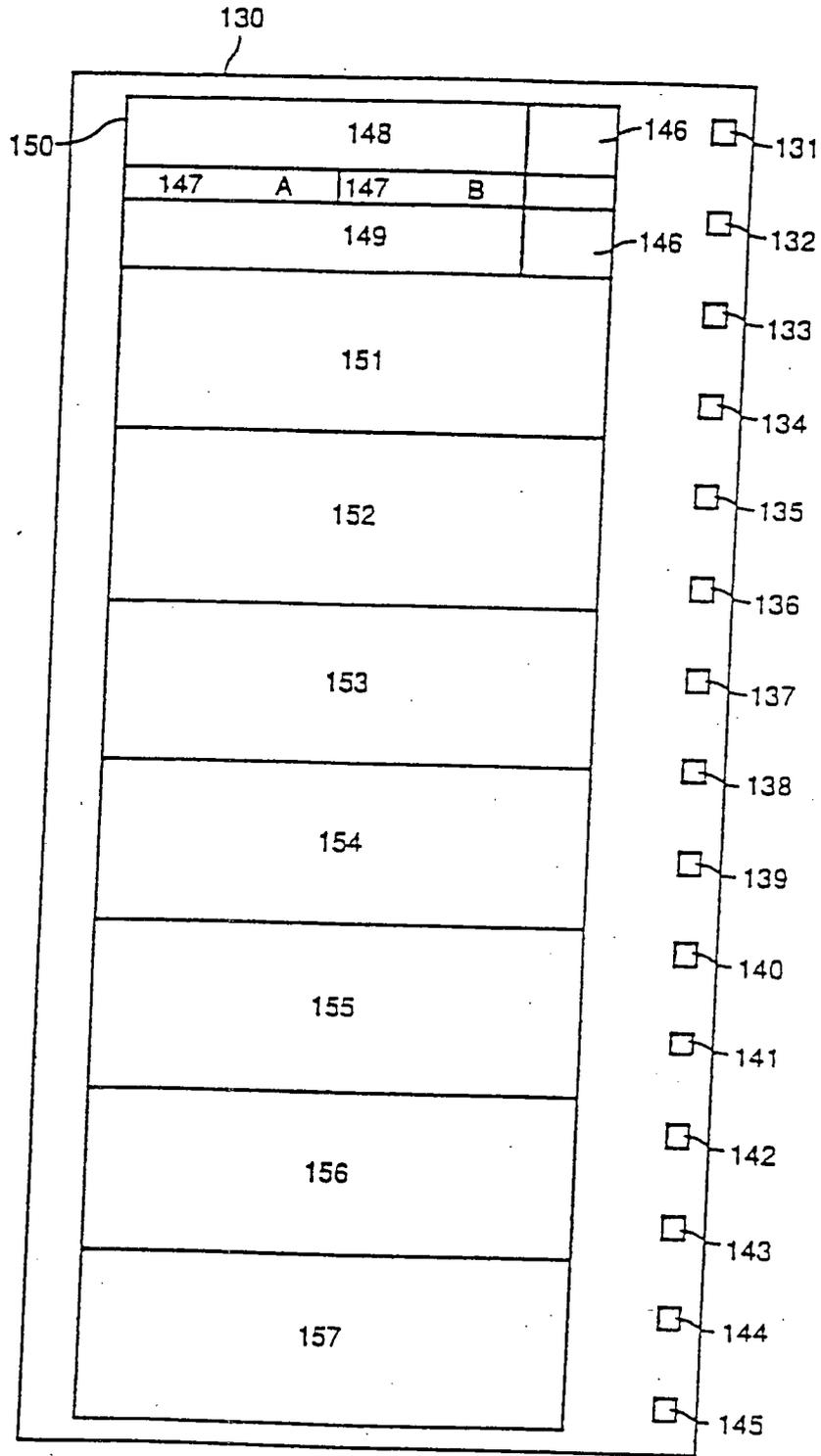


FIG. 15