



(12) **Patentschrift**

(21) Deutsches Aktenzeichen: **199 82 871.7**
(86) PCT-Aktenzeichen: **PCT/US99/01942**
(87) PCT-Veröffentlichungs-Nr.: **WO 1999/041667**
(86) PCT-Anmeldetag: **28.01.1999**
(87) PCT-Veröffentlichungstag: **19.08.1999**
(43) Veröffentlichungstag der PCT Anmeldung
in deutscher Übersetzung: **29.03.2001**
(45) Veröffentlichungstag
der Patenterteilung: **29.07.2010**

(51) Int Cl.⁸: **G06F 12/00** (2006.01)

Innerhalb von drei Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

(30) Unionspriorität:
09/023,234 13.02.1998 US

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

(74) Vertreter:
ZENZ Patent- und Rechtsanwälte, 45128 Essen

(72) Erfinder:
Holman, Thomas J., Portland, Oreg., US

(56) Für die Beurteilung der Patentfähigkeit in Betracht
gezogene Druckschriften:
US 57 15 207 A
US 57 01 438 A
US 40 99 231 A

Richard Crisp: "High Bandwidth RDRAM Technology Reduces System Cost", Proceedings of COMPCON'96, IEEE, 25-28. Februar 1996, S. 365-377

Richard Crisp: "Direct Rambus Technology: The new main memory standard", IEEE Micro, November/December 1997, pages 18-28

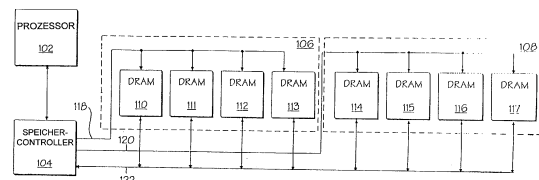
Zacharenia Xanthaki: "A Memory Controller for Access Interleaving over a single Rambus", FORTH-ICS/TR-124, July 1994

Dave Bursky: "Memory Systems Design and Applications", Hayden Book Company, Inc., 1980, ISBN 0-8104-0980-1, S. 213-220

(54) Bezeichnung: **Speichersystem mit jeweils einen Speichermodul-Controller enthaltenden Speichermodulen**

(57) Hauptanspruch: Speichersystem (300) mit einem mit einem Prozessor (302) gekoppelten System-Speicher-Controller (304) und mehreren über einen System-speicherbus (323, 1023, 1242) mit dem System-Speicher-Controller (304) verbundenen Speichermodulen (306, 308; 500; 600; 700), wobei jedes Speichermodul (306, 308; 500; 600; 700) eine Baueinheit mit einer Mehrzahl von Speicherbauelementen (312–315, 317–320; 1204–1207), die auf einer oder beiden Seiten einer Platine des Moduls angeordnet sind, einer Schnittstelle mit einem Verbinder zu dem Systemspeicherbus (323, 1023, 1242) und mit Signalleitungen (330–344; 1208–1227, 1238, 1240), die die Speicherbauelemente mit der Schnittstelle verbinden, bildet,
wobei der Systemspeicherbus (323, 1023, 1242) mit der Direct-Rambus-Busarchitektur kompatibel ist, dadurch gekennzeichnet, dass die Speicherbauelemente (312–315, 317–320; 1204–1207) SDRAM-Bauelemente sind, dass die Schnittstelle zu dem Systemspeicherbus (323,

1023, 1242) von einem mit den Signalleitungen gekoppelten und auf dem Modul angeordneten Speichermodul-Controller (310, 316; 1000; 1202) gebildet ist, und dass der Speichermodul-Controller (310, 316; 1000; 1202) aufweist:...



Beschreibung

[0001] Die vorliegende Erfindung bezieht sich auf ein Speichersystem mit einem mit einem Prozessor gekoppelten System-Speicher-Controller und mehreren über einen Systemspeicherbus mit dem System-Speicher-Controller verbundenen Speichermodulen, wobei jedes Speichermodul eine Baueinheit mit einer Mehrzahl von Speicherbauelementen, die auf einer oder beiden Seiten einer Platine des Moduls angeordnet sind, einer Schnittstelle mit einem Verbinder zu dem Systemspeicherbus und mit Signalleitungen, die die Speicherbauelemente mit der Schnittstelle verbinden, bildet, wobei der Systemspeicherbus mit der Direct-Rambus-Busarchitektur kompatibel ist.

[0002] Speichermodule, wie beispielsweise Single-in-line-Speichermodule (SIMMs) und Dual-in-line-Speichermodule (DIMMs) werden üblicherweise verwendet, um Daten, Befehle und andere Informationen in Computern oder anderen digitalen Systemen zu speichern. [Fig. 1](#) zeigt ein typisches Speichersystem, bei welchem ein Prozessor **102** mit dynamischen Speicherbauelementen mit wahlfreiem Zugriff (DRAM-Bauelementen) **110–117** auf Speichermodulen **106** bzw. **108** über einen Speicher-Controller **104** kommuniziert. Der Speicher-Controller **104** übermittelt geeignete Speicherbefehle (z. B. Schreiben, Lesen, Auffrischen, etc.) an das Speichermodul **106** über den Adress- und Kommandobus **118** und an das Speichermodul **108** über den Adress- und Kommandobus **120**. Daten werden aus dem Speicher-Controller **104** zu beiden Modulen über den Bus **122** übertragen.

[0003] Die Kopplung verschiedenartiger Speichermodule mit einem System-Speicher-Controller ist beispielsweise aus der Patentschrift US 5,701,438 A bekannt.

[0004] Aus dem Artikel von A. Aldereguia und D. Cromer: "Integrated Controller with Fine-Tuned Memory Timings" in IBM Technical Disclosure Bulletin, Vol. 36, No. 9, September 1993, ist die Einkopplung eines Speicher-Redrive-Moduls zwischen einem Speichercontroller und SIMM-Speichermodulen bekannt, wobei dieses Speicher-Redrive-Modul Verzögerungsleitungen zum Verzögern von Speichersteuersignalen (z. B. RAS und CAS) enthält. Beispielsweise werden die Flanken des RAS-Signals derart verzögert, dass weniger Takte für ein Voraufladen erforderlich sind.

[0005] Es gibt wenigstens zwei den herkömmlichen Speichersystemen gemäß [Fig. 1](#) anhaftende Nachteile: (1) Die Busse **118**, **120** und **122** sind Mehrfach-Abwurf-Busse (multi-drop buses), die hohe kapazitive Lasten aufweisen, welche große Treiber im Speicher-Controller **104** und in den DRAMs **110–117** (zum Treiben des Busses **122**) erfordern; und (2) es besteht eine Neigung zu einer großen FehlAbstimmung hinsichtlich der Belastung zwischen den Adress- und Kommandobussen und dem Datenbus **122**. Diese Nachteile wirken zusammen und verringern die maximale Betriebsfrequenz, erhöhen den Energieverbrauch und verringern die Leistung des Systems. Zusätzlich kann es sein, dass der Speicher-Controller **104** eine hohe Anzahl von Pins (z. B. etwa 190 Pins für ein 4-DIMM-SDRAM-System) enthält, um den Zugriff auf die Speichermodule **106** und **108** zu unterstützen. Die Busbreite des Datenbusses **122** beträgt oftmals 64 oder 72, um größere Bandbreiten, beispielsweise bis zu 100 Mega Übertragungen pro Sekunde (MT/s), zu unterstützen.

[0006] [Fig. 2](#) zeigt ein anderes typisches Speichersystem, das eine Technologie der Rambus, Inc. aus Mountain View, Kalifornien, verwendet. Bei diesem System kommuniziert der Prozessor **202** mit dynamischen Rambus-Speicherbauelementen mit wahlfreiem Zugriff (RDRAMTM-Bauelementen) **210–217** auf Speichermodulen **206** und **208** (auch Rambus-in-line-Speichermodule oder RIMMTM-Module genannt) über einen Speicher-Controller **204**. Der Speicher-Controller **204** übermittelt geeignete Speicherbefehle (z. B. Schreiben, Lesen, Auffrischen, etc.) und Daten in einer zeitlich gemultiplexten Weise zu den Speichermodulen **206** und **208** über einen speicherspezifischen Bus **226**. Jedes der RDRAMTM-Bauelemente **210–217** enthält eine Schnittstellenlogik **218–225**, die es den RDRAMTM-Bauelementen ermöglicht, eine Schnittstelle zum speicherspezifischen Bus **226** zu bilden. Der speicherspezifische Bus **226** kann bei einer hohen Frequenz (z. B. 250–400 MHz) arbeiten, weist eine geringe Busbreite (z. B. 16 oder 18 Signalleitungen) auf, weist eine symmetrische Belastung für Adress-, Kommando- und Dateninformationen auf und hat eine Bandbreite von bis zu 800 MT/s. Bei anderen Ausführungsbeispielen können die Adress- und Steuerinformationen von dem Datenbus bei dieser Umgebung getrennt (demultiplexed) sein.

[0007] RDRAM-Speichermodule sind beispielsweise in dem Artikel von Richard Crisp, "High Bandwidth RDRAM Technology Reduces System Cost", in IEEE Proceedings of COMPCON, 1996, Seiten 365–377, oder in dem Artikel von Richard Crisp, "Direct Rambus Technology: The New Main Memory Standard", in IEEE Micro, November/Dezember 1997, Seiten 18–28, beschrieben.

[0008] Das System gemäß [Fig. 2](#) weist eine Reihe von Nachteilen auf. Jedes RDRAM™-Bauelement enthält eine beträchtliche Menge an Logik in der Schnittstellenlogik, welche die Implementierung der RDRAM™-Bauelemente schwieriger und teurer macht als die anderer DRAM-Bauelemente und welche bewirkt, dass die RDRAM™-Bauelemente mehr Leistung ziehen und mehr Wärme erzeugen als dann, wenn die Logik nicht vorhanden wäre. Dies kann zu thermischen und Zuverlässigkeitsproblemen führen. Zusätzlich enthält jeder RDRAM™ eine Verzögerungsverriegelungsschleifen(DLL)-Schaltung, die mit einer Mehrzahl von Taktsignalen auf den Bus **226** gekoppelt ist. Die DLL-Schaltungen sind üblicherweise immer in Betrieb und ziehen eine beträchtliche Menge Energie, was zu thermischen und Zuverlässigkeitsproblemen beiträgt. Darüber hinaus muß der Speicher-Controller **204** relativ große Treiber enthalten, um den Bus **226** anzusteuern.

[0009] Bei jedem der in den [Fig. 1](#) und [Fig. 2](#) beschriebenen Speichersysteme muss der Speicher-Controller so konstruiert sein, dass er die richtigen Speichertransaktionen zu den Speicherbauelementen in einem vorgegebenen Format und zu vorgegebenen Zeiten bereitstellt. Die Speicherbauelemente und der Speicher-Controller müssen so konstruiert sein, dass sie miteinander arbeiten können. Aufgabe der Erfindung ist es, das Design der Speicherbauelemente und des Speicher-Controllers derart zu entkoppeln, dass unabhängige Fortschritte in der jeweiligen Technologie gemacht werden können, und darüber hinaus, den Betrieb der Speicherbauelemente von dem des Speicher-Controllers zu entkoppeln, um die von jedem Speicherbauelement und von dem gesamten Speichersystem gezogene Leistung zu verringern.

[0010] Diese Aufgabe wird erfindungsgemäß durch ein Speichersystem mit den Merkmalen des Anspruchs 1 gelöst.

[0011] Vorteilhafte und bevorzugte Weiterbildungen der Erfindung sind in den Unteransprüchen gekennzeichnet.

[0012] Die Merkmale und Vorteile der vorliegenden Erfindung werden beispielhaft in den Figuren der begleitenden Zeichnungen veranschaulicht.

[0013] [Fig. 1](#) ist ein Blockschaltbild eines Ausführungsbeispiels eines herkömmlichen Speichersystems.

[0014] [Fig. 2](#) ist ein Blockschaltbild eines anderen Ausführungsbeispiels eines herkömmlichen Speichersystems.

[0015] [Fig. 3](#) ist ein Blockschaltbild eines Ausführungsbeispiels eines Speichersystems gemäß der vorliegenden Erfindung.

[0016] [Fig. 4](#) ist ein Blockschaltbild eines anderen Ausführungsbeispiels eines Speichersystems gemäß der vorliegenden Erfindung.

[0017] [Fig. 5](#) ist ein Blockschaltbild eines Ausführungsbeispiels eines dynamischen Speichermoduls mit wahlfreiem Zugriff (DRAM-Moduls), das einen DRAM-Speichermodul-Controller enthält.

[0018] [Fig. 6](#) ist ein Blockschaltbild eines statischen Speichermoduls mit wahlfreiem Zugriff (SRAM-Moduls) ist, das einen SRAM-Speichermodul-Controller enthält.

[0019] [Fig. 7](#) ist ein Blockschaltbild eines nicht-flüchtigen Speichermoduls ist, das einen Speichermodul-Controller enthält.

[0020] [Fig. 8](#) ist ein Blockschaltbild eines Speichermodul-Controllers, der mit einem zeitlich gemultiplexten Systemspeicherbus gekoppelt ist.

[0021] [Fig. 9](#) ist ein Ablaufdiagramm eines von dem Speichermodul-Controller gemäß [Fig. 8](#) implementierten Prozesses.

[0022] [Fig. 10](#) ist ein Blockschaltbild eines Ausführungsbeispiels des erfindungsgemäßen Speichermodul-Controllers gemäß [Fig. 3](#), der mit einem demultiplexten Systemspeicherbus gekoppelt ist.

[0023] [Fig. 11](#) ist ein Ablaufdiagramm eines Ausführungsbeispiels des von dem Speichermodul-Controller gemäß [Fig. 10](#) implementierten Prozesses.

[0024] [Fig. 12](#) ist ein Blockschaltbild eines Ausführungsbeispiels des erfindungsgemäßen Speichermoduls, das einen Speichermodul-Controller enthält, der eine Schnittstelle zwischen SDRAM-Bauelementen und einem ein Direct Rambus™-Protokoll verwendenden Systemspeicherbus bildet.

[0025] [Fig. 13](#) ist ein Ablaufdiagramm eines Ausführungsbeispiels einer Lesetransaktion, die von dem Speichermodul-Controller gemäß [Fig. 12](#) implementiert wird.

[0026] [Fig. 14](#) ist ein Beispielzeitdiagramm der Lesetransaktion, die von dem Speichermodul-Controller gemäß [Fig. 12](#) implementiert wird.

[0027] [Fig. 15](#) ist ein Ablaufdiagramm eines Ausführungsbeispiels einer Schreibtransaktion, die von dem Speichermodul-Controller gemäß [Fig. 12](#) implementiert wird.

[0028] [Fig. 16](#) ist ein Beispielzeitdiagramm der Schreibtransaktion, die von dem Speichermodul-Controller gemäß [Fig. 12](#) implementiert wird.

[0029] [Fig. 17](#) ist ein Beispielzeitdiagramm einer Lesetransaktion, die von dem Speichermodul-Controller gemäß [Fig. 12](#) unter Verwendung eines modifizierten Protokolls implementiert wird.

[0030] [Fig. 18](#) ist ein Beispielzeitdiagramm einer Schreibtransaktion, die von dem Speichermodul-Controller gemäß [Fig. 12](#) unter Verwendung eines modifizierten Protokolls implementiert wird.

[0031] Ein Speichersystem, das eine verteilte Steuerung der Speicherbauelemente auf Speichermodulen enthält, wird beschrieben. Die vorliegende Erfindung umfasst ein Speichermodul-Controller-Bauelement auf jedem Speichermodul in dem System. Der Speichermodul-Controller kommuniziert mit dem System-Speicher-Controller über einen Systemspeicherbus und auf der Modulebene mit den einzelnen Speicherbauelementen. Der Systemspeicherbus kann ein Bus mit einer geringen Pin-Anzahl und einer hohen Frequenz sein. Der Speichermodul-Controller kann mit den einzelnen Speicherbauelementen über nicht gemultiplexte Signalleitungen größerer Anzahl und geringerer Frequenz kommunizieren. Da der System-Speicher-Controller direkt nur mit den Speichermodul-Controllern kommuniziert, kann die Belastung des Systemspeicherbusses und darüber hinaus die Größe der Bustreiber in dem System-Speicher-Controller reduziert werden.

[0032] Der Speichermodul-Controller auf jedem Speichermodul ist die Schnittstelle zwischen dem System-Speicher-Controller und den einzelnen Speicherbauelementen auf den Modulen. Diese Architektur entkoppelt die einzelnen Speicherbauelemente von dem Systemspeicherbus und dem System-Speicher-Controller. Dies kann die unabhängige Entwicklung der Speicherbauelementtechnologie ermöglichen. Beispielsweise können die Speicherbauelemente so weiterentwickelt werden, dass sie schneller, breiter, bei abweichenden Betriebsversorgungsspannungen oder mit verringerten Spannungshüben betreibbar sind gegenüber dem Fall, bei dem die Speicherbauelemente direkt mit dem System-Speicher-Controller kommunizieren würden.

[0033] Das Speichermodul kann seine eigene Speicheradressleitungen, Steuerleitungen und Datenleitungen zwischen dem Speichermodul-Controller und den einzelnen Speicherbauelementen aufweisen. Die Signalleitungen für diese Punkt-zu-Punkt- oder Busverbindungen können signifikant kürzer als die Systemverbindungsbusse sein, die bei herkömmlichen Speichersystemen verwendet werden. Dies kann eine verringerte Belastung, verbesserte Kontrolle der Kapazität an den Adress- und Datenleitungen ermöglichen und die maximale Betriebsfrequenz des Moduls erhöhen, während die von dem System gezogene Leistung reduziert wird. Darüber hinaus kann die Leistung weiter dadurch reduziert werden, dass die DLLs (Delay Locked Loops) aus den einzelnen Speicherbauelementen fortgelassen werden. Ein DLL kann in dem Speichermodul-Controller enthalten sein, wenn er zur Schnittstellenbildung zu dem Systemspeicherbus erforderlich ist.

[0034] [Fig. 3](#) ist ein Blockschaltbild eines Ausführungsbeispiels der vorliegenden Erfindung. Das System **300** enthält einen Prozessor **302**, einen System-Speicher-Controller **304** und Speichermodule **306** und **308**. Der Prozessor **302** kommuniziert mit dem System-Speicher-Controller **304**. Der Prozessor **302** kann eine beliebige Steuerlogik oder ein Mikroprozessor sein, die bzw. der Schreib-, Lese- und andere Speichertransaktionen zu dem System-Speicher-Controller **304** übermittelt. Der System-Speicher-Controller **304** übermittelt die Speichertransaktionen zu den Speichermodul-Controllern **310** und **316** der Speichermodule **306** bzw. **308** über den Systemspeicherbus **323**. Der System-Speicher-Controller **304** kann Teil eines Chipsatzes für beispielsweise einen Personalcomputer oder eine andere unabhängige Logik sein.

[0035] Die Speichermodule **306** und **308** können SIMMs, DIMMs, RIMM™-Module oder andere Arten von

Speichermodulen sein. Jedes Speichermodul enthält einen Speichermodul-Controller und eines oder mehrere Speicherbauelemente. Beispielsweise enthält das Speichermodul **306** den Speichermodul-Controller **310** und Speicherbauelemente **312–315**, und das Speichermodul **308** enthält einen Speichermodul-Controller **316** und Speicherbauelemente **317–320**. Wenn auch das veranschaulichte System **300** zwei Speichermodule enthält, kann eine beliebige Anzahl von Speichermodulen verwendet werden. Die Speicherbauelemente können auf einer Seite des Moduls, auf beiden Seiten des Moduls angeordnet sein und/oder sie können aufeinander gestapelt sein. Das Modul kann auch ein Mehr-Chip-Modul sein. Bei einem Ausführungsbeispiel kann ein Speichermodul eine ungefähre Höhe **350** von 1 bis 2 Zoll und eine ungefähre Länge **352** von 4 bis 6 Zoll haben.

[0036] Die Speichermodule **310** und **316** tauschen Speichertransaktionen mit dem System-Speicher-Controller **304** über den Systemspeicherbus **323** aus. Der Systemspeicherbus **323** kann ein Bus geringer Pin-Anzahl (beispielsweise etwa 16 bis 35 Adress-, Daten- und Steuersignalleitungen) sein, der bei einer hohen Frequenz arbeitet. Bei einem Ausführungsbeispiel kann der Systemspeicherbus **323** bei einer Frequenz von etwa 200–500 MHz betrieben werden. Andere Frequenzen können verwendet werden. Der Systemspeicherbus **323** umfasst einen Kommando/Adress-Bus **324** und einen Datenbus **328**. Der Kommando/Adress-Bus **324** kann Kommandos und Adressinformationen auf einer oder mehreren der Signalleitungen zeitlich multiplexen. Der Systemspeicherbus **323** kann optional einen Taktbus **322** enthalten, der eines oder mehrere Taktsignale auf den gleichen oder separaten Signalleitungen übertragen kann und einen Handshaking-Bus **326**, der ein Handshaking-Signal zwischen dem System-Speicher-Controller **304** und Speichermodul-Controllern **310** und **316** führen kann. Das Handshaking-Signal kann von dem System-Speicher-Controller **304** beispielsweise beim Start eines Speicheranforderungspakets oder dann, wenn er gültige Daten auf dem Datenbus **328** übermittelt, erzeugt werden. Das Handshaking-Signal kann von den Speichermodul-Controllern **310** und **316** erzeugt werden, wenn sie zum Datenbus **328** gültige Daten bereitstellen, die von dem System-Speicher-Controller **304** angefordert worden sind.

[0037] Der Systemspeicherbus **323** ist mit der Direct Rambus™-Busarchitektur kompatibel, welche 16 oder 18 Datenbits auf dem Bus **328**, ein oder mehrere Taktsignale auf dem Bus **322** und Kommando- und Adressinformationen auf dem Bus **324** enthält. Die Taktsignale bei diesem Ausführungsbeispiel können bei einer Frequenz von etwa 400 MHz laufen, was eine Datenübertragungsrate von bis zu 800 MT/s ermöglicht. Bei diesem Ausführungsbeispiel kann der System-Speicher-Controller **304** ein beliebiger Controller sein, der in der Lage ist, das Direct Rambus™-Protokoll für Speichertransaktionen auf dem Systemspeicherbus **323** zu implementieren. Zusätzlich kann der Systemspeicherbus **323** Rambus-Signalisierungspegel (RSL) für die auf dem Bus übertragenen Signale verwenden. Bei alternativen Ausführungsbeispielen können andere Arten geeigneter Signalisierungspegel verwendet werden, einschließlich TTL-, CMOS-, GTL-, SSTL-, CTT- oder anderen Signalisierungspegeln.

[0038] Der Systemspeicherbus **323** kann physisch auf einer gedruckten Schaltungsplatine (PCB) angeordnet sein, die den System-Speicher-Controller **304** und Verbindungssteckplätze für die Module **306** und **308**, wie sie in [Fig. 3](#) gezeigt sind, enthält. Alternativ kann der Systemspeicherbus **323** durch separate Kanäle der Speichermodule **306** und **308** geführt sein, wie beispielsweise die Kanäle **402**, **404**, **406** und **408**, die in [Fig. 4](#) gezeigt sind.

[0039] Die Speichermodul-Controller **310** und **316** arbeiten als Brücken zwischen dem Systemspeicherbus **323**, der in einem Protokoll oder Format betrieben wird, und lokalen oder Speichermodulbussen (z. B. Leitungen **330**, **332** und **334–337**), die bei einem zweiten abweichenden Protokoll oder Format betrieben werden. Jedes Speichermodul enthält die Steuerlogik, die erforderlich ist, um die Speichertransaktion auf dem Systemspeicherbus **323** zu interpretieren und diese Transaktion in die richtigen Steuer-, Adress- und Datensignale für seine Speicherbauelemente auf dem Speichermodul zu übersetzen. In ähnlicher Weise interpretieren die Speichermodule die Speichertransaktionen auf dem lokalen oder Speichermodulbus in das Format, das zum Übermitteln der Transaktion an den System-Speicher-Controller **304** über den Systemspeicherbus **323** erforderlich ist.

[0040] Die Speichermodul-Controller **310** und **316** können außerdem eine Schaltung enthalten, die erforderlich ist, um Auffrischoperationen der dynamischen Speicherbauelemente, Vorabrufoperationen, Fehlerkorrekturfunktionen, die beispielsweise ECC-Bits verwenden, durchzuführen, ferner eine Stromkalibrierung, sofern der Systemspeicherbus **323** ein Strombus ist, eine Seriell-Vorhanden-Erfassungsschaltung, eine DLL-Schaltung, eine Stromversorgungsmanagementschaltung, die die Taktsignale oder Taktfreigabesignale für die Speicherbauelemente abschalten oder andere Herunterschaltungen, Programmier- und Löschespannungen für nicht-flüchtige Speicherschaltungen bereitstellen kann, und/oder eine Levelizing-Schaltung zum Erzeugen von Signalen auf dem Systemspeicherbus **323** in der richtigen Taktdomäne (sofern es mehr als eine Takt-

domäne auf dem Systemspeicherbus **323** gibt).

[0041] Jedes Speicherbauelement auf dem Speichermodul übermittelt Speichertransaktionen zu seinem Speichermodul-Controller über Adress-, Daten- und Steuersignale. Speichertransaktionen sind solche Transaktionen, die für den jeweiligen Typ des verwendeten Speicherbauelements geeignet sind. Beispielsweise können die Speichertransaktionen ein Schreiben, Lesen, Auffrischen, Vorabrufen, Herunterschalten, Rücksetzen, Löschen und dergleichen umfassen. Das Speichermodul **306** beispielsweise kann Speicheradressen zu den Speicherbauelementen **312–315** über Signalleitungen **330**, Steuerinformationen (z. B. Chip-Freigabesignale, Schreibfreigabesignale, RAS, CAS, Ausgabefreigabesignale, etc.) über Signalleitungen **332** und Daten über Punkt-zu-Punkt-Verbindungen **334–337** übermitteln. Jede der Signalleitungen **330**, **332** und **334–337** kann eine oder mehrere Signalleitungen umfassen. Die Signalleitungen **330** und **332** können Busse sein, die mit sämtlichen vier Speicherbauelementen **312–315** oder mit Gruppen von Speicherbauelementen (z. B. zwei gleichzeitig) verbunden sind oder sie können Punkt-zu-Punkt-Verbindungen zum Speichermodul-Controller **310** sein. In ähnlicher Weise kann das Speichermodul **308** Speicheradressen zu den Speicherbauelementen **317–320** über Signalleitungen **338**, Steuerinformationen über die Signalleitungen **340** und Daten über Punkt-zu-Punkt-Verbindungen **341–344** übermitteln. Jede dieser Signalleitungen **338**, **340** und **341–344** kann eine oder mehrere Signalleitungen umfassen.

[0042] Bei einem Ausführungsbeispiel kann die Anzahl der Signalleitungen **334–337** die gleiche sein wie die Anzahl der Daten führenden Signalleitungen des Systemspeicherbusses **323**. Bei einem anderen Ausführungsbeispiel kann die Anzahl der Signalleitungen **334–337** zusammen größer sein als die Anzahl der Daten führenden Signalleitungen aus dem System-Speicher-Controller **304** zum Speichermodul-Controller **310** auf dem Systemspeicherbus **323**. Dies kann die Bandbreite der Datenübertragung auf dem Speichermodul erhöhen. Beispielsweise kann der Systemspeicherbus **323** 16 Datensignale parallel auf dem Bus **328** führen, und der Speichermodul-Controller **310** kann gleichzeitig 16 Datensignale jedem der Speicherbauelemente **312–315** zur Verfügung stellen. Da die Datenbusbreite breiter sein kann (z. B. 64 oder 72 Bits breit) als die des Systemspeicherbusses **323** (z. B. 16 Bits breit), kann es sein, dass der Speichermodul-Controller **310** ein Taktsignal auf dem Bus **330** zu den Speicherbauelementen **312–315** liefert, das eine geringere Frequenz aufweist als das auf dem Systemspeicherbus **323** laufende. Bei einem Ausführungsbeispiel kann die Taktfrequenz auf dem Bus **330** beispielsweise etwa 50 bis 200 MHz sein, wenn die Frequenz des Taktsignals auf dem Taktbus **322** etwa 100 bis 400 MHz beträgt. Die Frequenz des Taktsignals auf dem Taktbus **322** kann darüber hinaus doppelt gepumpt sein, d. h. eine Aktion (z. B. ein Laden von Daten) auf jeder Taktflanke ausführen.

[0043] Da man lokale Adress-, Steuer- und Datenleitungen auf den Speichermodulen hat, welche von dem Systembus **323** entkoppelt sind, kann die Länge dieser Signale gegenüber denjenigen auf dem Systemspeicherbus reduziert, die Belastung dieser Leitungen gegenüber der des Systemspeichers verringert und die Belastung genauer zwischen den verschiedenen Leitungen kontrolliert werden, so dass man die symmetrische Belastung zwischen den Signalleitungen exakter erreicht. Beispielsweise kann die Belastung der Datenleitungen und der Adress- und Steuerleitungen etwa die gleiche Belastung (z. B. etwa 10–40 pF) plus oder minus etwa 10% sein. Dies kann in vorteilhafter Weise die durch das Modul gezogene Leistung absenken und die maximale Betriebsfrequenz des Moduls für eine gegebene Speichertransaktion erhöhen.

[0044] Außerdem können durch Entkopplung der Speicherbauelemente von dem Systemspeicherbus **323** und dem System-Speicher-Controller **304** die Speicherbauelemente **312–315** und **317–320** bei verringerten oder abweichenden Stromversorgungsspannungen gegenüber denjenigen betrieben werden, die dem Rest des Systems **300** zur Verfügung gestellt werden. Beispielsweise können die Speicherbauelemente **312–315** und **317–320** (und/oder die Speichermodul-Controller **310** und **316**) mit Stromversorgungen von etwa 1,8 Volt bis 2,5 Volt betrieben werden, während der Rest der Bauelemente im System **300** bei 3,3 Volt oder 5,0 Volt betrieben wird. Dies kann ebenfalls dazu dienen, die von dem Modul und somit von dem System **300** gezogene Leistungsmenge zu senken. Darüber hinaus kann der Spannungshub der Signale auf den Leitungen **330**, **332** und **334–337** vorteilhafterweise so ausgewählt werden, dass er klein (z. B. etwa 1,0 bis 2,0 Volt) ist, um die von den Speicherbauelementen **312–315** gezogene Leistung weiter zu reduzieren.

[0045] Die Leistung kann für jedes Speicherbauelement und für das System im Ganzen weiter reduziert werden, wenn eine DLL erforderlich ist, um eine Schnittstelle zum Systemspeicherbus **323** zu bilden. Die DLL kann in dem Speichermodul-Controller **310** statt in jedem der Speicherbauelemente **312–315**, wie das bei herkömmlichen Systemen erforderlich gewesen sein kann, enthalten sein.

[0046] [Fig. 5](#) ist ein Blockschaltbild eines Speichermoduls **500**, das ein als DRAM-Speichermodul konfigurierbares Ausführungsbeispiel des Speichermoduls **306** ist. Das Modul **500** enthält einen DRAM-Speichermodul-

dul-Controller **510**, der eine Schnittstelle zwischen DRAM-Bauelementen **512–515** und dem Systemspeicherbus **323** zur Verfügung stellt. Der DRAM-Speichermodul-Controller **510** übersetzt aus dem Systemspeicherbus **323** empfangene Speichertransaktionen und erzeugt DRAM-Speicheroperationen (z. B. Schreiben, Lesen, Vorabrufen, Auffrischen), die Adressen auf dem Adressbus **516**, Datensignale auf Datenleitungen **523–526** und Steuersignale, wie beispielsweise Schreibfreigabe WE auf Leitung **518**, RAS auf Leitung **520** und CAS auf Leitung **522**, einschließen. Der DRAM-Speichermodul-Controller **510** kann darüber hinaus zusätzliche Steuersignale an die DRAMs **512–515**, die Taktsignale für synchrone Operationen (d. h. für SDRAMs), Speicherbankauswahlsignale und/oder Chipauswahl- oder Chipfreigabesignale einschließen, bereitstellen. Der DRAM-Speichermodul-Controller kann darüber hinaus Auffrischkommandos, Vorabrufkommandos und/oder Energieverwaltungsmanagementkommandos an die DRAMs **512–515** zur Verfügung stellen.

[0047] [Fig. 6](#) ist ein Blockschaltbild eines Speichermoduls **600**, das als SRAM-Speichermodul konfiguriert ist. Das Modul **600** enthält einen SRAM-Speichermodul-Controller **610**, der eine Schnittstelle zwischen SRAM-Bauelementen **612–615** und dem Systemspeicherbus zur Verfügung stellt. Der SRAM-Speichermodul-Controller **610** übersetzt aus dem Systemspeicherbus empfangene Speichertransaktionen und erzeugt SRAM-Speicheroperationen (z. B. Schreiben, Lesen, Zurücksetzen, Herunterschalten), die Adressen auf dem Adressbus **616**, Datensignale auf den Datenleitungen **623–626** und Steuersignale, wie beispielsweise Chipauswahl oder Chipfreigabe CE auf Leitung **618**, Schreibfreigabe WE auf Leitung **620** und Ausgabefreigabe auf Leitung **622**, einschließen. Der SRAM-Speichermodul-Controller **610** kann darüber hinaus zusätzliche Steuersignale an die SRAMs **612–615** bereitstellen, die Taktsignale für synchrone Operationen, Burst-Steuersignale, Interrupt-Signale, Rücksetz-Signale, Schreib- und Lesesignale (z. B. für ein FIFO-Bauelemente), Stromversorgungsmanagementsignale, Byte-Freigabesignale und/oder Expansionsignale einschließen.

[0048] [Fig. 7](#) ist ein Blockschaltbild eines Speichermoduls **700**, das als nicht-flüchtiges Speichermodul konfiguriert ist. Das Modul **700** enthält einen Speichermodul-Controller **710**, der eine Schnittstelle zwischen nicht-flüchtigen Bauelementen **712–715** und dem Systemspeicherbus bereitstellt. Der Speichermodul-Controller **710** für den nicht-flüchtigen Speicher übersetzt aus dem Systemspeicherbus empfangene Speichertransaktionen und erzeugt Speicheroperationen für nicht-flüchtige Speicher (z. B. Schreiben oder Programmieren, Lesen, Herunterschalten, Löschen, etc.), die Adressen auf dem Adressbus **716**, Datensignale auf den Datenleitungen **723–726** und Steuersignale einschließen, wie beispielsweise Chipauswahl- oder Chipfreigabe CE auf Leitung **718**, Schreibfreigabe WE auf Leitung **720**, Ausgabefreigabe auf Leitung **722** und eine Programmiervoltage auf Leitung **728**. Der Speichermodul-Controller **710** für nicht-flüchtigen Speicher kann darüber hinaus zusätzliche Steuersignale an die nicht-flüchtigen Speicherbauelementen **712–715** zur Verfügung stellen, einschließlich Taktsignalen für eine synchrone Operation, Burst-Steuersignalen, Rücksetzsignalen, Stromversorgungsmanagementsignalen oder anderen Signalen oder Kommandos.

[0049] Es wird wieder auf [Fig. 3](#) Bezug genommen; der System-Speicher-Controller **304** und die Speichermodul-Controller **310** und **316** kommunizieren miteinander unter Verwendung eines Protokolls. Das Protokoll ist nicht-gemultiplext, wie es in [Fig. 3](#) gezeigt ist, wobei die Daten über einen separaten Datenbus getrennt von Adress- und Steuerinformationen übermittelt werden.

[0050] [Fig. 8](#) ist ein Blockschaltbild des Speichermodul-Controllers **800**, der ein zeitlich gemultiplextes Protokoll verwendet. Beispiele eines zeitlich gemultiplexten Protokolls sind die Base and Concurrent Rambus™-Protokolle.

[0051] Der Speichermodul-Controller **800** bildet eine Schnittstelle zu dem Systemspeicherbus **823**, der eine oder mehrere Taktsignalleitungen **824**, einen Transaktionsbus **826** enthält und optional ein Gültig-Signal **828** enthält. Der Transaktionsbus **826** ist ein Mehrzweckbus, der Adress-, Daten- und Steuerinformationen für eine Speichertransaktion führen kann. Das Gültig-Signal **828** stellt einen oder mehrere Handshaking-Signale dar, die für ein entkoppeltes Protokoll verwendet werden können, um den Start einer Speichertransaktion anzuzeigen oder um anzuzeigen, wenn gültige Daten auf dem Transaktionsbus **826** vorhanden sind. Das Gültig-Signal **828** kann durch eine Handshake-Logik **806** unter der Steuerung der Steuerlogik **802** überwacht und erzeugt werden. Das Gültig-Signal **828** kann bei einem gekoppelten Protokoll fortgelassen werden. Der Speichermodul-Controller **800** bildet darüber hinaus eine Schnittstelle zu Speicherbauelementen auf einem Speichermodul, indem er ein Taktsignal **830**, Adresssignale **832**, Steuersignale **834** und Datensignale **836** zur Verfügung stellt. Das Taktsignal kann bei asynchronen Speicherbauelementen fortgelassen werden.

[0052] Der Speichermodul-Controller **800** enthält eine Anforderungsbehandlungslogik **804** zur Bildung einer Schnittstelle zu dem Transaktionsbus **826**. Die Anforderungsbehandlungslogik **804** kann eine Deserialisierungslogik enthalten, welche die gemultiplexten Steuer-, Adress- und Dateninformationen, die auf dem Trans-

aktionsbus **826** bereitgestellt werden, trennt und diese Signale der Steuerlogik **802** über die Leitungen **838**, **840** bzw. **842** zur Verfügung stellt. Die Anforderungsbehandlungslogik **804** kann außerdem eine Serialisierungslogik enthalten, die Steuer-, Adress- und Dateninformationen auf den Leitungen **838**, **840** bzw. **842** in zeitlich gemultiplexte Serien von Signalen serialisiert, die dem Transaktionsbus **826** zur Verfügung gestellt werden sollen.

[0053] Die Steuerlogik **802** ist die Intelligenz des Speichermodul-Controllers **800** und erzeugt die geeigneten Adress-, Steuer- und Datensignale für die Speichermodul-Speicherbauelemente in Erwiderung der aus der Anforderungsbehandlungslogik **804** empfangenen Signale. Jedes Speichermodul-Controller-Bauelement in jedem Modul kann eine unterschiedliche Steuerlogik enthalten, die eine spezielle Übersetzung zwischen den Signalarten und dem Protokoll auf dem Systemspeicherbus und den von den Speicherbauelementen auf dem Speichermodul erwarteten speziellen Speichersignalen und speziellen Protokoll implementiert. Beispielsweise kann die Steuerlogik **802** die geeigneten Adresssignale der Adressschnittstellenschaltung **808** über Leitungen **844**, die geeigneten Steuersignale der Steuerschnittstellenschaltung **820** über die Leitungen **846** und die geeigneten Datensignale der Daten-I/O-Schaltung **822** über die Leitungen **848** zur Verfügung stellen. Die Schnittstellenschaltungen können Puffer- und Registerelemente enthalten, um Adressen **832**, Speichersteuersignale **834** und Daten **836** anzusteuern.

[0054] Die Schnittstellenschaltungen **818**, **820** und **822** können durch ein von einem Taktgenerator **810** erzeugtes Taktsignal getaktet werden. Der Taktgenerator **810** kann darüber hinaus ein Taktsignal der Steuerlogik **802** und den Taktpuffern **816** zur Verfügung stellen, welche das Taktsignal **830** und/oder Taktfreigabesignale an die Speicherbauelemente auf einem Speichermodul zur Verfügung stellen. Der Taktgenerator **810** kann Taktsignale in Abhängigkeit von einem aus dem Systemspeicherbus **823** zur Verfügung gestellten Takt **824** erzeugen. CLK **830** kann eine andere Frequenz als Takt **824** haben. Der Speichermodul-Controller **800** oder Taktgenerator **810** kann ferner eine DLL, Taktpuffer oder Takteilerschaltung enthalten, die den Takt **824** formen oder verändern, bevor er den Taktgenerator **810** erreicht.

[0055] Der Speichermodul-Controller **800** kann darüber hinaus eine Stromversorgungsmanagereinheit **808** enthalten, die unter der Steuerung der Steuerlogik **802** den Taktgenerator **810** freigeben oder sperren kann. Dies kann wiederum den Takt **830** oder ein Taktfreigabesignal, die den Speicherbauelementen auf einem Speichermodul zur Verfügung gestellt werden, derart freigeben oder sperren, dass die von den Speicherbauelementen verbrauchte Energie kontrolliert wird.

[0056] Der Speichermodul-Controller **800** kann darüber hinaus eine Auffrischlogik **814** und Vorabruflogik **812** enthalten. Unter der Steuerung der Steuerlogik **802** kann die Auffrischlogik **814** Steuersignale an die Speicherbauelemente auf einem Speichermodul über die Steuerschnittstellenschaltung **820** senden. Die Steuerlogik **802** kann das Auffrischkommando zu geeigneten Zeitpunkten erzeugen oder sie kann ein Auffrischkommando in Erwiderung eines über den Transaktionsbus **826** von dem System-Speicher-Controller **304** gesendeten Auffrischkommandos erzeugen. Die Vorabruflogik **812** kann unter der Steuerung der Steuerlogik **802** eine Seite Daten aus den Speicherbauelementen (z. B. DRAM-Bauelementen) vorabrufen und die vorabgerufenen Daten zur Verwendung bei Speicherlesetransaktionen speichern. Der Speichermodul-Controller **800** kann darüber hinaus eine Seriell-vorhanden-Erfassungsschaltung, ECC-Schaltung, Stromkalibrierungsschaltung und andere Schaltungen enthalten, die entweder aus dem System-Speicher-Controller oder aus den Speicherbauelementen fortgelassen worden sind, um die Komplexität der Designs dieser Teile zu verringern. Dies kann die Anzahl der Pins des System-Speicher-Controllers verringern.

[0057] [Fig. 9](#) zeigt ein Ablaufdiagramm, das ein Ausführungsbeispiel der Operation des Speichermodul-Controllers **800** für Schreib- und Lesetransaktionen veranschaulicht. Der Prozess kann ein Pipeline-Prozess sein oder als Einzel-Stufen-Prozess implementiert sein. Es ist klar, dass der Speichermodul-Controller **800** neben der Übersetzung von Schreib- und Lesetransaktionen zwischen einem System-Speicher-Controller und den Speicherbauelementen auf einem Speichermodul viele weitere Funktionen durchführen kann.

[0058] Der Prozess beginnt im Schritt **902**. Beim Schritt **904** empfängt der Speichermodul-Controller **800** eine Speichertransaktionsanforderung auf dem Transaktionsbus **826**. Die Datenbehandlungslogik **804** deserialisiert die Transaktion im Schritt **906**, um die Adress-, Steuer- und Dateninformationen der Transaktion zu erlangen. Beim Schritt **908** prüft die Steuerlogik **802** die deserialisierten Informationen und bestimmt, ob die Transaktion an ein Speicherbauelement des Moduls, das von dem Speichermodul-Controller **800** gesteuert wird, gerichtet ist. Dies kann ausgeführt werden, indem die empfangene Speicheradresse überprüft wird, um nachzusehen, ob sie einer Adresse entspricht, die durch eines der Speicherbauelemente in dem Speichermodul belegt wird. Sofern die Transaktion nicht an ein von dem Speichermodul-Controller **800** gesteuertes Bauelement gerichtet

ist, kehrt der Prozess zum Schritt **902** zurück. Sofern die Transaktion an ein von dem Speichermodul-Controller **800** gesteuertes Speicherbauelement gerichtet ist, bestimmt die Steuerlogik **802** beim Schritt **910**, welche Art der Transaktion angefordert worden ist. Sofern die Transaktion eine Schreibtransaktion ist, erzeugt die Steuerlogik **802** die geeigneten Schreibsteuersignale (z. B. WE, CS, etc.) beim Schritt **912** und stellt diese Signale der Steuerschnittstellenschaltung **820** zur Verfügung. Beim Schritt **914** liefert die Steuerlogik **802** dann die Schreibdaten in dem richtigen Datenformat an die Daten-I/O-Schaltung **822** und die Adresse des ausgewählten Speicherbauelementes an die Adressschnittstellenschaltung **818**. Die Schreibdaten können dann in einem (nicht gezeigten) Schreibpuffer gepuffert werden. Die Daten können dann in den gewünschten Speicherplatz des ausgewählten Speicherbauelementes geschrieben werden. Beim Schritt **916** kann ein optionales Bestätigungssignal zurück zu dem System-Speicher-Controller gesendet werden, nachdem die Schreiboperation abgeschlossen ist. Der Prozess kehrt dann zum Schritt **902** zurück.

[0059] Wenn die Transaktion eine Lesetransaktion ist, dann stellt die Steuerlogik **802** die Leseadresse des ausgewählten Speicherbauelementes der Adressschnittstellenschaltung zur Verfügung. Beim Schritt **920** erzeugt die Steuerlogik **802** die richtigen Lesesteuersignale (z. B. WE, CS, OE, etc.) und stellt diese Signale der Steuerschnittstellenschaltung **820** zur Verfügung. Die Daten können dann aus dem gewünschten Speicherplatz des ausgewählten Speicherbauelementes gelesen werden. Die gelesenen Daten können in einem (nicht gezeigten) Lesepuffer gepuffert, in der Steuerlogik **802** gespeichert oder registriert werden. Beim Schritt **922** können die gelesenen Daten der Anforderungsbehandlungslogik **804** zur Verfügung gestellt werden, wo sie serialisiert werden und von anderen Daten eingerahmt werden können, einschließlich beispielsweise einer Anforderungsnummer, die diese spezielle Lesetransaktion anzeigt. Beim Schritt **924** können die gelesenen Daten dann zurück zu dem System-Speicher-Controller gesendet werden, wenn der Systemspeicherbus **823** frei ist oder wenn sämtliche anderen vorhergehenden Transaktionen abgeschlossen sind. Bei einem Ausführungsbeispiel kann eine Handshake-Logik **806** oder eine andere Logik (z. B. Anforderungsbehandlungslogik **804**, Steuerlogik **802** oder eine andere Busüberwachungslogik) die Aktivität auf dem Systemspeicherbus **823** überwachen und der Steuerlogik **802** anzeigen, wenn der Speichermodul-Controller **800** an der Reihe ist, seine gelesenen Daten zu dem System-Speicher-Controller auf dem Systemspeicherbus **823** zu senden. Die gelesenen Daten können mit dem Gültig-Signal **828** bei einem entkoppelten System zurückgesendet werden. Der Prozess kehrt dann zum Schritt **902** zurück.

[0060] [Fig. 10](#) ist ein Blockschaltbild eines erfindungsgemäßen Speichermodul-Controllers **1000**, der ein ein nicht gemultiplextes Protokoll verwendendes Ausführungsbeispiel des Speichermodul-Controllers **310** ist. Das nicht gemultiplexte Protokoll kann das Direct Rambus™-Protokoll sein. Andere mit dem Direct Rambus™-Protokoll kompatible Protokolle können ebenfalls verwendet werden.

[0061] Der Speichermodul-Controller **1000** bildet eine Schnittstelle zwischen dem Systemspeicherbus **1023**, der eines oder mehrere Taktsignalleitungen **1024**, einen Kommando- und Adressbus CMD/ADDR **1026**, einen Datenbus **1027** und optional ein Gültig-Signal **1028** enthalten kann. Der CMD/ADDR-Bus **1026** kann sowohl Adress- als auch Steuerinformationen für eine Speichertransaktion führen. Alternativ kann der CMD/ADDR-Bus **1026** in separate Kommando- und Adressbusse aufgeteilt sein. Das Gültig-Signal **1028** ist ein Handshaking-Signal, das für ein entkoppeltes Protokoll verwendet werden kann, um den Beginn einer Speichertransaktion anzuzeigen oder um anzuzeigen, wenn gültige Daten auf dem Transaktionsbus **1026** vorhanden sind. Das Gültig-Signal kann von der Handshake-Logik **1006** unter der Steuerung der Steuerlogik **1002** überwacht und erzeugt werden. Das Gültig-Signal **1028** kann für ein gekoppeltes Protokoll fortgelassen werden. Der Speichermodul-Controller **1000** bildet darüber hinaus eine Schnittstelle zu den Speicherbauelementen auf einem Speichermodul (wie in [Fig. 3](#)), indem er ein Taktsignal **1030**, Adresssignale **1032**, Steuersignale **1034** und Datensignale **1036** zur Verfügung stellt. Das Taktsignal **1030** kann bei asynchronen Speicherbauelementen fortgelassen werden.

[0062] Der Speichermodul-Controller **1000** enthält eine Anforderungsbehandlungslogik **1004** für eine Schnittstellenbildung zu dem Transaktionsbus **1026**. Die Anforderungsbehandlungslogik **1004** kann eine Deserialisierungslogik enthalten, die die auf dem CMD/ADDR-Bus **1026** zur Verfügung gestellten gemultiplexten Steuer- und Adressinformationen separieren und diese Signale der Steuerlogik **1002** über die Leitungen **1042** bzw. **1044** zur Verfügung stellen kann. Die Anforderungsbehandlungslogik **1004** kann darüber hinaus eine Serialisierungslogik enthalten, die Steuer- und Adressinformationen auf den Leitungen **1042** bzw. **1044** in eine Reihe von Signalen serialisieren kann, die dem CMD/ADDR-Bus **1026** zur Verfügung gestellt werden sollen.

[0063] Der Speichermodul-Controller **1000** enthält ferner eine Datenbehandlungslogik **1046**, die Daten aus dem Datenbus **1024** empfangen, die Daten in ein für die Speicherbauelemente des Speichermoduls geeignetes Format umformatieren und die umformatierten Daten dem Schreibpuffer **1012** zur Verfügung stellen kann.

Bei einem Ausführungsbeispiel kann die Datenbehandlungslogik **1046** eine Deserialisierungs- oder Entpacklogik enthalten, um die Übersetzung zwischen beispielsweise einem schmalen (z. B. 16-Bit-)Datenbus **1027** und einem breiteren (z. B. 64-Bit-)Speicherbauelementedatenbus **1036** auszuführen. Die Daten können in einem Schreibpuffer **1012** gespeichert werden, bis sie benötigt werden, um einem Speicherbauelement über die Daten-I/O-Schaltung **1022** zur Verfügung gestellt zu werden. Eine zugehörige Adresse für die Schreibdaten kann in einer Adressspeichereinheit **1014** gespeichert werden. Bei einem alternativen Ausführungsbeispiel kann der Schreibpuffer **1012** fortgelassen werden. Die Datenbehandlungslogik **1046** kann außerdem Daten aus den Speicherbauelementen eines Speichermoduls über die Daten-I/O-Schaltung **1022** und/oder den Lese-puffer **1038** empfangen. Die Datenbehandlungslogik kann dann die Daten in ein von dem Protokoll des Systemspeicherbusses **1023** erwartetes Format umformatieren. Bei einem Ausführungsbeispiel kann die Datenbehandlungslogik **1046** eine Serialisierungs- oder Packlogik enthalten, um die Übersetzung zwischen beispielsweise einem breiteren (z. B. 64-Bit-)Speicherbauelementedatenbus **1036** und einem schmaleren (z. B. 16-Bit-)Datenbus **1027** auszuführen. Bei noch einem anderen Ausführungsbeispiel kann die Datenbehandlungslogik fortgelassen werden und das Umformatieren der Daten kann durch die Steuerlogik **1002** ausgeführt werden.

[0064] Die Steuerlogik **1002** ist die Intelligenz des Speichermodul-Controllers **1000** und stellt die geeigneten Steuer-, Adress- und Datensignale für die Speichermodulspeicherbauelemente in Abhängigkeit von den aus der Anforderungsbehandlungslogik **1004** empfangenen Signalen zur Verfügung. Jedes Speichermodul-Controller-Bauelement in jedem Modul kann eine unterschiedliche Steuerlogik aufweisen, die eine spezielle Übersetzung zwischen den Signalarten und dem Protokoll auf dem Systemspeicherbus und den speziellen durch die Speicherbauelemente auf dem Speichermodul erwarteten Speichersignale und das spezielle Protokoll implementieren kann. Beispielsweise kann die Steuerlogik **1002** die richtigen Adresssignale der Adressschnittstellenschaltung **1018** über die Leitungen **1048**, die richtigen Steuersignale der Steuerschnittstellenschaltung **1020** über die Leitungen **1050** und die richtigen Datensignale der Daten-I/O-Schaltung **1022** zur Verfügung stellen, indem der Schreibpuffer **1012** über die Leitung **1052** und der Adressspeicher **1014** über Leitungen **1054** gesteuert wird. Die Steuerlogik **1002** kann darüber hinaus die geeigneten Steuersignale dem Lese-puffer **1038** zur Verfügung stellen, um zu kontrollieren, wenn aus einem Speicherbauelement auf einem Speichermodul gelesene Daten der Datenbehandlungslogik **1046** zur Verfügung gestellt werden. Die Schnittstellenschaltungen können Puffer und Register-elemente enthalten, um die Adresse **1032**, Speichersteuersignale **1034** und Daten **1036** anzusteuern.

[0065] Die Schnittstellenschaltungen **1018**, **1020** und **1022** können durch ein von dem Taktgenerator **1010** erzeugtes Taktsignal getaktet werden. Der Taktgenerator **1010** kann darüber hinaus ein Taktsignal der Steuerlogik **1002** und den Takt-puffern **1016** bereitstellen, welche das Taktsignal **1030** und/oder Taktfreigabesignale zu den Speicherbauelementen auf einem Speichermodul ansteuern können. Der Taktgenerator **1010** kann Taktsignale in Abhängigkeit von einem durch die DLL **1058** zur Verfügung gestellten Taktsignal erzeugen. Die DLL **1058** kann ein oder mehrere Taktsignale **1024** empfangen, die von dem Systemspeicherbus **1023** zur Verfügung gestellt werden. CLK **1030** kann eine andere Frequenz als Takt **1024** haben.

[0066] Der Speichermodul-Controller **1000** kann darüber hinaus eine Stromversorgungsmanagereinheit **1008** enthalten, die unter der Steuerung der Steuerlogik **1002** den Taktgenerator freigeben oder sperren kann. Dies wiederum kann den Takt **1030** oder ein Taktfreigabesignal sperren oder freigeben, das den Speicherbauelementen auf einem Speichermodul zur Verfügung gestellt wird, so dass die von diesen Speicherbauelementen verbrauchte Energie kontrolliert wird.

[0067] Der Speichermodul-Controller **1000** kann optional eine Adressspeichereinheit **1040** enthalten, die mit der Steuerlogik **1002**, der Adressschnittstellenschaltung **1018** und optional mit dem Taktgenerator **1010** gekoppelt ist. Die Adressspeichereinheit **1040** kann verwendet werden, um Adressinformationen zu speichern, die aus dem CMD/ADDR **1026** zur Verfügung gestellt werden.

[0068] Der Speichermodul-Controller **1000** kann ferner eine Auffrisch- und Vorabruflogik enthalten, wie sie in [Fig. 8](#) veranschaulicht ist, welche unter der Steuerung der Steuerlogik **1002** betrieben wird. Der Speichermodul-Controller **1000** kann ferner eine Seriell-vorhanden-Erfassungslogik, ECC-Schaltung, Stromkalibrierungsschaltung und andere Schaltungen enthalten, die entweder aus dem System-Speicher-Controller oder bei den Speicherbauelementen fortgelassen werden, um die Komplexität des Designs dieser Teile zu verringern. Das kann die Anzahl der Pins des System-Speicher-Controllers reduzieren.

[0069] [Fig. 11](#) zeigt ein Ablaufdiagramm, das ein Ausführungsbeispiel des Betriebs des Speichermodul-Controllers **1000** für Schreib- und Lesetransaktionen veranschaulicht. Der Prozess kann ein Pipeline-Prozess sein

oder als Einzel-Stufen-Prozess implementiert sein. Es ist klar, dass der Speichermodul-Controller **1000** neben dem Übersetzen von Schreib- und Lesetransaktionen zwischen einem System-Speicher-Controller und Speicherbauelementen auf einem Speichermodul viele weitere Funktionen ausführen kann.

[0070] Der Prozess startet beim Schritt **1102**. Beim Schritt **1104** empfängt der Speichermodul-Controller **1000** eine Speichertransaktionsanforderung auf dem CMD/ADDR-Bus **1026**. Beim Schritt **1006** entpackt oder deserialisiert die Datenbehandlungslogik **1004** die Kommando- und Adressinformationen und übermittelt diese Informationen an die Steuerlogik **1002**. Beim Schritt **1106** prüft die Steuerlogik **1002** die entpackten oder deserialisierten Informationen und bestimmt, ob die Transaktion an ein Speicherbauelement in dem Modul gerichtet ist, das von dem Speichermodul-Controller **1000** gesteuert wird. Dies kann ausgeführt werden, indem die empfangene Speicheradresse geprüft wird, um nachzusehen, ob sie einer durch eines der Speicherbauelemente in dem Speichermodul belegten Adresse entspricht. Sofern die Transaktion nicht an ein Speicherbauelement gerichtet ist, das von dem Speichermodul-Controller **1000** gesteuert wird, kehrt der Prozess zum Schritt **1102** zurück. Sofern die Transaktion an ein von dem Speichermodul-Controller **1000** gesteuertes Speicherbauelement gerichtet ist, bestimmt die Steuerlogik **1002** beim Schritt **1110**, welche Art der Transaktion angefordert wird. Sofern die Transaktion eine Schreibtransaktion ist, erzeugt die Steuerlogik **1002** die richtigen Schreibsteuersignale (z. B. WE, CS, etc.) beim Schritt **1112** und liefert diese Signale an die Steuerschnittstellenschaltung **1020**. Die Anforderungsbehandlungslogik **1004** kann darüber hinaus die Schreibadresse dem Adressspeicher **1014** zur Verfügung stellen. Alternativ kann die Steuerlogik **1002** die Schreibadresse der Adressschnittstellenschaltung **1018** zur Verfügung stellen, beispielsweise dann, wenn der Schreibpuffer **1012** nicht verwendet wird. Beim Schritt **1114** akzeptiert die Datenbehandlungslogik **1046** die Schreibdaten aus **1027**, entpackt oder deserialisiert die Schreibdaten und liefert die Schreibdaten an den Schreibpuffer **1012**. Beim Schritt **1116** werden die Schreibdaten in dem richtigen Datenformat der Daten-I/O-Schaltung **1022** zur Verfügung gestellt. Die Daten können dann in den gewünschten Speicherplatz des ausgewählten Speicherbauelements geschrieben werden. Beim Schritt **1118** kann ein optionales Bestätigungssignal zurück zu dem System-Speicher-Controller gesendet werden, nachdem die Schreiboperation abgeschlossen ist. Der Prozess kehrt dann zum Schritt **1102** zurück.

[0071] Sofern die Transaktion eine Lesetransaktion ist, erzeugt die Steuerlogik **1002** die richtigen Lesesteuersignale (WE, CS, OE, etc.) beim Schritt **1120** und liefert diese Signale der Steuerschnittstellenschaltung **1020**. Beim Schritt **1122** liefert die Steuerlogik **1002** dann die Leseadresse des ausgewählten Speicherbauelements an die Adressschnittstellenschaltung **1018**. Die Daten können dann aus dem gewünschten Speicherplatz des ausgewählten Speicherbauelements gelesen werden. Die gelesenen Daten können in dem Lesepuffer **1038** gepuffert, in der Steuerlogik **1002** gespeichert oder registriert werden. Beim Schritt **1124** können die gelesenen Daten der Datenbehandlungslogik **1046** zur Verfügung gestellt werden, wo sie serialisiert oder gepackt werden und durch andere Daten eingerahmt werden können, einschließlich beispielsweise einer Anforderungsnummer, die diese spezielle Lesetransaktion anzeigt. Beim Schritt **1126** können die gelesenen Daten zurück zu dem System-Speicher-Controller gesendet werden, wenn der Systemspeicherbus frei ist oder wenn sämtliche anderen vorhergehenden Transaktionen abgeschlossen worden sind. Bei einem Ausführungsbeispiel kann eine Handshake-Logik **1006** oder eine andere Logik (z. B. die Anforderungsbehandlungslogik **1004**, die Datenbehandlungslogik **1046**, die Steuerlogik **1002** oder eine andere Busüberwachungslogik) die Aktivität auf dem Systemspeicherbus **1023** überwachen und der Steuerlogik **1002** anzeigen, wenn der Speichermodul-Controller **1000** an der Reihe ist, seine gelesenen Daten an den System-Speicher-Controller auf dem Systemspeicherbus **1023** zu senden. Die gültigen gelesenen Daten können dann bei einem entkoppelten System mit einem Gültig-Signal **1028** zurückgesendet werden. Der Prozess kehrt dann zum Schritt **1102** zurück.

[0072] [Fig. 12](#) ist ein Blockschaltbild eines Speichermoduls **1200**, das ein Ausführungsbeispiel des erfindungsgemäßen Speichermoduls **306** gemäß [Fig. 3](#) ist. Das Speichermodul **1200** ist ein SDRAM-Modul, das SDRAM-Bauelemente **1204–1207** und einen Speichermodul-Controller **1202** enthält. Der Speichermodul-Controller **1202** schafft eine Brücke zwischen dem Systemspeicherbus **1242**, der ein Direct Rambus™-Protokoll verwendet (und mit einem System-Speicher-Controller gekoppelt ist) und SDRAM-Bauelementen **1204–1207**. Das Direct Rambus™-Protokoll enthält Taktsignale CLK **1228**, /CLK **1230** (wobei das Symbol "/" ein komplementäres Signal anzeigt), Datenleitungen **1232**, Zeilenadressleitungen **1234** und Spaltenadressleitungen **1236**. Die Daten können 16 Signalleitungen enthalten, die Zeilenadresse kann drei Signalleitungen enthalten und die Spaltenadresse kann fünf Signalleitungen enthalten, wie es in dem Advance Information Direct RDRAM™ 64/72-Mbit(256 K × 16/18 × 16d)-Datenblatt beschrieben ist. Bei einem Ausführungsbeispiel können CLK **1228** und /CLK **1230** eine Frequenz von etwa 400 MHz haben. Andere von diesem Protokoll verwendete Signale sind nicht gezeigt, um die vorliegende Erfindung nicht zu verdunkeln.

[0073] Der Speicher-Modul-Controller **1202** stellt aus dem Systemspeicherbus **1242** empfangene Speicher-

transaktionen den SDRAMs **1204–1207** zur Verfügung. Bei einem Ausführungsbeispiel können die SDRAMs **1204–1207** 64 Mb-SDRAMs sein, wie sie beispielsweise in dem im November 1997 veröffentlichten vorläufigen Datenblatt von IBM-Produkten IBM0364804C, IBM0364164C, IBM0364404C und IBM03644B4C beschrieben sind. Andere SDRAMs können ebenfalls verwendet werden.

[0074] Der Speichermodul-Controller **1202** stellt zunächst Steuerinformationen an die SDRAMs **1204** und **1205** über den Steuerbus **1238** und zweitens Steuerinformationen an die SDRAMs **1206** und **1207** über den Steuerbus **1240** zur Verfügung. Der Steuerbus **1238** kann ein Taktsignal CLK **1208**, ein Taktfreigabesignal CKE **1209**, ein Schreibfreigabesignal WE **1210**, ein Spaltenadress-Strobe-Signal CAS **1211**, ein Zeilenadress-Strobe-Signal RAS **1212** und ein oder mehrere Chipfreigabesignale **1213** enthalten. In ähnlicher Weise kann der Steuerbus **1240** ein Taktsignal CLK **1216**, ein Taktfreigabesignal CKE **1217**, ein Schreibfreigabesignal WE **1218**, ein Spaltenadress-Strobe-Signal CAS **1219**, ein Zeilenadress-Strobe-Signal RAS **1220** und ein oder mehrere Chipauswahlsignale CS **1221** enthalten. Bei einem Ausführungsbeispiel kann die Frequenz von CLK **1208** und CLK **1216** etwa 100 MHz bis 200 MHz sein. Andere Frequenzen können verwendet werden. Der Speichermodul-Controller übermittelt darüber hinaus Speicheradresssignale an die SDRAMs **1204** und **1205** über den Adressbus **1214** und an die SDRAMs **1206** und **1207** über den Adressbus **1222**. Bei einem Ausführungsbeispiel kann jeder Adressbus **12** Adressleitungen und zusätzlich zwei Bank-Auswahlleitungen zum Auswählen von Banken des Speichers in jedem SDRAM-Bauelement enthalten. Bei einem alternativen Ausführungsbeispiel kann der Steuerbus **1238** und **1240** derselbe Adressbus sein, wie es in [Fig. 3](#) gezeigt ist. In ähnlicher Weise können die Adressbusse **1214** und **1222** derselbe Bus sein. Der Speichermodul-Controller übermittelt darüber hinaus Lese- und Schreibdaten an die SDRAMs **1204**, **1205**, **1206** und **1207** über die Signalleitungen **1224**, **1225**, **1226** bzw. **1227**. Bei einem Ausführungsbeispiel kann jede Gruppe von Signalleitungen **16** Datenleitungen enthalten.

[0075] [Fig. 13](#) zeigt ein Ausführungsbeispiel eines von dem Speichermodul-Controller **1202** implementierten Prozesses zum Ausführen einer Lesetransaktion aus einem oder mehreren der SDRAMs **1204–1207** nach dem Empfangen einer Leseanforderung aus dem Systemspeicherbus **1244**. [Fig. 13](#) wird mit Hilfe des beispielhaften Zeitdiagramms gemäß [Fig. 14](#) beschrieben. Es ist klar, dass [Fig. 14](#) zeigt, dass CLK **1228** etwa die doppelte Frequenz von CLK **1208** aufweist. Bei einem Ausführungsbeispiel kann CLK **1228** eine Frequenz von etwa 400 MHz und CLK **1208** eine Frequenz von etwa 200 MHz haben. Bei anderen Ausführungsbeispielen können abweichende Frequenzverhältnisse und/oder abweichende Frequenzen verwendet werden.

[0076] Der Speichermodul-Controller **1202** kann den Prozess gemäß [Fig. 13](#) unter Verwendung der in dem Speichermodul **1000** gemäß [Fig. 10](#) gezeigten Schaltungsanordnung implementieren. Beispielsweise entspricht CLK **1024** dem CLK **1228**, CMD/ADDR **1026** entspricht dem Zeilensignal **1234** und dem Spaltensignal **1236**, die Daten **1027** entsprechen den Daten **1232**, CLK **1030** entspricht CLK **1208**, Adresse **1032** entspricht Adresse **1214**, Speichersteuersignale **1034** entsprechen dem Steuerbus **1238** und die Daten **1036** entsprechen den Daten **1224**. Bei anderen Ausführungsbeispielen können andere Schaltungen verwendet werden, um den Prozess gemäß [Fig. 13](#) zu implementieren.

[0077] Der Prozess startet beim Schritt **1302**. Beim Schritt **1304** und vom Zeitpunkt t_0 bis Zeitpunkt t_1 wird ein Aktivierungskommando auf den Zeilenleitungen **1234** gesendet, das eine Bauelement-, Bank- und Zeilenadresse des ausgewählten Speicherplatzes in einem oder mehreren der SDRAMs **1204–1207** einschließt. Beim Schritt **1306** und vom Zeitpunkt t_2 bis zum Zeitpunkt t_3 deserialisiert die Datenanforderungslogik **1004** die Kommando- und Adressinformationen und stellt diese Informationen der Steuerlogik **1002** zur Verfügung. Dann stellt die Steuerlogik **1002** die Zeilenadresse den ausgewählten SDRAMs über den Adressbus **1214** und/oder **1222** und das richtige Bankaktivierungskommando den ausgewählten SDRAMs über den Steuerbus **1238** und/oder **1240** zur Verfügung. Bei einem Ausführungsbeispiel umfasst das richtige Bankaktivierungskommando das Sperren der CAS-Leitungen **1211** und/oder **1219** und WE-Leitungen **1210** und/oder **1218** und das Freigeben der CS-Leitungen **1213** und/oder **1221** und der RAS-Leitungen **1212** und/oder **1220** bei der ansteigenden Flanke von CLK **1208** und/oder **1216**.

[0078] Da das Speicheradressierschema-Protokoll, das von Direct RDRAMTM-Bauelementen verwendet wird, von dem, das durch die ausgewählten SDRAMs verwendet wird, abweichen kann, kann das Aktivierungskommando, das auf den Zeilenadressleitungen **1234** angelegt wird, Spaltenadressinformationen für die ausgewählten SDRAMs einschließen. Bei diesem Beispiel können diese Spaltenadressinformationen in der Adressspeichereinheit **1040** bis zu dem Zeitpunkt gespeichert werden, zu dem die Spaltenadressinformationen an die ausgewählten SDRAMs ausgegeben werden.

[0079] Beim Schritt **1308** und vom Zeitpunkt t_4 bis zum Zeitpunkt t_6 wird ein Lesekommando auf den Spal-

tenleitungen **1236** gesendet. Das Lesekommando umfasst Bauelement-, Bank- und Spaltenadressinformationen für den gewünschten Leseort in den ausgewählten SDRAMs. Beim Schritt **1310** und vom Zeitpunkt t6 bis zum Zeitpunkt t7 deserialisiert die Anforderungsbehandlungslogik **1004** das Lesekommando und stellt diese Informationen an die Steuerlogik **1002** bereit. Dann liefert die Steuerlogik **1002** die Spaltenadresse und das Lesekommando an die ausgewählten SDRAMs über den Adressbus **1214** und/oder **1222** bzw. über den Steuerbus **1238** und/oder **1240**. Bei einem Ausführungsbeispiel umfasst das geeignete Lesekommando das Sperren der RAS-Leitungen **1212** und/oder **1222** und der WE-Leitungen **1210** und/oder **1218** und das Freigeben der CS-Leitungen **1213** und/oder **1221** und der CAS-Leitungen **1211** und/oder **1219** bei der ansteigenden Flanke von CLK **1208** und/oder CLK **1216**.

[0080] Sofern es eine Differenz zwischen der Anzahl der in den ausgewählten SDRAMs ausgewählten Bytes und der Anzahl derjenigen Bytes gibt, die aus einer einzelnen Lesetransaktion von dem System-Speicher-Controller erwartet werden, dann werden die Schritte **1308** und **1310** ggf. so oft wiederholt, bis die Anzahl von Bytes übereinstimmt. Beispielsweise, wie es in [Fig. 14](#) gezeigt ist, wird dann, wenn das erste Lesekommando READ1 an der Spalte **1236** (zusammen mit dem Aktivierungskommando an Zeile **1234**) 16 Bytes Daten adressiert und das Lesekommando an die ausgewählten SDRAMs **32** Bytes Daten adressiert, ein zweites Lesekommando READ2 vom Zeitpunkt t6 bis zum Zeitpunkt t9 ausgegeben, um die zweiten 16 Bytes zu adressieren.

[0081] Beim Schritt **1312** und vom Zeitpunkt t8 bis zum Zeitpunkt t15 kann die Daten-I/O-Schaltung **1022** die aus den ausgewählten SDRAMs gelesenen Daten empfangen. Beispielsweise können 32 Bytes Daten aus den vier SDRAMs **1204–1207** auf folgende Weise gelesen werden: Vom Zeitpunkt t8 bis zum Zeitpunkt t10 kann die Daten-I/O-Schaltung **1022** ein Datenpaket D0 empfangen, das die ersten acht Bytes enthält, zwei Bytes aus jedem der vier SDRAMs **1204–1207**; vom Zeitpunkt t10 bis zum Zeitpunkt t12 kann sie das Datenpaket D1 empfangen, das die zweiten acht Bytes enthält, zwei Bytes aus jedem der vier SDRAMs; vom Zeitpunkt t12 bis zum Zeitpunkt t13 das Datenpaket D2, das die dritten acht Bytes enthält, zwei Bytes aus jedem der vier SDRAMs; und vom Zeitpunkt t13 bis zum Zeitpunkt t14 das Datenpaket D3, das die letzten acht Bytes enthält, jeweils zwei Bytes aus jedem der vier SDRAMs. Es ist klar, dass bei anderen Ausführungsbeispielen eine abweichende Anzahl von Bytes und eine abweichende Anzahl von Zyklen erforderlich sein können, um die ausgewählten Daten aus den ausgewählten SDRAMs zu lesen.

[0082] Wenn die gelesenen Daten empfangen sind, können sie in einem Lesepuffer **1038** gespeichert werden, bevor die Daten der Datenbehandlungslogik **1046** zur Verfügung gestellt werden. Alternativ können sie direkt der Datenbehandlungslogik **1046** zur Verfügung gestellt werden. Beim Schritt **1314** kann die Datenbehandlungslogik die Daten in das richtige, von dem System-Speicher-Controller auf dem Datenbus **1232** erwartete Format serialisieren oder packen. Bei einem Ausführungsbeispiel können die Datenpakete D0 und D1, die eine Gesamtzahl von 16 Bytes Daten aus den SDRAMs **1204–1207** einschließen, in eine ununterbrochene Übertragung DATA1 gepackt werden, die dem Datenbus **1232** beim Schritt **1316** und vom Zeitpunkt t11 bis zum Zeitpunkt t15 zur Verfügung gestellt wird. In ähnlicher Weise können die Datenpakete D2 und D3, die die zweiten 16 Bytes Daten aus den SDRAMs **1204** bis **1207** enthalten, in eine ununterbrochene Übertragung DATA2 gepackt werden, die dem Datenbus **1232** beim Schritt **1316** und vom Zeitpunkt t14 bis zum Zeitpunkt t16 zur Verfügung gestellt wird.

[0083] [Fig. 15](#) zeigt ein Ausführungsbeispiel eines durch einen Speichermodul-Controller **1202** implementierten Prozesses zum Durchführen einer Schreibtransaktion zu einem oder mehreren der SDRAMs **1204–1207** nach dem Empfangen einer Schreibanforderung aus dem Systemspeicherbus **1242**. Der gleiche Prozess kann verwendet werden, um Daten zu beliebigen der SDRAMs zu schreiben. [Fig. 15](#) wird unter Zuhilfenahme des beispielhaften Zeitdiagramms der [Fig. 16](#) beschrieben. Der Speichermodul-Controller **1202** kann den Prozess gemäß [Fig. 13](#) unter Verwendung der im Speichermodul-Controller **1000** gemäß [Fig. 10](#) gezeigten Schaltungsanordnung implementieren.

[0084] Es ist klar, dass [Fig. 15](#) zeigt, dass CLK **1228** etwa die doppelte Frequenz von CLK **1208** hat. Bei einem Ausführungsbeispiel kann CLK **1228** eine Frequenz von etwa 400 MHz und CLK **1208** eine Frequenz von etwa 200 MHz aufweisen. Bei anderen Ausführungsbeispielen können abweichende Frequenzverhältnisse und/oder abweichende Frequenzen verwendet werden.

[0085] Der Prozess beginnt beim Schritt **1502**. Beim Schritt **1504** und vom Zeitpunkt t0 bis zum Zeitpunkt t1 wird ein Aktivierungskommando auf den Zeilenleitungen **1234** gesendet, das eine Bauelement-, Bank- und Zeilenadresse des Speicherplatzes in einem oder mehreren ausgewählter SDRAMs **1204–1207** einschließt. Beim Schritt **1506** deserialisiert die Datenanforderungslogik **1004** die Kommando- und Adressinformationen und stellt diese Informationen der Steuerlogik zur Verfügung.

[0086] Da das von DirectRDRAM™-Bauelementen verwendete Speicheradressierungsschema-Protokoll von demjenigen abweichen kann, das von den SDRAMs **1204–1207** verwendet wird, kann das auf den Zeilenadressleitungen **1234** gelieferte Aktivierungskommando Spaltenadressinformationen für die ausgewählten SDRAMs enthalten. Bei diesem Beispiel können diese Spaltenadressinformationen in der Adressspeichereinheit **1040** solange gespeichert werden, bis die Spaltenadressinformationen an die ausgewählten SDRAMs ausgegeben werden.

[0087] Beim Schritt **1508** und vom Zeitpunkt t2 bis zum Zeitpunkt t3 wird ein Schreibkommando auf den Spaltenleitungen **1236** gesendet. Das Schreibkommando enthält Bauelemente-, Bank- und Spaltenadressinformationen für die gewünschten Schreiborte in den ausgewählten SDRAMs. Beim Schritt **1510** deserialisiert die Anforderungsbehandlungslogik **1004** das Schreibkommando und stellt diese Informationen der Steuerlogik **1002** zur Verfügung. Sofern es eine Differenz zwischen der zum Schreiben in die ausgewählten SDRAMs ausgewählte Anzahl von Bytes und der durch den System-Speicher-Controller ausgewählten Anzahl von Bytes gibt, so werden die Schritte **1508** und **1510** so oft wiederholt, bis die Anzahlen der Bytes übereinstimmen. Wenn beispielsweise, wie es in [Fig. 16](#) gezeigt ist, das erste Schreibkommando WRITE1 auf Spalte **1236** (zusammen mit dem Aktivierungskommando auf Zeile **1234**) 16 Bytes Daten auswählt und das Schreibkommando zu den ausgewählten SDRAMs 32 Bytes Daten adressiert, so wird ein zweites Schreibkommando WRITE2 vom Zeitpunkt t3 bis zum Zeitpunkt t4 ausgegeben, um die zweiten 16 Bytes zu adressieren.

[0088] Beim Schritt **1512** und vom Zeitpunkt t5 bis zum Zeitpunkt t6 kann die Datenbehandlungslogik **1046** Schreibdaten DATA1 vom Datenbus **1232** empfangen und die Schreibdaten entpacken oder deserialisieren. Die Datenbehandlungslogik **1046** kann die Schreibdaten DATA1 in den Schreibpuffer **1012** speichern, und der Adressspeicher **1014** kann die Schreibadresse speichern, die von der Anforderungsbehandlungslogik **1004** ausgegeben wird. Wenn ein nachfolgendes Schreib- oder Lesekommando empfangen wird, kann die Steuerlogik **1002** den Schreibpuffer **1012** und den Adressspeicher **1014** für die angeforderte Adresse überprüfen. Wenn die Adresse in dem Adressspeicher **1014** angeordnet ist, dann können die zugehörigen Daten im Schreibpuffer **1012** gelesen oder überschrieben werden. Wenn ein nachfolgendes Schreibkommando WRITE2 ausgegeben wird, dann können die vom Zeitpunkt t8 bis zum Zeitpunkt t10 empfangenen und diesem Kommando zugeordneten Daten DATA2 entpackt oder deserialisiert und ebenfalls im Schreibpuffer **1012** gespeichert werden.

[0089] Beim Schritt **1516** und vom Zeitpunkt t6 bis zum Zeitpunkt t7 stellt die Steuerlogik **1002** dann die Zeilenadresse an die ausgewählten SDRAMs über den Adressbus **1214** und/oder **1222** und das geeignete Bankaktivierungskommando an die ausgewählten SDRAMs über den Steuerbus **1238** und/oder **1240** zur Verfügung. Bei einem Ausführungsbeispiel umfasst das geeignete Bankaktivierungskommando ein Sperren der CAS-Leitungen **1211** und/oder **1219** und WE-Leitungen **1210** und/oder **1218** und ein Freigeben der CS-Leitungen **1213** und/oder **1221** und der RAS-Leitungen **1212** und/oder **1220** bei der ansteigenden Flanke von CLK **1208** und/oder CLK **1216**. Beim Schritt **1518** und vom Zeitpunkt t9 bis zum Zeitpunkt t11 stellt die Steuerlogik **1002** dann die Spaltenadresse und das Schreibkommando den ausgewählten SDRAMs über den Adressbus **1214** und/oder **1222** bzw. den Steuerbus **1238** und/oder **1240** zur Verfügung. Bei einem Ausführungsbeispiel umfasst das richtige Schreibkommando ein Sperren der RAS-Leitungen **1212** und/oder **1210** und ein Freigeben der CS-Leitungen **1213** und/oder **1221**, der RAS-Leitungen **1212** und/oder **1210** und der WE-Leitungen **1210** und/oder **1218** bei der ansteigenden Flanke von CLK **1208** und/oder CLK **1216**.

[0090] Beim Schritt **1520** und vom Zeitpunkt t9 bis zum Zeitpunkt t14 kann die Steuerlogik veranlassen, dass die Schreibdaten den ausgewählten SDRAMs auf den Datenleitungen **1224**, **1225**, **1226** und/oder **1227** aus der Daten-I/O-Schaltung **1222** zur Verfügung gestellt werden. Beispielsweise können 32 Bytes Daten zu den vier SDRAMs **1204–1207** auf folgende Weise geschrieben werden: Vom Zeitpunkt t9 bis zum Zeitpunkt t11 kann die Daten-I/O-Schaltung **1022** ein Datenpaket D0 bereitstellen, das die ersten acht Bytes enthält, jeweils zwei Bytes für jeden der vier SDRAMs; und vom Zeitpunkt t11 bis zum Zeitpunkt t12 das Datenpaket D1, das die zweiten acht Bytes enthält, jeweils zwei Bytes für jeden der SDRAMs **1204–1207**. D0 und D1 können DATA1 entsprechen. Zusätzlich kann die Daten-I/O-Schaltung **1022** vom Zeitpunkt t12 bis zum Zeitpunkt t13 ein Datenpaket D2 bereitstellen, das die dritten acht Bytes enthält, jeweils zwei Bytes für jeden der SDRAMs **1204–1207**, und vom Zeitpunkt t13 bis zum Zeitpunkt t14 das Datenpaket D3, das die letzten acht Bytes enthält, jeweils zwei Bytes für jeden der SDRAMs **1204–1207**. D2 und D3 können DATA2 entsprechen. Es ist klar, dass bei anderen Ausführungsbeispielen eine abweichende Anzahl von Bytes und eine abweichende Anzahl von Zyklen erforderlich sein können, um die ausgewählten Daten in die ausgewählten SDRAMs zu schreiben. Obwohl bei diesem Beispiel in sämtliche DRAMs geschrieben wird, kann bei einem alternativen Ausführungsbeispiel ein partielles Schreiben zu einem (oder zu mehreren, aber nicht allen) der SDRAMs ausgeführt werden. Bei einem alternativen Ausführungsbeispiel kann der Speichermodul-Controller **1202** außerdem eine Brü-

cke zwischen dem System Speicherbus **1242** und den SDRAMs **1204–1207** unter Verwendung eines modifizierten Protokolls zur Verfügung stellen. Das modifizierte Protokoll kann ein nicht gemultiplextes sein mit einem Datenbus und einem separaten Steuerbus, der Steuerinformationen und Adressen zwischen einem System-Speicher-Controller (z. B. dem System-Speicher-Controller **304** gemäß [Fig. 3](#)) und dem Speichermodul-Controller **1202** übermittelt.

[0091] Bei diesem Ausführungsbeispiel kann eine Lesetransaktion aus einem oder mehreren der SDRAMs **1204–1207** durchgeführt werden, wie es in [Fig. 17](#) gezeigt ist, wobei der Zeilenbus **1234** und Spaltenbus **1236** als ein Steuerbus **1235** verwendet werden, der acht Signalleitungen aufweist. Bei anderen Ausführungsbeispielen kann eine abweichende Anzahl von Signalleitungen verwendet werden.

[0092] Vom Zeitpunkt t0 bis zum Zeitpunkt t1 wird ein Aktivierungskommando über den Steuerbus **1235** gesendet, das eine Bauelemente-, Bank- und Zeilenadresse der ausgewählten Speicherplätze in den SDRAMs **1204–1207** einschließt. Bei diesem Ausführungsbeispiel kann sich das Aktivierungskommando über zwei Zyklen (z. B. Zyklus 0 und Zyklus 1 in der unten angegebenen Tabelle 1) von CLK **1228**, in welchen Daten bei jeder Taktflanke von CLK **1228** übertragen werden (d. h. doppelt-gepumpt), erstrecken. Bei einem Ausführungsbeispiel ist das Paketformat so, wie in Tabelle 1 gezeigt, wo RQ[7:0] den Signalleitungen des Steuerbusses **1235** entspricht.

Cycle	RQ[7]	RQ[6]	RQ[5]	RQ[4]	RQ[3]	RQ[2]	RQ[1]	RQ[0]
0	ST	SF	D[4]	D[3]	D[2]	D[1]	D[0]	B[0]
0	R[6]	R[5]	R[4]	R[3]	R[2]	R[1]	R[0]	B[1]
1	R[13]	R[12]	R[11]	R[10]	R[9]	R[8]	R[7]	B[2]

TABELLE 1

[0093] Das Aktivierungskommando enthält verschiedene Felder:

B[2:0]Bankauswahl: Wählt eine von acht Banken in den SDRAMs **1204–1207** aus;

D[4:0]Bauelementeauswahl: Wählt einen von 2⁵ oder 32 Speichermodul-Controllern aus;

R[13:0]Zeilenadresse: Stellt eine Zeilenadresse für einen oder mehrere in den SDRAMs **1204–1207** ausgewählten Orten zur Verfügung; und

ST, SF Codierter Rahmen/Schnelles Kommando: Bits, die verwendet werden, um anzuzeigen, ob ein Kommando gerahmt ist, wie in Tabelle 2 angezeigt. Die erweiterten Kommandos können Keine-Operation-(noop), Vorauflade-, Auffrisch- und andere Kommandos umfassen.

ST	SF	Operation
0	0	Keine Rahmenbildung
0	1	Rahme ein Aktiv-Kommando (Seite aktivieren)
1	0	Rahme ein Lese/Schreib-Kommando (Seitentreffer)
1	1	Rahme ein Erweitertes Kommando

TABELLE 2

[0094] Vom Zeitpunkt t2 bis zum Zeitpunkt t3 deserialisiert die Datenanforderungslogik **1004** die Kommando- und Adressinformationen und stellt diese Informationen der Steuerlogik **1002** zur Verfügung. Die Steuerlogik **1002** stellt dann die Zeilenadresse den ausgewählten SDRAMs über den Adressbus **1214** und/oder **1222** und das richtige Bankaktivierungskommando den ausgewählten SDRAMs über den Steuerbus **1238** und/oder **1240** zur Verfügung (z. B. wie es oben unter Bezugnahme auf die [Fig. 13](#) und [Fig. 14](#) beschrieben wurde).

[0095] Vom Zeitpunkt t4 bis zum Zeitpunkt t5 wird ein Lesekommando auf den Steuerbus **1235** ausgesendet. Das Lesekommando enthält Bauelemente-, Bank- und Spaltenadressinformationen für die gewünschte Leseseite in den SDRAMs. Bei diesem Ausführungsbeispiel kann das Lesekommando sich über zwei Zyklen (d. h. den Zyklus 0 und den Zyklus 1 in Tabelle 3 unten) von CLK **1228** erstrecken und weist ein Paketformat, wie

es in Tabelle 3 gezeigt ist.

Cycle	RQ[7]	RQ[6]	RQ[5]	RQ[4]	RQ[3]	RQ[2]	RQ[1]	RQ[0]
0	ST	SF	D[4]	D[3]	D[2]	D[1]	D[0]	B[0]
0	C[6]	C[5]	C[4]	C[3]	C[2]	C[1]	C[0]	C[1]
1	AP	R/W	C[11]	C[10]	C[9]	C[8]	C[7]	C[2]

TABELLE 3

[0096] Das Lesekommando enthält mehrere unterschiedliche Felder, die nicht in dem Aktivierungskommando vorhanden sind:

C[11:0] Zeilenadresse: Stellt eine Spaltenadresse für einen oder mehrere in den SDRAMs **1204–1207** ausgewählte Orte zur Verfügung;

R/W Lesen/Schreiben: Zeigt an, ob das Kommando eine Lesetransaktion oder eine Schreibtransaktion ist. Wenn dieses Signal sich in dem einen logischen Zustand befindet, zeigt es eine Lesetransaktion an, und wenn es sich in dem komplementären logischen Zustand befindet, zeigt es eine Schreibtransaktion an; und AP Auto-Voraufladen: Zeigt an, ob die ausgewählte Spalte (oder Zeile) der ausgewählten Speicherzelle nach dem Lesen voraufgeladen werden sollte.

[0097] Vom Zeitpunkt t6 bis zum Zeitpunkt t8 deserialisiert die Anforderungsbehandlungslogik **1004** das Lesekommando und stellt diese Informationen der Steuerlogik **1002** zur Verfügung. Die Steuerlogik **1002** stellt dann die Spaltenadresse und das Lesekommando den ausgewählten SDRAMs über den Adressbus **1214** und/oder **1222** bzw. über den Steuerbus **1238** und/oder **1240** zur Verfügung (z. B. wie oben unter Bezugnahme auf die [Fig. 13](#) und [Fig. 14](#) beschrieben wurde). Sofern das erste Lesekommando READ1 auf dem Steuerbus **1235** weniger Bytes Daten als das an die SDRAMs **1204–1207** ausgegebene Lesekommando adressiert, so kann ein zweites Lesekommando READ2 vom Zeitpunkt t7 bis zum Zeitpunkt t9 ausgegeben werden.

[0098] Vom Zeitpunkt t10 bis zum Zeitpunkt t16 kann die Daten-I/O-Schaltung **1022** die aus den ausgewählten SDRAMs gelesenen Daten empfangen. Beispielsweise können 32 Bytes Daten aus den vier SDRAMs **1204–1207** auf folgende Weise gelesen werden: Vom Zeitpunkt t10 bis zum Zeitpunkt t11 kann die Daten-I/O-Schaltung ein Datenpaket D0 empfangen, das die ersten acht Bytes, jeweils zwei Bytes aus jedem der vier SDRAMs **1204–1207** enthält; vom Zeitpunkt t11 bis zum Zeitpunkt t12 das Datenpaket D1, das die zweiten acht Bytes, jeweils zwei Bytes aus jedem der vier SDRAMs, enthält; vom Zeitpunkt t13 bis zum Zeitpunkt t14 ein Datenpaket D2, das die dritten acht Bytes, jeweils zwei Bytes aus jedem der vier SDRAMs, enthält; und vom Zeitpunkt t14 bis zum Zeitpunkt t16 das Datenpaket D3, das die letzten acht Bytes, jeweils zwei Bytes aus jedem der vier SDRAMs, enthält. Es ist klar, dass bei anderen Ausführungsbeispielen eine abweichende Anzahl von Bytes und eine abweichende Anzahl von Zyklen erforderlich sein können, um die ausgewählten Daten aus den ausgewählten SDRAMs zu lesen.

[0099] Wenn die gelesenen Daten empfangen sind, können sie in einem Lesepuffer **1038** gespeichert werden, bevor die Daten der Datenbehandlungslogik **1046** zur Verfügung gestellt werden. Alternativ können sie direkt der Datenbehandlungslogik **1046** zur Verfügung gestellt werden. Beim Schritt **1314** kann die Datenbehandlungslogik die Daten in das richtige, von dem System-Speicher-Controller auf dem Datenbus **1232** erwartete Format serialisieren oder packen. Bei einem Ausführungsbeispiel können die Datenpakete D0 und die D1, die eine Gesamtzahl von 16 Bytes Daten aus den SDRAMs **1204–1207** enthalten, in eine ununterbrochene Übertragung DATA1 gepackt werden, die auf dem Datenbus **1232** vom Zeitpunkt t12 bis zum Zeitpunkt t15 zur Verfügung gestellt wird. In ähnlicher Weise können die Datenpakete D2 und D3, die eine zweite Anzahl von 16 Bytes Daten aus den SDRAMs **1204–1207** enthalten, in eine ununterbrochene Übertragung DATA2 gepackt werden, die dem Datenbus **1232** vom Zeitpunkt t15 bis zum Zeitpunkt t16 zur Verfügung gestellt werden kann.

[0100] Bei diesem Ausführungsbeispiel kann eine Schreibtransaktion zu einem oder mehreren der SDRAMs **1204–1207** so durchgeführt werden, wie es in [Fig. 18](#) gezeigt ist. Vom Zeitpunkt t0 bis zum Zeitpunkt t1 wird ein Aktiviere-Kommando, wie es in Tabelle 1 gezeigt ist, auf dem Steuerbus **1235** gesendet, das eine Bauelemente-, Bank- und Zeilenadresse der Speicherplätze in einem oder mehreren der SDRAMs **1204–1207** einschließt. Die Datenanforderungslogik **1004** deserialisiert die Kommando- und Adressinformationen und stellt diese Informationen der Steuerlogik zur Verfügung.

[0101] Vom Zeitpunkt t2 bis zum Zeitpunkt t3 wird ein Schreibkommando auf den Steuerbus **1235** ausgesen-

det, das Bauelemente-, Bank- und Spaltenadressinformationen für die gewünschten Schreiborte in den SDRAMs **1204–1207** einschließt. Das Schreibkommando verwendet das in Tabelle 3 gezeigte Paketformat, wobei das R/W-Bit so gesetzt ist, dass es eine Schreibtransaktion anzeigt. Die Anforderungsbehandlungslogik **1004** deserialisiert das Schreibkommando und stellt diese Informationen der Steuerlogik **1002** zur Verfügung. Wenn es eine Differenz zwischen der zum Schreiben in die SDRAMs **1204–1207** ausgewählte Anzahl von Bytes (z. B. 32 Bytes) und der durch die von dem System-Speicher-Controller ausgegebene einzelne Schreibtransaktion ausgewählten Anzahl von Bytes (z. B. 16 Bytes) gibt, dann kann das erste Schreibkommando WRITE1 auf dem Steuerbus **1235** einen ersten Teil (z. B. 16 Bytes) der Daten schreiben und ein zweites Schreibkommando WRITE2 kann vom Zeitpunkt t4 bis zum Zeitpunkt t5 ausgegeben werden, um einen zweiten Teil der Daten zu schreiben (z. B. weitere 16 Bytes).

[0102] Vom Zeitpunkt t6 bis zum Zeitpunkt t9 kann die Datenbehandlungslogik **1046** Schreibdaten DATA1 aus dem Datenbus **1232** empfangen und die Schreibdaten entpacken oder deserialisieren. Die Datenbehandlungslogik kann die Schreibdaten DATA1 in den Schreibpuffer **1012** speichern, und der Adressspeicher **1014** kann die durch die Anforderungsbehandlungslogik **1004** ausgegebene Schreibadresse speichern. Wenn ein nachfolgendes Schreib- oder Lesekommando empfangen wird, kann die Steuerlogik **1002** den Schreibpuffer **1012** und Adressspeicher **1014** hinsichtlich der angeforderten Adresse überprüfen. Wenn die Adresse im Adressspeicher **1014** angeordnet ist, dann können die entsprechenden Daten im Schreibpuffer gelesen oder überschrieben werden. Wenn ein nachfolgendes Schreibkommando WRITE2 ausgegeben wird, dann können die vom Zeitpunkt t9 bis zum Zeitpunkt t11 empfangenen und diesem Kommando zugeordneten Daten DATA2 entpackt oder deserialisiert werden und ebenfalls im Schreibpuffer **1012** gespeichert werden.

[0103] Vom Zeitpunkt t7 bis zum Zeitpunkt t8 stellt die Steuerlogik **1002** dann die Zeilenadresse an die ausgewählten SDRAMs über den Adressbus **1214** und/oder **1222** und das richtige Bankaktivierungskommando an die SDRAMs **1204–1207** über den Steuerbus **1238** und/oder **1240** zur Verfügung (beispielsweise wie es oben unter Bezugnahme auf die [Fig. 15](#) und [Fig. 16](#) beschrieben wurde). Vom Zeitpunkt t11 bis zum Zeitpunkt t12 stellt die Steuerlogik **1002** dann die Spaltenadresse und das Schreibkommando an die ausgewählten SDRAMs über den Adressbus **1214** und/oder **1222** bzw. über den Steuerbus **1238** und/oder **1240** zur Verfügung (beispielsweise wie es oben unter Bezugnahme auf die [Fig. 15](#) und [Fig. 16](#) beschrieben wurde).

[0104] Vom Zeitpunkt t10 bis zum Zeitpunkt t15 kann die Steuerlogik veranlassen, dass die Schreibdaten den ausgewählten SDRAMs auf den Datenleitungen **1224**, **1225**, **1226** und/oder **1227** aus der Daten-I/O-Schaltung **1022** zur Verfügung gestellt werden. Beispielsweise können 32 Bytes Daten in die vier SDRAMs **1204–1207** auf folgende Weise eingeschrieben werden: Vom Zeitpunkt t10 bis zum Zeitpunkt t12 kann die Daten-I/O-Schaltung **1022** ein Datenpaket D0 zur Verfügung stellen, das die ersten acht Bytes, jeweils zwei Bytes für jeden der vier SDRAMs, enthält; und vom Zeitpunkt t12 bis zum Zeitpunkt t13 ein Datenpaket D1, das die zweiten acht Bytes, jeweils zwei Bytes für jeden der SDRAMs **1204–1207**, enthält. D0 und D1 können DATA1 entsprechen. Zusätzlich kann die Daten-I/O-Schaltung **1022** vom Zeitpunkt t13 bis zum Zeitpunkt t14 ein Datenpaket D2 bereitstellen, das die dritten acht Bytes, jeweils zwei Bytes für jeden der SDRAMs **1204–1207**, enthält, und vom Zeitpunkt t14 bis zum Zeitpunkt t15 ein Datenpaket D3, das die letzten acht Bytes enthält, jeweils zwei Bytes für jeden der SDRAMs **1204–1207**. D2 und D3 können DATA2 entsprechen. Es ist klar, dass bei anderen Ausführungsbeispielen eine abweichende Anzahl von Bytes und eine abweichende Anzahl von Zyklen erforderlich sein können, um die ausgewählten Daten in die ausgewählten SDRAMs zu schreiben. Obwohl bei diesem Beispiel in sämtliche DRAMs eingeschrieben wird, kann bei alternativen Ausführungsbeispielen ein partielles Schreiben in einen (oder weniger als allen) der SDRAMs ausgeführt werden.

Patentansprüche

1. Speichersystem (**300**) mit einem mit einem Prozessor (**302**) gekoppelten System-Speicher-Controller (**304**) und mehreren über einen Systemspeicherbus (**323**, **1023**, **1242**) mit dem System-Speicher-Controller (**304**) verbundenen Speichermodulen (**306**, **308**; **500**; **600**; **700**), wobei jedes Speichermodul (**306**, **308**; **500**; **600**; **700**) eine Baueinheit mit einer Mehrzahl von Speicherbauelementen (**312–315**, **317–320**; **1204–1207**), die auf einer oder beiden Seiten einer Platine des Moduls angeordnet sind, einer Schnittstelle mit einem Verbinder zu dem Systemspeicherbus (**323**, **1023**, **1242**) und mit Signalleitungen (**330–344**; **1208–1227**, **1238**, **1240**), die die Speicherbauelemente mit der Schnittstelle verbinden, bildet, wobei der Systemspeicherbus (**323**, **1023**, **1242**) mit der Direct-Rambus-Busarchitektur kompatibel ist, **dadurch gekennzeichnet**, dass die Speicherbauelemente (**312–315**, **317–320**; **1204–1207**) SDRAM-Bauelemente sind, dass die Schnittstelle zu dem Systemspeicherbus (**323**, **1023**, **1242**) von einem mit den Signalleitungen gekoppelten und auf dem Modul angeordneten Speichermodul-Controller (**310**, **316**; **1000**; **1202**) gebildet ist, und

dass der Speichermodul-Controller (**310, 316; 1000; 1202**) aufweist:
 eine erste Schnittstellenschaltung (**1004, 1046**) zum Empfangen der Speichertransaktionen in einem mit der Direct-Rambus-Busarchitektur kompatiblen ersten Format von dem Systemspeicherbus,
 eine zweite Schnittstellenschaltung (**1012, 1020, 1022**), die über die Signalleitungen Speichertransaktionen in einem an die SDRAM-Speicherbauelemente angepassten zweiten Format, das sich von dem ersten Format unterscheidet, an die Speicherbauelemente sendet, und
 eine mit der ersten Schnittstellenschaltung und der zweiten Schnittstellenschaltung gekoppelte Steuerlogik (**1002**) zum Konvertieren der Speichertransaktionen des ersten Formats in Speichertransaktionen des zweiten Formats,
 so dass der Systemspeicherbus (**323, 1023, 1242**) von den Signalleitungen (**330–344; 1208–1227, 1238, 1240**) entkoppelt wird.

2. Speichersystem (**300**) nach Anspruch 1, dadurch gekennzeichnet, dass die Signalleitungen (**330–344; 1208–1227, 1238, 1240**) einen Modul-Speicherbus umfassen, der zwischen dem Speichermodul-Controller (**310, 316; 1000; 1202**) und der Mehrzahl von Speicherbauelementen (**312–315, 317–320; 1204–1207**) eingekoppelt ist.

3. Speichersystem (**300**) nach Anspruch 2, dadurch gekennzeichnet, dass der Modul-Speicherbus separate Adress-, Daten- und Steuersignalleitungen aufweist.

4. Speichersystem (**300**) nach Anspruch 2, dadurch gekennzeichnet, dass der Modul-Speicherbus ein Taktsignal (**1208, 1216**) aufweist.

5. Speichersystem (**300**) nach Anspruch 2, dadurch gekennzeichnet, dass der Speicherbus (**323, 1242**) bei einer ersten Datenrate und der Modul-Speicherbus bei einer zweiten, abweichenden Datenrate arbeitet.

6. Speichersystem (**300**) nach einem der Ansprüche 1 bis 5, dadurch gekennzeichnet, dass die Speichermodule (**306, 308**) Dual-in-line-Speichermodule (DIMM) sind.

7. Speichersystem (**300**) nach einem der Ansprüche 1 bis 5, dadurch gekennzeichnet, dass die Speichermodule (**306, 308**) Single-in-line-Speichermodule (SINN) sind.

8. Speichersystem (**300**) nach einem der Ansprüche 1 bis 7, dadurch gekennzeichnet, dass die erste Schnittstellenschaltung eine Anforderungsbehandlungslogik (**1004**) aufweist, die die Adreß- und Kommandoinformationen trennt und die separaten Adreß- und Kommandoinformationen der Steuerlogik (**1002**) zur Verfügung stellt.

9. Speichersystem (**300**) nach einem der Ansprüche 1 bis 8, dadurch gekennzeichnet, dass die erste Schnittstellenschaltung eine Handshaking-Logik (**1006**) aufweist, die ein Handshake-Signal an den Speicherbus (**1023**) liefert, welches anzeigt, daß der Speichermodul-Controller (**1000**) Daten an den Speicherbus (**1023**) übermittelt.

10. Speichersystem (**300**) nach einem der Ansprüche 1 bis 9, dadurch gekennzeichnet, dass die erste Schnittstellenschaltung eine Datenbehandlungslogik (**1046**) aufweist, die Daten der Speichertransaktionen in dem ersten Format von dem Systemspeicherbus (**1023**) empfangen und in Daten der Speichertransaktionen in dem zweiten Format umformatieren kann.

11. Speichersystem (**300**) nach einem der Ansprüche 1 bis 10, dadurch gekennzeichnet, dass der Speichermodul-Controller (**1000**) einen Schreibpuffer (**1012**) aufweist, der mit der ersten Schnittstellenschaltung gekoppelt ist und Daten speichert, die mit Speichertransaktionen in dem ersten Format gesendet worden sind.

12. Speichersystem (**300**) nach Anspruch 11, dadurch gekennzeichnet, dass der Speichermodul-Controller (**1000**) eine mit dem Schreibpuffer (**1012**) und der ersten Schnittstellenschaltung gekoppelte Adressspeichereinheit (**1014**) aufweist, wobei die Adressspeichereinheit (**1014**) den in dem Schreibpuffer (**1012**) gespeicherten Schreibdaten zugeordnete Adressen speichert.

13. Speichersystem (**300**) nach einem der Ansprüche 1 bis 12, dadurch gekennzeichnet, dass der Speichermodul-Controller (**1000**) ferner einen mit der Steuerlogik (**1002**) gekoppelten Lesepuffer (**1038**) aufweist, der Daten speichert, die aus wenigstens einem der mehreren Speicherbauelemente (**312–315, 317–320; 1204–1207**) gelesen worden sind.

14. Speichersystem (**300**) nach einem der Ansprüche 1 bis 13, dadurch gekennzeichnet, dass der Speichermodul-Controller (**1000**) ferner einen Taktgenerator (**1010**) aufweist, der mit der Steuerlogik (**1002**) gekoppelt ist und ein erstes Taktsignal (**1024**) aus dem Speicherbus (**1023**) empfängt und ein zweites Taktsignal (**1030**) für die Speicherbauelemente (**312–315, 317–320; 1204–1207**) erzeugt.

Es folgen 17 Blatt Zeichnungen

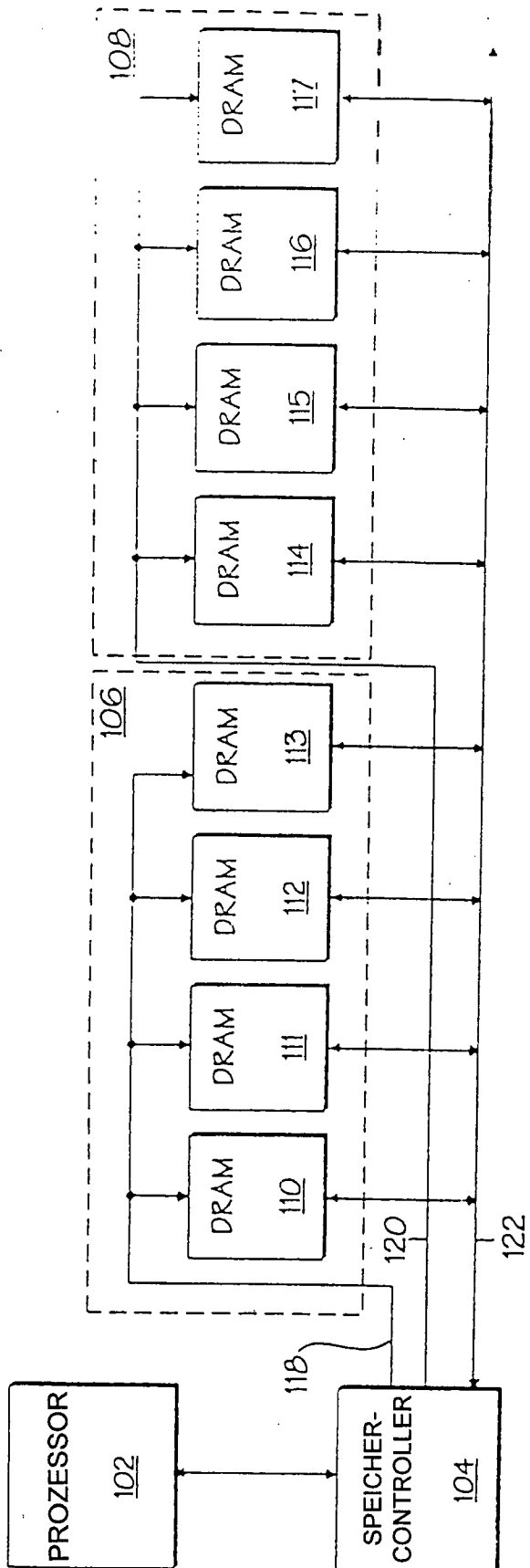


FIG. 1
(STAND DER TECHNIK)

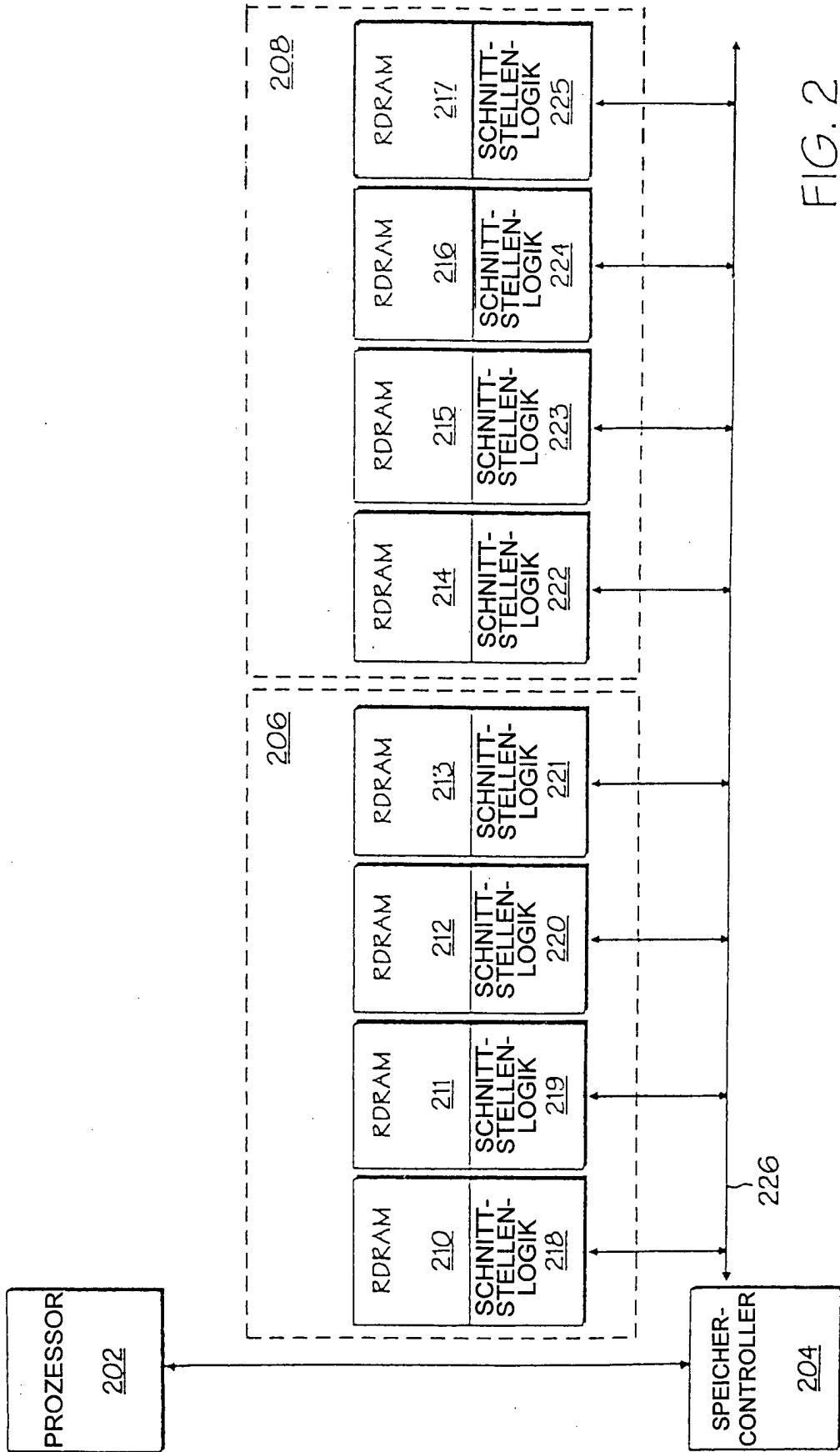


FIG. 2
(STAND DER TECHNIK)

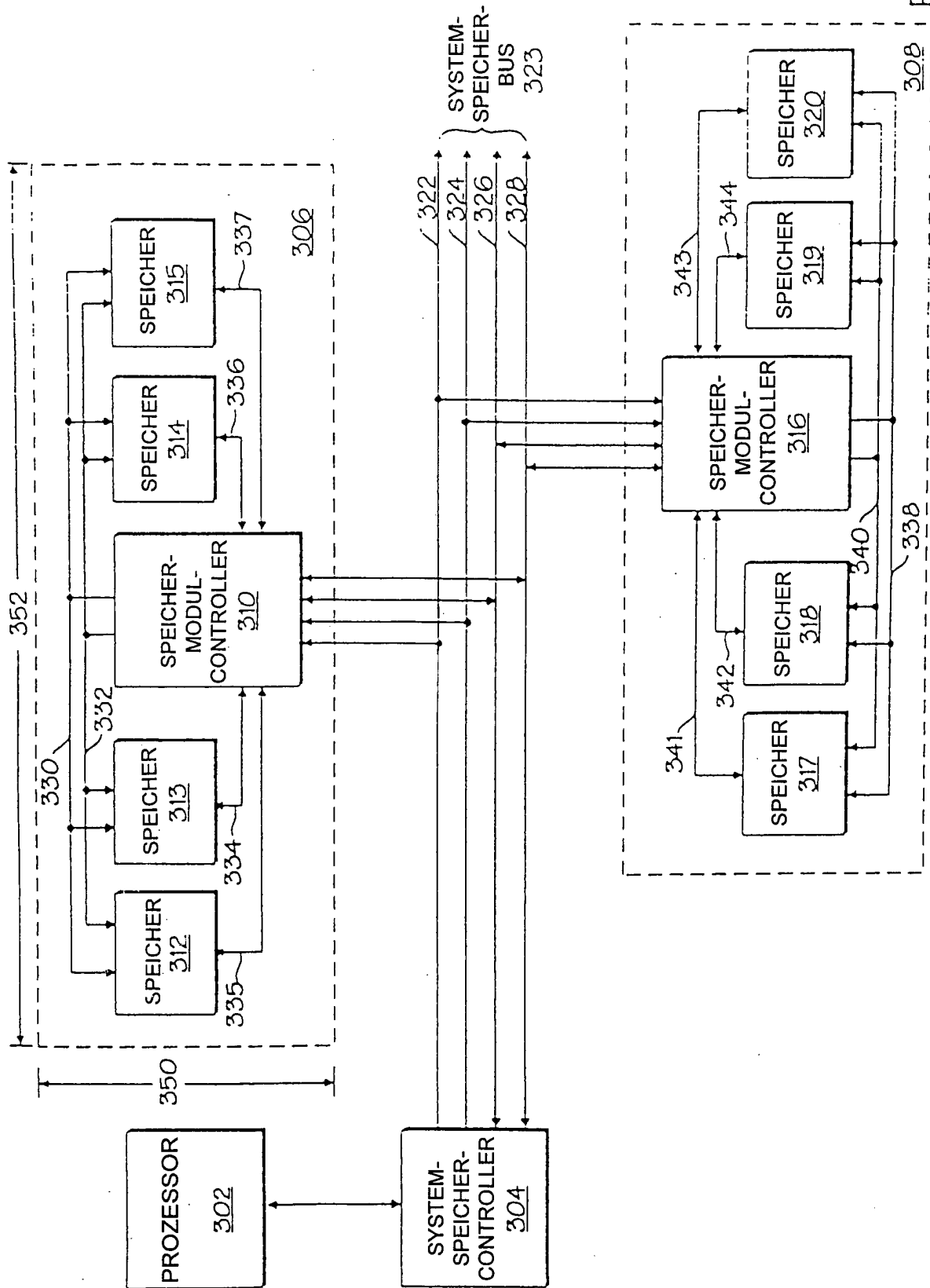


FIG. 3

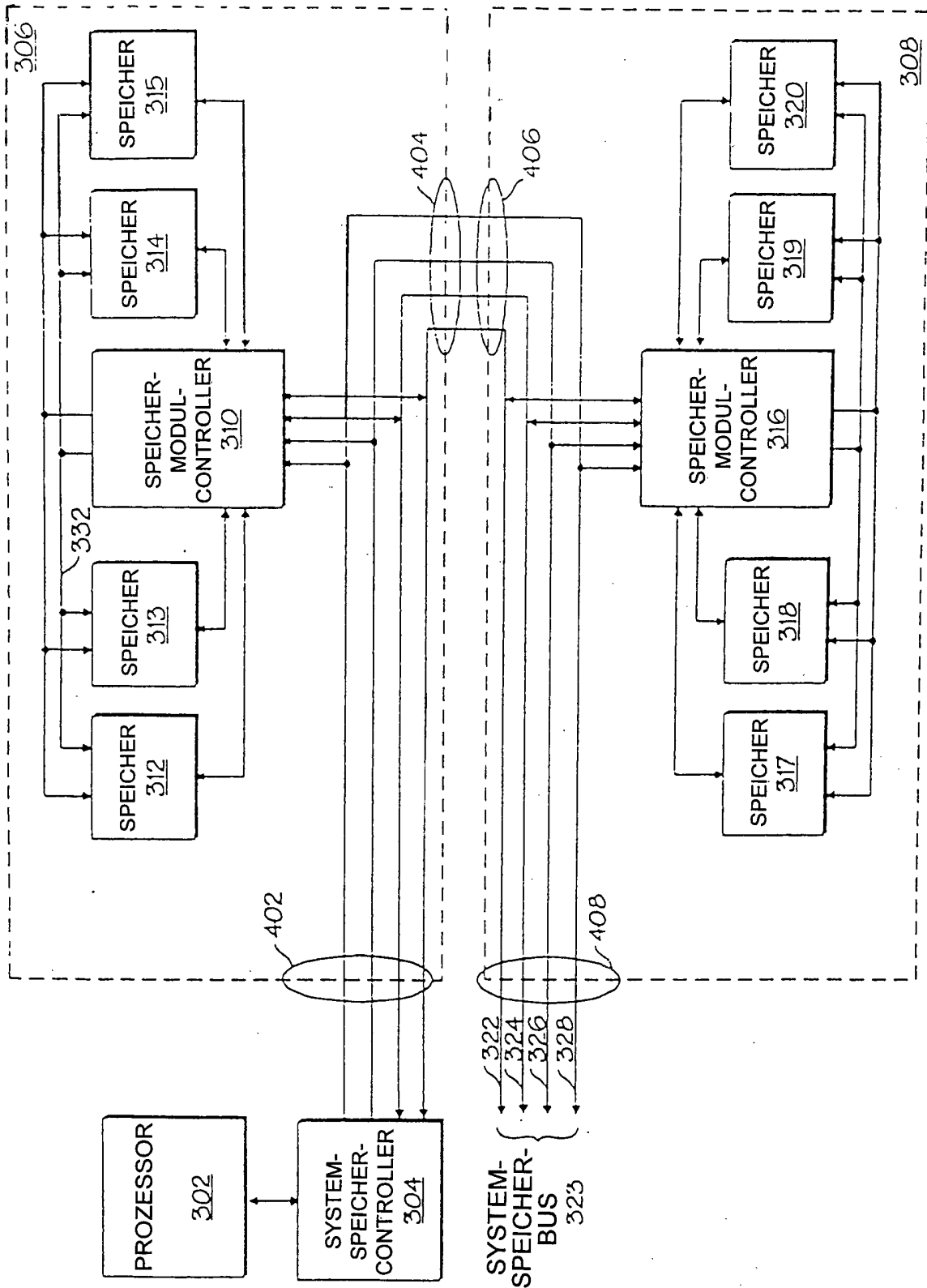
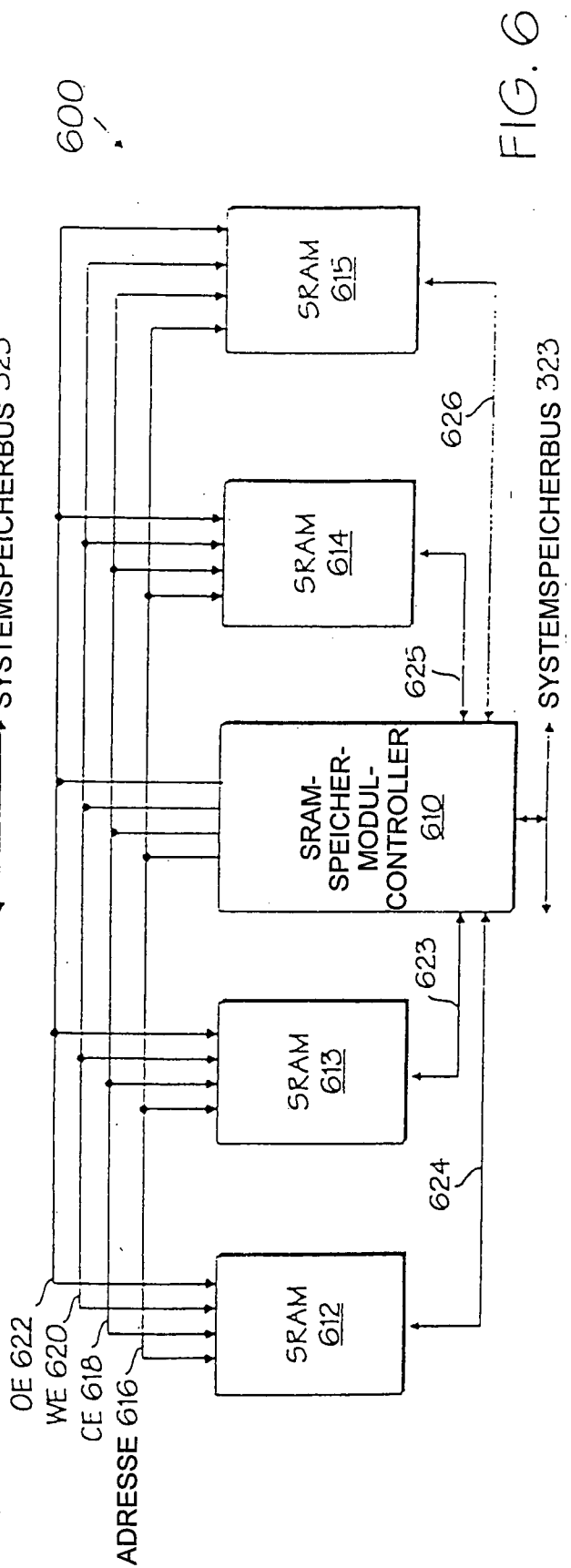
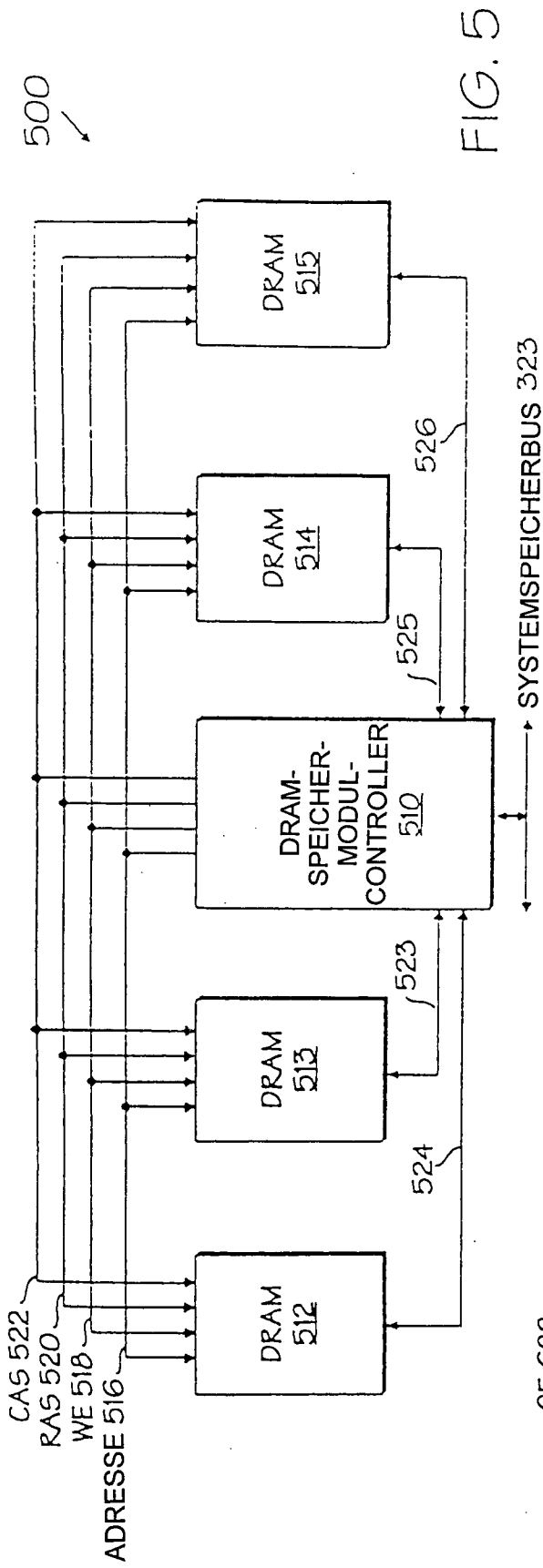


FIG. 4



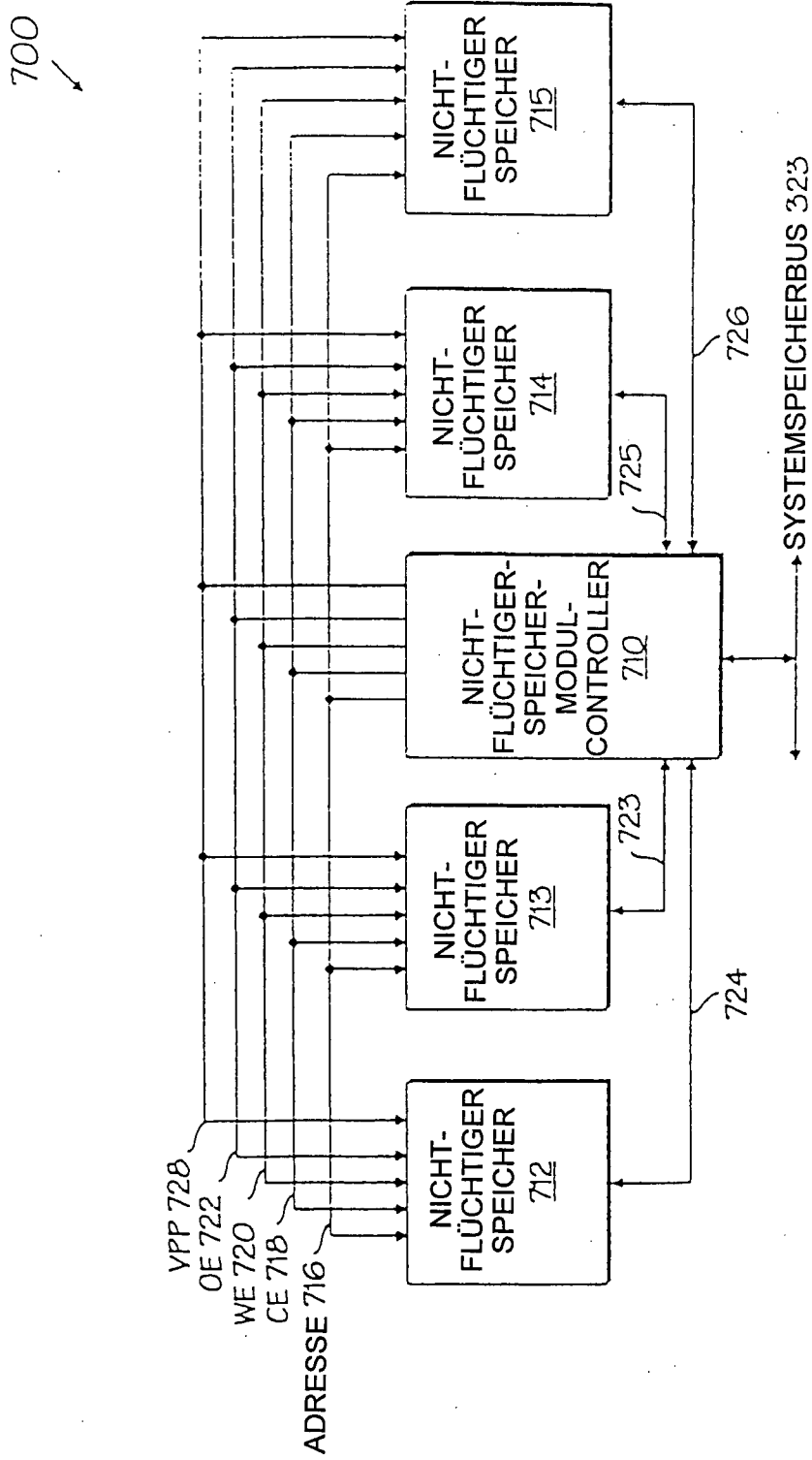


FIG. 7

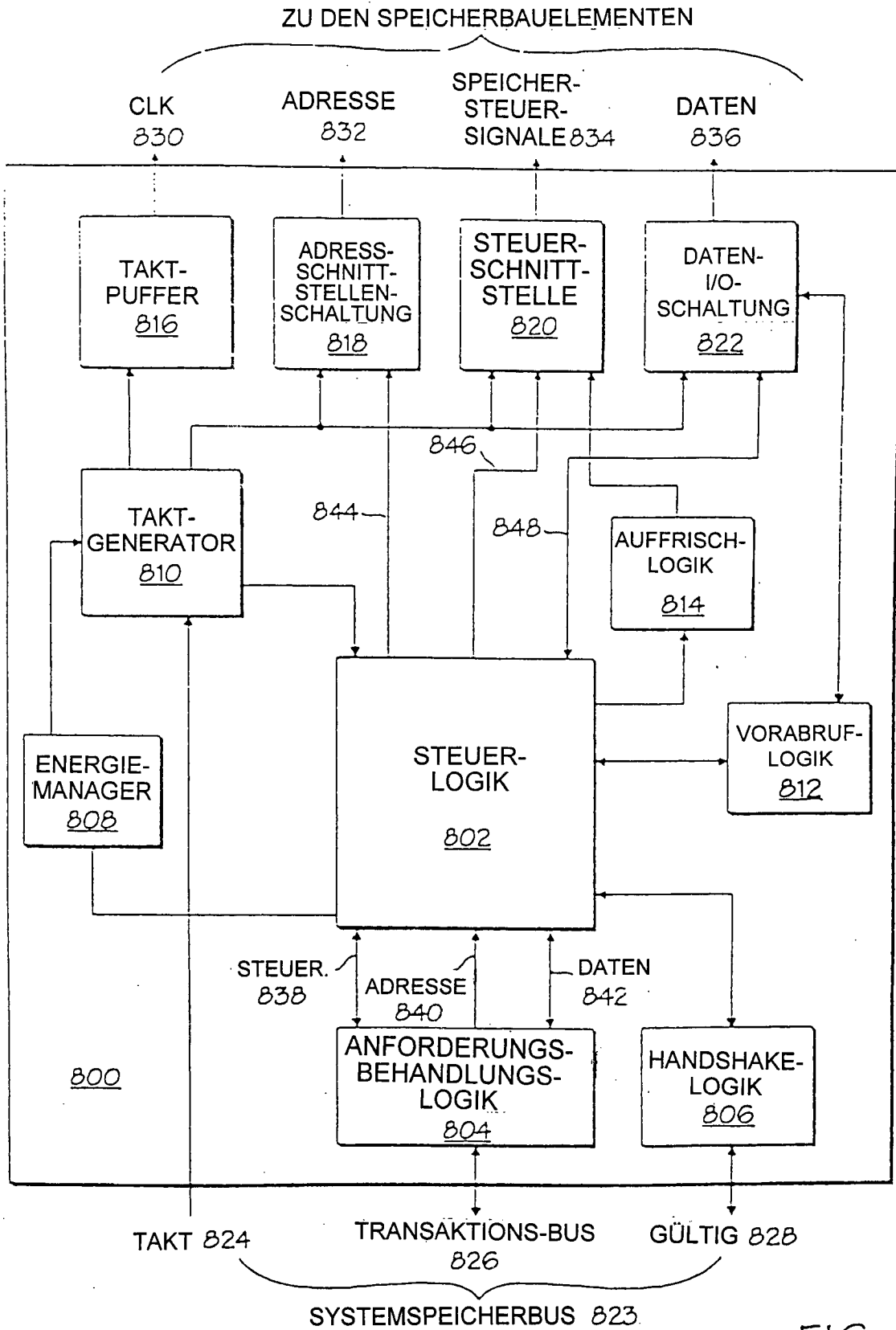


FIG. 8

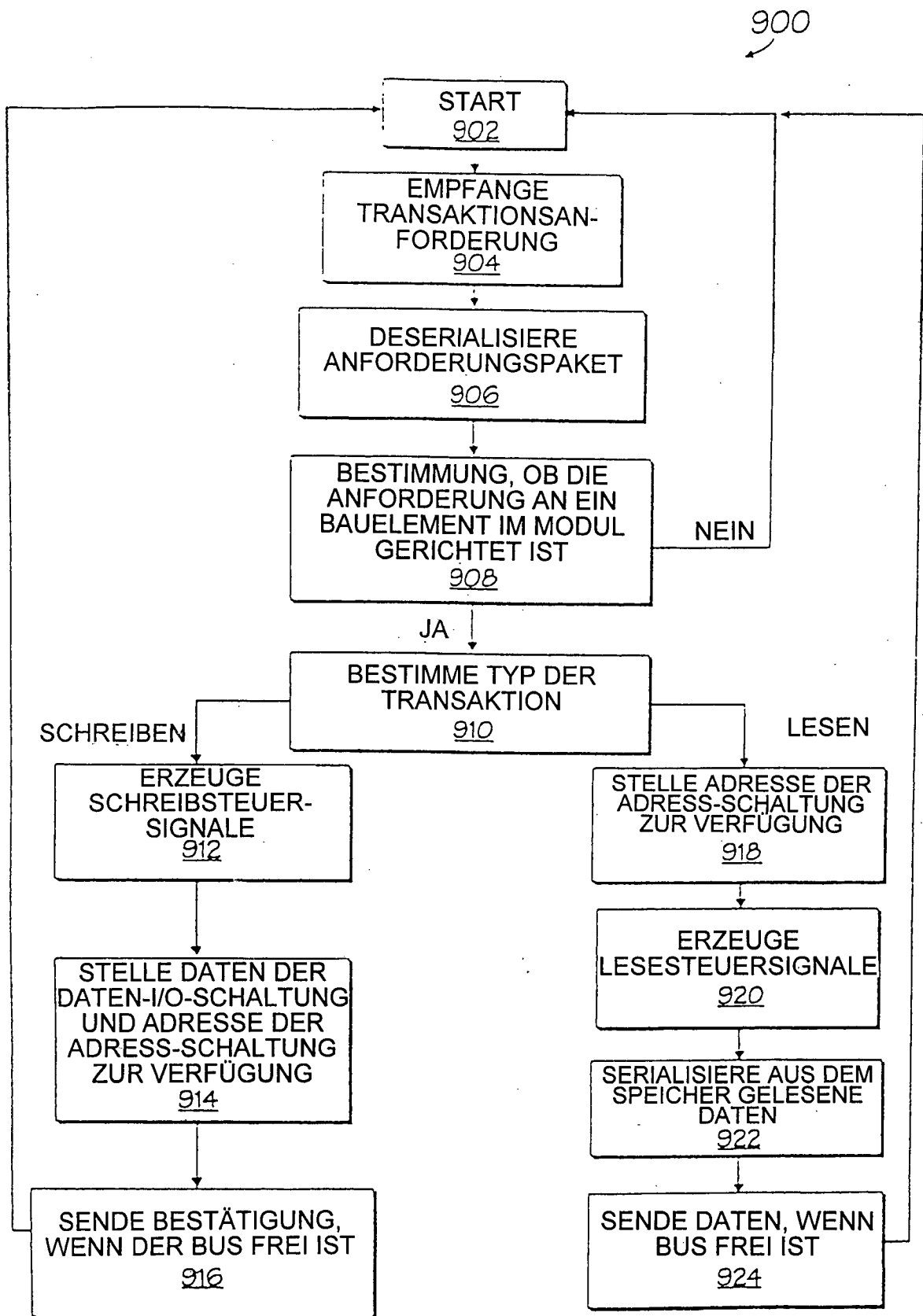


FIG. 9

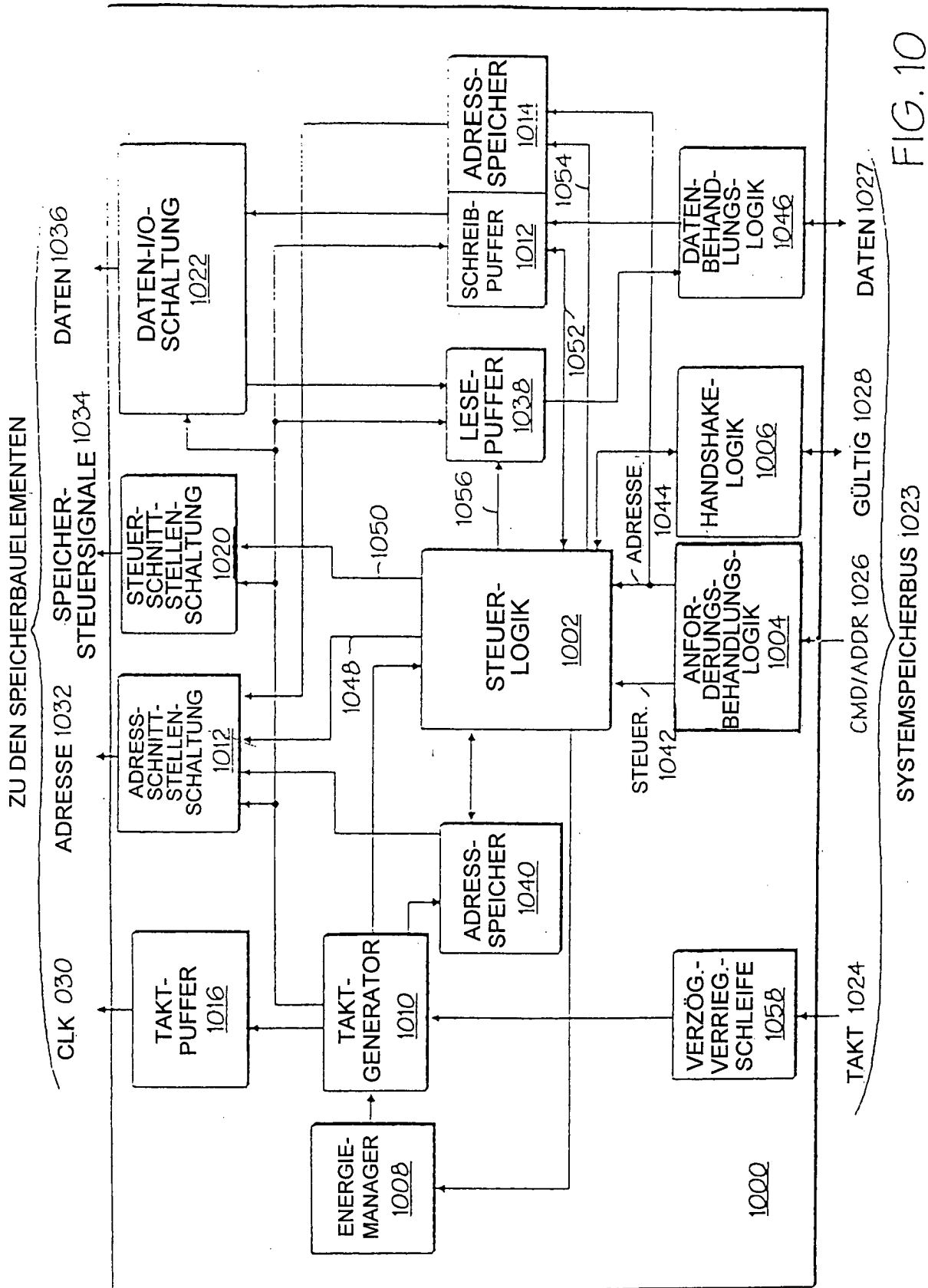


FIG. 10

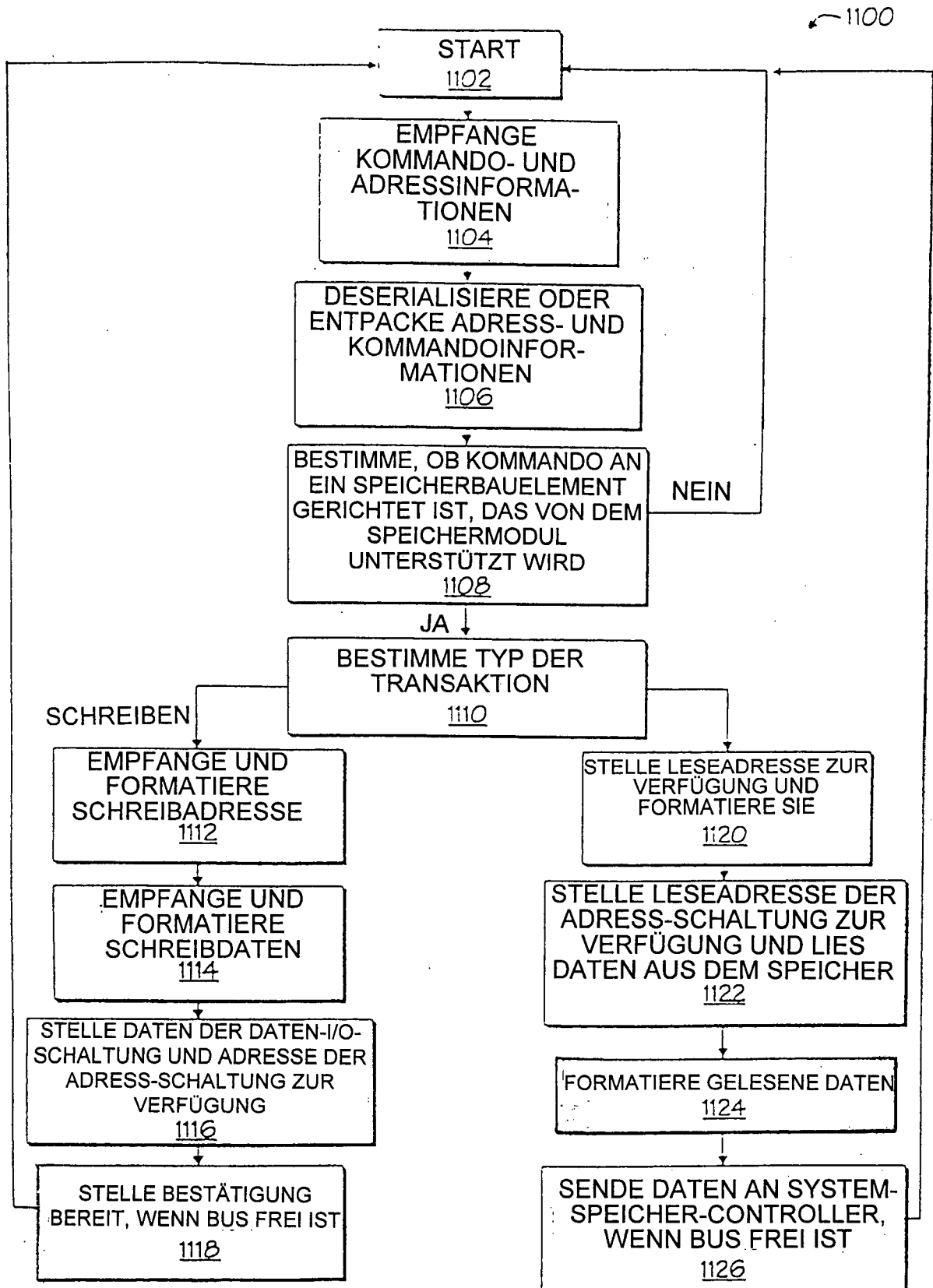


FIG. 11

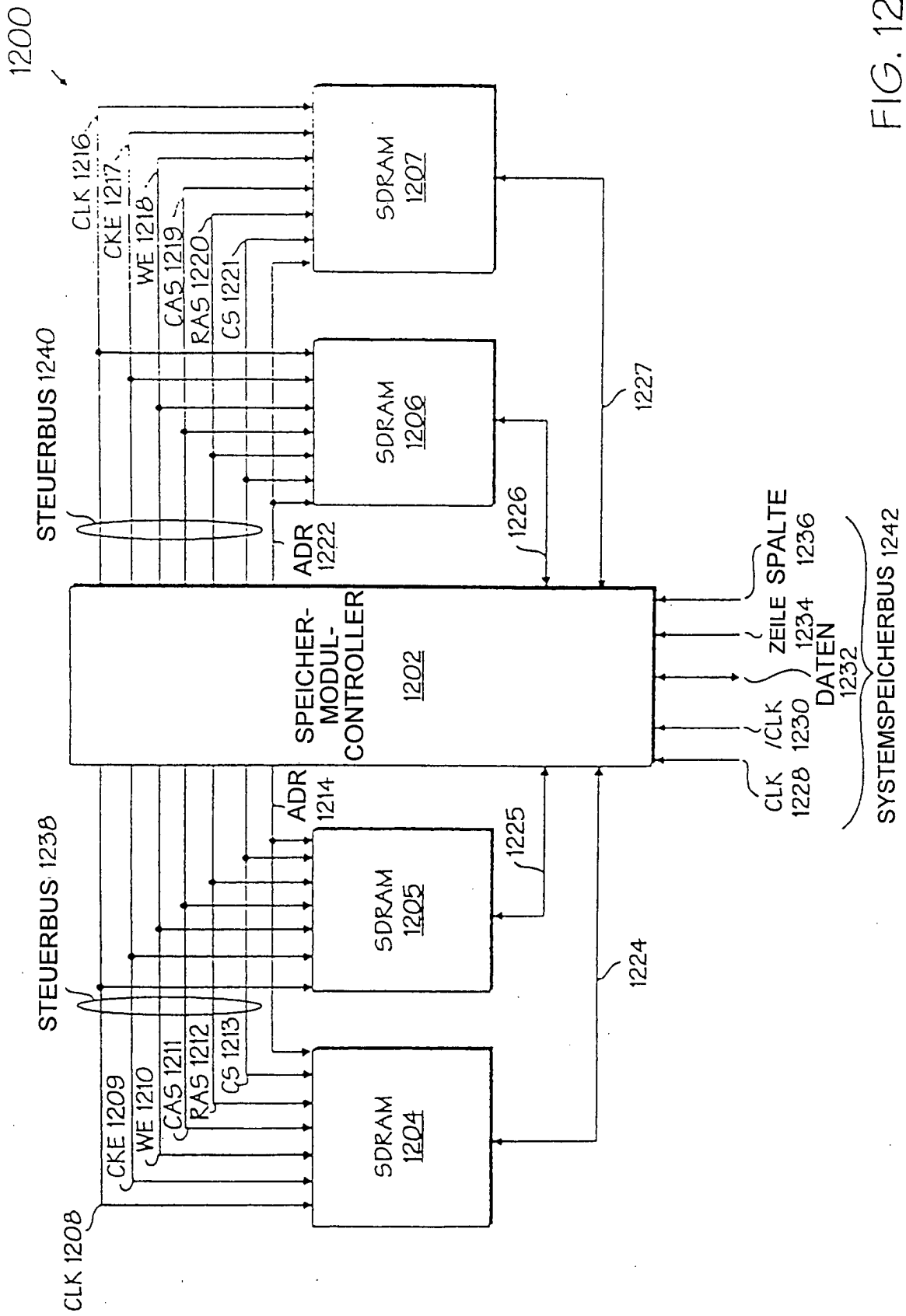


FIG. 12

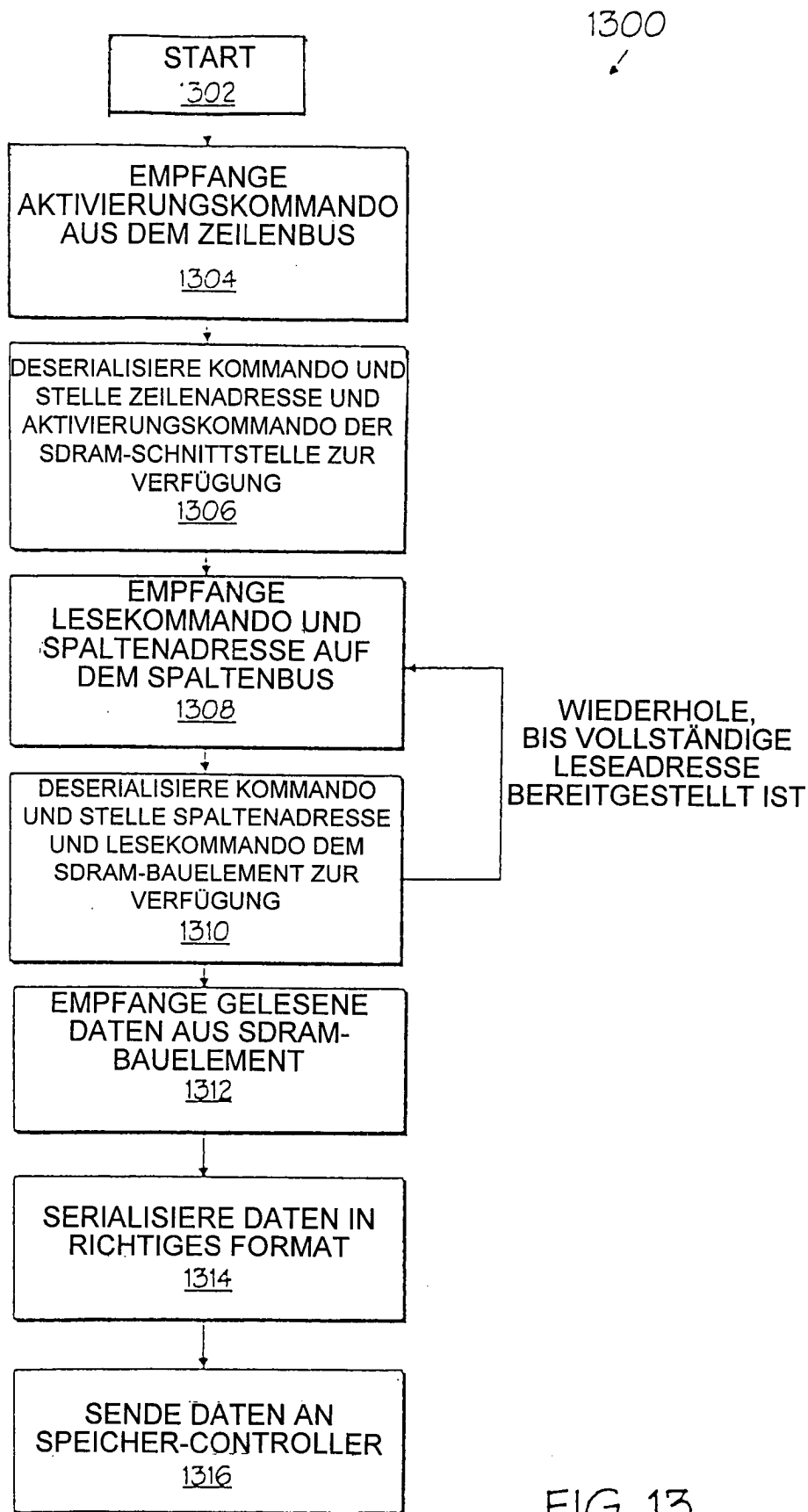


FIG. 13

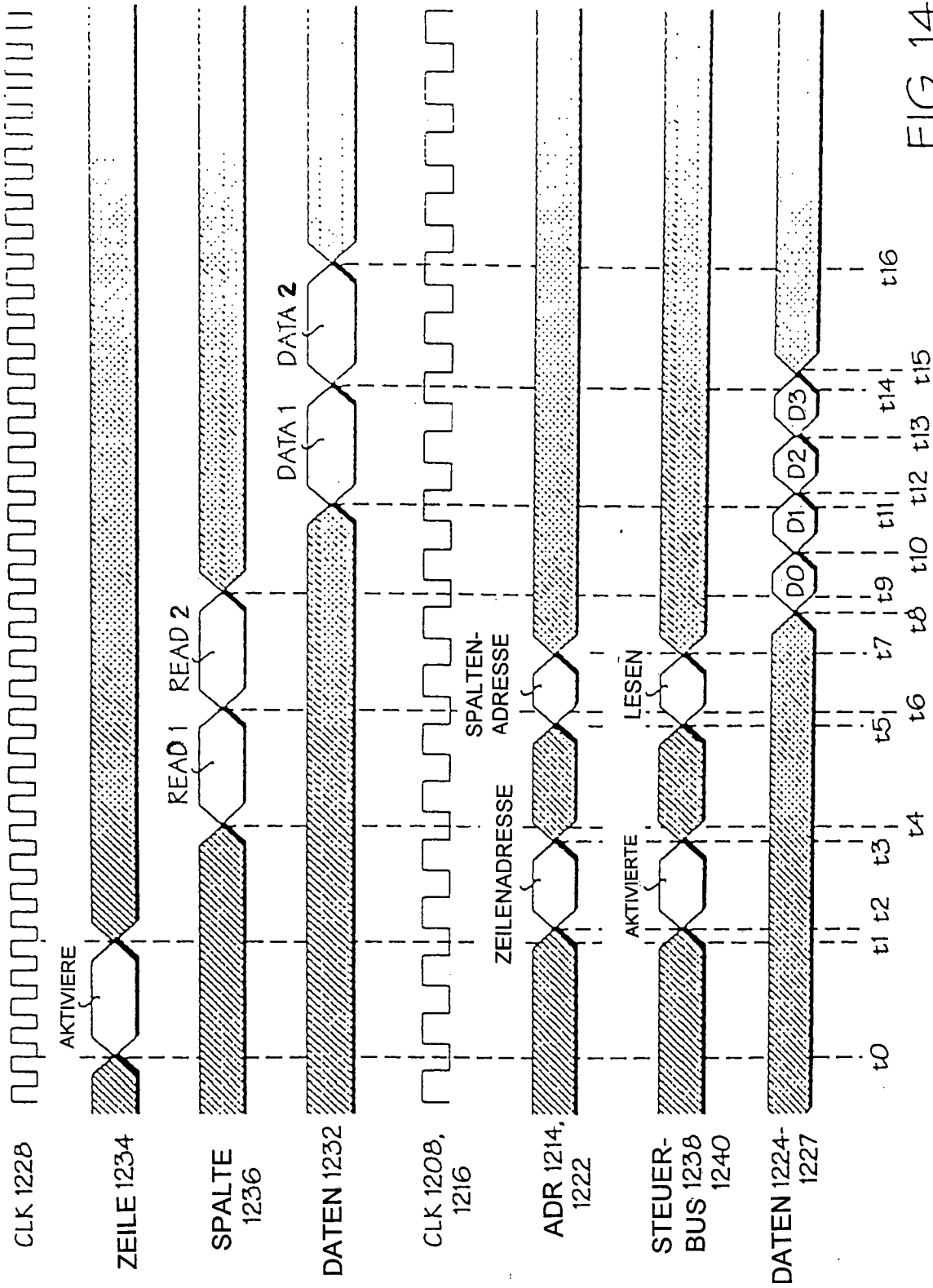


FIG. 14

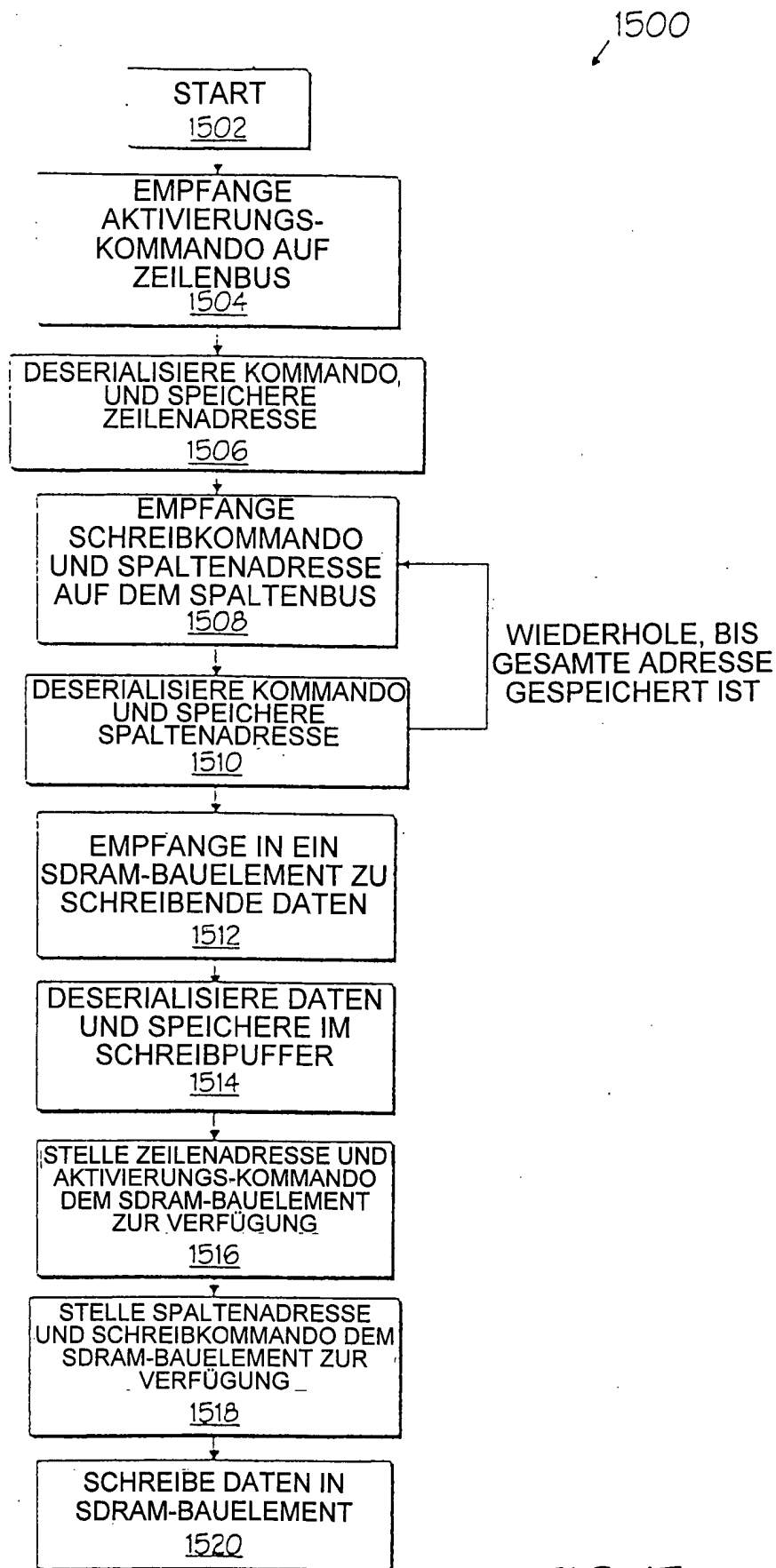


FIG. 15

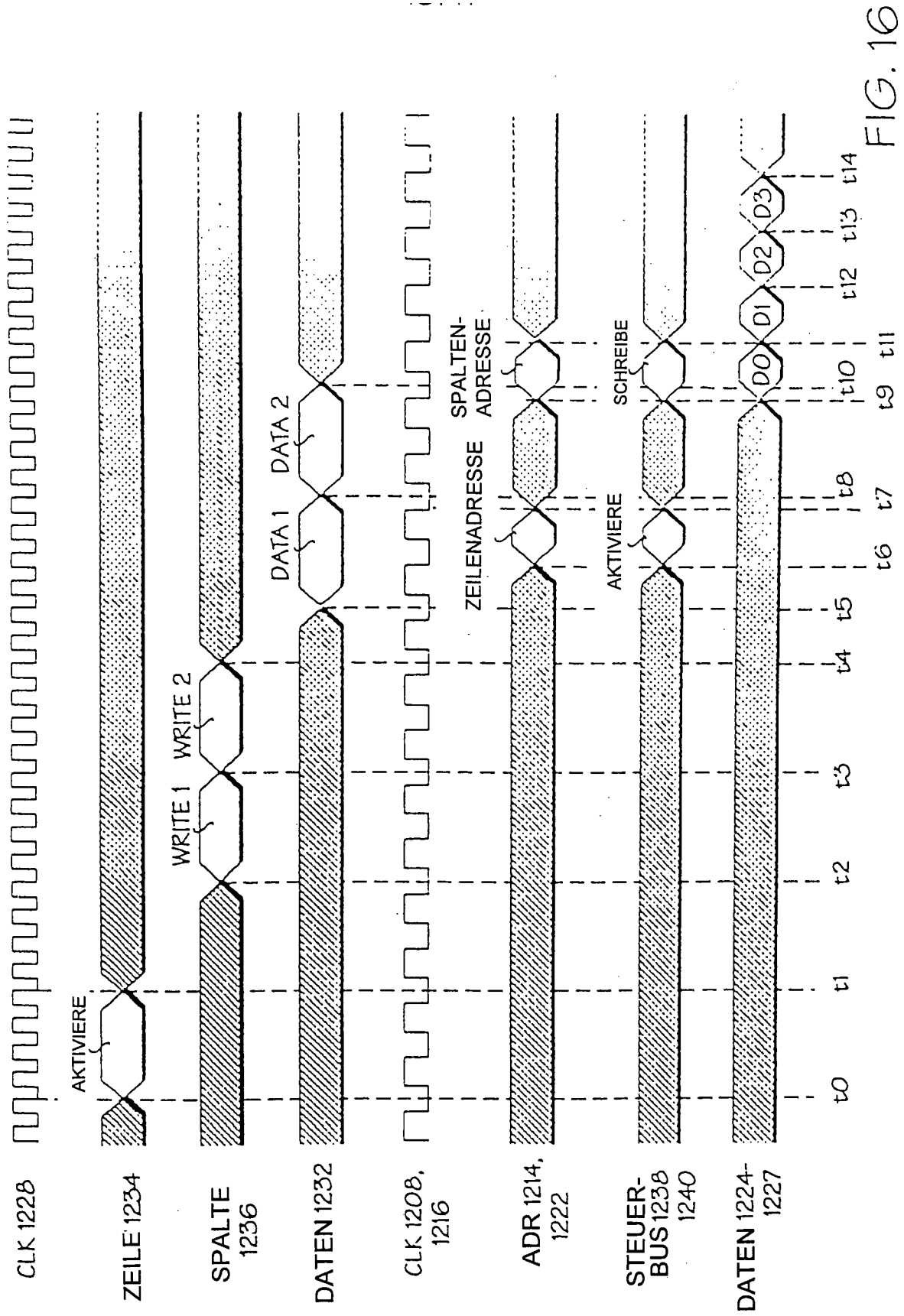


FIG. 16

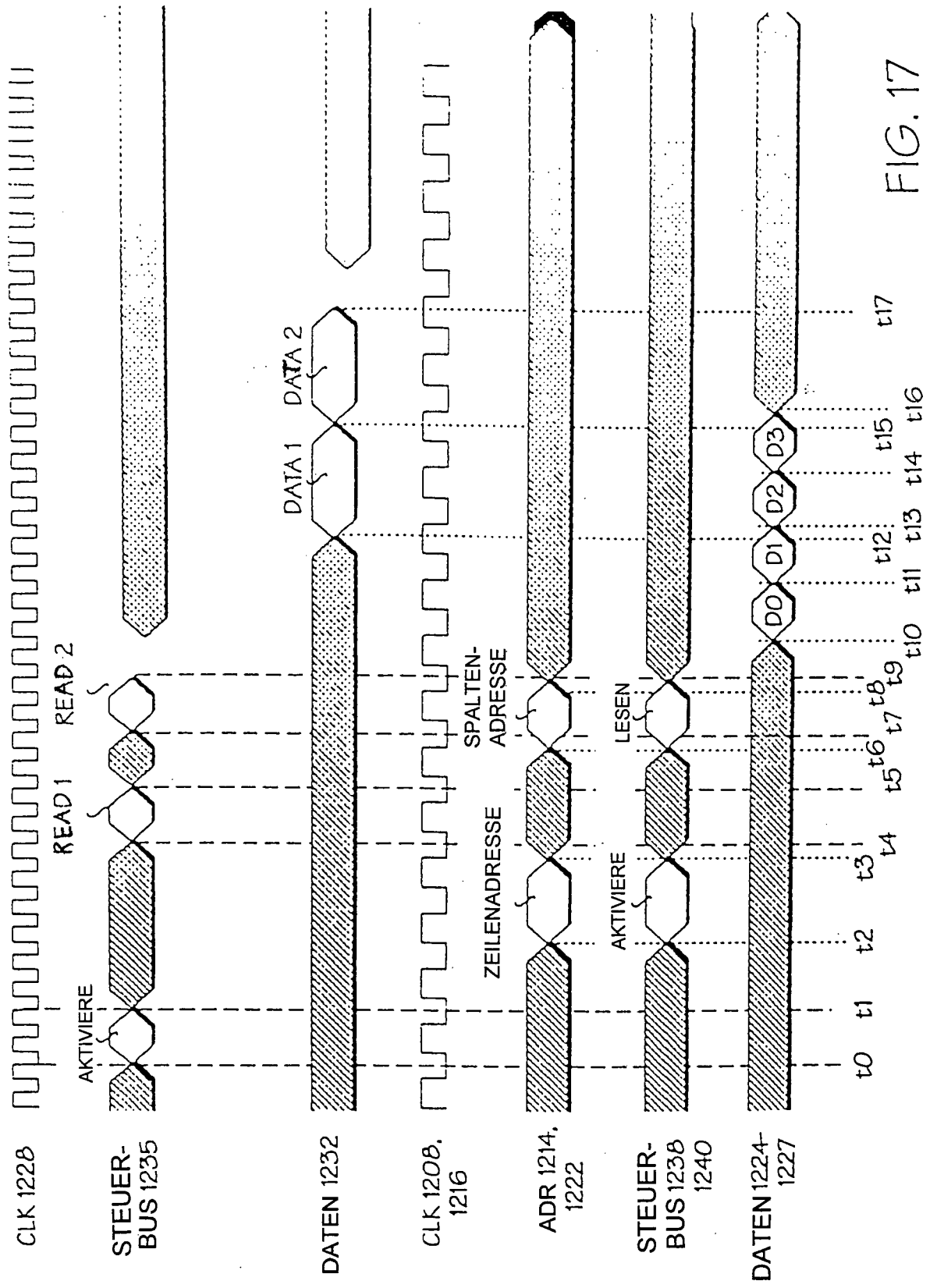


FIG. 17

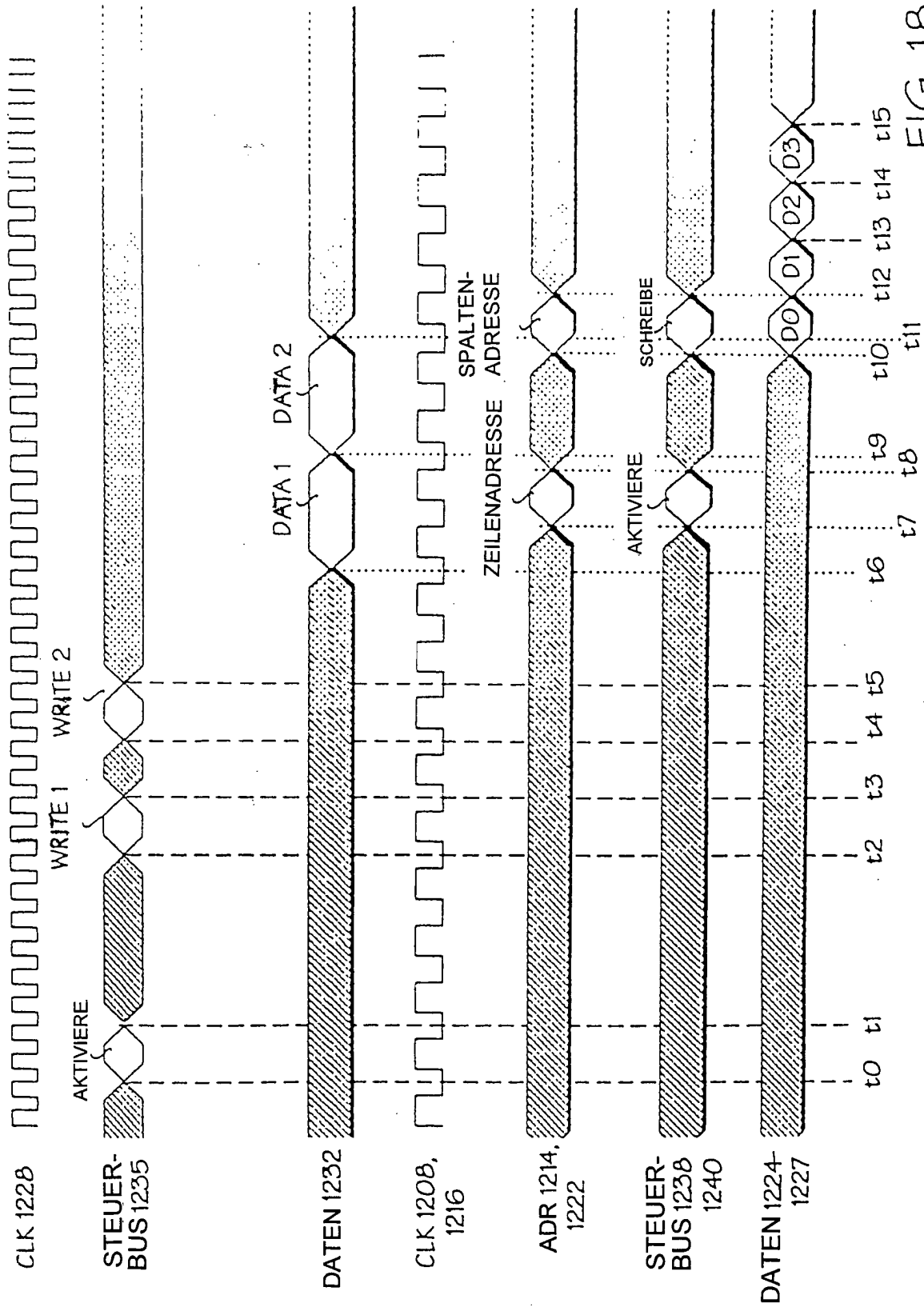


FIG. 18