(19) Deutsches Patent- und Markenamt

(10) **DE 11 2011 103 208 T5** 2013.10.02

(12) # Veröffentlichung

der internationalen Anmeldung mit der
(87) Veröffentlichungs-Nr.: **WO 2012/040649**
in deutscher Übersetzung (Art. III § 8 Abs. 2 IntPatÜG)
(21) Deutsches Aktenzeichen: **11 2011 103 208.0**
(86) PCT-Aktenzeichen: **PCT/US2011/053129**
(86) PCT-Anmeldetag: **23.09.2011**
(87) PCT-Veröffentlichungstag: **29.03.2012**
(43) Veröffentlichungstag der PCT Anmeldung
in deutscher Übersetzung: **02.10.2013**

(51) Int Cl.: ***G06F 13/14*** (2013.01)
G06F 13/16 (2013.01)
G06F 12/00 (2013.01)
G11C 7/10 (2013.01)

(30) Unionspriorität:
61/386,237 **24.09.2010** **US**

(71) Anmelder:
Texas Memory Systems, Inc., Houston, Tex., US

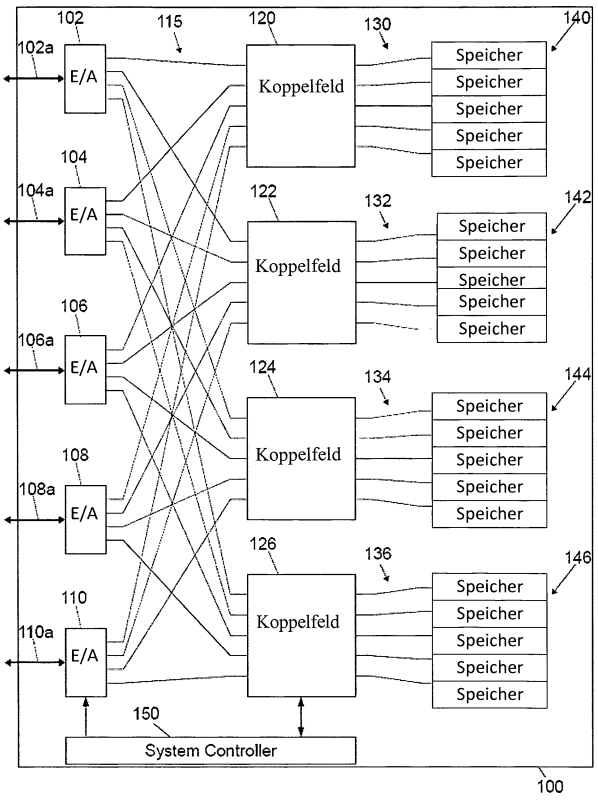
(74) Vertreter:
BARDEHLE PAGENBERG Partnerschaft Patentanwälte, Rechtsanwälte, 81675, München, DE

(72) Erfinder:
Frost, Holloway H., Houston, Tex., US; Hutsell, Rebecca, Houston, Tex., US

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Hochgeschwindigkeits-Speichersystem**

(57) Zusammenfassung: Die offen gelegten Ausführungsbeispiele betreffen ein Flash-basiertes Speichermodul mit serieller Hochgeschwindigkeits-Kommunikation. Das Flash-basierte Speichermodul umfasst unter Anderem eine Vielzahl von Ein-/Ausgabe-Modulen (E/A-Modulen), jedes von ihnen konfiguriert, um mit einem externen Gerät über eine oder mehrere externe Kommunikationsverbindungen zu kommunizieren, eine Vielzahl Flash-basierter Speicherkarten, von denen jede eine Vielzahl von Flash-Speichervorrichtungen umfasst, und eine Vielzahl von Koppelfeldelementen, von denen jedes mit einer jeweiligen der Flash-basierten Speicherkarten verbunden und so konfiguriert ist, dass es die Kommunikation eines jeden der E/A-Module mit der jeweiligen betreffenden Flash-basierten Speicherkarte erlaubt. Jedes E/A-Modul ist mit jedem Koppelfeldelement über eine serielle Hochgeschwindigkeits-Kommunikationsverbindung verbunden, und jedes Koppelfeldelement ist mit der jeweiligen der Flash-basierten Speicherkarten durch eine Vielzahl paralleler Kommunikationsverbindungen verbunden.



Beschreibung

QUERVERWEIS AUF
VERWANDTE ANWENDUNGEN:

[0001] Diese Anmeldung beansprucht die Priorität der am 24. September 2010 eingereichten vorläufigen US-Anmeldung Nr. 61/386,237 mit dem Titel „High-Speed Memory System"; die vorläufige Anmeldung wird hiermit durch Bezugnahme in ihrer Gesamtheit zum Bestandteil dieses Dokuments.

ERKLÄRUNG ÜBER DURCH [US]-
BUNDESMITTEL GEFÖRDERTE FORSCHUNG:

Nicht zutreffend.

VERWEIS AUF ANHANG:

Nicht zutreffend.

HINTERGRUND DER ERFINDUNG

[0002] Gebiet der Erfindung: Diese Offenlegung bezieht sich allgemein auf Speichersysteme, auf die von externen Host-Geräten und/oder externen Kommunikationsgeräten zugegriffen wird.

Beschreibung einschlägiger Technik:

[0003] Zur Entgegennahme eingehnder Daten und Datenanforderungen und zur Verarbeitung dieser Anforderungen, Daten zu speichern oder abzurufen, setzen Speichersysteme häufig eine Vielzahl von Verfahren und Vorrichtungen ein. Oft sind solche Speichersystemen in ihrer Bandbreite begrenzt, da die Anzahl ihrer Ein-/Ausgabe-Ports begrenzt ist und/oder Engstellen im System vorhanden sind. Solche Engstellen können durch die Verwendung relativ langsamer Datenbusse wie auch durch die Verwendung komplizierter Schaltungs- oder Übertragungsstrukturen bzw. -protokolle entstehen.

[0004] Dementsprechend wird ein effektiveres, leistungsfähigeres und optimales Hochgeschwindigkeits-Speichersystem benötigt.

ZUSAMMENFASSUNG DER ERFINDUNG:

[0005] Die offen gelegten Ausführungsformen beziehen sich auf Verfahren und Vorrichtungen für die Bereitstellung eines effektiveren, leistungsfähigeren und optimalen Hochgeschwindigkeits-Speichersystems. Im Allgemeinen beziehen sich die offen gelegten Ausführungsformen in einem Aspekt auf ein Flash-basiertes Speichermodul mit serieller Hochgeschwindigkeits-Kommunikation. Das Flash-basierte Speichermodul umfasst unter anderem eine Vielzahl von E/A-Modulen, wobei jedes E/A-Modul konfiguriert ist, um mit einem externen Gerät über eine oder mehrere externe Kommunikationsverbindungen zu kommunizieren, eine Vielzahl von Flash-basierten Speicherkarten, jede Flash-basierte Speicherkarte bestehend aus einer Vielzahl von Flash-Speichervorrichtungen, wobei jede Flash-Speichervorrichtung über einen physikalischen Speicherraum verfügt, der in Blöcke unterteilt ist, wobei jeder Block weiter in Seiten unterteilt ist und jede Seite eine einzeln adressierbare Speicherstelle darstellt, auf der Speicheroperationen durchgeführt werden und mehrere solcher Speicherstellen gleichzeitig in Ein-Block-Gruppierungen löschbar sind, sowie eine Vielzahl von Koppelfeldelementen, wobei jedes Koppelfeldelement mit einer jeweiligen der Flash-basierten Speicherkarten verbunden und konfiguriert ist, um jedem einzelnen der E/A-Module die Kommunikation mit der jeweiligen der Flash-basierten Speicherkarten zu erlauben. Jedes E/A-Modul ist mit einem Koppelfeldelement über eine serielle Hochgeschwindigkeits-Kommunikationsverbindung verbunden, wobei jede serielle Hochgeschwindigkeits-Kommunikationsverbindung jedem E/A-Modul ermöglicht, Befehle, Anweisungen bzw. Daten darstellende Bits an jedes Koppelfeldelement zu übertragen und von diesem zu empfangen; und jedes Koppelfeldelement ist mit einer jeweiligen der Flash-basierten Speicherkarten über eine Vielzahl paralleler Kommunikationsverbindungen verbunden, wobei jede parallele Kommunikationsverbindung ein Koppelfeldelement mit einer der Flash-Speichervorrichtungen der jeweiligen Flash-basierten Speicherkarte verbindet.

[0006] Im Allgemeinen betreffen die offen gelegten Ausführungsformen in einem weiteren Aspekt einen erweiterbaren Hochgeschwindigkeits-Speicher. Der erweiterbare Hochgeschwindigkeits-Speicher umfasst unter Anderem eine Leiterplatte, auf der Leiterplatte aufgebrachte Schnittstellenschaltkreise, so konfiguriert, um der Hochgeschwindigkeits-Speicherkarte den Empfang von Anweisungen, Befehle bzw. Daten darstellenden Bits von einem oder mehreren externen Geräten über eine oder mehrere serielle Hochgeschwindigkeits-Kommunikationsverbindungen zu erlauben, eine Vielzahl von auf der Leiterplatte montierten Speichervorrichtungen, wobei jede Speichervorrichtung über einen physikalischen Speicherraum verfügt, in welchem die Speicheroperationen durchgeführt werden, und einen auf der Leiterplatte montierten und mit den Schnittstellenschaltkreisen und der Vielzahl von Speichervorrichtungen verbundenen Controller, wobei der Controller so konfiguriert ist, dass er die Kommunikation zwischen den Schnittstellenschaltkreisen und jeder einzelnen Speichervorrichtung zur Ausführung von Speicheroperationen steuert. Die Schnittstellenschaltkreise sind mit dem Controller über eine Vielzahl serieller Hochgeschwindigkeits-Kommunikationsleitungen verbunden, wobei jede serielle Hochgeschwindigkeits-Kommunikationsleitung einer der seriellen Hochgeschwindigkeits-Kommunika-

tionsverbindungen entspricht, und wobei der Controller mit der Vielzahl der Speichervorrichtungen über eine vordefinierte Anzahl paralleler Kommunikationsleitungen verbunden ist, der Controller so konfiguriert, dass er die Befehle, Anweisungen bzw. Daten darstellenden Bits aus den seriellen Hochgeschwindigkeits-Kommunikationsverbindungen aus einem seriellen Format in ein paralleles Format umwandelt.

[0007] Im Allgemeinen bezieht sich noch ein weiterer Aspekt der offen gelegten Ausführungsformen auf ein Speichermodul mit serieller Hochgeschwindigkeits-Kommunikation. Das Speichermodul umfasst unter Anderem eine erste Vielzahl von Eingabeverarbeitungsblöcken und eine zweite Vielzahl von Eingabeverarbeitungsblöcken, die einzelnen Eingabeverarbeitungsblöcke so konfiguriert, dass sie Bits empfangen, welche Befehle, Anweisungen bzw. Daten in einem seriellen Format darstellen, und die Befehle, Anweisungen bzw. Daten darstellenden Bits entsprechend einem parallelen Format neu anordnen, eine Vielzahl von Speichervorrichtungen, wobei jede Speichervorrichtung über einen physikalischen Speicherraum verfügt, in welchem Speicheroperationen ausgeführt werden, wie auch einen mit der ersten und der zweiten Vielzahl der Eingabeverarbeitungsblöcke und mit den Speichervorrichtungen verbundenen Controller, der Controller so konfiguriert, dass er die Kommunikation zwischen der ersten und der zweiten Vielzahl von Eingabeverarbeitungsblöcken und jeder einzelnen Speichervorrichtung zur Ausführung der Speicheroperationen steuert. Der Controller umfasst unter Anderem: (a) einen ersten und einen zweiten mit der ersten bzw. mit der zweiten Vielzahl von Eingabeverarbeitungsblöcken verbundenen Mehrkanalpuffer, jeder Mehrkanalpuffer konfiguriert, um die Bits, welche Befehle, Anweisungen bzw. Daten darstellen, aus der ersten bzw. aus der zweiten Vielzahl von Eingabeverarbeitungsblöcken in parallelem Format zu empfangen und aus den Befehle, Anweisungen bzw. Daten darstellenden Bits eine vorher festgelegte Anzahl von Wörtern zu bilden, wobei jedes einzelne Wort aus einer vorher festgelegten Anzahl von Bits zusammengesetzt ist; (b) einen mit dem ersten bzw. mit dem zweiten Mehrkanalpuffer verbundenen ersten und zweiten Fehlerkorrektur- und Datenschutzschaltkreis, die ersten und zweiten Fehlerkorrektur- und Datenschutzschaltkreise konfiguriert, um die Wörter aus dem ersten bzw. aus dem zweiten Mehrkanalpuffer zu empfangen, mithilfe der Wörter einen oder mehrere Fehlerkorrektur-Codebits zu generieren, die Fehlerkorrektur-Codebits für jedes einzelne Wort an das Wort anzufügen und jedes Wort mit den daran angefügten Fehlerkorrektur-Codebits auszugeben; (c) einen mit den ersten bzw. den zweiten Fehlerkorrektur- und Datenschutzschaltkreisen verbundenen ersten Ausgabepuffer und zweiten Ausgabepuffer, wobei der erste und der zweite Ausgangspuffer so konfiguriert sind, dass sie abwechselnd die Wörter mit den hinzugefügten Fehlerkorrektur-Codebits aus den ersten und zweiten Fehlerkorrektur- und Datenschutzschaltkreisen dergestalt empfangen, dass ein erstes Wort aus einem der ersten und zweiten Fehlerkorrektur- und Datenschutzschaltkreise einem der ersten und der zweiten Ausgangspuffer zur Verfügung gestellt wird, und ein nächstes Wort von einem weiteren der ersten und zweiten Fehlerkorrektur- und Datenschutz-Schaltkreise einem anderen der ersten und zweiten Ausgabepuffer zur Verfügung gestellt wird, und (d) einen Speicherpuffer, konfiguriert, um die Wörter mit den ihnen aus den ersten und zweiten Ausgabepuffern beigefügten Fehlerkorrektur-Codebits zu empfangen und eine vorbestimmte Anzahl von Wörtern auf vorbestimmte Art und Weise zur Bildung eines Super-Wortes zu kombinieren.

KURZBESCHREIBUNG DER ZEICHNUNGEN:

[0008] Abb. 1 veranschaulicht ein Hochgeschwindigkeits-Speichersystem, welches in Übereinstimmung mit bestimmten Lehren dieser Offenlegung aufgebaut ist.

[0009] Abb. 1A–Abb. 1C veranschaulichen Beispiele von DMA Read-Modify-Write-(Lese-Ändern-Schreib-)Operationen, die im Hochgeschwindigkeits-Speichersystem dieser Offenlegung durchgeführt werden können.

[0010] Abb. 2 veranschaulicht ein Ausführungsbeispiel eines E/A-Moduls, das im Hochgeschwindigkeits-Speichersystem dieser Offenlegung verwendet werden kann.

[0011] Abb. 3 veranschaulicht ein alternatives Ausführungsbeispiel eines E/A-Moduls, das im Hochgeschwindigkeits-Speichersystem dieser Offenlegung verwendet werden kann.

[0012] Abb. 4 veranschaulicht einen beispielhaften Ansatz zur Umsetzung eines Koppelfeldmoduls des oben in Verbindung mit Abb. 1 beschriebenen Typs.

[0013] Abb. 5 veranschaulicht ein Ausführungsbeispiel einer Speicherkarte, die im System der vorliegenden Offenlegung verwendet werden kann.

[0014] Abb. 6 veranschaulicht eine alternative Ausführungsform einer Speicherkarte, die im Hochgeschwindigkeits-Speichersystem der vorliegenden Offenlegung verwendet werden kann.

[0015] Abb. 7 veranschaulicht ein alternatives Hochgeschwindigkeits-Speichersystem, in welchem die E/A-Module direkt mit den Speichersteckkarten kommunizieren, ohne die Koppelfeldmodule zu benutzen.

[0016] Abb. 8 veranschaulicht ein Ausführungsbeispiel einer Hochgeschwindigkeits-Speichersteckkarte, die im Hochgeschwindigkeits-Speichersystem von Abb. 7 verwendet werden kann.

[0017] Abb. 9 veranschaulicht Teile eines Beispiels einer Hochgeschwindigkeits-Speichersteckkarte in größerem Detail.

[0018] Abb. 10A veranschaulicht ein Ausführungsbeispiel eines Serialisierungs-/Deserialisierungs- und Verpackungsmoduls für eine Hochgeschwindigkeits-Speichersteckkarte.

[0019] Abb. 10B–Abb. 10D veranschaulichen ein Ausführungsbeispiel eines first-in-first-out (FIFO) Speicherpuffers für eine Hochgeschwindigkeits-Speichersteckkarte.

[0020] Abb. 11 veranschaulicht ein Ausführungsbeispiel eines FIFO-Multikanalpuffers für eine Hochgeschwindigkeits-Speichersteckkarte.

[0021] Abb. 12 und Abb. 12A veranschaulichen einen Beispielablauf für das Verschieben von Daten aus einem Fehlerkorrektur- und Datenschutz-Schaltkreis in den FIFO-Ausgangspuffer einer Hochgeschwindigkeits-Speichersteckkarte.

[0022] Abb. 13A und Abb. 13B veranschaulichen einen Beispielablauf für das Verschieben von Daten über mehrere einzelne serielle Hochgeschwindigkeits-Kommunikationsverbindungen.

[0023] Abb. 14A–Abb. 14D veranschaulichen Beispielabläufe für die Vermeidung der unsachgemäßen „Vermischung" von Daten aus unterschiedlichen SCHREIB-Operationen beim Verschieben der Daten über die seriellen Hochgeschwindigkeits-Kommunikationsverbindungen.

[0024] Abb. 15 veranschaulicht einen Beispielablauf für die Arbitration zwischen LESE- und SCHREIB-Zugang für Daten, die über die seriellen Hochgeschwindigkeits-Kommunikationsverbindungen $|I|_{[VH1]}$ das Hochgeschwindigkeits-Speichersystem dieser Offenlegung verschoben werden.

[0025] Abb. 16 veranschaulicht einen Beispiel-Controller und die Verbindungen zwischen dem Controller und dem physikalischen RAM-Speicher im Hochgeschwindigkeits-Speichersystem dieser Offenlegung.

[0026] Abb. 17 veranschaulicht ein Operationsbeispiel des Controllers von Abb. 16 über zwei grundlegende Speicherzyklen im Hochgeschwindigkeits-Speichersystem dieser Offenlegung.

DETAILLIERTE BESCHREIBUNG:

[0027] Die oben beschriebenen Abbildungen und die Textbeschreibung spezifischer Strukturen und Funktionen unten erfolgen nicht zu dem Zweck, den Umfang der Erfindung der Anmelder oder den Umfang der beiliegenden Ansprüche einzuschränken. Vielmehr dienen die Abbildungen und Textbeschreibungen dazu, dem Fachmann die Herstellung und Verwendung der Erfindungen zu vermitteln, für welche der Schutz des Patentrechts beantragt wird. Fachleute auf dem Gebiet werden erkennen, dass aus Klarheits- und Verständnisgründen nicht alle Merkmale einer kommerziellen Ausführungsform der Erfindungen beschrieben oder dargestellt sind. Ebenso werden Fachleute auf diesem Gebiet erkennen, dass die Entwicklung einer tatsächlichen kommerziellen Ausführungsform, in der Aspekte der vorliegenden Erfindung enthalten sind, zahlreiche umsetzungsspezifische Entscheidungen erfordert, damit das ultimative Ziel des Erfinders zur kommerziellen Ausführung erreicht wird. Zu solchen umsetzungsspezifischen Entscheidungen zählen gegebenenfalls die Konformität mit systembezogenen, geschäftsbezogenen, behördenbezogenen und anderen Einschränkungen, die je nach spezifischer Umsetzung, jeweiligem Standort und jeweiligem Zeitpunkt unterschiedlich sein können, und sind wahrscheinlich nicht nur auf diese beschränkt. Zwar sind die Bemühungen eines Entwicklers gegebenenfalls komplex und zeitaufwändig, solche Bemühungen würden jedoch dessen ungeachtet für einen Fachmann, der den Vorteil dieser Offenlegung hat, ein Routinevorhaben darstellen. Es ist darauf hinzuweisen, dass die in diesem Dokument offen gelegten und erläuterten Erfindungen zahlreichen und unterschiedlichen Veränderungen und alternativen Formen unterliegen können. Nicht zuletzt darf die Verwendung eines Begriffs im Singular wie „ein/e/s" nicht so betrachtet werden, als dass durch ihn die Anzahl eines Elements beschränkt wird. Auch werden Begriffe, die eine räumliche Beziehung beschreiben, wie „oben", „unten", „links", „rechts", „obere/r/s", „untere/r/s", „nach unten", „nach oben", „seitlich" und ähnliche in der Textbeschreibung zur Klarheit beim spezifischen Verweis auf die Abbildungen verwendet, dienen jedoch nicht dazu, den Umfang der Erfindung oder der beigefügten Ansprüche einzuschränken.

BEISPIEL EINE SPEICHERSYSTEMS

[0028] In den Zeichnungen und insbesondere in Abb. 1 ist ein in Übereinstimmung mit bestimmten Lehren dieser Offenlegung aufgebautes Hochgeschwindigkeits-Speichersystem **100** veranschaulicht. Im Allgemeinen empfängt das Hochgeschwindigkeits-Speichersystem **100** datenbezogene Anforderungen wie LESE- und SCHREIB-Anforderungen von externen Host-Geräten und verarbeitet diese Anforderungen, um Daten im physikalischen Speicher

zu speichern bzw. aus dem physikalischen Speicher abzurufen.

[0029] Das beispielhafte Speichersystem **100** umfasst und nutzt eine Vielzahl von E/A-Modulen, seriellen Hochgeschwindigkeits-Kommunikationskanälen, konfigurierbaren Koppelfeldmodulen, parallelen Speicherbussen und physikalischen Speicherkarten zur Bereitstellung eines verknüpfenden Speichersystems mit hoher Bandbreite. Weitere Details des Systems und seiner vielen neuartigen Aspekte sind unter nachzulesen.

[0030] Unter Bezugnahme auf Abb. 1 umfasst das System **100** eine Vielzahl von Eingabe-/Ausgabe-Schnittstellenmodulen (E/A-Schnittstellenmodule) **102**, **104**, **106**, **108** und **110**. Zwar sind im betreffenden Beispiel fünf E/A-Module veranschaulicht, die Anzahl der E/A-Module ist jedoch nicht von entscheidender Bedeutung und unterliegt auch Änderungen. Jedes E/A-Modul ist über einen Host-Kommunikationskanal (**102a**, **104a**, **106a**, **108a** bzw. **110a**) mit einem oder mehreren externen Host-Geräten gekoppelt. Die Kanäle **102a**, **104a**, **106a**, **108a** und **110a** erlauben externen Geräten – beispielsweise Servern oder anderen Geräten, die als Host betrieben werden können – Befehle, Anweisungen und Daten von den E/A-Schnittstellenmodulen **102**, **104**, **106**, **108** und **110** zu empfangen oder an diese zu senden.

[0031] Die Host-Kommunikationskanäle können unterschiedliche Formen annehmen und sich nach einer Vielzahl unterschiedlicher Protokolle wie Fibre Channel, InfiniBand, Ethernet und Front Panel Data Port (FPDP) richten. Zwar ist der genaue physikalische und logische Aufbau der Kommunikationsverbindungen für das offen gelegte System nicht von entscheidender Bedeutung, jedoch ist es für die vollständige Nutzung der vom offen gelegten System gebotenen Vorteile wünschenswert, dass die Host-Kommunikationsverbindungen in der Lage sind, Datenübertragungen von hoher Bandbreite zu unterstützen.

[0032] Im Beispiel von Abb. 1 ist jedes der E/A-Module **102**, **104**, **106**, **108** und **110** über eine Vielzahl serieller Hochgeschwindigkeits-Kommunikationskanäle (in Abb. 1 gemeinschaftlich mit **115** gekennzeichnet) mit vier konfigurierbaren Koppelfeldmodulen **120**, **122**, **124** und **126** gekoppelt. Diese Kanäle ermöglichen den E/A-Modulen die Kommunikation von Befehlen, Anweisungen bzw. Daten an die Koppelfeldmodule, wie unten näher beschrieben.

[0033] Im veranschaulichten Beispiel gibt es vier Koppelfeldmodule **120**, **122**, **124** und **126**, wobei jedoch anzumerken ist, dass die Anzahl der Koppelfeldmodule nicht von entscheidender Bedeutung ist und variieren kann. In Abb. 1 ist jedes der vier Koppelfeldmodule über einen eigenen seriellen System-Kommunikationskanal mit jedem der fünf E/A-Module gekoppelt. Es gibt Vorstellungen alternativer Ausführungsformen, in denen eines oder mehrere der Koppelfeldmodule nicht mit allen E/A-Modulen gekoppelt ist/sind.

[0034] Wie in Abb. 1 gezeigt, ist im Beispielsystem jedes Koppelfeldmodul über eine Vielzahl parallel geschalteter Speicherbusse an eine Gruppierung von Speicherkarten gekoppelt. Beispielsweise ist im veranschaulichten Beispiel das Koppelfeld **120** über eine erste Gruppe von parallel geschalteten Speicherbussen **130** an eine erste Gruppe von Speicherkarten **140** gekoppelt. In ähnlicher Weise sind die Koppelfeldmodule **122**, **124** und **126** über die parallel geschalteten Speicherbusgruppen **132**, **134** bzw. **136** an die Speicherkartengruppen **142**, **144** und **146** gekoppelt. Im veranschaulichten Beispiel ist jedes Koppelfeld über fünf parallel geschaltete Multiplexbusse, von denen jeder zur Weitergabe von Steuerungsdaten, Befehlen oder Überwachungsdaten wie auch zur Datenübertragung verwendet werden kann, an eine Gruppe von fünf Speicherkarten gekoppelt. Es versteht sich, dass die Anzahl der Speicherkarten und die Art des Speicherschnittstellenbusses unterschiedlich sein können, ohne dass dadurch von den Lehren dieser Offenlegung abgewichen wird.

[0035] Im veranschaulichten Beispiel entspricht im Allgemeinen jede Speicherkarte und somit jede Gruppe von Speicherkarten einem bestimmten Bereich von Speicherorten. Bei dem spezifischen Bereich kann es sich um einen tatsächlichen physikalischen Adressbereich (beispielsweise um einen Bereich von Adressen, in welchem jede Adresse einem mit der Speicherkarte verbundenen spezifischen physikalischen Speicherort entspricht) oder um einen logischen Adressbereich handeln (d. h., um einen Bereich logischer Adressen, in welchem jede logische Adresse von einem Controller auf der Speicherkarte – oder von einem externen Controller – einem mit der Speicherkarte verbundenen physikalischen Speicherort zugewiesen werden kann).

[0036] Im Beispiel von Abb. 1 präsentiert das Gesamtsystem **100** den externen Hosts eine Anzahl verfügbarer Speicheradressen, wobei jede Speicherkarte im Allgemeinen einem bestimmten Bereich der vom System **100** präsentierten Adressen entspricht. Da jede Gruppe von Speicherkarten mit einem bestimmten Bereich von Adressen zusammenhängt, hängt darüber hinaus jedes Koppelfeld mit jenem Adressbereich zusammen, der der Gruppe von Speicherkarten entspricht, mit denen es verbunden ist.

[0037] Im System **100** von Abb. 1 ist ein System-Controller **150** mit jedem der E/A-Module **102**, **104**, **106**, **108** und **110** und mit jedem der Koppelfeldmodule **120**, **122**, **124** und **126** gekoppelt. Der System-

Controller ist zwar mit allen E/A-Modulen in Abb. 1 verbunden, jedoch ist nur eine einzelne solche Verbindung dargestellt. In gleicher Weise ist nur eine der vier Verbindungen zwischen dem System-Controller und den Koppelfeldmodulen dargestellt. Im Allgemeinen liefert der System-Controller **150** Zeitgebungs- und Synchronisierungssignale, er könnte jedoch für Überwachungs-, Melde- und sonstige Aufsichtsaufgaben verwendet werden.

[0038] Im allgemeinem Betrieb empfängt jedes der E/A-Module über seine Host-Kommunikationsverbindungen externe datenbezogene Anfragen von einem externen Host-Gerät, mit welchem das E/A-Modul verbunden ist. Jedes E/A-Modul verarbeitet diese datenbezogenen Anforderungen, um die Speicherung von Daten in das Speichersystem **100** oder den Abruf von Daten aus dem Speichersystem **100** zu ermöglichen. In der Ausführungsform von Abb. 1 werden Datenanforderungen zwar in Form deutlich unterscheidbarer „Kommunikationspakete" empfangen und Antworten in der gleichen Form zurückgegeben, es können jedoch auch alternative Kommunikationsprotokolle verwendet werden.

[0039] Die genauen Protokolldetails sind zwar von der Anwendung abhängig, jedoch sollte ein externes Protokoll mit geringem zusätzlichen Kommunikationsaufwand gewählt werden, um einen hohen Datendurchsatz zu gewährleisten. Im System von Abb. 1 enthalten die von den E/A-Modulen empfangenen und verarbeiteten Kommunikationspakete im Allgemeinen einen Dateikopf und, falls das Paket Daten transportieren soll, eine Datennutzlast. Zwar sind der genaue Inhalt und das Format des Datenkopfes je nach Anwendung unterschiedlich, meist enthält der Datenkopf jedoch: (a) eine Kennung, die den Anforderungstyp anzeigt (beispielsweise, ob die Anforderung eine SCHREIB-Anforderung zur Speicherung von Daten im System oder eine LESE-Anforderung zur Abfrage von früher im System gespeicherten Daten ist); (b) eine Kennung, welche eine mit der Anforderung verbundene bestimmte ADRESSE anzeigt (beispielsweise eine spezifische logische oder physikalische „Adresse", die im Falle einer SCHREIB-Anforderung im Allgemeinen einen Ort kennzeichnet, an welchem die Speicherung der mit der Anforderung verbundenen Daten beginnen soll, oder im Falle einer LESE-Anforderung einen Ort, an dem der Abruf der mit der Anforderung verbundenen Daten beginnen soll, wobei jede dieser Adressen gelegentlich als „ZIELADRESSE" bezeichnet wird), und bei einigen Systemen (c) eine Identifizierung der Quelle der datenbezogenen Anforderung (gelegentlich als „QUELLADRESSE" bezeichnet), sowie (d) eine Kennung, welche die mit der Anforderung verbundene Datenmenge angibt (beispielsweise eine Angabe über die Anzahl von Bytes, Seiten oder andere Datenmengen, die im Rahmen der Anforderung im System gespeichert oder aus dem System abgerufen werden soll). Bei einigen Kommunikationsprotokollen wird davon ausgegangen, dass alle externen datenbezogenen Anfragen mit einer bestimmten Datenmenge verbunden sind, beispielsweise einer Datenseite von bestimmter Größe oder einer bestimmten Anzahl von Datenwörtern. In solchen Protokollen ist gegebenenfalls die ausdrückliche Angabe der mit einer bestimmten externen Anforderung verbundenen Datenmenge nicht erforderlich, da es implizit ist, dass die Anforderung die Standard-Datenmenge für das betreffende Protokoll betrifft.

[0040] Ebenso, wie externe Geräte datenbezogene Anforderungen an ihre angeschlossenen E/A-Module ausgeben, geben E/A-Module ihre als DMA-Anforderungen (Direct Memory Access, direkter Speicherzugang) oder einfach als DMA bezeichneten Anforderungen an ihre angeschlossenen Koppelfeldmodule und an ihre angeschlossenen Speicherkarten aus. Jede externe Anforderung erfordert möglicherweise Zugriff auf einen Bereich von Systemadressen, der mehrere Speicherkarten und Koppelfeldmodule umspannt. Selbst wenn eine bestimmte externe Anforderung Zugang zu lediglich einer Speicherkarte erfordert, übersteigt die mit der externen Anforderung verbundene Datenmenge unter Umständen die mit einer einzelnen DMA-Anforderung verbundene maximale Datenmenge. Aus diesem Grund kann ein E/A-Modul für jede eingehende externe Anforderung mehrere DMA-Anforderungen ausgeben. In diesem Szenario werden die DMA-Anforderungen solcherart an Speicherkarten weitergegeben, dass die Datenspeicherung oder der Datenabruf an den korrekten physikalischen Orten innerhalb des Systems erfolgt. So wird das E/A-Modul beispielsweise im Falle einer externen LESE-Anforderung eine oder mehrere DMA LESE-Anforderungen ausgeben, um die angeforderten Daten von den verschiedenen Orten „aufzusammeln", an denen sie gespeichert sind. Bei einer externen SCHREIB-Anforderung wird das E/A-Modul im Allgemeinen eine oder mehrere DMA SCHREIB-Anforderungen ausgeben, um alle empfangenen Daten an die verschiedenen Orte „zuzustellen", an welchen sie gespeichert werden sollen.

[0041] Im Allgemeinen werden DMA SCHREIB-Anforderungen in „Pakete" geordnet, wobei sich die Anforderung und die dazugehörigen Daten im selben Paket befinden. DMA LESE-Anforderungen hingegen enthalten im Allgemeinen nur die Daten der Anforderung (Adresse, Datenmenge usw.). Die Antworten auf jede dieser Anforderungsarten sind in etwa komplementär. Für jede vom E/A-Modul ausgegebene DMA SCHREIB-Anforderung gibt das Koppelfeld ein DMA SCHREIB-Antwortpaket zurück, welches zumindest Statusinformation über den Erfolg oder Fehlschlag der SCHREIB-Operation enthält. Für jede vom E/A-Modul ausgegebene DMA LESE-Anforderung gibt das Koppelfeld ein DMA LESE-Antwortpaket zurück, welches sowohl die angeforderten

Daten und zumindest Statusinformation über die Gültigkeit der zurückgegebenen Daten enthält.

[0042] In einer Ausführungsform, bei welcher jede der Speicherkarten im System einen Flash-Speicher benutzt, legt jede einzelne DMA-Anforderung die Übertragung einer festen Datenmenge fest, wobei die feste Menge der auf einer einzelnen „Seite" eines Flash-Speichers gespeicherten Datenmenge entspricht. In diesem Beispiel stellt eine Seite im Allgemeinen die kleinste Informationsmenge dar, die diskret in den Flash-Speicher geschrieben werden kann.

[0043] Im System **100** von Abb. 1 werden DMA-Anforderungen im Allgemeinen so definiert, dass sie eine feste Menge an Daten (4 Kilobyte) an oder von Adressen übertragen, die an festen Adressengrenzen (4 Kilobyte) ausgerichtet sind. In einigen Fällen ist es möglich, dass der von einer externen SCHREIB-Anforderung umfasste Adressbereich nicht an diesen festen Adressengrenzen von DMA-Anforderungen ausgerichtet ist. Wie in Abb. 1A–Abb. 1C dargestellt, kann eine externe SCHREIB-Anforderung an einer Adresse beginnen, die der zulässigen Startadresse einer DMA-Anforderung nicht entspricht. In gleicher Weise kann eine DMA SCHREIB-Anforderung an einer Adresse enden, die der Endadresse einer DMA-Anforderung nicht entspricht. Wenn eine solche „Fehlausrichtung" eintritt, führt die E/A-Karte gegebenenfalls eine oder mehrere RMW-Operationen (Read-Modify-Write-Operationen: Lesen, Ändern, Schreiben) aus. In einer RMW-Operation gibt das E/A-Modul zum Abruf von Daten aus dem Systemspeicher eine DMA LESE-Anforderung aus, deren Adressbereich den Beginn bzw. das Ende einer externen SCHREIB-Anforderung überspannt. Das E/A-Modul ändert (ersetzt) dann einen Teil dieser abgerufenen Daten mit externen Daten aus der nicht ausgerichteten SCHREIB-Anforderung und sendet dann diese geänderten Daten durch Ausgabe einer komplementären DMA SCHREIB-Anforderung an den Systemspeicher zurück. Auf diese Weise ermöglicht das E/A-Modul die Speicherung von Daten im Systemspeicher in Adressbereichen, die die Ausrichtungsbeschränkungen der einzelnen DMA-Anforderungen nicht befolgen.

[0044] Abb. 1A zeigt ein Beispiel eines Falles von „enthalten in", bei welchem eine externe SCHREIB-Anforderung eine Adresse und eine Datenmenge **160** angibt, die zur Gänze in einer Seite enthalten ist, jedoch eine RMW-Operation veranlassen wird, da die Startadresse nicht an einer DMA-Adressgrenze ausgerichtet ist. In diesem Fall ist lediglich eine einzige DMA LESE-Anforderung erforderlich. Die neuen (geänderten) Daten werden anstelle der vorher gespeicherten Daten mithilfe einer ausgegebenen DMA SCHREIB-Anforderung gespeichert, die dieselbe (ausgerichtete) Adresse wie die DMA LESE-Anforderung verwendet. Abb. 1B zeigt eine externe SCHREIB-Anforderung mit einer angegebenen Adresse und einer Datenmenge **170**, welche zwei Flash-Seiten überspannt. Dieser Fall der „Überspannung" generiert zwei DMA LESE-Anforderungen zum Abruf der Daten und anschließend zwei DMA SCHREIB-Anforderungen zur Rückgabe des neuen Datenbildes. Abb. 1C zeigt eine externe SCHREIB-Anforderung, deren angegebene Adresse und Datenmenge **180** über drei Seiten reicht. In diesem „seitenübergreifenden" Fall gibt die E/A-Karte für die erste Seite eine DMA LESE-Anforderung aus, blendet die neuen Daten ein und gibt eine DMA SCHREIB-Anforderung aus, um die Änderungen festzuschreiben. Da die nächste Seite von den neuen Daten vollständig überschrieben wird, muss die E/A-Karte auf dieser Seite keine RMW-Operation ausführen, sondern gibt einfach eine DMA SCHREIB-Anforderung an die ausgerichtete Adresse aus. Der Rest des neuen Datensatzes füllt jedoch die dritte Seite nicht vollständig aus. Daher wird die E/A-Karte auf dieser Seite eine RMW-Operation in der oben beschriebenen Art und Weise ausführen. Es versteht sich, dass diese drei Fälle nur eine Teilmenge der verschiedenen Arten von externen SCHREIB-Anforderungsoperationen darstellen, die ausgeführt werden können; auch dienen sie lediglich der Veranschaulichung der RMW-Operation.

[0045] Zwar ist die Struktur externer Anforderungen im Allgemeinen vom Protokoll abhängig, jedoch ist das interne DMA-Anforderungs-/-Antwortprotokoll für ein bestimmtes System meist unveränderlich und auf die Maximierung der Systemleistung hin ausgelegt. Im System von Abb. 1 sind die DMA-Anforderungen und DMA-Antworten als Pakete strukturiert, die einen Dateikopf und, falls das Paket Daten transportieren soll, eine Datennutzlast enthalten.

[0046] DMA-Anforderungen umfassen im Allgemeinen: (a) eine Kennung, die den DMA-Anforderungstyp anzeigt (beispielsweise, ob die DMA-Anforderung eine SCHREIB-Anforderung zur Speicherung von Daten im System oder eine LESE-Anforderung zur Abfrage von früher im System gespeicherten Daten ist); (b) eine Kennung, welche eine mit der DMA-Anforderung verbundene bestimmte ADRESSE anzeigt (beispielsweise eine spezifische logische oder physikalische „Adresse", die im Falle einer DMA SCHREIB-Anforderung im Allgemeinen einen Ort kennzeichnet, an welchem die Speicherung der mit der Anforderung verbundenen Daten beginnen soll, oder im Falle einer DMA LESE-Anforderung einen Ort, an dem der Abruf der mit der Anforderung verbundenen Daten beginnen soll, (c) eine Kennung des E/A-Moduls, aus welchem die DMA-Anforderung stammt, sowie (d) eine Kennzeichnung („Tag"), die bei jedem E/A-Modul spezifische von diesem Modul stammende DMA-Anforderungen eindeutig iden-

tifiziert. Bei einer DMA SCHREIB-Anforderung umfasst die Anforderung auch die Daten, die für diese Anforderung im Systemspeicher gespeichert werden sollen. Im System **100** von Abb. 1 ist die Menge der mit jeder DMA-Anforderung verbundenen Daten fest eingestellt. Bei Systemen, in welchen die Datenmenge variabel ist, kann der Datenkopf auch (e) eine Kennung enthalten, welche die mit der Anforderung verbundene Datenmenge anzeigt.

[0047] DMA-Antworten umfassen im Allgemeinen (a) eine Kennung, die den Typ der DMA-Antwort anzeigt (d. h., ob die Antwort einer DMA SCHREIB-Anforderung oder einer DMA LESE-Anforderung entspricht), (b) eine Kennzeichnung, die bei jedem E/A-Modul vom betreffenden E/A-Modul zur eindeutigen Zuweisung jeder einzelnen DMA-Antwort an die entsprechende DMA-Anforderung verwendet werden kann, und (c) eine Statusanzeige, die bei einer DMA SCHREIB-Antwort den Erfolg oder Fehlschlag der entsprechenden Schreiboperation oder bei einer DMA LESE-Anforderung die Gültigkeit der abgerufenen Daten anzeigt. Bei einer DMA LESE-Antwort umfasst die Antwort auch die von der ursprünglichen DMA LESE-Anforderung angeforderten Daten.

[0048] Im System **100** von Abb. 1 antwortet ein E/A-Modul auf eine externe datenbezogene Anforderung unter Anderem dadurch, dass die Anforderung minimal verarbeitet wird, um festzustellen, ob es sich bei ihr um eine LESE-Anforderung oder eine SCHREIB-Anforderung handelt, und um die Koppelfelder (und möglicherweise die Speichersteckkarten) zu ermitteln, an welche die DMA-Anforderungen zur Befriedigung der externen Anforderungen gerichtet werden müssen.

[0049] Nach der minimalen Verarbeitung der externen datenbezogenen Anforderung durch das E/A-Modul, wie oben festgelegt, treten die Komponenten des Systems in Betrieb, um die Anforderung zu befriedigen. Beispielsweise verarbeitet ein E/A-Modul zunächst jede externe datenbezogene Anforderung, um ihren Typ (LESEN oder SCHREIBEN) wie auch die spezifischen DMA-Anforderungen zu ermitteln, die zur Befriedigung der externen Anforderung ausgegeben werden müssen. Das E/A-Modul liefert dann die entsprechenden DMA-Anforderungen zur weiteren Bearbeitung an die betreffenden Koppelfeldmodule. Die Koppelfeldmodule ihrerseits (a) ermitteln die Speicherkarte, an die die einzelnen DMA-Anforderungen zu leiten sind, (b) wandeln jede eingegangene DMA-Anforderung aus dem seriellen Format in ein zur Darstellung auf der Speicherkarte geeignetes paralleles Format um und (c) leiten jede DMA-Anforderung zur Gänze oder zum Teil an die Speicherkarte weiter, deren Adressbereich die mit der DMA-Anforderung verbundenen Adresse überspannt. Wie unten genauer beschrieben, wird die DMA-Anforderung, entweder (bei einer DMA SCHREIB-Anforderung) die gelieferten Daten im physikalischen Speicher zu speichern oder (bei einer DMA LESE-Anforderung) Daten aus dem physikalischen Speicher abzurufen, durch eine Schaltung in der Speicherkarte weiter verarbeitet. Hier soll besonders darauf hingewiesen werden, dass die mit jeder einzelnen DMA SCHREIB-Anforderung verbundene Datennutzlast im Allgemeinen als Bestandteil der Anforderung zugestellt wird. In einigen Ausführungsformen gibt es gegebenenfalls ein Bestätigungssignal (ACK) zwischen den Koppelfeldmodulen und den Speicherkarten zur Bestätigung, dass die Dateien erfolgreich auf die Speicherkarten übertragen wurden.

[0050] Durch die Verwendung der Kombination aus seriellen Hochgeschwindigkeits-System-Kommunikationsverbindungen für die E/A-Module und aus den Koppelfeldmodulen, konfigurierbaren Koppelfeldmodulen und parallel geschalteten Hochgeschwindigkeits-Kommunikationsverbindungen für die Koppel Feldmodule und die Speicherkarten stellt das Speichersystem **100** ein System von extrem hoher Bandbreite dar.

[0051] Im oben besprochenen Beispiel werden die seriellen Hochgeschwindigkeits-Kommunikationsverbindungen, welche die E/A-Schnittstellenmodule an die Koppelfeldmodule koppeln, zur Übertragung sowohl von Befehlsdaten als auch von Steuerdaten (z. B. Information der Art, wie sie im Datenkopf einer DMA-Anforderung vorhanden ist) und Daten (z. B. in einer Datennutzlast gelieferte Information) benutzt. Es sind alternative Ausführungsformen vorgesehen, bei welchen einige zusätzliche minimale seriell geschaltete Kommunikationsverbindungen eingesetzt werden, um Kommunikation zu leisten, bei welcher keine Übertragung von Daten des in einer Datennutzlast eingehenden Typs involviert ist. So sind beispielsweise Ausführungsformen vorgesehen, bei welchen eine serielle Kommunikationsverbindung mit geringer Geschwindigkeit und relativ geringer Bandbreite für bestimmte Kategorien von Kommunikation benutzt werden können; damit könnten die Hochgeschwindigkeitsverbindungen um Datenverkehr einer bestimmten Menge entlastet werden, wodurch eine entsprechende Steigerung der Leistung dieser Verbindungen erreichbar wäre. Beispielsweise könnte jede DMA LESE-Anforderung und DMA SCHREIB-Antwort auf diese Weise übertragen werden, womit die seriellen Hochgeschwindigkeitsverbindungen für DMA SCHREIB-Anforderungen und DMA LESE-Antworten reserviert würden, da beide die Daten in ihren jeweiligen Paketen transportieren. Da die DMA LESE-Anforderung und die DMA SCHREIB-Antwort keine Datennutzlast übertragen, sind sie von der Größe her im Allgemeinen viel kleiner und könnten ohne signifikanten Leistungsverlust eine langsamere Kommunikationsverbindung benutzen. In einer alternativen Ausführungsform könnten DMA LESE-Anforderungen und DMA SCHREIB-Antworten über einen

alternativen seriellen Kanal mit geringer Bandbreite übertragen oder über eine separate gemeinsame Hochgeschwindigkeitsverbindung von allen/an alle Koppelfelder(n) weitergegeben werden.

[0052] Eine serielle Hochgeschwindigkeitsverbindung könnte auch zur Verbindung aller Koppelfelder mit allen E/A-Modulen benutzt werden, wobei ein Multi-Drop-Bus diese Seitenbanddaten mit einer den Daten ähnlichen Signalrate liefern würde. Alternativ könnte die Seitenbandverbindung als Ringstruktur umgesetzt werden. In beiden Fällen würde das die Anzahl der Punkt-zu-Punkt-Verbindungen im System zulasten zusätzlicher Arbitrierungslogik und Hardware verringern, hätte jedoch den zusätzlichen Vorteil eines höheren Datendurchsatzes und einer größeren Bandbreite.

[0053] Weitere Einzelheiten bezüglich der oben ausgeführten spezifischen Elemente sowie alternativer Ausführungsformen der Elemente bzw. des Gesamtsystems sind unten nachzulesen.

DIE E/A-MODULE

[0054] Die E/A-Module von Abb. 1 können viele Formen annehmen. Ein Ausführungsbeispiel eines geeigneten E/A-Moduls ist in Abb. 2 dargestellt.

[0055] In Abb. 2 ist das Beispiel eines E/A-Moduls **200** veranschaulicht. In diesem Beispiel ist am E/A-Modul **200** eine Schnittstelle mit zwei separaten Fibre-Channel-Ports (**201** und **203**) vorgesehen, wobei jeder von ihnen dem spezifischen E/A-Modul **200** den Anschluss an einen (nicht dargestellten) Fibre-Channel-Host-Bus-Adapter ermöglicht und dadurch einem externen Host den Zugang zum System **100** als SCSI-Gerät erlaubt. Jeder Port (**201** oder **203**) umfasst eine einzelne Sendeverbindung und eine einzelne Empfangsverbindung, wobei jede Verbindung entweder bei 2,125 Gigabit pro Sekunde (Gb/s) oder oder bei 4,25 Gb/s betrieben wird. Jeder Port kann Point-to-Point- und Arbitrated-Loop-Fibre-Channel-Protokolle unterstützen. Es versteht sich, dass diese physikalische Schnittstelle und die SCSI- und Fibre Channel(FC)-Protokolle beispielhaft sind und dass andere physikalische Schnittstellen und Protokolle verwendet werden können.

[0056] In Abb. 2 umfasst das beispielhafte E/A-Modul **200** optisch-elektrische Wandler **202a** und **202b** zur Umwandlung von Information zwischen optischen und elektrischen Formaten. Im Allgemeinen wandelt jedes dieser Geräte (**202a** und **202b**) die in seinen externen optischen Empfangsverbindungen (**201** oder **203**) empfangenen Signale in elektrische Signale um, die dann an einen Hochgeschwindigkeits-Controller **204** weitergeleitet werden. In gleicher Weise wandelt jedes dieser Geräte die aus dem Hochgeschwindigkeits-Controller **204** empfangenen elektrischen Signale in optische Signale um, die über seine externen optischen Übertragungsleitungen übertragen werden. Dazu können bekannte optisch-elektrische Wandler eingesetzt werden.

[0057] Die oben angeführten optisch-elektrischen Wandler sind mit einem Hochgeschwindigkeits-Controller **204** gekoppelt. Im Beispiel von Abb. 2 handelt es sich beim Hochgeschwindigkeits-Controller **204** um ein konfiguriertes Field Programmable Gate Array (FPGA), beispielsweise einen Xilinx Virtex-4 FPGA-Baustein. Der Controller **204** ist über eine Kommunikationsverbindung mit einem programmierbaren Mikroprozessor (CPU) **206**, beispielsweise einem Freescale MPC8547 Prozessor **206**, und mit dem Speicher **208** gekoppelt. Im dargestellten Beispiel wird der Speicher **208** aus DDR-Speicherkomponenten gebildet.

[0058] Wie ersichtlich ist, stellt der Controller **204** vier serielle Hochgeschwindigkeits-System-Kommunikationskanäle **210**, **212**, **214** und **216** zur Verfügung. Der Prozessor **206** und der Controller **204** bilden zusammen die Protokollumwandlungsfunktion zwischen den externen Fibre Channel Ports (**201** und **203**) und den vier seriellen System-Kommunikationskanälen **210**, **212**, **214** und **216**. Im dargestellten Beispiel umfasst jeder der seriellen System-Kommunikationskanäle eine physikalische Vollduplexschicht, welche sowohl einen Sende-(TX) wie auch einen Empfangs-(RX)Teilkanal umfasst. Jeder Teilkanal umfasst darüber hinaus zwei verschiedene serielle Kommunikationsverbindungen von 5 Gb/s, die miteinander verklebt sind, um einen einzelnen Teilkanal mit einer Datenübertragungsgeschwindigkeit von 10 Gb/s zu bilden. In diesem Beispiel werden Daten 8 B/10 B kodiert, bevor sie über die seriellen System-Kommunikationskanäle übertragen werden, was in einer Datenübertragungsrate von 8 Gb/s (oder 1 Gigabyte pro Sekunde (GB/s)) resultiert. Ebenso könnten alternative Kodierungssysteme (z. B. 64 B/66 B-Kodierung) benutzt werden. Im Beispiel von Abb. 2 wird jeder serielle Kommunikationskanal unter Einsatz eines Multi-Gigabit-Transceiver-Moduls (MGT) umgesetzt, welches in einem Xilinx Virtex-4 FPGA-Baustein enthalten ist, obwohl zur Kenntnis zu nehmen ist, dass andere ähnliche Umsetzungsverfahren eingesetzt werden könnten, ohne von den Lehren dieser Offenlegung abzuweichen. Vorhandene Verfahren zur Verklebung von Kommunikationsverbindungen, beispielsweise die von der Virtex-4 Familie von FPGAs ermöglichten, können zur Bildung der Kommunikationskanäle angewendet werden. Alternativ lassen sich mit anderen Serialisierungs-/Deserialisierungsprotokollen (SerDes) verbundene andere Verklebungsmodelle nutzen.

[0059] Im Beispiel von Abb. 2 verfügt das E/A-Schnittstellenmodul **200** über vier serielle System-Kommunikationskanäle **210**, **212**, **214** und **216**, von

denen jeder mit einem anderen Koppelfeldmodul verbunden ist. Der Einsatz von vier seriellen Vollduplex-Systemkommunikationskanälen stellt insgesamt für jedes E/A-Modul eine potenzielle Lese-Bandbreite von 4 GB/s und eine Schreib-Bandbreite von 4 GB/s zur Verfügung. Angesichts der Tatsache, dass das System **100** von **Abb. 1** fünf E/A-Module benutzt, würde das dargestellte Beispielsystem über eine Lese-Gesamtbandbreite von 20 GB/s und eine Schreib-Gesamtbandbreite von 20 GB/s verfügen.

[0060] Im allgemeinen empfängt und verarbeitet der Hochgeschwindigkeits-Controller **204** externe datenbezogene Anforderungen über die FC-Schnittstellen **201** und **203**. Wie oben beschrieben, enthält eine externe datenbezogene Anforderung in den meisten Fällen einen Dateikopf mit folgenden Angaben: (a) ob es sich bei der Anforderung um eine LESE- oder eine SCHREIB-Anforderung handelt, und (b) die ZIELADRESSE. Wahlweise kann der Dateikopf (c) die QUELLADRESSE sowie (d) Angaben über die mit der Anforderung verbundene Datenmenge enthalten. Bei SCHREIB-Anforderungen umfasst die Anforderung auch eine Datennutzlast. Der Hochgeschwindigkeits-Controller **204** verarbeitet die Anforderung und kann dem Prozessor oder der CPU **206** einen Großteil der Dateikopfinformation (jedoch keine Datennutzlast) liefern. Die CPU **206** wird zumindest einen Teil des FC-Protokolls handhaben und bei der Handhabung der Gesamtsystemschnittstelle unterstützend mitwirken.

[0061] Wie bereits oben besprochen, arbeiten der Hochgeschwindigkeits-Controller **204** und die CPU **206** bei der Verwaltung der externen Fibre-Channel-Verbindungen zusammen. Abgesehen von einiger minimaler Unterstützung der CPU ist der Hochgeschwindigkeits-Controller **204** in den meisten Ausführungsformen lediglich für die Ausgabe der internen DMA-Anforderungen verantwortlich, die zur Erfüllung der einzelnen externen datenbezogenen Anforderungen erforderlich sind. Insbesondere bestimmt der Hochgeschwindigkeits-Controller **204** die speziellen DMA-Anforderungen, welche auszustellen sind, und auf welche Koppelfelder die DMA-Anforderungen zur Erfüllung der einzelnen externen Anforderungen auszustellen sind. Sobald der Hochgeschwindigkeits-Controller die zur Erfüllung einer externen Anforderung erforderlichen entsprechenden Koppelfeldmodule und DMA-Anforderungen bestimmt hat, beginnt er mit der Ausstellung von DMA-Anforderungen an die betreffenden Koppelfelder. Im Allgemeinen ist die Anzahl der ausstehenden DMA-Anforderungen (Anforderungen, für die noch keine Antwort eingegangen ist), die von einem E/A-Modul **200** ausgestellt werden können, durch die Größe des Tag-Feldes im DMA-Dateikopf begrenzt. Bei relativ kleinen Übertragungen ist die Anzahl der erforderlichen DMA-Anforderungen gering; der Hochgeschwindigkeits-Controller **204** kann sie alle ausstellen, ohne auf den Eingang von Antworten zu warten. Bei großen Übertragungen ist unter Umständen die Anzahl der erforderlichen DMA-Anforderungen groß; der Hochgeschwindigkeits-Controller **204** kann gezwungen sein, die Ausstellung von DMA-Anforderungen zu verzögern. Angesichts eines Tag-Werts von beispielsweise 8 Bit könnte der Hochgeschwindigkeits-Controller zunächst 256 DMA-Anforderungen ohne Wiederverwendung von Tag-Werten ausstellen. Jede ausstehende DMA-Anforderung würde eindeutig markiert werden; so wäre der Hochgeschwindigkeits-Controller in der Lage, jede DMA-Antwort ihrer entsprechenden DMA-Anforderung zuzuweisen. Mit der Rückgabe von Antworten auf DMA-Anforderungen könnte der Hochgeschwindigkeits-Controller die Tags erfüllter DMA-Anforderungen (jene Anforderungen, für welche eine DMA-Antwort eingegangen ist) erneut verwenden und damit die Ausstellung nachfolgender DMA-Anforderungen ermöglichen.

[0062] Im Allgemeinen werden DMA-Anforderungen und ihre entsprechenden DMA-Antworten über die seriellen System-Kommunikationskanäle **210**, **212**, **214** und **216** übertragen. Da jedes E/A-Modul vier serielle Kommunikationskanäle vorsieht, kann es eigenständig Daten an vier verschiedene Koppelfeldmodule gleichzeitig oder beinahe gleichzeitig übertragen bzw. von diesen Modulen empfangen.

[0063] Wie oben beschrieben, ist der Hochgeschwindigkeits-Controller **204** für die Lieferung der LESE- und SCHREIB DMA-Anforderungen an die korrekten Koppelfeldmodule zur Erfüllung der einzelnen externen datenbezogenen Anforderungen verantwortlich. In einer Ausführungsform bestimmt der Hochgeschwindigkeits-Controller **204** anhand einer im Speicher **208** des Hochgeschwindigkeits-Controllers gepflegten Wertetabelle, welches Koppelfeldmodul die jeweilige DMA-Anforderung erhalten soll. In dieser Ausführungsform greift der Controller **204** auf eine Wertetabelle zu und pflegt diese; sie enthält Information, die einen bestimmten Adressbereich innerhalb des Gesamtbereichs der vom System angebotenen Adressen mit einem bestimmten Koppelfeldmodul korreliert. Ein Vorteil dieses Ansatzes besteht darin, dass er bei der Konfiguration und Umkonfiguration des Systems ein gewisses Maß an Flexibilität gewährt. In einer noch weiteren Ausführungsform verbindet die Wertetabelle im Speicher **208** darüber hinaus einen bestimmten Bereich präsentierter Adressen mit einer bestimmten Speicherkarte und die verschiedenen Speicherkarten mit den Koppelfeldmodulen, so dass eine empfangene Adresse mit einem gegebenen Koppelfeldmodul und einer mit dem gegebenen Koppelfeldmodul gekoppelten bestimmten Speicherkarte korreliert.

[0064] In einer alternativen Ausführungsform ist die Beziehung zwischen einem gegebenen Koppelfeld und einer empfangenen Adresse fest verdrahtet, so

dass ein gegebener Adressbereich immer einem bestimmten Koppelfeldmodul entspricht.

[0065] In einer noch anderen Ausführungsform ist die Beziehung zwischen einem gegebenen Koppelfeld und einer empfangenen Adresse dynamisch; dadurch wird ermöglicht, dass eine fehlerhafte oder anderweitig nicht nutzbare Speicherkarte oder ein entsprechendes Koppelfeld einer anderen Speicherkarte oder einem anderen Koppelfeld neu zugeordnet wird.

[0066] Von einem E/A-Modul, beispielsweise dem E/A-Schnittstellenmodul **200** von Abb. 2, ausgestellte DMA-Übertragungen können gegebenenfalls die Übertragung einer festen oder variablen Datenmenge vorgeben. Bei Ausführungsformen, in denen die Größe der DMA-Übertragung festgeschrieben ist, ist es gegebenenfalls nicht erforderlich, dass DMA-Anforderungen Angaben bezüglich der Menge der zu übertragenden Daten liefern. In der Ausführungsform, in welcher die Speicherkarten einen Flash-Speicher benutzen, ist die Größe der DMA-Übertragung auf die auf einer einzigen Flash-Speicherseite gespeicherte Datenmenge festgelegt.

[0067] Bei einer DMA-Operation können unterschiedliche Datenmengen übertragen werden. In solchen Ausführungsformen umfasst jede DMA-Anforderung im Allgemeinen eine Angabe über die zu übertragende Datenmenge (häufig ausgedrückt als Vielfaches eines minimalen Datenquantums).

[0068] Im Laufe von DMA SCHREIB-Operationen, unabhängig davon, ob DMA-Übertragungen unter Anwendung einer festen oder variablen Übertragungsgröße durchgeführt werden, reiht der Hochgeschwindigkeits-Controller **204** die bei ihm von einem angehängten externen Host-Gerät eingehenden Daten in eine Warteschlange ein. Bei größeren, aus mehreren DMA SCHREIB-Anforderungen bestehenden Datenübertragungen muss die E/A-Schnittstelle nicht auf den Empfang aller externen Daten warten, bevor sie mit der Ausstellung von DMA SCHREIB-Anforderungen beginnen kann. Stattdessen kann der Hochgeschwindigkeits-Controller mit der Ausstellung von DMA SCHREIB-Anforderungen beginnen und die Ausstellung von DMA SCHREIB-Anforderungen fortsetzen, sobald ausreichend Daten zur Verfügung stehen, um die einzelnen DMA SCHREIB-Anforderungen zu unterstützen.

[0069] Im Laufe von DMA LESE-Operationen reiht der Hochgeschwindigkeits-Controller die (aus dem Systemspeicher) in die entgegengesetzte Richtung fließenden Daten in eine Warteschlange. In diesem Fall stellt der Hochgeschwindigkeits-Controller eine Reihe von DMA LESE-Anforderungen aus und wartet dann auf die Rückgabe von DMA LESE-Antworten aus den einzelnen Koppelfeldmodulen. Da diese Antworten unter Umständen in einer zu den entsprechenden DMA LESE-Anforderungen unterschiedlichen Reihenfolge eingehen, muss der Hochgeschwindigkeits-Controller die vom Koppelfeld eingehenden Daten in einer Warteschlange ordnen, bis die Daten in der richtigen Reihenfolge zugestellt werden können. Bei einem Szenario, in welchem eine externe Datenanforderung die Ausstellung von 256 DMA LESE-Anforderungen erfordert, ist es möglich, dass die letzte DMA LESE-Antwort der ersten DMA LESE-Anforderung entspricht; dadurch würde eine Einordnung der Daten aus 256 DMA-Übertragungen in die richtige Reihenfolge durch die E/A-Schnittstelle erforderlich werden. Die E/A-Schnittstelle kann Daten nur so lange beim externen Anfrager (Host-Gerät) abladen, als die Daten sequenziell zugestellt werden können.

[0070] In der oben beschriebenen Art und Weise kann das E/A-Schnittstellenmodul **200** Datenanforderungen von einem externen Host empfangen, zur Erfüllung der Anfrage auf Speicher innerhalb des Systems **100** zugreifen und die angeforderten Daten extern an Host-Geräte liefern.

[0071] Es sollte zur Kenntnis genommen werden, dass es sich bei dem E/A-Schnittstellenmodul von Abb. 2 lediglich um ein Beispiel eines E/A-Schnittstellenmoduls handelt, das im System der vorliegenden Offeblegung verwendet werden kann. Es können alternative E/A-Schnittstellen verwendet werden. Ein Beispiel einer alternativen Schnittstelle ist in Abb. 3 dargestellt.

[0072] Abb. 3 veranschaulicht ein alternatives E/A-Schnittstellenmodul **300**, welches in der Lage ist, datenbezogene Anfragen über eine Vielzahl von Hochgeschwindigkeits-InfiniBand-Ports (gemeinschaftlich als **301** bezeichnet) zu empfangen.

[0073] In Abb. 3 umfasst das beispielhafte alternative E/A-Modul **300** einen InfiniBand-Schnittstellenchip **302**, der in der Lage ist, datenbezogene Anfragen und Antworten über eine Vielzahl von InfiniBand-Ports **301** zu empfangen und zu übertragen. Als InfiniBand-Schnittstellenchip kann ein Standard-InfiniBand-Chip, beispielsweise einer der bei Mellanox erhältlichen, dienen.

[0074] Im Allgemeinen empfängt der InfiniBand-Schnittstellen-Chip **302** datenbezogene Anforderungen über die InfiniBand-Ports **301** und dekodiert die eingegangenen Anforderungen minimal zumindest in jenem Umfang, in dem er die Befehlskopfzeile und bei SCHREIB-Anforderungen die Datennutzlast erkennt. Der Schnittstellen-Chip **302** macht dann die Angaben des Befehlskopfs einer Steuer-CPU **306** durch die Verwendung eines Schaltelements, eines Routing-Elements oder eines Shared-Memory-Elements **305** verfügbar, welches – im Beispiel von

Abb. 3 – ein PCI Express-(PCIe-)Schalter ist. Unter Verwendung des Schalt-/Routing-/Speicherelements **305** stellt der Schnittstellen-Chip **302** die empfangenen Daten auch einem Hochgeschwindigkeits-Controller **304** (dies kann ein konfiguriertes FPGA sein) zur Verfügung. Der Hochgeschwindigkeits-Controller **304** ist mit mehreren seriellen System-Kommunikationskanälen **310**, **312**, **314** und **316** zu den Koppelfeldmodulen ausgestattet und operiert mit den empfangenen datenbezogenen Anforderungen zur Lieferung und zum Empfang von Information zu und von den Koppelfeldmodulen in einer Art und Weise, die der oben in Verbindung mit dem Hochgeschwindigkeits-Controller **200** von Abb. 2 beschriebenen ähnlich ist.

[0075] Es ist anzumerken, dass die oben besprochenen E/A-Schnittstellenmodule **200** und **300** lediglich beispielhaft sind, und dass andere Arten von E/A-Schnittstellenmodulen, einschließlich derjenigen mit unterschiedlichen physikalischen Schichten, unterschiedlichen Kommunikationsprotokollen und diskreten Bauelementen (im Gegensatz zu konfigurierten FPGAs) verwendet werden könnten, ohne von der vorliegenden Offenlegung abzuweichen.

[0076] Im Allgemeinen besteht die Hauptanforderung an ein E/A-Modul darin, dass es in der Lage ist, extern gelieferte datenbezogene Anforderungen zu empfangen und auf sie zu antworten, und dass es in der Lage ist, diese Anforderungen über eine Vielzahl von Hochgeschwindigkeits-System-Kommunikationskanälen an mehrere Koppelfeldmodule zu leiten (und Informationen sowie Daten zu empfangen, um die Anforderungen zu beantworten).

DIE KOPPELFELDMODULE

[0077] Wie oben beschrieben, besteht die allgemeine Gesamtfunktion der einzelnen Koppelfeldmodule im System der vorliegenden Offenlegung darin: (a) DMA-Anforderungen von den E/A-Modulen zu empfangen und diese DMA-Anforderungen (samt etwaigen dazugehörigen Datennutzlasten) an die betreffenden mit dem Koppelfeldmodul verbundenen Speicherkarten zu leiten, und (b) aus den mit dem Koppelfeld verbundenen Speicherkarten abgerufene Daten zu empfangen und die abgerufenen Daten (sowie möglicherweise einige dazugehörige Informationen) an die mit den abgerufenen Daten verbundenen E/A-Module zu liefern.

[0078] Zur Umsetzung der Koppelfeldmodule des offen gelegten Systems können verschiedene Ansätze verfolgt werden. Abb. 4 veranschaulicht einen beispielhaften Ansatz für die Umsetzung eines Koppelfeldmoduls des oben in Verbindung mit Abb. 1 beschriebenen Typs.

[0079] In Abb. 4 ist das Beispiel eines Koppelfeldmoduls **400** veranschaulicht. Zwar lässt sich das Koppelfeldmodul durch die Verwendung diskreter Schaltkreise, anwendungsspezifischer integrierter Schaltkreise (Application Specific Integrated Circuits, ASICs) oder eines Mix aus beiden verwirklichen, in der Ausführungsform von Abb. 4 jedoch ist das Koppelfeldmodul **400** durch die Verwendung eines konfigurierten FPGA, beispielsweise eines Xilinx Virtex 4 FPGA, verwirklicht.

[0080] In diesem Beispiel ist das konfigurierte FPGA so konfiguriert, dass es fünf serielle System-Kommunikations-Ports **402**, **404**, **406**, **408** und **410** vorsieht. Jeder serielle Kommunikations-Port ist konfiguriert und verbunden, um bidirektionale Hochgeschwindigkeits-Kommunikationskanäle zwischen sich selbst und einem der Hochgeschwindigkeits-Kommunikations-Ports eines E/A-Moduls bereitzustellen. Da in diesem Beispiel das Koppelfeldmodul **400** fünf Vollduplex-Kommunikations-Ports bereitstellt, kann es daher mit jedem einzelnen der E/A-Module im System **100** gleichzeitig und eigenständig kommunizieren. Daher kann der Kommunikation-Port **402** an einen der seriellen Kommunikations-Ports des E/A-Schnittstellenmoduls **102** gekoppelt werden, Port **404** an einen Port des E/A-Schnittstellenmoduls **104** usw., so dass zwischen jedem der E/A-Module und dem Koppelfeldmodul **400** eine serielle Verbindung besteht.

[0081] Neben der Bereitstellung der oben beschriebenen fünf seriellen Kommunikations-Ports stellt das Koppelfeldmodul **400** von Abb. 4 auch mehrere parallele Kommunikationsbusse zur Verfügung, welche dazu verwendet werden, die Kommunikation zwischen dem Koppelfeldmodul **401** und einer Vielzahl von Speicherkarten zu ermöglichen. Im Beispiel von Abb. 4 gibt es fünf Speicherkarten **412**, **414**, **416**, **418** und **420**, wobei das Koppelfeldmodul **400** unter Zuhilfenahme paralleler Busse mit den Speicherkarten kommuniziert. Im Beispiel von Abb. 4 wird die Kommunikation zwischen dem Koppelfeldmodul **400** und jeder der einzelnen Speicherkarten dadurch erreicht, dass ein paralleler Bus verwendet wird, welcher im Beispiel für jede einzelne Speicherkarte einen 16-Bit-(sechzehn Bit)Datenbus und einen 5-Bit-(fünf Bit)Steuerbus umfasst.

[0082] Im allgemeinen Betrieb dient das Koppelfeldmodul für 100 hauptsächlich als Mittel zum Empfang von DMA-Anforderungen über eine der seriellen Systemverbindungen, zur Identifizierung spezifischer Speicherkarten, an welche die DMA-Anforderungen zu leiten sind, zur Umwandlung der DMA-Anforderungen aus einem seriellen Format in ein paralleles Format und der Bereitstellung der parallel formatierten DMA-Anforderungen an die jeweiligen Speicherkarten. Daneben empfängt das Koppelfeldmodul **400** im Allgemeinen die von einer Speicherkarte bereitge-

stellten Daten, stellt fest, für welches E/A-Modul die Daten bestimmt sind, und überträgt die empfangenen Daten über eine der seriellen System-Kommunikationsverbindungen als DMA-Antwort an das entsprechende E/A-Modul.

[0083] In den Beispielfällen kommuniziert das Koppelfeldmodul **400** mit jedem einzelnen E/A-Modul mithilfe eines seriellen Schnittstellenbusses, sowie mit jeder einzelnen Speicherkarte mithilfe eines parallelen Schnittstellenbusses. Hier ist darauf hinzuweisen, dass die Umsetzung des E/A-Schnittstellenbusses vom E/A-Modul abhängt, während der Speicherschnittstellenbus von der Speicherkarte abhängt. Theoretisch könnte jeder der beiden Busse entweder als serielle oder als parallele Schnittstelle ausgeführt werden, sofern der Bus die Leistung den erforderlichen Datenraten entsprechend erbringen kann. Die Schnittstellenbusse sind anwendungsabhängig, wobei die Aufbauentscheidungen, welche sich auf die Wahl eines bestimmten Busses auswirken, den Fachleuten auf diesem Gebiet bekannt sein sollten. So könnte beispielsweise zur Verminderung der Anschlussdichte die Schnittstelle der Speicherkarte als eine in ihrer Charakteristik dem E/A-Modul ähnliche serielle Hochgeschwindigkeitsverbindung ausgeführt werden, was die Schnittstelle Koppelfeld – Speicher vereinfachen würde.

[0084] Da es sich bei jedem der Busse, der einen Koppelfeld-Port mit einem E/A-Modul verbindet, um einen Punkt-zu-Punkt-Bus handelt, ist die Arbitrierung minimal und wird durch einen Mechanismus reguliert, der so entworfen ist, dass er dem Koppelfeld erlaubt, den E/A-Bus in Wartestellung zu halten, falls dies zur Verhinderung eines Überlaufs erforderlich ist.

[0085] Die Konfiguration eines FPGA zur Durchführung dieser Aufgaben sollte einem durchschnittlichen Fachmann auf diesem Gebiet, der den Vorteil dieser Offenlegung hat, ersichtlich sein.

DIE SPEICHERKARTEN

[0086] Die in dem offen gelegten System einsetzbaren Speicherkarten können viele Formen annehmen. Sie können eine Vielzahl von Speichern (einschließlich RAM-Speicher verschiedenster Art (DDR RAM, DDR2 RAM, DDR3 RAM und ähnliche) sowie Flash-Speicher verschiedenster Art (einschließlich MLC Flash-Speicher, SLC Flash-Speicher und dergleichen) nutzen. Die Struktur der Speicherkarten kann auch unterschiedlich ausfallen; dies hängt zum Teil von der Art des eingesetzten Speichers ab.

[0087] Abb. 5 veranschaulicht eine exemplarische Ausführungsform einer Speicherkarte **500**, die einen Flash-Speicher benutzt, und die im System der vorliegenden Offenlegung eingesetzt werden kann. Im Beispiel von Abb. 5 ist der Flash-Speicher ein SLC Flash-Speicher, auch wenn Ausführungsformen, in welchen ein MLC Flash-Speicher eingesetzt wird, ebenso in Betracht kommen.

[0088] Unter Bezugnahme auf Abb. 5 umfasst die Flash-Speicherkarte **500** einen System-Controller **502**, der Information über einen Parallelbus **504** empfängt, mit dem der System-Controller mit einem der Koppelfeldmodule (in Abb. 5 nicht dargestellt) gekoppelt ist. Wie oben erwähnt, umfasst der parallele Bus **504** einen 16-Bit breiten Datenbus und einen 5-Bit breiten Steuerbus. Die vom Koppelfeldmodul gelieferten DMA-Anforderungen werden vom System-Controller **502** empfangen und verarbeitet. Der System-Controller **502** handhabt die Kommunikationsprotokolle zwischen der Speicherkarte und dem Koppelfeldmodul; auch kann er Funktionalitäten wie Fehlerkorrektur zur Behandlung von Bus-Fehlern umsetzen. Daneben kann der System-Controller **502** gegebenenfalls auch die eingegangene ZIELADRESSE teilweise bearbeiten, um festzustellen, welches der spezifischen Speicherelemente auf der Speicherkarte mit der ZIELADRESSE verbunden ist.

[0089] Im Beispielsystem kommuniziert der System-Controller **502** mit einer Anzahl individueller Flash-Controller **506**, **508**, **510** und **512** über direkte Verbindungen. Der System-Controller **502** kommuniziert auch mit einer CPU **503**, die ebenso mit den einzelnen Flash-Controllern kommunizieren kann. Diese Kommunikation kann über eine direkte Verbindung mit den Flash-Controllern erfolgen, wie in Abb. 5 dargestellt, oder durch eine so genannte „Pass-through"-Verbindung, bei welcher die CPU **503** mit dem System-Controller **502** kommuniziert, um auf die Flash-Controller zuzugreifen. Dies wird im Allgemeinen gemacht, um Bus-Fanout zu verringern, und ist eine Entwurfskriterium, das Fachleuten in diesem Gebiet offensichtlich ist.

[0090] Im illustrierten Beispiel wird die Kommunikation zwischen dem System-Controller **502** und den Flash-Controllern **506**, **508**, **510** und **512** durch den Einsatz unabhängiger paralleler Busse mit einer Breite von 16 Bit erreicht, wobei ein solcher unabhängiger paralleler Bus zwischen dem System-Controller **502** und jedem der einzelnen Flash-Controller **506**, **508**, **510** und **512** gekoppelt ist. Ähnliche eigenständige parallele Busse mit einer Breite von 16 Bit können in Ausführungsformen eingesetzt werden, in denen die CPU **503** mit jedem Flash-Controller **506**, **508**, **510** und **512** direkt kommuniziert.

[0091] Jeder der einzelnen Flash-Controller ist mit einem physikalischen Flash-Speicherraum **513**, **514**, **515** bzw. **516** und mit dem Controller-Speicher **517**, **518**, **519**, und **520** verbunden (welcher beispielsweise ein DDR RAM-Speicher sein kann). Im dargestellten Beispiel wird jeder einzelne Flash-Speicherraum

(**513**–**516**) aus zehn unabhängigen Flash-Speicherchips gebildet. Im Betrieb verarbeitet der Flash-Controller **502** DMA-Anforderungen zur Speicherung bereitgestellter Daten in einem bestimmten physikalischen Flash-Speicher sowie zum Abruf der angeforderten Daten und der Bereitstellung derselben an den System-Controller **502**. Es kann eine Anzahl unterschiedlicher Typen von Flash-Controllern eingesetzt werden. Ein bevorzugter Controller und sein Betrieb sind ausführlicher in den am 5. September 2009 eingereichten und ebenfalls anhängigen US-Patentanmeldungen Nr. 12/554,888, 12/554,891 und 12/554,892 beschrieben, welche hiermit durch Bezugnahme zum Bestandteil dieses Dokuments werden.

[0092] Da insbesondere das System **100** der vorliegenden Offenlegung in einer Art und Weise operiert, dass es sich bei den von den E/A-Schnittstellenmodulen empfangenen ZIELADRESSEN um die den Speicherkarten bereitgestellten Adressen handelt, kann das vorliegende System vor allem ohne weiteres sowohl mit einem Flash-Speicher (welcher letztendlich die Übersetzung der empfangenen ZIELADRESSE in einen physikalischen Flash-Adressort erfordert) und mit einem RAM-Speicher (oder einem anderen Speicher, der keine Übersetzung der logischen Adresse in eine physikalische Adresse erfordert) benutzt werden. Da eine Umwandlung einer ZIELADRESSE in eine physikalische Flash-Adresse erst erfolgt, nachdem die DMA-Anforderung den Speicherkarten bereitgestellt wurde, erlaubt das vorliegende System darüber hinaus den Einsatz sowohl von Flash-Speicherkarten als auch von RAM-Speicherkarten. Dies ist vor allem deshalb möglich, da der Betrieb der Flash-Controller bei der Zuordnung einer empfangenen ZIELADRESSE an eine bestimmte physikalische Flash-Adresse für den Schnittstellen-Bus, der die Koppelfeldmodule mit den Speicherkarten koppelt, zum Großteil transparent ist.

[0093] Abb. 6 veranschaulicht eine alternative Ausführungsform einer Speicherkarte **600**, welche einen RAM-Speicher benutzt, und welche im System der vorliegenden Offenlegung eingesetzt werden kann. In dieser Ausführungsform wird die Adresse, die der Speicherkarte vom Koppelfeldmodul zur Verfügung gestellt wird, als physikalische Adresse für den Zugriff auf den RAM-Speicher eingesetzt. Wie in der Abbildung ersichtlich, umfasst die Speicherkarte **600** eine Anzahl von Elementen, die alle auf einer einzigen mehrschichtigen Leiterplatte positioniert und an dieser befestigt werden können.

[0094] Im Allgemeinen umfasst die Speicherkarte **600** einen System-Controller **602** der, wie der Controller **502** in Abb. 5, eine DMA-Anforderung aus dem Koppelfeldmodul erhält und die Anfforderung unter Verwendung des für das betreffende System übernommenen Protokolls (welches gegebenenfalls eine Fehlerkorrektur enthält oder mit einer solchen kombiniert ist) verarbeitet. Der System-Controller leitet dann die Anforderung und eine etwaige Datennutzlast an einen Speicher-Controller **604** weiter, der für schnelle DMA-Übertragungen der empfangenen Daten an die der ZIELADRESSE entsprechende physikalische Adresse (bei einer SCHREIB-Anforderung) sorgen oder Daten aus dem RAM-Speicher **606** (welcher ein DDR, DDR2, DDR3 oder jeder andere RAM-Hochgeschwindigkeitsspeicher sein kann) mithilfe einer DMA-Übertragung abrufen und diese Daten dem System-Controller **602** (bei einer LESE-Anforderung) bereitstellen kann. Es versteht sich, dass die Ausrichtung der in Abb. 6 dargestellten Pfeile nur der Darstellung des Beispiels einer SCHREIB-Operation dienen soll. Ebenso versteht sich, dass für Zwecke der Umsetzung der Controller **602** und der Controller **604** in einem FPGA-Bauelement kombiniert werden können.

ALTERNATIVE SYSTEMAUSFÜHRUNGSFORM:

[0095] Im oben beschriebenen System **100** werden Koppelfeldmodule eingesetzt, um seriell übertragene datenbezogene Anforderungen zu empfangen und diese Anforderungen in parallel übertragene Anforderungen umzuwandeln, welche den Speicherkarten bereitgestellt werden. Es werden alternative Ausführungsformen in Betracht gezogen, bei welchen die Koppelfeldmodule entfallen und die datenbezogenen Anforderungen seriell direkt aus den E/A-Modulen an die Speicherkarten übertragen werden.

[0096] Abb. 7 veranschaulicht ein alternatives Speichersystem **700**, bei welchem E/A-Module oder E/A-Bausteine mithilfe serieller Hochgeschwindigkeits-Kommunikationsverbindungen ohne den Einsatz von Koppelfeldmodulen direkt mit Speichersteckkarten kommunizieren können.

[0097] Bezugnehmend auf Abb. 7 umfasst das alternative Speichersystem **700** eine Vielzahl von E/A-Modulen **701**, **702**, **703**, **704**, **705**, **706**, **707**, **708**, **709**, **710**, **711** und **712**, wobei jedes von ihnen über einen oder mehrere Kommunikationskanäle mit einem oder mehreren externen Host-Geräten kommuniziert. Im dargestellten Beispiel verfügt jedes E/A-Modul über eine Vielzahl bidirektionaler serieller Vollduplex-Hochgeschwindigkeits-Kommunikationskanäle, die den einzelnen E/A-Modulen die Kommunikation mit jedem einer Vielzahl von Speichersteckkarten **740**, **742**, **744** und **746** erlauben. Im Beispiel von Abb. 7 verfügt jedes E/A-Modul **701**–**712** über acht serielle Kommunikations-Ports, wobei jede Linie, die ein E/A-Modul und eine Speichersteckkarte verbindet, zwei separate Vollduplex-Kommunikationskanäle darstellen soll.

[0098] Die E/A-Module **701**–**712** sind mit den Speichersteckkarten **740**, **742**, **744** und **746** so gekoppelt, dass Kommunikation mit extrem hoher Bandbreite er-

möglicht wird. So versorgt beispielsweise das erste E/A-Modul **701** acht serielle Datenports (zwei für jede Speichersteckkarte). Damit beträgt die vom E/A-Modul **701** bewältigbare maximale Datenrate das Achtfache der maximalen Bandbreite der seriellen Kommunikationskanäle.

[0099] Im Betrieb operieren die E/A-Module **701**–**712** ähnlich den oben in Verbindung mit Abb. 2 und Abb. 3 beschriebenen E/A-Modulen. Da jedoch die E/A-Module von Abb. 7 eine größere Anzahl serieller Kommunikationskanäle versorgen, ermöglichen sie die gleichzeitige und unabhängige Kommunikation von mehr datenbezogenen Anforderungen oder Antworten. Dadurch ergibt sich für das Speichersystem **700** allgemein sowie spezifisch zwischen den E/A-Modulen **701**–**712** und den Speichersteckkarten **740**, **742**, **744** und **746** eine größere Bandbreite. Da beispielsweise das E/A-Modul **701** acht serielle Kommunikations-Ports (zwei für jede der Speichersteckkarten) versorgt, könnte es gleichzeitig (oder beinahe gleichzeitig) und eigenständig an jede der Speichersteckkarten **740**, **742**, **744** und **746** zwei SCHREIB-Anforderungen für insgesamt acht gleichzeitig (oder beinahe gleichzeitig) verarbeitete SCHREIB-Anforderungen ausgeben. Ohne die Koppelfelder befinden sich die gleiche Arbitrierung und der gleiche Rückhaltemechanismus, wie diese im Originalsystem **100** angeführt sind, in einigen Ausführungsformen auf jeder Speichersteckkarte.

[0100] Im Beispiel von Abb. 7 wird jede serielle System-Kommunikationsverbindung dazu benutzt, sowohl Daten wie auch Steuerinformation an die E/A-Module, die Kommunikationsbausteine und die Speichersteckkarten weiterzugeben oder von diesen abzugeben. Unter Anderem empfängt jede Speichersteckkarte digitale Daten, antwortet auf SCHREIB-Anforderungen (digitale Daten an bestimmten Orten im physikalischen Speicherraum auf der Speichersteckkarte zu speichern) und antwortet auf LESE-Anforderungen (an bestimmten Orten des physikalischen Speichers auf der Platte gespeicherte digitale Daten abzurufen und bereitzustellen). Da jede serielle Hochgeschwindigkeits-Kommunikationsverbindung in der Lage ist, serielle Daten mit einer sehr hohen Geschwindigkeit (im vorliegenden Beispiel mit 625 MB/s) zu empfangen oder zu übertragen, kann das System von Abb. 7 Daten mit einer sehr hohen Datenrate speichern und abrufen.

[0101] Abb. 8 veranschaulicht Einzelheiten bezüglich der Struktur einer beispielhaften Speichersteckkarte **800** für den Einsatz im System **700** von Abb. 7.

[0102] Betrachtet man Abb. 8, umfasst die Speichersteckkarte **800** eine Speichersteuereinheit **802** und einen physikalischen Speicherraum **804**, geformt aus einer Vielzahl einzelner Speicherchips. Die Speichersteckkarte umfasst darüber hinaus entsprechende Schnittstellenschaltkreise **806**, um der Speichersteckkarte den Empfang einer Vielzahl serieller Kommunikationsverbindungen zu ermöglichen. In Abb. 8 ermöglichen die Schnittstellenschaltkreise **806** der Speichersteckkarte **800**, Eingaben von vierundzwanzig seriellen Kommunikationsverbindungen zu empfangen. Im Beispiel von Abb. 8 sind sämtliche verschiedenen Bauteile auf derselben Leiterplatte **800** angeordnet

[0103] Die Speichersteuereinheit **802** kann durch den Einsatz eines so genannten Field Programmable Gate Array oder „FPGA", beispielsweise des bei Xilinx erhältlichen und mit 333 MHz betriebenen Virtex-6 FPGA (XC6VLX240T-2FFG1156CES), gebildet werden. Es wird jedoch darauf hingewiesen, dass die Speichersteuereinheit **802** alternativ unter Verwendung anderer Typen von FPGA-Bausteinen, diskreten Schaltungen, einem programmierten Mikroprozessor oder einer Kombination aus beliebigen oder allen oben genannten Elementen umgesetzt werden könnte.

[0104] Im dargestellten Beispiel ist der physikalische Speicherraum **804** unter Verwendung einer Vielzahl so genannter Double Data Rate Dynamic Random Access Speicherchips („DDR"), beispielsweise des bei der Micron Technologies, Inc. erhältlichen und mit 333 MHZ betriebenen DDR3-800 (MT4J128M8BY-25E, aufgebaut. Es versteht sich jedoch, dass andere Formen von Speichern wie alternative DDR-Bauteile, Flash und andere Speichertypen verwendet werden können, ohne dass dadurch von den Lehren dieser Offenlegung abgewichen wird.

[0105] Im vorliegenden Beispiel werden 288 externe parallele Datenleitungen zum physikalischen Speicherraum **804** (in Abb. 9 besser erkennbar) von der Speichersteuereinheit **802** angetrieben. Da der Speicherraum mit 333 MHz getaktete DDR-Speicher benutzt, ist die Hochgeschwindigkeits-Speichersteckkarte von Abb. 8 in der Lage, für jeden Taktzyklus bis zu 576 Bits digitaler Information (512 Bits Daten und 64 Bits ECC) im physikalischen Speicherraum zu speichern, und bietet damit bei Daten und ECC eine Gesamtspeichergeschwindigkeit von etwa 24000 MB/s oder bei Daten allein von 20,83 GB/s.

[0106] Abb. 9 veranschaulicht Teile der Struktur einer beispielhaften Hochgeschwindigkeits-Speichersteckkarte in größerem Detail. Insbesondere veranschaulicht Abb. 9 Strukturen innerhalb der Speichersteuereinheit **802** von Abb. 8, die dazu verwendet werden kann, digitale Daten aus einer seriellen Kommunikationsverbindung zu empfangen und mit diesen Daten für die Speicherung innerhalb des physikalischen Speicherraumes **804** zu arbeiten.

[0107] In Abb. 9 ist eine der seriellen Hochgeschwindigkeits-Kommunikationsverbindungen, die serielle

Verbindung **900** dargestellt, gekoppelt mit der Speichersteuereinheit **802**. Die serielle Hochgeschwindigkeits-Kommunikationsverbindung **900** ist zu einem Serialisierungs-/Deserialisierungs- und Packmodul **912** vorgesehen, das innerhalb der Speichersteuerungsschaltung **802** angeordnet sein wird. Dieses in **Abb. 10A** weiter erläuterte Serialisierungs-/Deserialisierungs- und Packmodul **912** zeigt die serielle Verbindung, welche einen Eingang zu einem Multi-Gigabit-Transceiver (MGT) **1010** darstellt, der innerhalb der Speichersteuereinheit **802** angeordnet sein wird. Der MGT **1010** empfängt serielle Daten aus der seriellen Kommunikationsverbindung und wandelt die empfangenen seriellen Daten in regelmäßigen Abständen in parallele Daten um. Weiter zum Beispiel von **Abb. 10A**; dort empfängt der MGT **1010** serielle Daten und stellt parallele 32-Bit breite Daten bei 156,25 MHz bereit

[0108] Im Beispiel von **Abb. 10A** werden die 32-Bit breiten parallelen Daten aus dem MGT **1010** einem 256-Bit breiten, mehrere Wörter tiefen First-In-First-Out(FIFO-)Speicherpuffer **1012** auf solche Art und Weise bereitgestellt, dass die empfangenen 32-Bit breiten Daten aus dem MGT **1010** in 256-Bit breite Datenwörter verpackt werden. Dieser Vorgang ist in **Abb. 10B**–**Abb. 10D** allgemein dargestellt.

[0109] In **Abb. 10B** ist der 256-Bit breite FIFO-Speicher **1012** schematisch dargestellt, wie er mit einem 256-Bit breiten Wort A1–A8 gepackt ist. Das Verfahren, mit dem das dargestellte System die 32-Bit breiten Daten aus MGT **1010** zur Bildung von 256-Bit breiten Wörtern verpackt, ist wie folgt. Zunächst geht ein anfängliches 32-Bit breites Wort, A1 des Beispiels, aus dem MGT **1010** ein. Dieses anfängliche 32-Bit breite Wort A1 wird an einem anfänglichen Datenort gespeichert, wie in **Abb. 10B** dargestellt. Das nächste aus dem MGT **1010** eingegangene 32-Bit breite Wort, A2 des Beispiels, wird an einem zweiten Ort gespeichert, das dritte Wort A3 an einem dritten Ort usw., bis acht Wörter A1–A8 eingegangen sind und im FIFO **1012** gespeichert wurden, so dass sie das 256-Bit breite Wort A1–A8 bilden, wie in **Abb. 10B** dargestellt.

[0110] Sobald das 256-Bit breite Wort A1–A8 durch den oben beschriebenen Vorgang gebildet wurde und ein weiteres 32-Bit breites Wort, B1 des Beispiels, vom MGT **1010** eingeht, wird das anfängliche Wort A1–A8 nach „unten" an einen anderen Ort innerhalb des FIFO **1012** verschoben und das neu eingegangene 32-Bit breite Wort am anfänglichen Ort gespeichert, wie in **Abb. 10C** dargestellt. Dieser Vorgang wiederholt sich, bis ein zweites vollständiges 256-Bit breites Wort B1–B8 aufgebaut ist, wie in **Abb. 10C** dargestellt. Der Vorgang des Empfangen von 32-Bit-Wörtern aus dem MGT **1010**, des Packens der eingegangenen 32-Bit-Wörter in 256-Bit-Wörter und des Verschiebens der 256-Bit breiten Wörter durch den FIFO **1012** setzt sich fort, bis die 256-Bit breiten Speicherorte innerhalb des FIFO **1012** vollständig beschrieben sind. Ein Beispiel für einen stärker mit Daten beschriebenen FIFO **1012** ist in **Abb. 10D** dargestellt.

[0111] Zwar kann die „Tiefe" des Stackings von FIFO **1012** je nach Ausführungsform unterschiedlich sein, im Beispiel der **Abb. 9** und **Abb. 10A**–**Abb. 10D** ist jedoch der FIFO **1012** mindestens zweiunddreißig 256-Bit breite Wörter tief. Dies liegt daran, dass im vorliegenden Beispiel eine Gruppe von zweiunddreißig 256-Bit breiten Wörtern die Basiseinheit für Datenübertragungen in den/aus dem physikalischen Speicherraum **804** ist. Als solche wird jede Gruppe von zweiunddreißig 256-Bit breiten Wörtern in diesem Dokument als Basic Memory Cycle(„BMC", Speicherzyklusbasis-)Einheit bezeichnet. In diesem Beispiel umfasst jede BMC-Einheit 8.192 Datenbits (32 × 256) oder 1 KB Daten.

[0112] Unter Bezugnahme auf **Abb. 9** werden die Daten aus dem Pack-FIFO **912** an einen Mehrkanal-FIFO-Puffer **914** bereitgestellt – von diesen ist in **Abb. 9** nur ein einzelner Kanal dargestellt. Der Mehrkanal-FIFO-Puffer **914** ist im dargestellten Beispiel 256 Bit breit, kann Daten mit einer Rate von 256 Bit/s empfangen und ist mit einer Rate von 27,7 MHz getaktet. Daten werden aus dem Serialisierungs-/Deserialisierungs- und Packmodul **912** an den Mehrkanal-FIFO-Puffer **914** mithilfe einer „Burst"-Übertragung bereitgestellt, in welcher eine vollständige BMC-Einheit von Daten über zweiunddreißig Taktzyklen übertragen wird.

[0113] Sobald eine vollständige BMC-Einheit von Daten aus dem Serialisierungs-/Deserialisierungs- und Packmodul **912** an den Mehrkanal-FIFO-Puffer **914** übertragen ist, umfasst der Mehrkanal-FIFO-Puffer **914** eine vollständige BMC-Einheit von Daten in Form von 32 aufeinanderfolgenden 256-Bit-„Wörtern". Im dargestellten Beispiel ist die Tiefe des Mehrkanal-FIFO-Puffers **914** so gestaltet, dass er in der Lage ist, mehrere BMC-Einheiten von Daten zu speichern. Dem wird allgemein in **Abb. 11** Rechnung getragen, wo der Inhalt eines Mehrkanal-FIFO-Puffers **914** abgebildet ist, welcher 3 BMC-Einheiten von Daten umfasst (dargestellt durch die Blöcke **1140a**, **1140b** und **1140c**).

[0114] Mit Rückbezug auf **Abb. 9** werden die im Mehrkanal-FIFO-Puffer **914** gespeicherten Daten über einen 256-Bit breiten und mit 333 MHz operierenden parallelen Datenbus an einen Fehlerkorrektur- und Datenschutzschaltkreis **916** bereitgestellt. Der Fehlerkorrektur- und Datenschutzschaltkreis **916** verarbeitet die eingegangenen Daten, um einen oder mehrere Fehlerkorrektur-Bits (Error Correction Code Bits, „ECC"-Bits) einzuführen und die Daten für den erhöhten Schutz gegen Datenbeschä-

digung zu verarbeiten. Beispiele von ECC-Verarbeitung sind Fachleuten mit durchschnittlichen Kenntnissen dieses Bereichs bekannt; auch kann jeder beliebige ECC-Verarbeitungsmechanismus eingesetzt werden. Daneben implementiert der Fehlerkorrektur- und Datenschutzschaltkreis gegebenenfalls andere Datenschutzmethoden oder Datenschutzumwandlungen wie „Chipkill" und andere gemeinhin für den erweiterten Datenschutz eingesetzte Methoden. Bei Chipkill, wie er dem durchschnittlichen Fachmann auf dem Gebiet bekannt ist, handelt es sich um eine von IBM entwickelte fortschrittliche ECC-Verarbeitung, welche Computerspeichersysteme gegen das Versagen eines einzelnen Speicherchips wie auch gegen Multibit-Fehler aus einem Teil eines einzelnen Speicherchips schützt. Nähere Informationen zu Chipkill sind in der nachfolgenden Arbeit nachzulesen, welche durch Bezugnahme hiermit zum Bestandteil dieses Dokuments wird: Timothy J. Dell, A White Paper on the Benefits of Chipkill-Correct ECC for PC Server Main Memory, (1997), IBM Microelectronics Division.

[0115] Als Ergebnis der Operationen des Fehlerkorrektur- und Datenschutzschaltkreises **916** werden jedem dem Schaltkreis bereitgestellten 256-Bit „Wort" Bits hinzugefügt; die sich daraus ergebende Ausgabe des Schaltkreises ist dann im Beispiel 288 Bits für jede Eingabe von 256 Bits. Diese Ausgabe von 288 Bits entspricht den dem ECC- und Datenschutzschaltkreis **916** bereitgestellten Eingabedaten und den vom Schaltkreis **916** hinzugefügten Schutz- und ECC-Bits.

[0116] Die 288-Bit breiten Ausgaben aus dem Datenschutzschaltkreis **916** werden über einen 288-Bit breiten Bus an zwei FIFO-Ausgangspuffer **918** und **920** geliefert. Im Beispiel ist dieser parallele Bus mit 333 MHz getaktet. Die Daten aus dem Datenschutzschaltkreis werden den FIFO-Ausgangspuffern **918** und **920** im „Ping-Pong"-Stil dergestalt zur Verfügung gestellt, dass das erste 288-Bit „Wort" aus dem Schaltkreis einem der FIFO-Ausgangspuffer (beispielsweise FIFO-Ausgangspuffer **918**) geliefert wird, während das nächste 288-Bit „Wort" im darauffolgenden Taktzyklus an den anderen der FIFO-Ausgangspuffer (beispielsweise FIFO-Puffer **920**) ergeht. Auf diese Weise werden Daten, die einer vollständigen BMC-Einheit entsprechen, so in die zwei FIFO-Ausgangspuffer **918** und **920** geschrieben, dass sich in jedem FIFO-Ausgangspuffer die Hälfte der Daten befindet. Im Beispiel weist jeder der FIFO-Ausgangspuffer **918** und **920** eine solche Tiefe auf, dass jeder Puffer Daten speichern kann, die mehreren BMC-Einheiten entsprechen.

[0117] Dieser Vorgang der Verschiebung von Daten aus dem Schaltkreis **916** zum FIFO-Ausgangspuffer **918** und **920** ist in Abb. 12 und Abb. 12A allgemein veranschaulicht, in welchen beispielhafte Inhalte der FIFO-Ausgangspuffer **918** und **920** nach der Übertragung einer BMC-Einheit dargestellt sind. Wie in den Bildern dargestellt, sind die Daten der beispielhaften BMC-Einheit in beiden Puffern enthalten; auch sind die Daten, welche die gesamte BMC-Einheit bilden, über die beiden FIFO-Ausgangspuffer **918** und **920** miteinander „verschachtelt".

[0118] Zurück zu Abb. 9: Die Daten aus den zwei FIFO-Ausgangspuffern **918** und **920** werden an einen 576-Bit breiten Hochgeschwindigkeits-Speicherausgangspuffer **922** geliefert. Im Beispiel von Abb. 9 liefert jeder FIFO-Ausgangspuffer **918** und **920** mit jedem Taktzyklus 288 Bit Daten- und Fehlerschutzinformation an den Hochgeschwindigkeits-Speicherausgangspuffer **922**, während die Eingabe in den Hochgeschwindigkeits-Speicherpuffer **922** im Beispiel mit 333 MHz getaktet ist. Bei dieser Geschwindigkeit und Datenübertragungsrate kann eine vollständige BMC-Einheit von Daten (32 Wörter) in 16 Taktzyklen übertragen werden. Hier werden 16 Wörter in jedem der FIFO-Ausgangspuffer **918** und **920** gespeichert; so sind zur Übertragung des gesamten BMC 16 Taktzyklen erforderlich.

[0119] Die Koppelung zwischen den FIFO-Ausgangspuffern **918** und **920** und dem Hochgeschwindigkeits-Speicherausgangspuffer **922** ist so ausgeführt, dass die vorher verschachtelten und in den FIFO-Ausgangspuffern **918** und **920** gespeicherten Daten zusammengelegt werden und so ein einziges ordnungsgemäß geordnetes „Superwort" von 576 Bits bilden, in welchem die Daten die Reihenfolge darstellen, in welcher sie ursprünglich über die serielle Hochgeschwindigkeits-Kommunikationsverbindung **900** empfangen wurden.

[0120] Zurück zu Abb. 9: Die Daten aus dem Hochgeschwindigkeits-Ausgangspuffer werden an den (nicht dargestellten) physikalischen Speicher **804** über einen 288-Bit breiten parallelen Bus bereitgestellt, der mit 333 MHz DDR getaktet ist. Da im DDR-Takt Daten an beiden Flanken des Taktes übertragen werden, erfolgt die Übertragung der Daten aus dem Ausgangs-Speicherpuffer **922** mit eine effektiven Übertragungsrate von 667 MHz. Daher beträgt die effektive Datenübertragungsrate des Schaltkreises von Abb. 9 zum Speicher 20,83 GB/s.

[0121] Zur Erläuterung: Im Beispiel von Abb. 9 ist die Operation des vorliegenden Systems in Verbindung mit lediglich einem einzelnen seriellen Hochgeschwindigkeitseingang, im Beispiel serieller Eingang **701** (siehe Abb. 7), veranschaulicht und beschrieben. Im vollständigen beispielhaften System können bis zu 24 individuelle serielle Hochgeschwindigkeits-Kommunikationsverbindungen vorgesehen werden. Die Art und Weise, in welcher Daten aus einem solchen mit mehreren seriellen Kommunikationsverbindungen ausgestatteten System vom vor-

liegenden System verarbeitet werden, ist in den Abb. 13A–Abb. 13B wiedergegeben.

[0122] Bezugnehmend zuerst auf Abb. 13A: Hier ist ein System veranschaulicht, das dem oben in Verbindung mit Abb. 9 besprochenen sehr ähnlich ist. Im Beispiel von Abb. 13A jedoch wurden der Eingangs-MGT **1010** und FIFO-Packungspuffer **1012** für den beispielhaften Hochgeschwindigkeitseingang kombiniert, so dass sie einen Eingabeverarbeitungsblock **1370** bilden. Zusätzlich sind 11 andere serielle Eingänge **1371**–**1381** dargestellt, von denen jeder über seinen eigenen entsprechenden Eingabeverarbeitungsblock verfügt. Jeder dieser Eingabeverarbeitungsblöcke arbeitet wie die oben in Verbindung mit Abb. 9 und Abb. 10A–Abb. 10D beschriebenen Schaltkreise, um in der Lage zu sein, dem Mehrkanal-FIFO-Puffer **914** eine vollständige BMC-Einheit von Daten in der Art eines Burst-Modus zu liefern.

[0123] Wegen des Aufbaus des Systems von Abb. 13A kann das System Daten mit einer sehr hohen Rate dergestalt empfangen, dass unter besten Betriebsbedingungen Daten aus der SCHREIB-Operation beinahe immer an den Mehrkanal-FIFO-Puffer **914** und Daten aus dem Mehrkanal-FIFO-Puffer **914** beinahe immer an den Fehlerkorrektur- und Datenschutzschaltkreis **916** geliefert werden.

[0124] Das System von Abb. 13A verarbeitet jedoch vorzugsweise Daten aus einer Hälfte der vom beschriebenen System bereitgestellten seriellen Hochgeschwindigkeits-Kommunikationsverbindungen. Wie in Abb. 13B veranschaulicht, umfasst das beispielhafte System einen weiteren Satz von Schaltkreisen ähnlich den oben beschriebenen, einschließlich eines Mehrkanal-FIFO-Puffers **914'**, der Daten aus den verbleibenden 12 seriellen Kommunikationsverbindungen verarbeitet und diese Daten einem zweiten Fehlerkorrektur- und Datenschutzschaltkreis **916'** bereitstellt.

[0125] Wie in Abb. 13B dargestellt, liefert der zweite Fehlerkorrektur- und Datenschutzschaltkreis **916'** seine Ausgabe im „Ping-Pong"-Stil an die FIFO-Ausgangspuffer **918** und **920**. Im beispielhaften System arbeitet der Vorgang zur Übertragung von Daten aus dem zweiten Fehlerkorrektur- und Datenschutzschaltkreis **916'** jedoch phasenungleich mit dem ersten Fehlerkorrektur- und Datenschutzschaltkreis **916**, so dass während der Datenübertragung des ersten Datenschutzschaltkreises **916** an einen der FIFO-Ausgangspuffer (z. B. an den FIFO-Ausgangspuffer **918**) der zweite Fehlerkorrektur- und Datenübertragungsschaltkreis **916'** Daten an den anderen FIFO-Ausgangspuffer (z. B. an den FIFO-Ausgangspuffer **920**) überträgt. Während des nächsten Taktzyklus wechselt die Übertragung. Auf diese Art und Weise können Daten während eines jeden Taktzyklus immer an beide FIFO-Ausgangspuffer **918** und **920** übertragen werden.

[0126] Um die Vermischung von Daten aus unterschiedlichen SCHREIB-Operationen zu vermeiden, werden die aus dem ersten und aus dem zweiten Fehlerkorrektur- und Datenschutzschaltkreis **916** und **916'** bereitgestellten Daten vorzugsweise auf eine solche Art und Weise zum FIFO-Ausgangspuffer **918** und **920** übertragen, dass sie erneut zusammengefügt werden, um wie Daten zu erscheinen, die über eine einzige serielle Kommunikationsverbindung eingegangen sind. Ein Ansatz, mit dem dies erreicht wird, ist in Abb. 14A und Abb. 14B dargestellt.

[0127] In Abb. 14A und Abb. 14B wird während einer ersten Datenübertragungsoperation das erste gerade Wort der BMC-Einheit, welches bereit zur Übertragung vom ersten Fehlerkorrektur- und Datenschutzschaltkreis **916** ist (z. B. Wordo, ao), vom ersten Schaltkreis **916** an einen Teil des für den ersten Schaltkreis **916** reservierten FIFO-Ausgangspuffers **918** übertragen. Gleichzeitig wird das erste zur Übertragung vom zweiten Fehlerkorrektur- und Datenschutzschaltkreis **916'** bereitstehende gerade Wort (Wordo, bo) aus der BMC-Einheit an einen Teil des für die Speicherung der Daten aus dem zweiten Schaltkreis **916'** reservierten FIFO-Ausgangspuffers **920** übertragen. Während des nächsten Taktzyklus wird das erste ungerade Wort (z. B. Word1, a1) aus dem BMC im ersten Schaltkreis **916** in einem reservierten Platz des FIFO-Ausgangspuffers **920** gespeichert, während das erste gerade Wort (z. B. Word1, b1) aus dem BMC in FIFO **916'** an einem reservierten Platz des FIFO-Ausgangspuffers **918** gespeichert wird. Auf diese Weise werden Daten immer so an beide FIFO-Ausgangspuffer **918** und **920** übertragen, dass die maximale Bandbreite erhalten bleibt.

[0128] In einem alternativen Verfahren kann das Schreiben von Daten aus dem zweiten Fehlerkorrektur- und Datenschutzschaltkreis **916'** um einen Taktzyklus verzögert werden, damit alle geraden Wörter im FIFO-Ausgangspuffer **918** und alle ungeraden Wörter im FIFO-Ausgangspuffer **920** gespeichert werden (siehe Abb. 14C und Abb. 14D).

[0129] Neben der Bereitstellung der Schaltkreise für den Empfang und die Verarbeitung von Daten, wie in Abb. 9 und in den anderen oben besprochenen Abbildungen dargestellt, kann jede Speicherkarte auch ähnliche (im Wesentlichen in umgekehrter Richtung funktionierende) Strukturen zum Abrufen von Daten aus dem RAM-Speicher mit einer hohen Datenrate und zum Entpacken der eingegebenen Daten enthalten, so dass sie über einen der seriellen Hochgeschwindigkeits-Kommunikationskanäle an ein E/A-Schnittstellenmodul übertragen werden können, welches dann die angeforderten Daten für das entsprechende Host-Gerät bereitstellen kann.

[0130] Im dargestellten Beispiel sind die Daten- und Adressleitungen, über die der Speicher-Controller **802** mit dem physikalischen RAM-Speicher **804** gekoppelt ist, so ausgeführt, dass zu jedem gegebenen Zeitpunkt nur ein LESE-Zugriff oder nur ein SCHREIB-Zugriff erfolgen kann. Somit muss der Controller **802** zur Erzielung der optimalen Leistung Arbitration irgendeiner Art vornehmen, wie in **Abb. 15** allgemein dargestellt. In einer Ausführungsform wird diese Arbitration erreicht, indem im Controller **802** ein Arbitrationsmodul **1504** enthalten ist, welches die Befehlskopfinformation für jede datenbezogene Anforderung empfängt und die Information in einem Puffer speichert, der mit dem E/A-Schnittstellenmodul zusammenhängt, welches die Anfrage gestellt hat. In einer bevorzugten Ausführungsform unterhält das Arbitrationsmodul **1504** separate Puffer **1500** und **1502** für LESE-Operationen und separate Puffer **1501** und **1503** für SCHREIB-Operationen, so dass der Controller **802** in einem gepufferten Speicher auf einem E/A-Schnittstellenmodul auf E/A-Schnittstellenmodulbasis eine Liste der LESE- und SCHREIB-Anforderungen oder der von den einzelnen E/A-Schnittstellenmodulen erhaltenen Befehle führt. Im Allgemeinen wird diese Liste gegebenenfalls auf Zeitstempelbasis oder zeitlich geordneter Basis gepflegt, wobei jeder Eintrag in der Liste Befehlsinformation enthalten kann, die mit jeder einzelnen von einem E/A-Schnittstellenmodul eingegangenen datenbezogenen Anforderung verbunden ist, nämlich die ZIELADRESSE, die Anzahl der Wörter (oder eine andere Anzeige der Menge der zu übertragenden Daten) und eine Anzeige der Übertragungsrichtung (z. B., ob LESE- oder SCHREIB-Operation).

[0131] Zur Optimierung der Leistung können die LESE und SCHREIB-Anforderungen wie folgt verarbeitet werden:
Allgemeinen gewährt das Arbitrationsmodul **1504** den LESE-Anforderungen Priorität und verarbeitet die LESE-Anforderungen in den LESE-Anforderungspuffern für die verschiedenen E/A-Schnittstellenmodule in der Reihenfolge ihres Eintreffens, sofern nicht bestimmte unten besprochene Bedingungen erfüllt sind.

[0132] Wenn das Arbitrationsmodul **1504** und der Controller **802** alle LESE-Anforderungen so verarbeitet haben, dass keine weiteren LESE-Anforderungen ausstehen, dann verarbeitet das Arbitrationsmodul **1504** und der Controller **802** etwaige in den SCHREIB-Anforderungspuffern ausstehende SCHREIB-Anforderungen in der zeitlichen Reihenfolge.

[0133] Um zu vermeiden, dass zu viele SCHREIB-Anforderungen ausstehend sind, und zur Unterstützung von Adresslatenz- und Kohärenzproblemen verarbeiten das Arbitrationsmodul **1504** und der Controller **802** eine SCHREIB-Anforderung, wenn festgestellt wird, dass die Anzahl der ausstehenden SCHREIB-Anforderungen einen bestimmten Grenzwert überschritten haben. Das Arbitrationsmodul **1504** führt diese Feststellung der Grenzwertüberschreitungen durch, indem es folgende Kriterien berücksichtigt: (i) die Gesamtanzahl ausstehender SCHREIB-Anforderungen von allen E/A-Schnittstellenmodulen zusammen, (ii) die Gesamtanzahl ausstehender SCHREIB-Anforderungen eines bestimmten E/A-Schnittstellenmoduls, oder (iii) eine beliebige Kombination der obigen. Wenn beispielsweise sowohl die Gesamtanzahl der ausstehenden SCHREIB-Anforderungen wie auch die Anzahl der SCHREIB-Anforderungen eines bestimmten E/A-Schnittstellenmoduls berücksichtigt würden, könnte das Arbitrationsmodul **1504** den Controller **802** zur Verarbeitung einer SCHREIB-Anforderung veranlassen, wenn entweder: (i) die Gesamtanzahl der ausstehenden SCHREIB-Anforderungen einen ersten Grenzwert übersteigt (in welchem Fall ausstehende SCHREIB-Anforderungen auf der Grundlage der zeitlichen Reihenfolge gehandhabt werden könnten), oder (ii) die Anzahl der ausstehenden SCHREIB-Anforderungen eines bestimmten E/A-Schnittstellenmoduls einen zweiten Grenzwert übersteigt (der niedriger als der erste Grenzwert sein kann) in welchem Fall das Arbitrationsmodul **1504** den Controller **802** veranlassen würde, SCHREIB-Anforderungen von dem mit dem Grenzwertüberschreitungspuffer verbundenen E/A-Schnittstellenmodul auf der Grundlage der zeitlichen Reihenfolge zu übernehmen.

[0134] Daneben kann der Arbitrator eine SCHREIB-Anforderung verarbeiten, wenn festgestellt wird, dass die gespeicherten Daten in einem der mit dem Empfang der Daten verbundenen FIFO-Ausgangspuffer einen bestimmten Grenzwert, beispielsweise eine Menge überschritten haben, welche der Hälfte der Speicherkapazität des FIFO-Ausgangspuffers entspricht, in welchem Fall eine mit diesem FIFO-Ausgangspuffer verbundene SCHREIB-Anforderung verarbeitet würde, so dass das E/A-Schnittstellenmodul weiterhin Daten und datenbezogene Anforderungen ohne Unterbrechung oder Drosselung übersenden kann.

[0135] Zusätzlich zu Obigem kann das Arbitrationsmodul **1504** auch steuern, wie LESE- und SCHREIB-Anforderungen zu verarbeiten sind, damit die Datenkohärenz erhalten bleibt. Wird beispielsweise festgestellt, dass eine ausstehende LESE-Anforderung an eine bestimmte Adresse gerichtet ist, und dass eine zeitlich frühere SCHREIB-Anforderung ausstehend ist, kann das Arbitrationsmodul **1504** den Controller **802** veranlassen, die ausstehende SCHREIB-Anforderung zu verarbeiten, um dafür zu sorgen, dass in Beantwortung der SCHREIB-Anforderung die richtigen Daten zurückgegeben werden.

SPEICHERCHIP-ZUGANG

[0136] Für die Übertragung von Daten an den physikalischen RAM-Speicher oder aus dem physikalischen RAM-Speicher kann der Controller **802** unterschiedliche Ansätze und Verfahren nutzen. In einer Ausführungsform kann der Controller **802** als DDR3-Speichermodul funktionieren oder ein solches einschließen, welches vom Arbitrationsmodul **1504** neben einem Startindikator zur zeitliche Steuerung der Datenübertragung die Befehlsinformation erhält. Im Allgemeinen kann die vom Arbitrationsmodul **1504** bereitgestellte Befehlsinformation Folgendes enthalten: die gleiche Information, wie sie in den für die Zwecke der Arbitration benutzten Puffern gespeichert ist, nämlich die ZIELADRESSE und noch spezifischer die Startadresse für die Übertragung, einen Identifikator der zu übertragenden Datenmenge, der aus einer Wörterzählung bestehen kann, sowie die Übertragungsrichtung. Der Controller **802** aktiviert dann in seiner Eigenschaft als Speichersteuerungsmodul die entsprechenden von der ZIELADRESSE (Startadresse) angezeigten Speicherchips.

[0137] Wenn durch die Übertragungsrichtung ein Befehl SCHREIBE in den Speicher angezeigt wird, koordiniert der Controller **802** die Ausgabe von Befehlen an die Speicherchips (beispielsweise zeitliche Abfolge und Adressbefehle), während er Daten aus dem Ausgangsspeicherpuffer **922** überträgt.

[0138] Wird durch die Übertragungsrichtung ein Befehl LESE aus dem RAM-Speicher angezeigt, dann koordiniert der Controller **802** die Ausgabe der entsprechenden Adress- und Timing-Befehle an die Speicherchips, während er Daten aus den Speicherchips an einen FIFO-Ausgangspuffer überträgt, der dem E/A-Schnittstellenmodul zugeordnet ist, an welches die Daten gerichtet sind.

[0139] In einem Beispiel empfängt der Controller **802** den nächsten Satz an Befehlsdaten, bevor die mit dem vorherigen Befehl verbundene Datenübertragung abgeschlossen ist; er ermöglicht damit, dass das System bei maximaler Bandbreite operiert.

[0140] Der allgemeine Betrieb des Controllers **802** bei der Steuerung der Übertragung von Daten an den physikalischen RAM-Speicher und von diesem ist in den **Abb. 16** und **Abb. 17** allgemein dargestellt.

[0141] Bezugnehmend zunächst auf **Abb. 16** ist eine bestimmte Anordnung des physikalischen RAM-Speichers dargestellt. In der Ausführungsform von **Abb. 16** ist ein Controller **802** veranschaulicht; auch sind die Verbindungen zwischen dem Controller **802** und dem physikalischen RAM-Speicher dargestellt. In diesem Beispiel nimmt der an den Controller **802** gekoppelte physikalische RAM-Speicher die Form von 72 (zweiundsiebzig) Speicherchips an, die in vier Abschnitte (**1602**, **1604**, **1606** und **1608**) unterteilt ist, wobei jeder Abschnitt aus 18 Speicherchips besteht. In diesem Beispiel weist jeder Speicherchip vier Bänke auf (A, B, C, D). Speicherchips sind so miteinander gekoppelt, dass sie Folgendes gemeinsam nutzen: (a) Adress- und Steuerleitungen (22 im vorliegenden Beispiel), sowie (b) Datenleitungen (72 im vorliegenden Beispiel). Die in **Abb. 16** dargestellte Koppelung ist lediglich eines der zur Verbindung der physikalischen Speicherchips mit dem Controller **802** möglichen Verfahren. Alle vier Abschnitte arbeiten im Gleichklang, obwohl sie verschiedene Steuerbusse haben; auch werden ihnen Befehle gleichzeitig und für dieselbe Bank erteilt.

[0142] Weitere Details bezüglich der Operation des Speicher-Controllers **802** sind in **Abb. 17** dargestellt. **Abb. 17** veranschaulicht den Betrieb des Controllers **802** über zwei grundlegende Speicherzyklen (wobei der zweite grundlegende Speicherzyklus hier als unvollständig dargestellt ist). Im Allgemeinen ist bei jedem Speicherzyklus die Übertragung von 32 288-Bit-Wörtern an den (oder vom) DDR-Speicher involviert. Im spezifischen Beispiel findet die Übertragung über 16 tatsächliche Taktzyklen statt. Bei 333 MHz ergibt dies für jede Speicherkarte eine Übertragungsrate von 1 Kilobyte Daten alle 48 ns oder von 20,83 Gigabyte/Sekunde (nur Daten). Wenn man berücksichtigt, dass ein System wie das in **Abb. 7** dargestellte unter Umständen über mehrere Speicherkarten verfügt, kann die gesamte Bandbreite des Systems signifikant über 20,83 GB/s liegen und bei Einsatz von fünf oder mehr Speicherkarten 100 GB/s übersteigen.

[0143] Bezugnehmend auf **Abb. 17** wird durch Aktivierung von Bank A des Speichers über die mit jedem Abschnitt verbundenen Befehlsbusse ein grundlegender Speicherzyklus eingeleitet. Im Beispiel von **Abb. 17** geschieht dies beim Taktzyklus 1. Vier Taktzyklen später, bei Taktzyklus 5, wird Bank B ein Aktivierungsbefehl bereitgestellt. Abermals vier Zyklen später, bei Taktzyklus 9, wird Bank C aktiviert, und wiederum vier Zyklen nach diesem Ereignis, bei Taktzyklus 13, erfolgt die Aktivierung von Bank D. Dieser Ansatz der selektiven Aktivierung der verschiedenen Bänke lässt eine Verletzung der physikalischen Vorladezeiten nicht zu, da eine Vorladung der Bank A zwischen den Zyklen 6 und 19 erfolgt, sobald der darauffolgende Aktivierungsbefehl für Bank A ausgegeben wird.

[0144] Für jede der verschiedenen Bänke wird fünf Zyklen nach erfolgter Aktivierung der Bänke für die betreffende Bank der Befehl geltend gemacht, mit dem angezeigt wird, ob es sich um eine LESE- oder eine SCHREIB-Übertragung handelt. Daher geschieht dies für Bank A bei Taktzyklus 6, für Bank B bei Taktzyklus 10, für Bank C bei Taktzyklus 14 und für Bank D bei Taktzyklus 18.

[0145] Fünf Taktzyklen nach der Bereitstellung der Angabe, ob es sich bei der Übertragung um eine LESE- oder eine SCHREIB-Übertragung handelt, werden die Daten den Datenleitungen für vier aufeinanderfolgende Zyklen bereitgestellt. Da die Datenübertragung mit der doppelten Datenrate (DDR) stattfindet, werden in jedem Speichertaktzyklus zwei Wörter übertragen. Auf diese Weise werden für jede Bank über die Gesamtheit des grundlegenden Speicherzyklus acht 288-Bit-Wörter an Daten übertragen (der Vermerk beim Datenbus in **Abb. 17**, Wo, umfasst zwei 288-Bit-Wörter).

[0146] Während der Datenübertragung für den hier besprochenen grundlegenden Speicherzyklus wird der grundlegende Speicherzyklus für den nächsten Speicherzyklus durch die Geltendmachung des Aktivierungsbefehls für Bank A bei Taktzyklus 19 eingeleitet. Der Rest des nachfolgenden grundlegenden Speicherzyklus folgt der obigen Beschreibung bezüglich des ersten grundlegenden Speicherzyklus.

[0147] Wie oben beschrieben, erfolgt in jedem grundlegenden Speicherzyklus die Übertragung eines kompletten Kilobyte (1 KB) Daten. In einer Ausführungsform kann das System „halbe Schreibzyklen" vorsehen, wenn nur 512 Byte übertragen werden. Bei solchen „halben Schreibzyklen" werden die Aktivierungen für lediglich zwei aufeinanderfolgende Bänke (zum Beispiel Bank A u. B oder Bank C u. D) geltend gemacht. Die anderen Bänke bleiben während des Zyklus im Leerlauf. Die ZIELADRESSE für die Übertragungen stellt fest, welche zwei der vier Bänke für die Übertragung aktiv sind.

[0148] Die obigen Ausführungsformen dienen lediglich der Darstellung, nicht jedoch der Einschränkung. Es können andere und weitere Ausführungsformen erdacht werden, die einen oder mehrere Aspekte der oben beschriebenen Erfindungen nutzen, ohne vom Geist der offen gelegten Ausführungsformen abzuweichen. Daneben kann die hier beschriebene Reihenfolge von Schritten in unterschiedlichen Abfolgen vorgenommen werden, sofern dies nicht ausdrücklich eingeschränkt ist. Die verschiedenen hier beschriebenen Schritte können mit anderen Schritten kombiniert, zwischen die beschriebenen anderen Schritte eingeschoben bzw. in mehrere Schritte aufgeteilt werden. Auf ähnliche Weise sind Elemente funktional beschrieben und können als separate Komponenten integriert oder in Komponenten mit mehreren Funktionen eingebaut werden.

[0149] Die offen gelegten Ausführungsformen wurden im Rahmen der bevorzugten und anderer Ausführungsformen beschrieben, es wurde jedoch nicht jede Ausführungsform der Erfindung beschrieben. Dem durchschnittlichen Fachmann auf diesem Gebiet stehen offensichtliche Änderungen an den beschriebenen Ausführungsformen zur Verfügung. Die offen gelegten und nicht offen gelegten Ausführungsformen sind nicht dazu bestimmt, Umfang oder Anwendbarkeit der von den Anmeldern ersonnenen Erfindung zu begrenzen oder einzuschränken. Die Anmelder beabsichtigen, jene Änderungen und Verbesserungen, die in den Umfang oder Bereich der Äquivalente der nachfolgenden Ansprüche fallen, in vollem Umfang zu schützen.

ZITATE ENTHALTEN IN DER BESCHREIBUNG

Diese Liste der vom Anmelder aufgeführten Dokumente wurde automatisiert erzeugt und ist ausschließlich zur besseren Information des Lesers aufgenommen. Die Liste ist nicht Bestandteil der deutschen Patent- bzw. Gebrauchsmusteranmeldung. Das DPMA übernimmt keinerlei Haftung für etwaige Fehler oder Auslassungen.

Zitierte Nicht-Patentliteratur

- Timothy J. Dell, A White Paper on the Benefits of Chipkill-Correct ECC for PC Server Main Memory, (1997), IBM Microelectronics Division [**0114**]

Patentansprüche

1. Ein Flash-basiertes Speichermodul mit serieller Hochgeschwindigkeits-Kommunikation, umfassend:
eine Vielzahl von Eingabe-/Ausgabemodulen (E/A-Modulen), jedes so konfiguriert, um mit einem externen Gerät über eine oder mehrere externe Kommunikationsverbindungen zu kommunizieren;
eine Vielzahl Flash-basierter Speicherkarten, jede Flash-basierte Speicherkarte umfassend eine Vielzahl von Flash-Speichervorrichtungen, wobei jede Flash-Speichervorrichtung über einen physikalischen Speicherraum verfügt, der in Blöcke unterteilt ist, wobei jeder Block weiter in Seiten unterteilt ist und jede Seite einen individuell adressierbaren Speicherort darstellt, an welchem Speicheroperationen ausgeführt werden, wobei eine Vielzahl solcher Speicherorte gleichzeitig in Gruppierungen von je einem Block löschbar sind, sowie
eine Vielzahl von Koppelfeldelementen, wobei jedes Koppelfeldelement mit jeweils einer bestimmten Flash-basierten Speicherkarte verbunden und so konfiguriert ist, dass jedes der in E/A-Module mit der entsprechenden Flash-basierten Speicherkarte kommunizieren kann;
bei welchem jedes E/A-Modul über eine serielle Hochgeschwindigkeits-Kommunikationsverbindung mit jedem Koppelfeldelement verbunden ist, wobei jede serielle Hochgeschwindigkeits-Kommunikationsverbindung jedem E/A-Modul erlaubt, Befehle, Anweisungen bzw. Daten darstellende Bits an jedes und von jedem Koppelfeldelement zu übertragen und zu empfangen, und
bei welchen jedes Koppelfeldelement mit der jeweiligen Flash-basierten Speicherkarte durch eine Vielzahl paralleler Kommunikationsverbindungen verbunden ist, wobei jede parallele Kommunikationsverbindung ein Koppelfeldelement mit einer bestimmten Flash-Speichervorrichtung der jeweiligen Flash-basierten Speicherkarte verbindet.

2. Das Flash-basierte Speichermodul von Anspruch 1, in welchem die von den E/A-Modulen übertragenen Bits, welche Befehle, Anweisungen bzw. Daten darstellen, Bestandteil einer Anforderung auf direkten Speicherzugriff (Direct Memory Access, DMA) sind, wobei eine solche DMA-Anforderung eine Lese-Modifizier-Schreib-(Read-Modify-Write, RMW-) DMA-Anforderung einschließt.

3. Das Flash-basierte Speichermodul von Anspruch 1, bei welchem die E/A-Module konfiguriert sind, um mit einem externen Gerät mithilfe eines Hochgeschwindigkeits-Kommunikationsprotokolls über eine oder mehrere externe Kommunikationsverbindungen unter Verwendung eines der folgenden Protokolle zu kommunizieren: Fibre Channel, InfiniBand, Ethernet oder Front Panel Data Port.

4. Das Flash-basierte Speichermodul von Anspruch 3, bei welchem jedes E/A-Modul eine Hochgeschwindigkeits-Schnittstelle umfasst, so konfiguriert, dass sie mit dem E/A-Modul die Kommunikation über eine oder mehrere externe Kommunikationsverbindungen unter Verwendung eines Hochgeschwindigkeits-Kommunikationsprotokolls ermöglicht.

5. Das Flash-basierte Speichermodul von Anspruch 4, bei welchem jedes E/A-Modul ferner einen Hochgeschwindigkeits-Controller mit mehreren Multi-Gigabit-Transceivern umfasst, wobei für jede mit dem E/A-Modul verbundene Hochgeschwindigkeits-Kommunikationsverbindung ein Multi-Gigabit-Transceiver vorhanden ist, und wobei jedes E/A-Modul konfiguriert ist, um an jedes/von jedem Koppelfeld mithilfe der Multi-Gigabit-Transceiver Bits zu übertragen/zu empfangen, welche Befehle, Anweisungen bzw. Daten darstellen.

6. Das Flash-basierte Speichermodul von Anspruch 5, bei welchem jedes E/A-Modul ferner eine CPU und ein gemeinsames Schaltelement umfasst, das gemeinsame Schaltelement so konfiguriert, dass es die Kommunikation zwischen der CPU, dem externen Gerät über eine oder mehrere externe Kommunikationsverbindungen sowie mit den Multi-Gigabit-Transceivern regelt.

7. Das Flash-basierte Speichermodul von Anspruch 1, bei welchem jedes Koppelfeldelement einen mit mehreren Multi-Gigabit-Transceivern ausgestatteten Hochgeschwindigkeits-Controller umfasst, je einen Multi-Gigabit-Transceiver für jede mit dem Koppelfeld verbundene serielle Hochgeschwindigkeits-Kommunikationsverbindung, wobei jedes Koppelfeldelement konfiguriert ist, um an jedes/von jedem E/A-Modul mithilfe der Multi-Gigabit-Transceiver Bits zu übertragen/zu empfangen, welche Befehle, Anweisungen bzw. Daten darstellen.

8. Eine erweiterbare Hochgeschwindigkeits-Speicherkarte, umfassend:
eine Leiterplatte;
Schnittstellenschaltkreise, die auf der Leiterplatte montiert sind und der Hochgeschwindigkeits-Speicherkarte den Empfang von Bits, welche Anweisungen, Befehle bzw. Daten darstellen, von einem oder mehreren externen Geräten über eine oder mehrere serielle Hochgeschwindigkeits-Kommunikationsverbindungen ermöglichen;
eine auf der Leiterplatte montierte Vielzahl von Speichervorrichtungen, wobei jede Speichervorrichtung über einen physikalischen Speicherraum verfügt, in welchem Speicheroperationen ausgeführt werden, und
einen auf die Leiterplatte montierten und mit den Schnittstellenschaltkreisen und der Vielzahl von Speichervorrichtungen verbundenen Controller, wobei der Controller so konfiguriert ist, dass er die Kom-

munikation zwischen den Schnittstellenschaltkreisen und jeder einzelnen Speichervorrichtung zur Durchführung von Speicheroperationen steuert;
bei welcher die Schnittstellen-Schaltkreise mit dem Controller über eine Vielzahl serieller Hochgeschwindigkeits-Kommunikationsleitungen verbunden ist, wobei jede serielle Hochgeschwindigkeits-Kommunikationsleitung einer der seriellen Hochgeschwindigkeits-Kommunikationsverbindungen entspricht; und
bei welcher der Controller über eine vorher festgelegte Anzahl paralleler Kommunikationsleitungen mit der Vielzahl von Speichervorrichtungenn verbunden ist, der Controller so konfiguriert, dass er die Befehle, Anweisungen bzw. Daten darstellenden Bits aus den seriellen Hochgeschwindigkeits-Kommunikationsverbindungen aus einem seriellen Format in ein paralleles Format umwandelt.

9. Die erweiterbare Hochgeschwindigkeits-Speicherkarte von Anspruch 8, bei welcher eine oder mehrere serielle Hochgeschwindigkeits-Kommunikationsverbindungen 24 serielle Hochgeschwindigkeits-Kommunikationsverbindungen umfassen, und bei welcher jede serielle Hochgeschwindigkeits-Kommunikationsverbindung aus zwei seriellen Vollduplex-Kommunikationskanälen besteht, jeder serielle Kommunikationskanal so konfiguriert, dass er Kommunikation unabhängig vom anderen seriellen Kommunikationskanal überträgt.

10. Die erweiterbare Hochgeschwindigkeits-Speicherkarte von Anspruch 8, bei welcher die vorher festgelegte Anzahl paralleler Kommunikationsleitungen, welche den Controller mit der Vielzahl von Speichervorrichtungen verbinden, 288 parallele Kommunikationsleitungen umfassen.

11. Die erweiterbare Hochgeschwindigkeits-Speicherkarte von Anspruch 8, bei welcher der Controller ein Serialisierungs-/Deserialisierungs- und Packmodul umfasst, das Serialisierungs-/Deserialisierungs- und Packmodul konfiguriert, Befehle, Anweisungen bzw. Daten darstellende Bits aus den Schnittstellenschaltkreisen einem seriellen Format entsprechend einzugeben und die Befehle, Anweisungen bzw. Daten darstellenden Bits einem parallelen Format entsprechend neu zu ordnen.

12. Die erweiterbare Hochgeschwindigkeits-Speicherkarte von Anspruch 11, bei welcher das Serialisierungs-/Deserialisierungs- und Packmodul die Befehle, Anweisungen bzw. Daten darstellenden Bits einem parallelen Format entsprechend mithilfe eines Multi-Gigabit-Transceivers und einem im Serialisierungs-/Deserialisierungs- und Packmodul befindlichen Speicherpuffer neu ordnet.

13. Die erweiterbare Hochgeschwindigkeits-Speicherkarte von Anspruch 11, bei welcher der Controller ferner einen Mehrkanalpuffer umfasst, so konfiguriert, um die vom Serialisierungs-/Deserialisierungs- und Packmodul im parallelen Format neu geordneten Befehle, Anweisungen bzw. Daten darstellenden Bits zu empfangen und aus den Befehle, Anweisungen bzw. Daten darstellenden Bits eine vorher festgelegte Anzahl von Wörtern zu bauen, wobei jedes Wort aus einer vorher festgelegten Anzahl von Bits zusammengesetzt ist.

14. Die erweiterbare Hochgeschwindigkeits-Speicherkarte von Anspruch 13, bei welcher der Controller ferner mindestens einen (1) Fehlerkorrektur- und Datenschutzschaltkreis umfasst, so konfiguriert, dass er die Wörter aus dem Mehrkanalpuffer empfängt, mithilfe der Wörter ein oder mehrere Fehlerkorrekturbits erzeugt, das Fehlerkorrekturbit eines jeden Wortes dem Wort hinzufügt und jedes Wort mit den in das Wort eingefügten Fehlerkorrekturbits ausgibt.

15. Die erweiterbare Hochgeschwindigkeits-Speicherkarte von Anspruch 14, bei welcher der Controller ferner mit mindestens einem der Fehlerkorrektur- und Datenschutzschaltkreise verbundene Ausgabepuffer umfasst, wobei die Ausgabepuffer konfiguriert sind, um abwechselnd die Wörter mit den in sie eingefügten Fehlerkorrektur-Codebits von dem mindestens einen Fehlerkorrektur- und Datenschutzschaltkreis so zu empfangen, dass ein erstes Wort aus dem mindestens einen Fehlerkorrektur- und Datenschutzschaltkreis einem der Ausgabepuffer bereitgestellt wird, und dass ein nächstes Wort aus dem mindestens einen Fehlerkorrektur- und Datenschutzschaltkreis einem anderen der Ausgabepuffer bereitgestellt wird.

16. Die erweiterbare Hochgeschwindigkeits-Speicherkarte von Anspruch 15, bei welcher der Controller ferner einen Speicherpuffer umfasst, so konfiguriert, dass er die Wörter mit den in sie eingefügten Fehlerkorrektur-Codebits aus den Ausgabepuffern empfängt und eine vorher festgesetzte Anzahl von Wörtern in einer vorher festgelegten Art und Weise kombiniert, dass sie ein Superwort bilden.

17. Ein Speichermodul mit serieller Hochgeschwindigkeits-Kommunikation, umfassend:
eine erste Vielzahl von Eingabeverarbeitungsblöcken und eine zweite Vielzahl von Eingabeverarbeitungsblöcken, jeder der Eingabeverarbeitungsblöcke konfiguriert, um die Befehle, Anweisungen bzw. Daten darstellenden Bits einem seriellen Format entsprechend zu empfangen und die Befehle, Anweisungen bzw. Daten darstellenden Bits einem parallelen Format entsprechend neu zu ordnen;
eine Vielzahl von Speichervorrichtungen, wobei jede Speichervorrichtung über einen physikalischen Speicherraum verfügt, in welchem Speicheroperationen ausgeführt werden, und einem mit der ersten Vielzahl und der zweiten Vielzahl von Eingabeverarbeitungsblöcken und den Speichervorrichtungenn verbunde-

ner Controller, der Controller so konfiguriert, dass er die Kommunikation zwischen der ersten und der zweiten Vielzahl von Eingabeverarbeitungsblöcken und jeder Speichervorrichtung steuert, um die Speicheroperationen auszuführen, der Controller umfassend:
(a) einen mit der ersten bzw. der zweiten Vielzahl von Eingabeverarbeitungsblöcken verbundenen ersten Mehrkanalpuffer und zweiten Multikanalpuffer, jeder Mehrkanalpuffer konfiguriert, um die Befehle, Anweisungen bzw. Daten darstellenden Bits aus der ersten bzw. zweiten Vielzahl von Eingabeverarbeitungsblöcken im parallelen Format zu empfangen und aus den Befehle, Anweisungen bzw. Daten darstellenden Bits eine vorher festgelegte Anzahl von Wörtern zu bauen, wobei jedes Wort aus einer vorher festgelegten Anzahl von Bits zusammengesetzt ist;
(b) einen ersten Fehlerkorrektur- und Datenschutzschaltkreis und einen zweiten Fehlerkorrektur- und Datenschutzschaltkreis, verbunden mit dem ersten bzw. mit dem zweiten Mehrkanalpuffer, der erste und der zweite Fehlerkorrektur- und Datenschutzschaltkreis so konfiguriert, dass er die Wörter aus dem ersten bzw. dem zweiten Mehrkanalpuffer empfängt, mithilfe der Wörter ein oder mehrere Fehlerkorrekturbits erzeugt, die Fehlerkorrekturbits eines jeden Wortes dem Wort hinzufügt und jedes Wort mit den in das Wort eingefügten Fehlerkorrekturbits ausgibt;
(c) einen ersten Ausgabepuffer und einen zweiten Ausgabepuffer, verbunden mit dem ersten und dem zweiten Korrektur- und Datenschutzschaltkreis, der erste und der zweite Ausgabepuffer so konfiguriert, dass sie abwechselnd die Wörter mit den in sie eingefügten Fehlerkorrektur-Codebits aus dem ersten und dem zweiten Fehlerkorrektur- und Datenschutzschaltkreis so empfangen, dass ein erstes Wort aus einem der ersten oder zweiten Fehlerkorrektur- und Datenschutzschaltkreise einem der ersten oder der zweiten Ausgabepuffer bereitgestellt wird, und dass ein nächstes Wort aus einem weiteren der ersten und zweiten Fehlerkorrektur- und Datenschutzschaltkreis einem anderen der ersten und zweiten Ausgabepuffer bereitgestellt wird, und
(d) ein Speicherpuffer, so konfiguriert, dass er die Wörter mit den in sie eingefügten Fehlercodebits von den ersten und den zweiten Ausgabepuffern empfängt und eine vorher festgesetzte Anzahl von Wörtern in einer vorher festgelegten Art und Weise kombiniert, sodass sie ein Superwort bilden.

18. Das Speichermodul von Anspruch 17, bei welchem der Controller ferner ein Arbitrationsmodul umfasst, so konfiguriert, um zu ermitteln, ob jede Speicheroperation eine LESE-Operation oder eine SCHREIB-Operation ist, das Arbitrationsmodul ferner konfiguriert, um Speicheroperationen den Vorrang zu geben, die LESE-Operationen sind, sofern Speicheroperationen, bei denen es sich um SCHREIB-Operationen handelt, nicht eine vorher festgesetzte Bedingung erfüllen.

19. Das Speichermodul von Anspruch 17, bei welchem das Speichermodul ein Flash-basiertes Speichermodul ist, wobei das Flash-basierte Speichermodul eine oder mehrere der folgenden Typen von Flash-Speichern einschließt: Single-Level Cell Flash-Speicher und Multi-Level Cell Flash-Speicher.

20. Das Speichermodul von Anspruch 17, bei welchem das Speichermodul ein RAM-basiertes Speichermodul ist, wobei das RAM-basierte Modul einen oder mehrere der folgenden Typen von RAM einschließt: DDR RAM, DDR2 RAM und DDR3 RAM.

Es folgen 25 Blatt Zeichnungen

Anhängende Zeichnungen

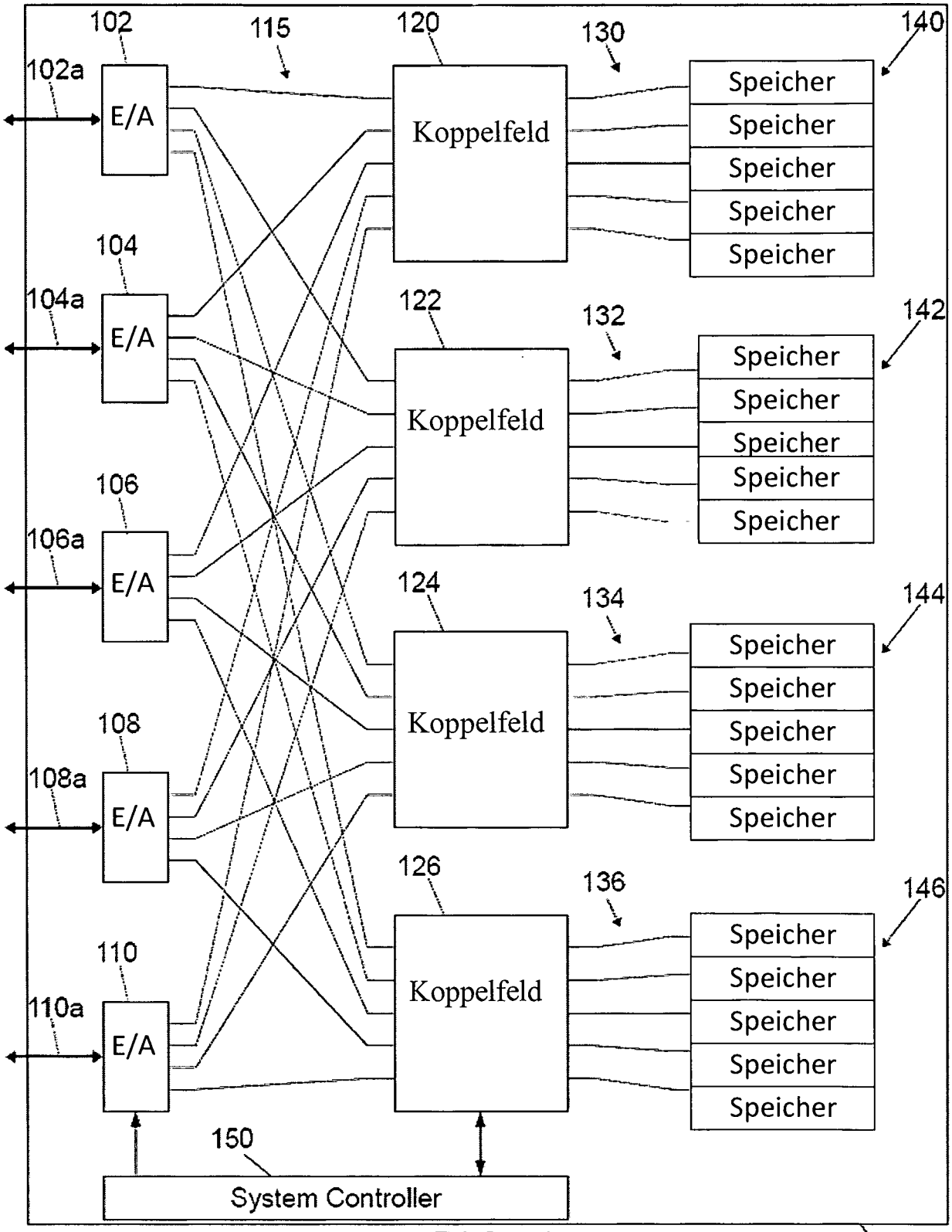


FIG. 1

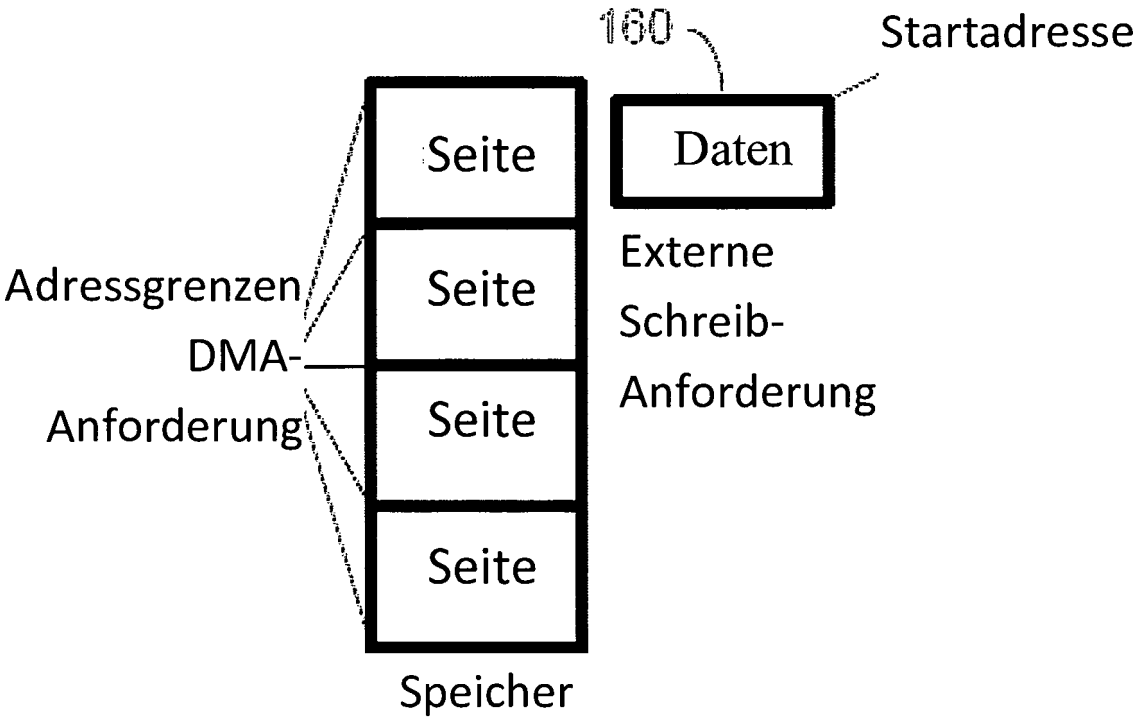

160
Startadresse
Seite
Seite
Seite
Seite
Daten
Externe
Schreib-
Anforderung
Adressgrenzen
DMA-
Anforderung
Speicher

FIG. 1A

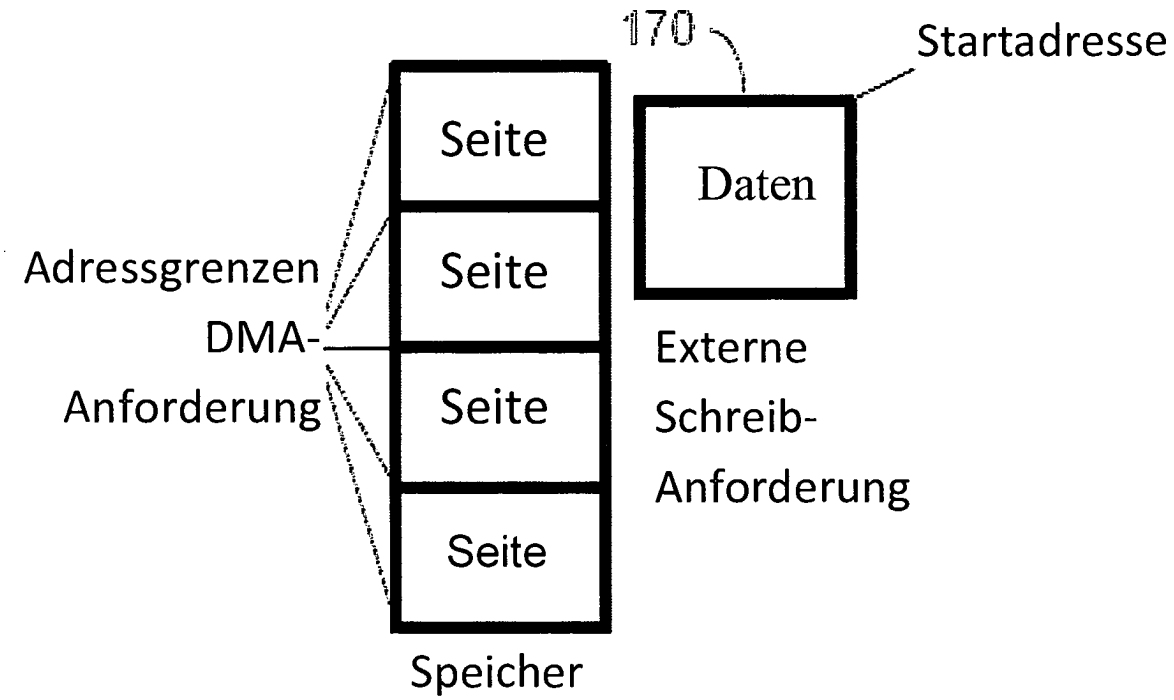

170
Startadresse
Seite
Seite
Seite
Seite
Daten
Adressgrenzen
DMA-
Anforderung
Externe
Schreib-
Anforderung
Speicher

FIG. 1B

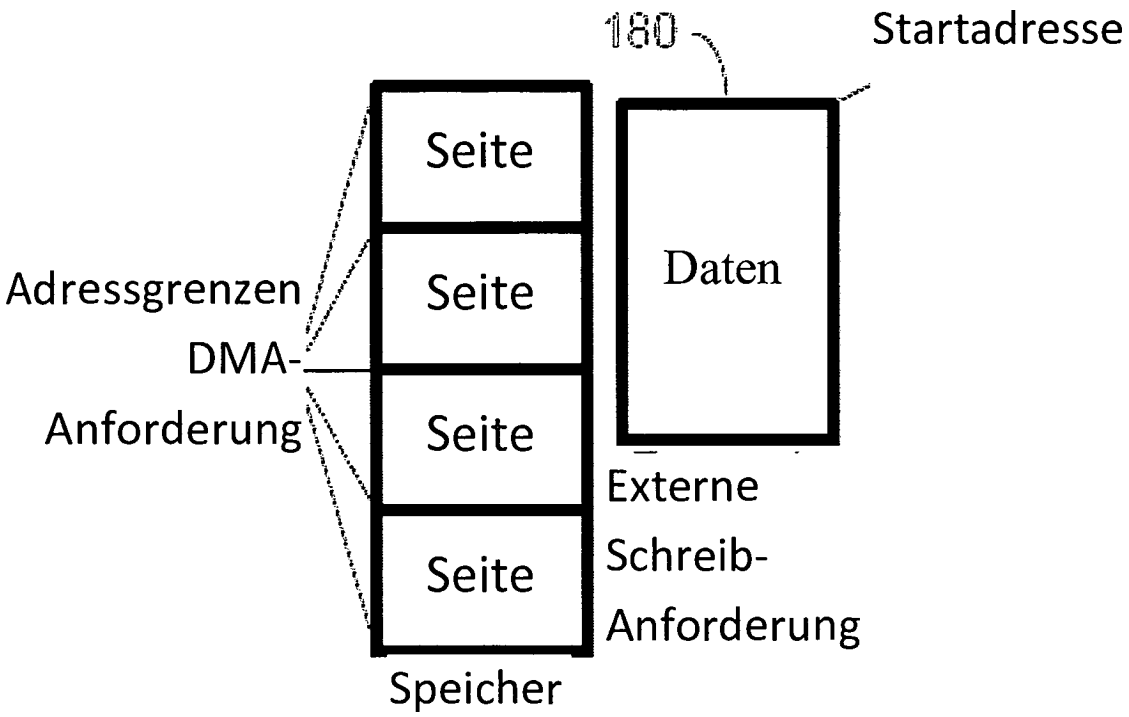

180
Startadresse
Seite
Seite
Seite
Seite
Daten
Adressgrenzen
DMA-
Anforderung
Externe
Schreib-
Anforderung
Speicher

FIG. 1C

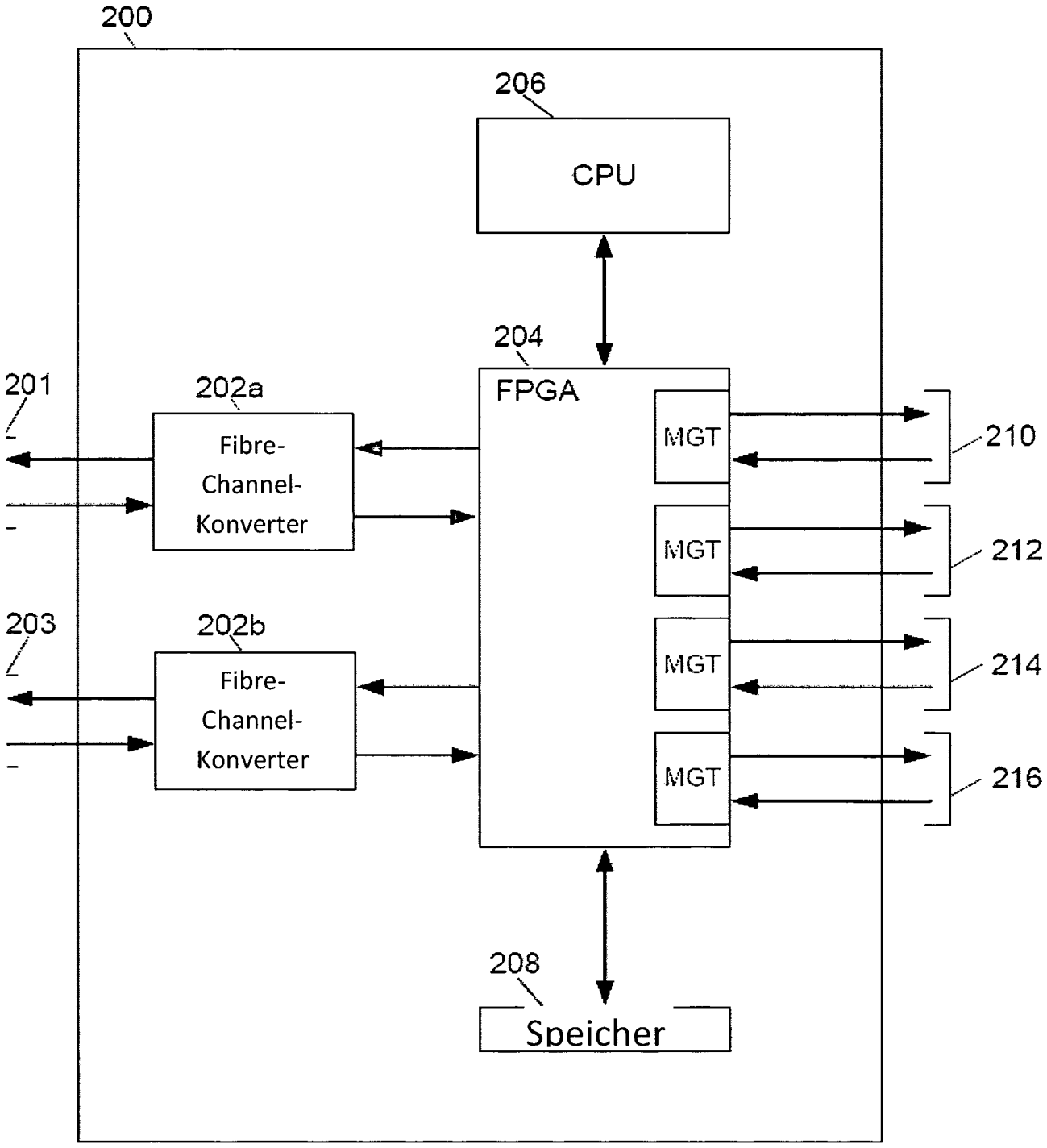

200
206
CPU
204
FPGA
201
202a
Fibre-
Channel-
Konverter
MGT
210
MGT
212
203
202b
Fibre-
Channel-
Konverter
MGT
214
MGT
216
208
Speicher

FIG. 2

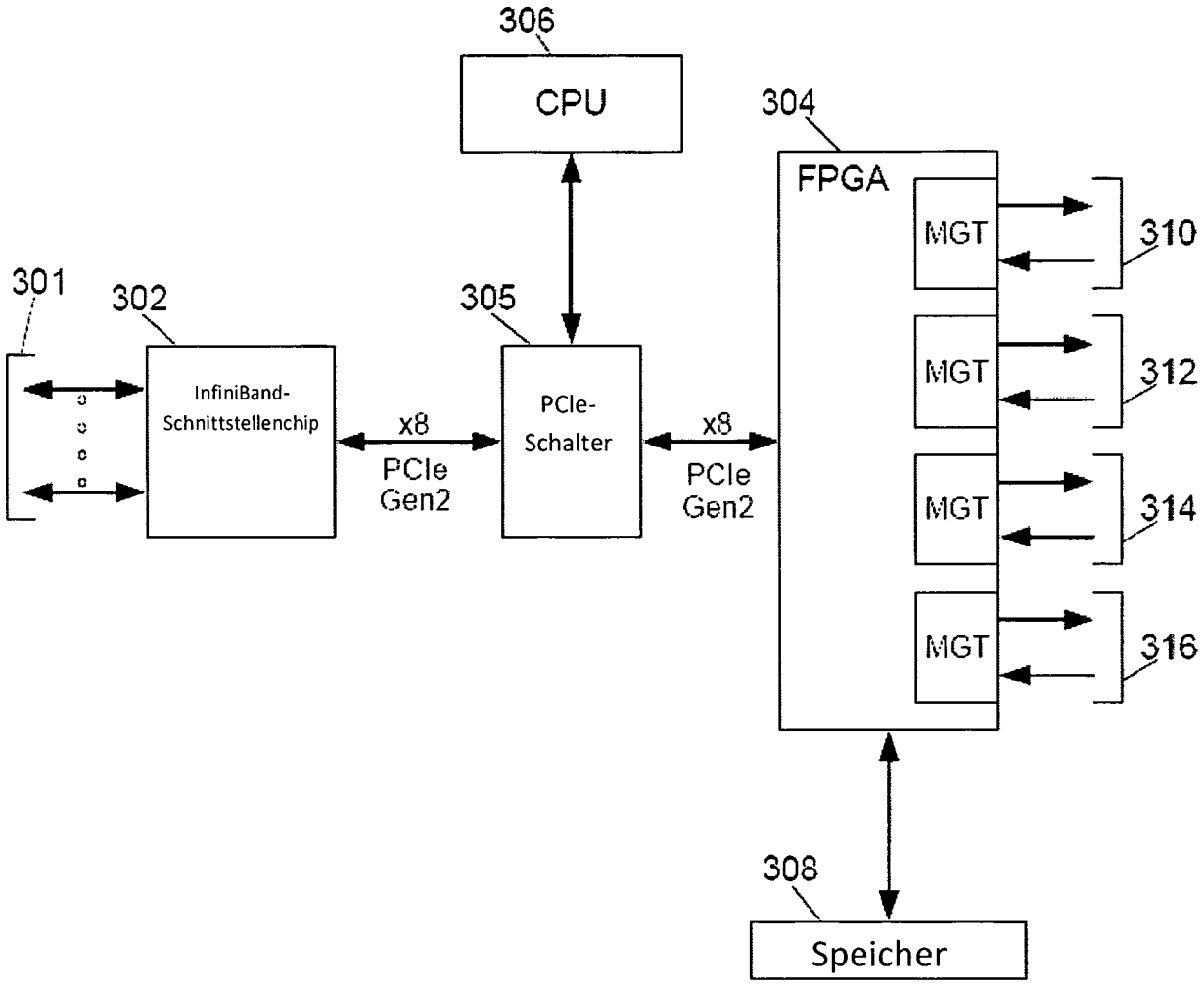

306
CPU
304
FPGA
MGT
310
MGT
312
MGT
314
MGT
316
301
302
305
InfiniBand-
Schnittstellenchip
x8
PCIe
Gen2
PCIe-
Schalter
x8
PCIe
Gen2
308
Speicher

FIG. 3

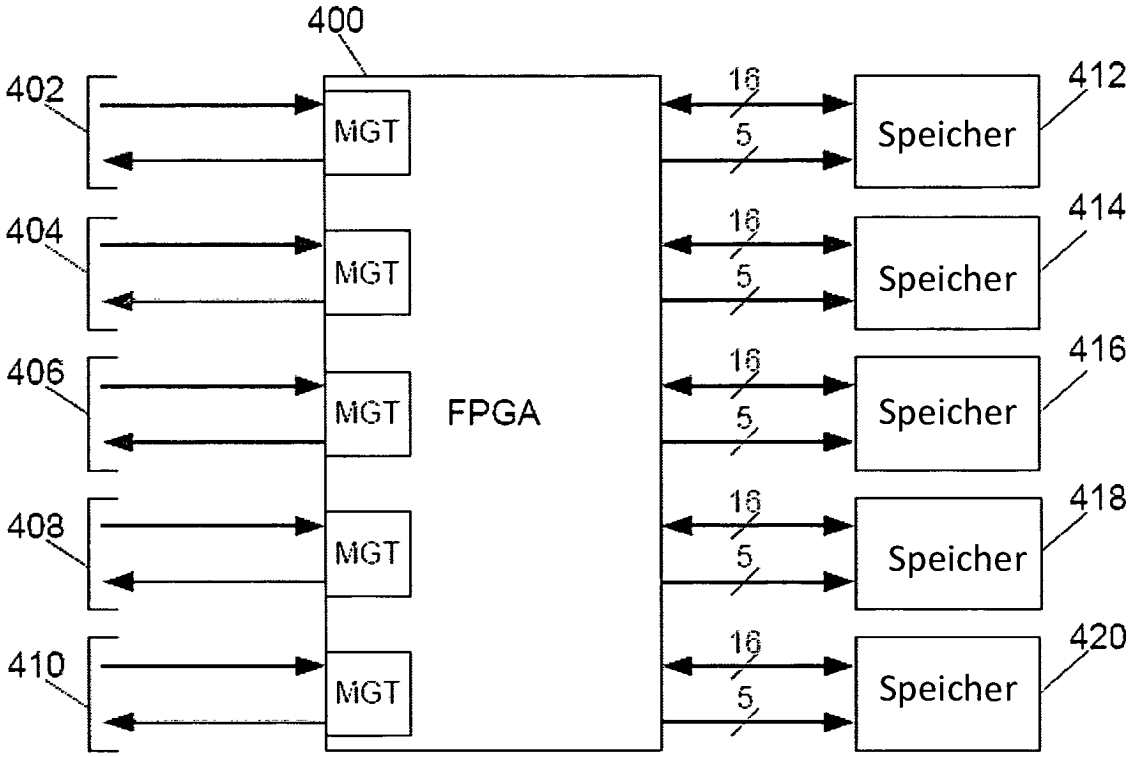

400
402
404
406
408
410
MGT
MGT
MGT
MGT
MGT
FPGA
16
5
16
5
16
5
16
5
16
5
Speicher
Speicher
Speicher
Speicher
Speicher
412
414
416
418
420

FIG. 4

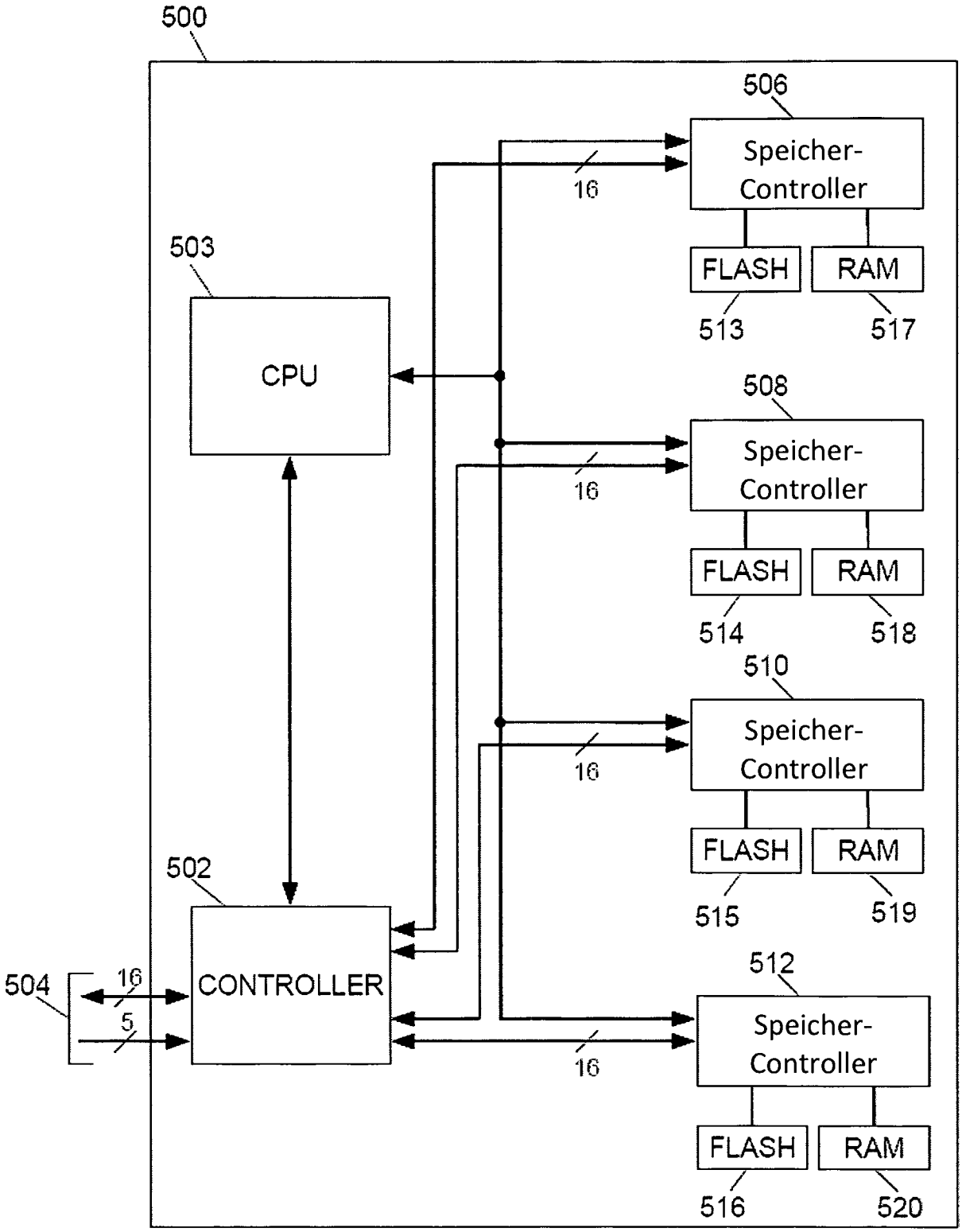

500
506
Speicher-
Controller
16
503
CPU
FLASH
RAM
513
517
508
Speicher-
Controller
16
FLASH
RAM
514
518
510
Speicher-
Controller
16
FLASH
RAM
515
519
502
504
16
5
CONTROLLER
512
Speicher-
Controller
16
FLASH
RAM
516
520

FIG. 5

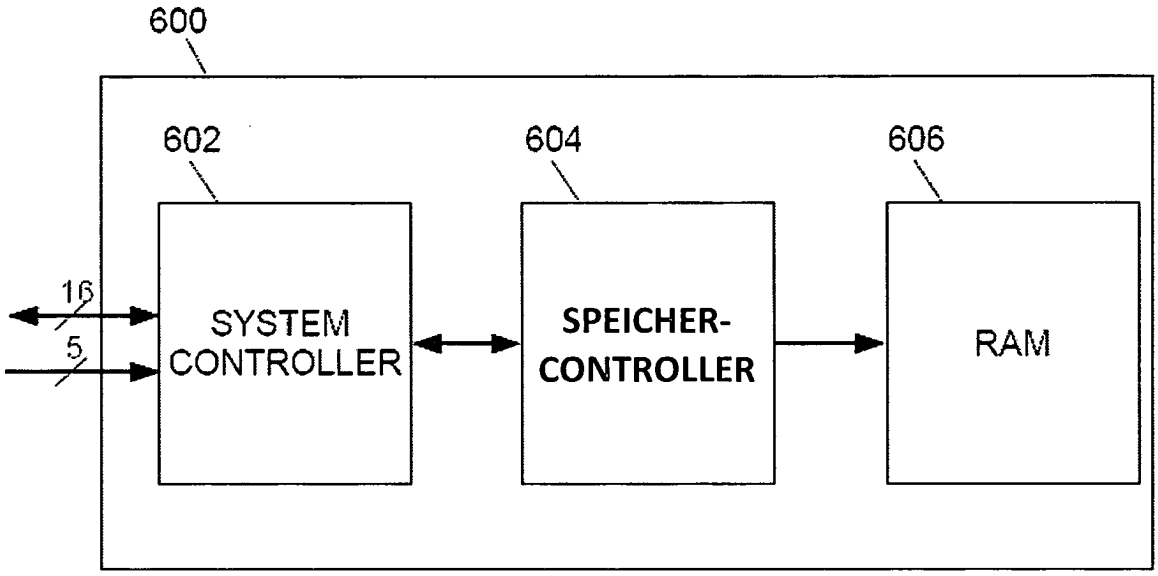

600
602
604
606
16
5
SYSTEM CONTROLLER
SPEICHER-CONTROLLER
RAM

FIG. 6

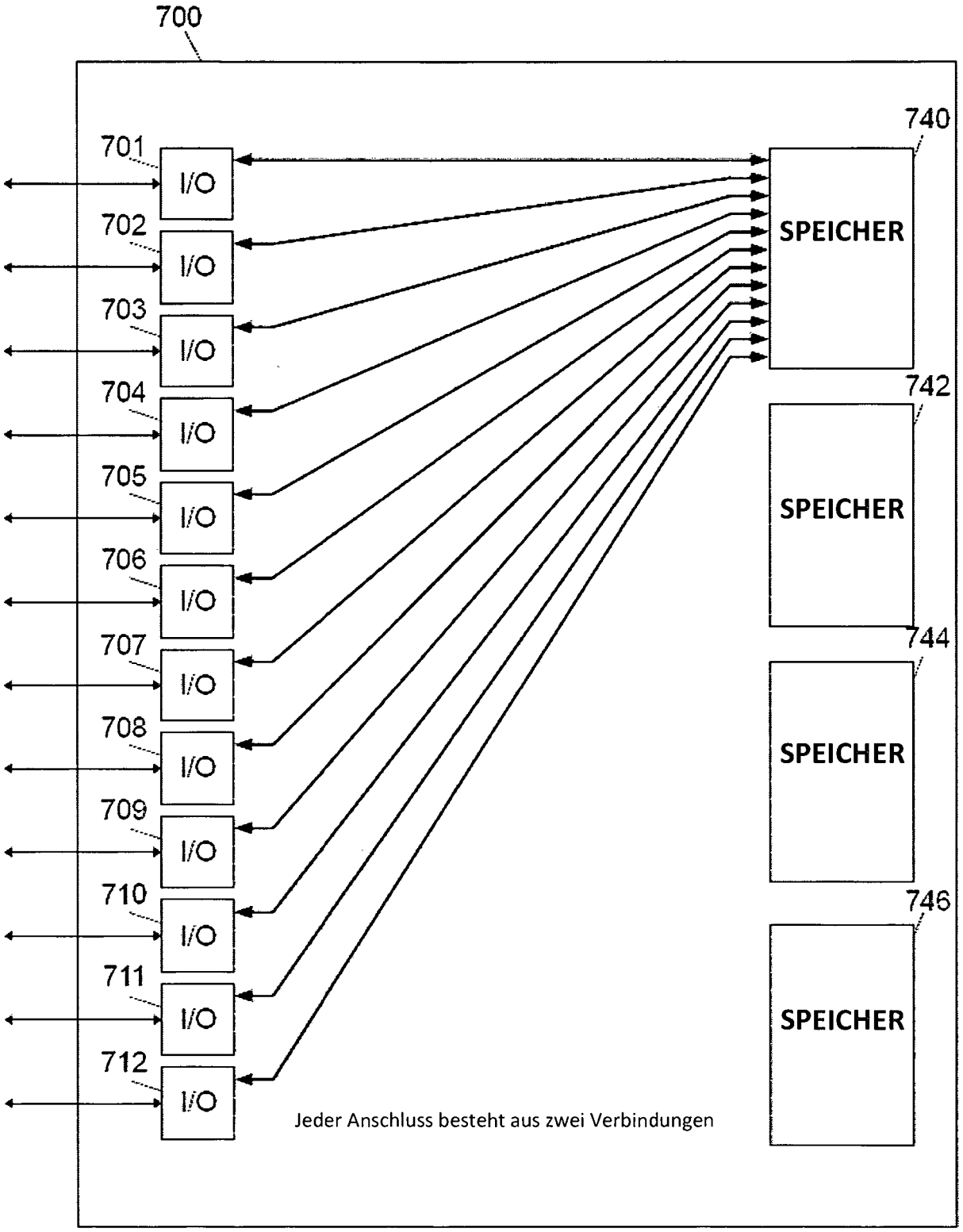

700
701
I/O
702
I/O
703
I/O
704
I/O
705
I/O
706
I/O
707
I/O
708
I/O
709
I/O
710
I/O
711
I/O
712
I/O
740
SPEICHER
742
SPEICHER
744
SPEICHER
746
SPEICHER
Jeder Anschluss besteht aus zwei Verbindungen

FIG. 7

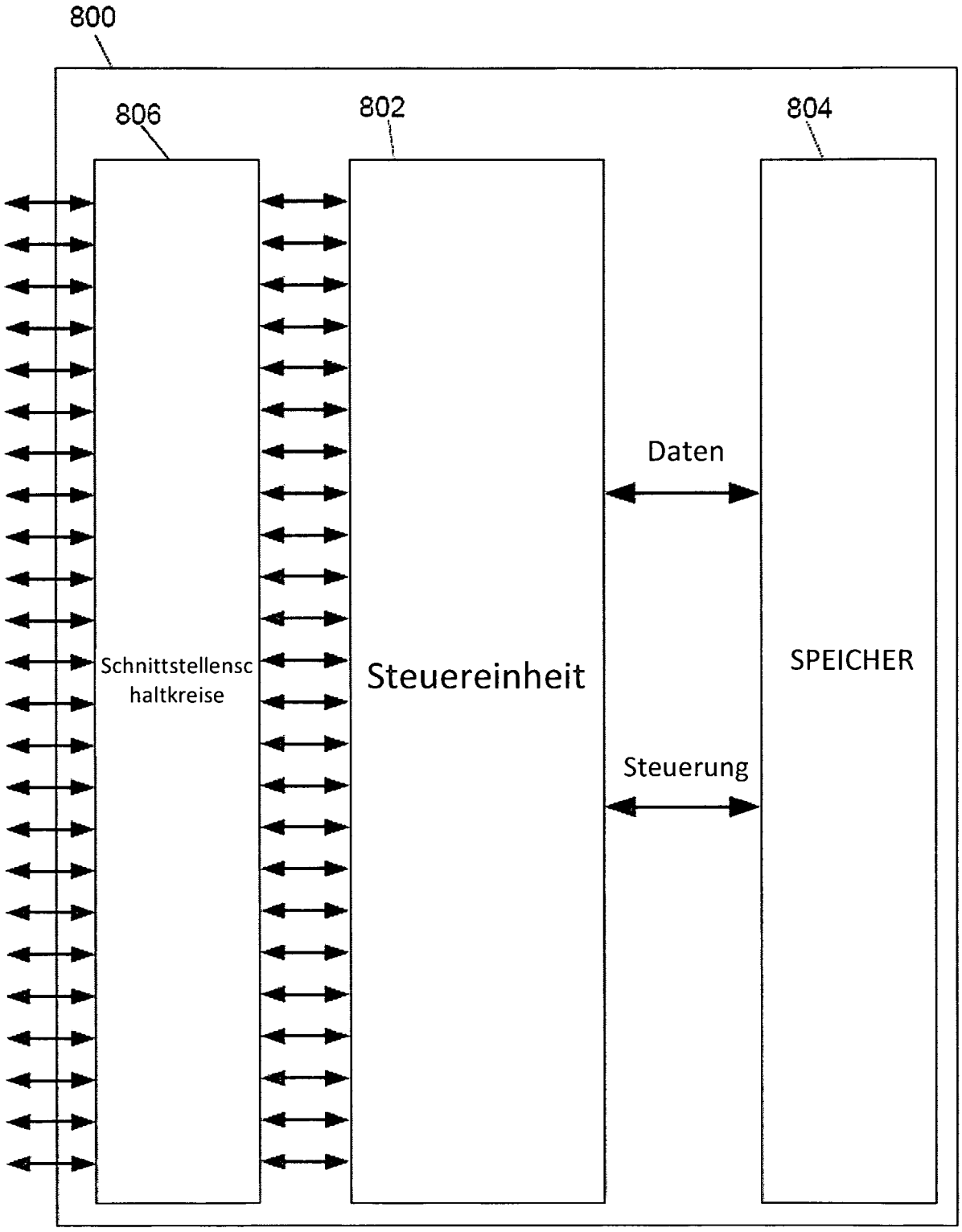

800
806
802
804
Schnittstellensc
haltkreise
Steuereinheit
Daten
Steuerung
SPEICHER

FIG. 8

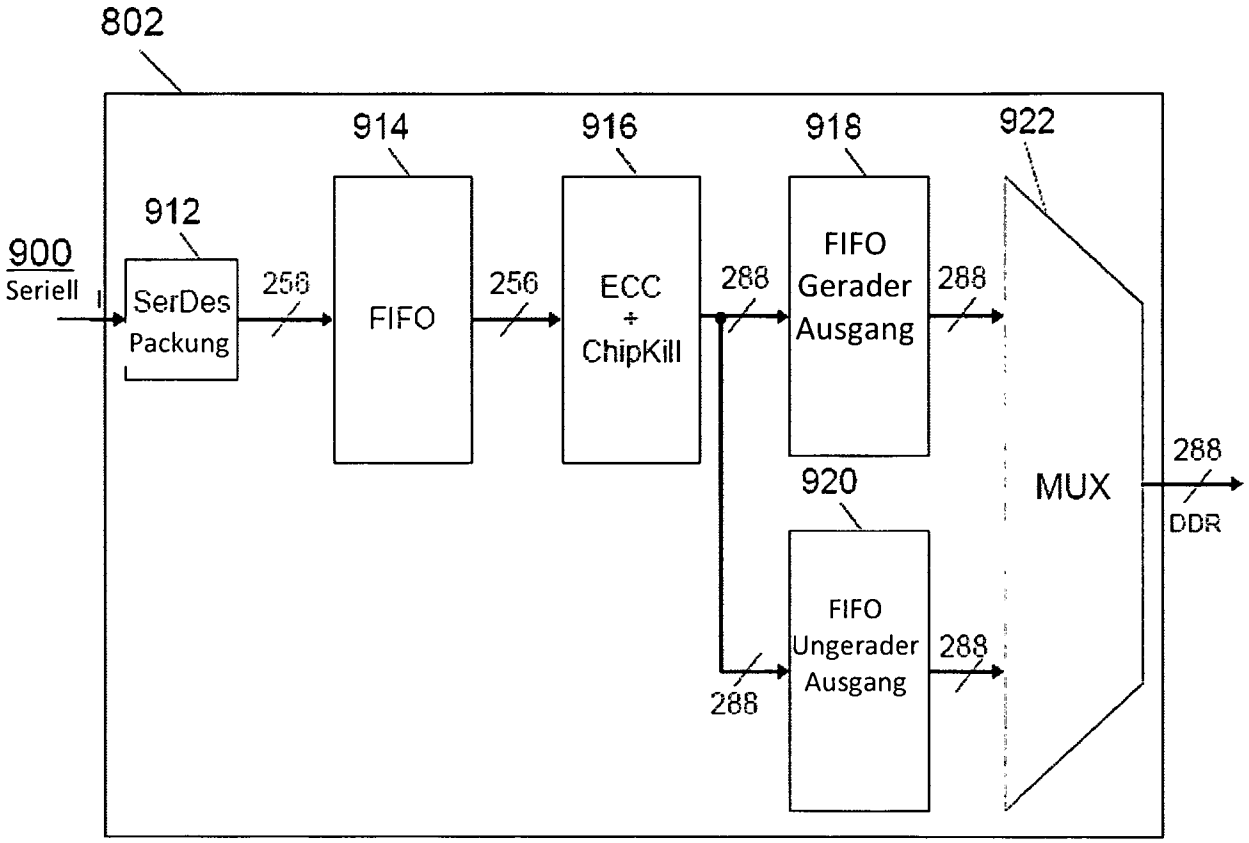

802
914
916
918
922
912
900
Seriell
SerDes
Packung
256
FIFO
256
ECC
÷
ChipKill
288
FIFO
Gerader
Ausgang
288
MUX
288
DDR
920
FIFO
Ungerader
Ausgang
288
288

FIG. 9

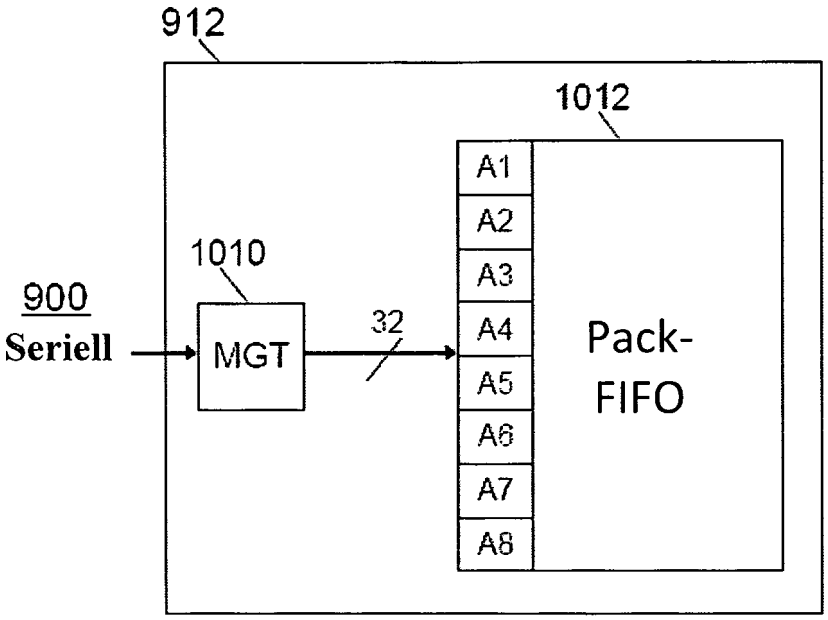

912
1012
A1
A2
A3
A4
A5
A6
A7
A8
1010
900
Seriell
MGT
32
Pack-
FIFO

FIG. 10A

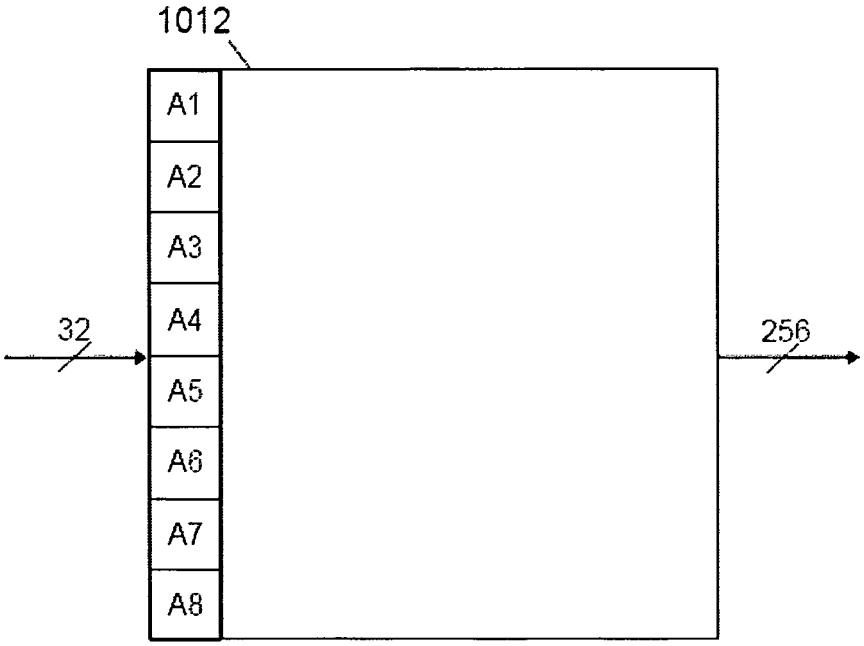


FIG. 10B

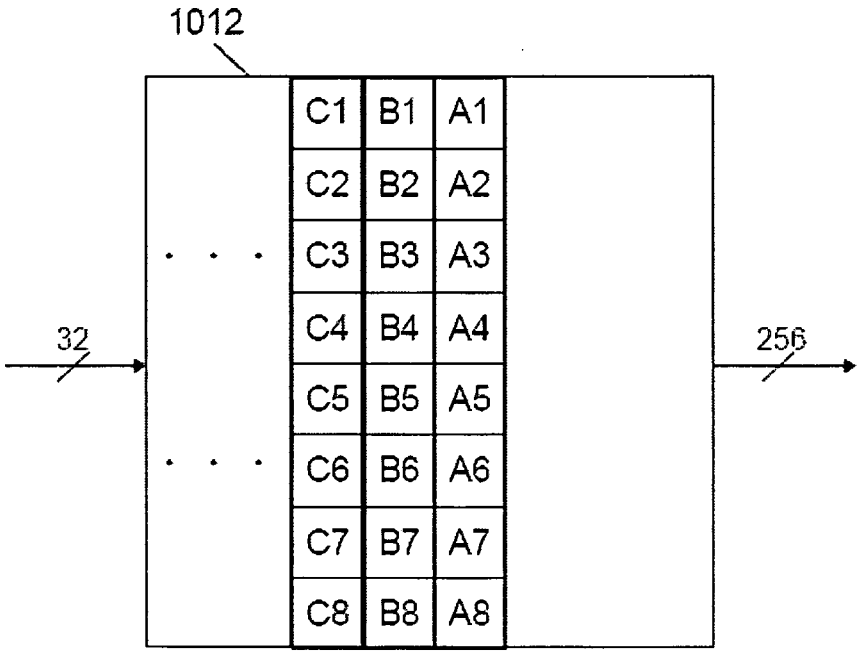

1012
C1 B1 A1
C2 B2 A2
C3 B3 A3
C4 B4 A4
C5 B5 A5
C6 B6 A6
C7 B7 A7
C8 B8 A8
32
256

FIG. 10C

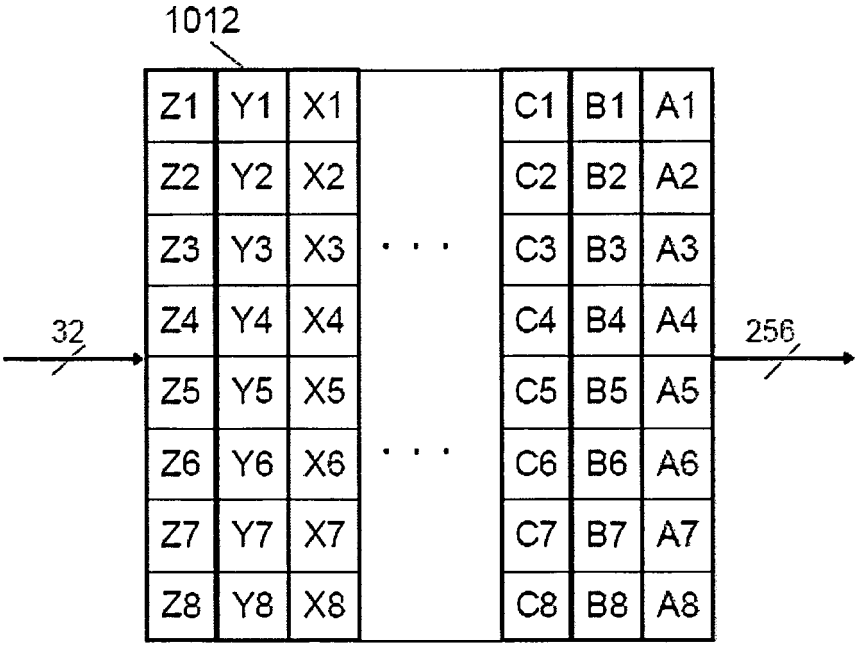

1012
32
256
Z1 Y1 X1 C1 B1 A1
Z2 Y2 X2 C2 B2 A2
Z3 Y3 X3 . . . C3 B3 A3
Z4 Y4 X4 C4 B4 A4
Z5 Y5 X5 C5 B5 A5
Z6 Y6 X6 . . . C6 B6 A6
Z7 Y7 X7 C7 B7 A7
Z8 Y8 X8 C8 B8 A8

FIG. 10D

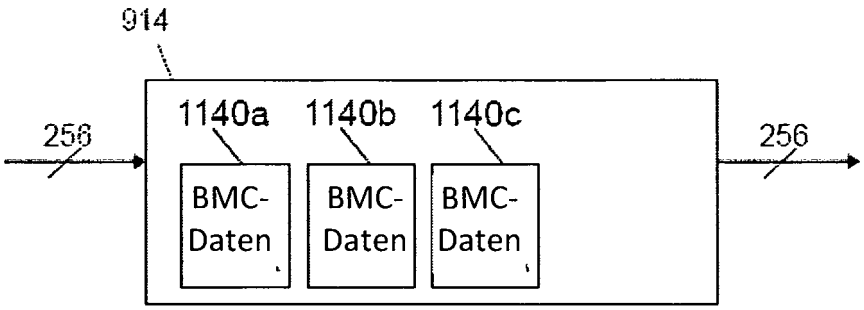


FIG. 11

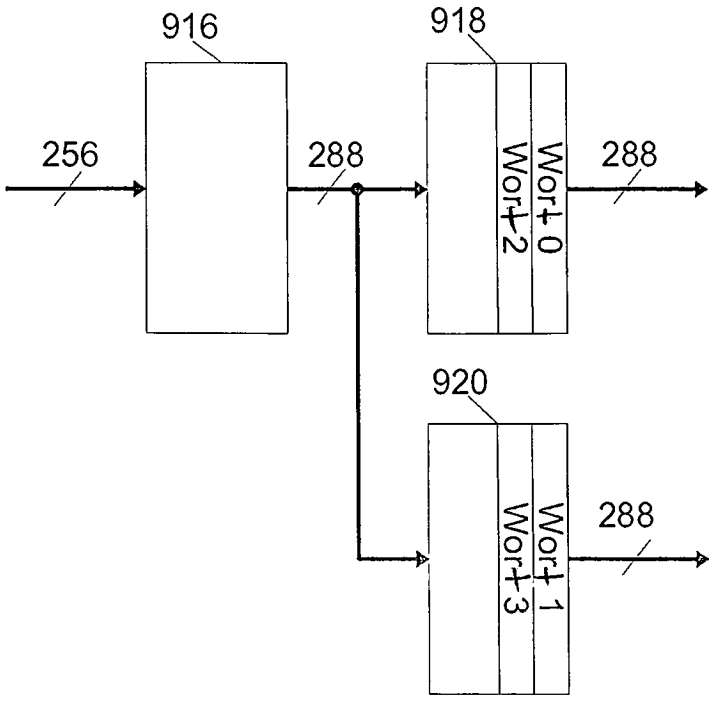


FIG. 12

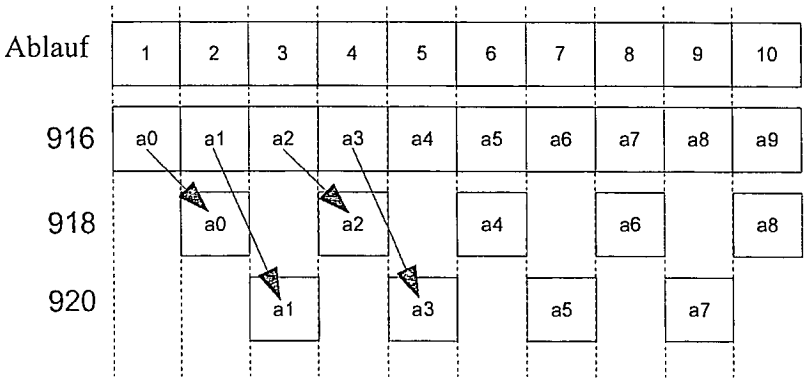


FIG. 12A

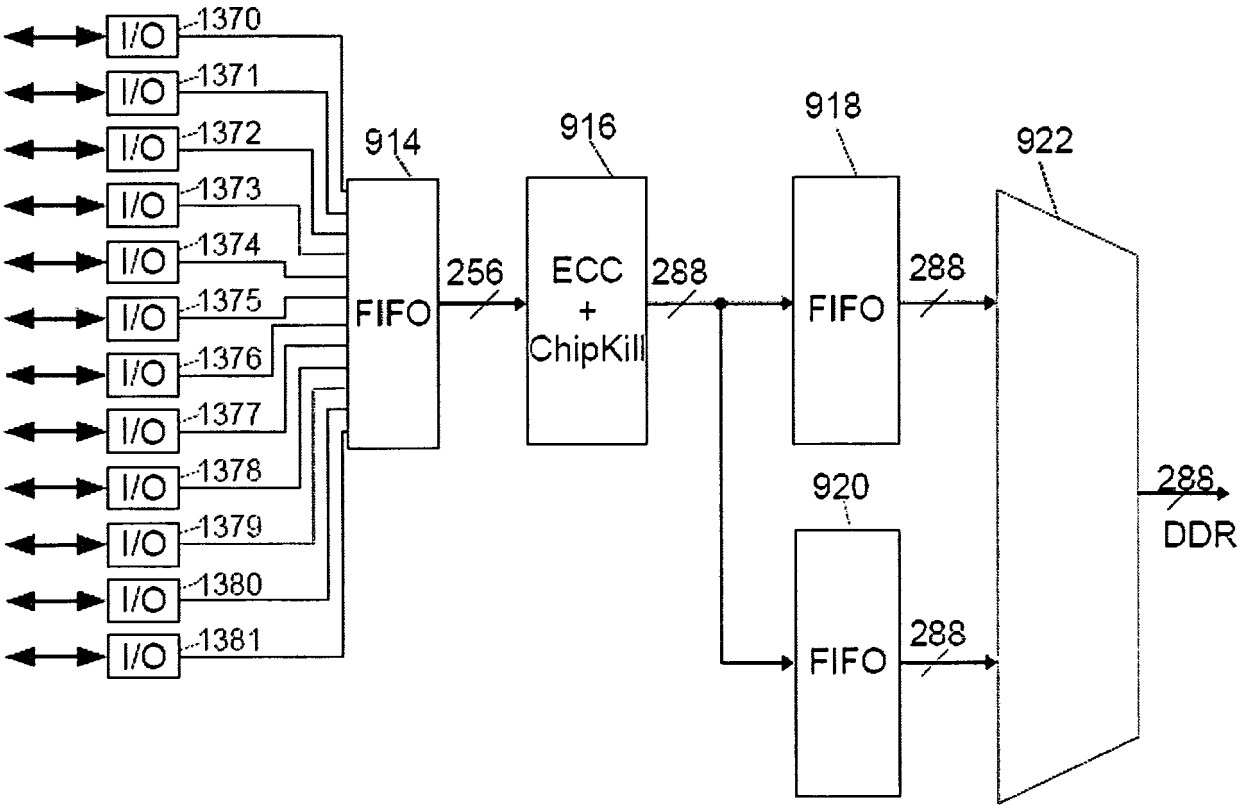

I/O
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
914
FIFO
256
916
ECC
+
ChipKill
288
918
FIFO
288
920
FIFO
288
922
288
DDR

FIG. 13A

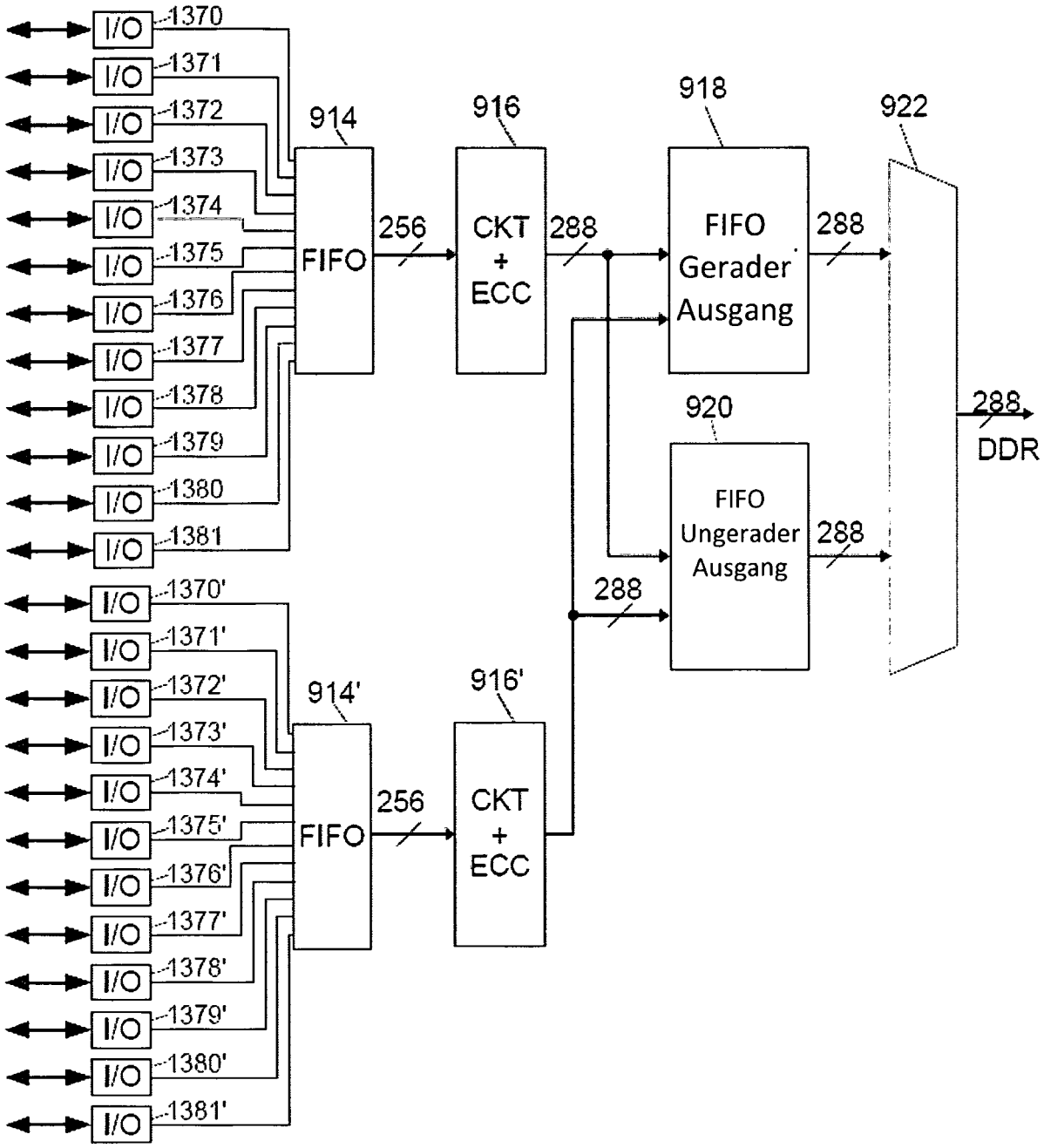

I/O
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
914
FIFO
256
916
CKT
+
ECC
288
918
FIFO
Gerader
Ausgang
288
922
920
FIFO
Ungerader
Ausgang
288
288
DDR
1370'
1371'
1372'
1373'
1374'
1375'
1376'
1377'
1378'
1379'
1380'
1381'
914'
FIFO
256
916'
CKT
+
ECC

FIG. 13B

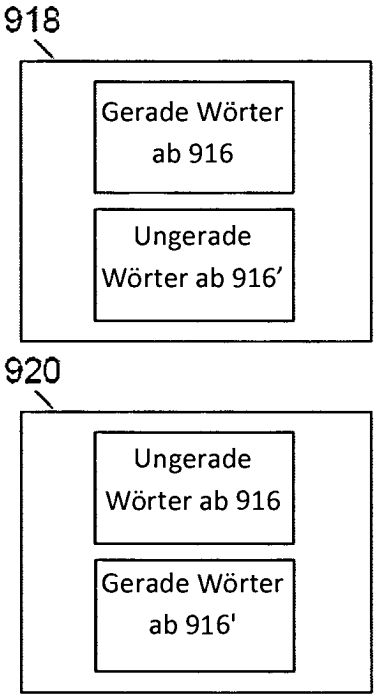


FIG. 14A

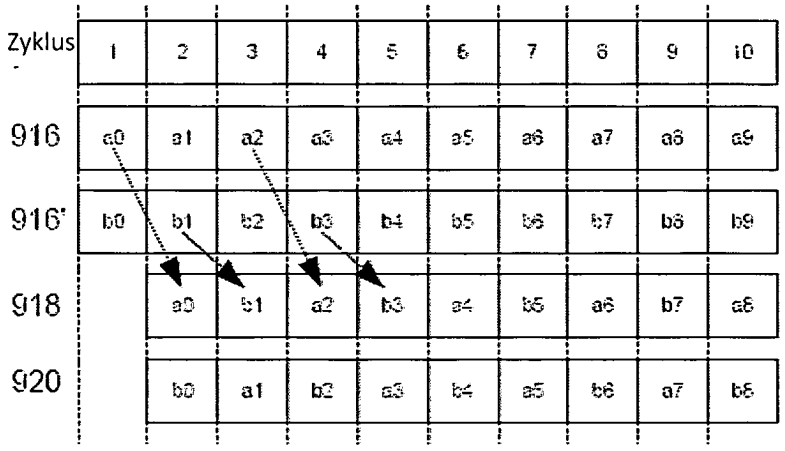


FIG. 14B

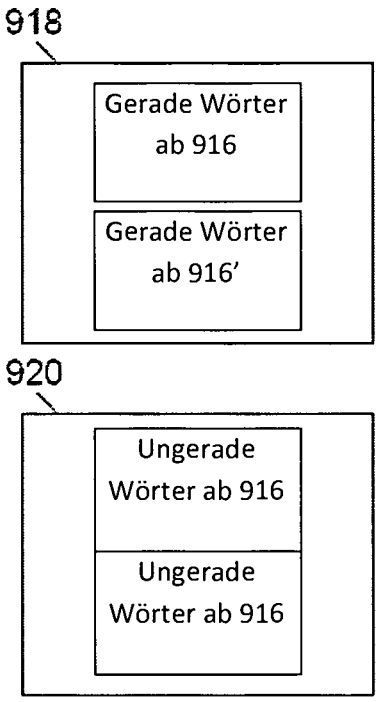


FIG. 14C

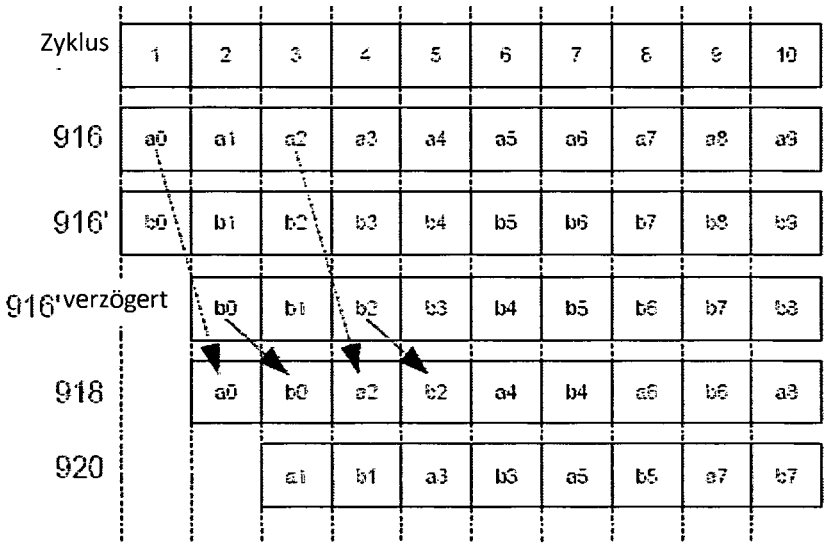


FIG. 14D

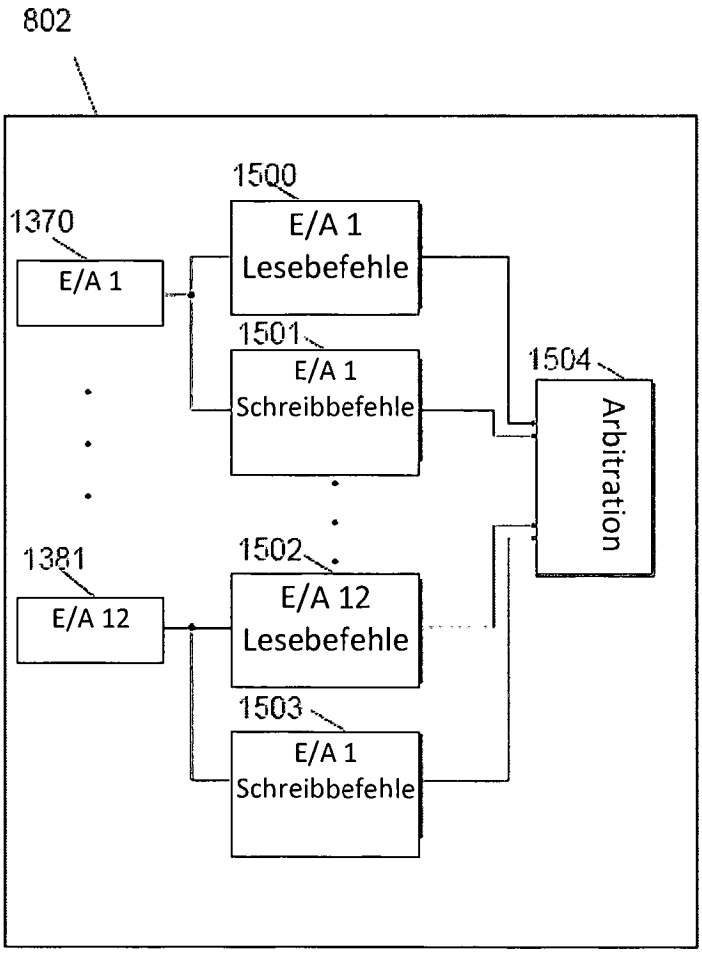

802
1500
E/A 1
Lesebefehle
1370
E/A 1
1501
E/A 1
Schreibbefehle
1504
Arbitration
1381
E/A 12
1502
E/A 12
Lesebefehle
1503
E/A 1
Schreibbefehle

FIG. 15

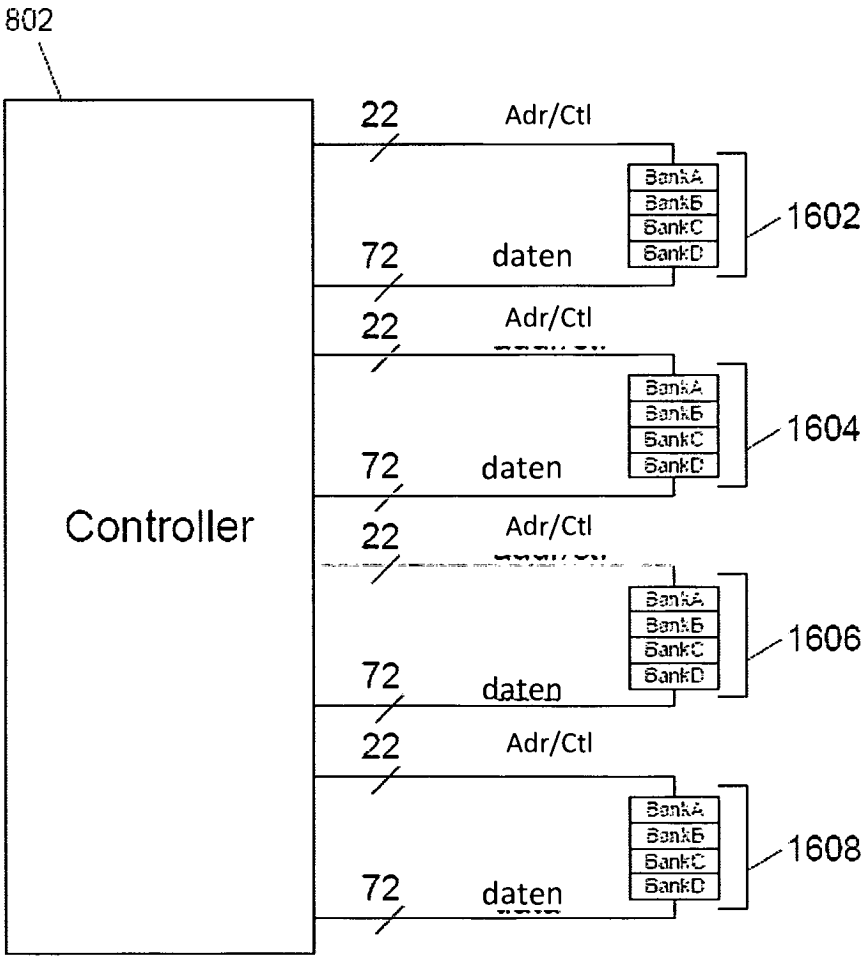

802
Controller
22
Adr/Ctl
72
daten
BankA
BankB
BankC
BankD
1602
1604
1606
1608

FIG. 16

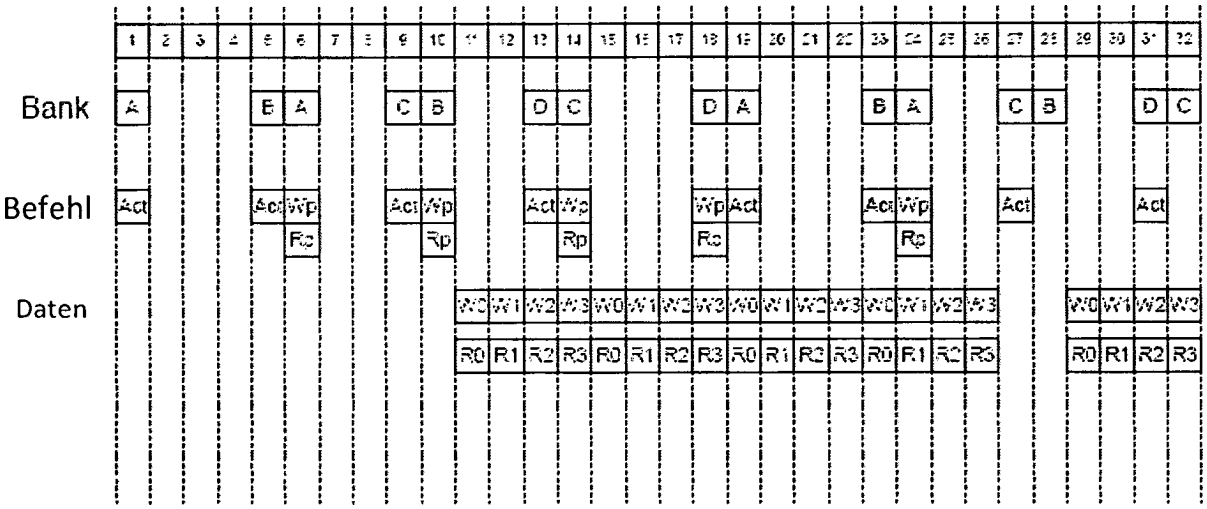


FIG. 17