



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 600 26 836 T2** 2006.09.21

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 188 106 B1**

(51) Int Cl.⁸: **G06F 3/06** (2006.01)

(21) Deutsches Aktenzeichen: **600 26 836.5**

(86) PCT-Aktenzeichen: **PCT/US00/12433**

(96) Europäisches Aktenzeichen: **00 930 436.1**

(87) PCT-Veröffentlichungs-Nr.: **WO 2000/067107**

(86) PCT-Anmeldetag: **05.05.2000**

(87) Veröffentlichungstag
der PCT-Anmeldung: **09.11.2000**

(97) Erstveröffentlichung durch das EPA: **20.03.2002**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **22.03.2006**

(47) Veröffentlichungstag im Patentblatt: **21.09.2006**

(30) Unionspriorität:

| | | |
|-----------------|-------------------|-----------|
| 132712 P | 05.05.1999 | US |
| 547567 | 12.04.2000 | US |
| 548330 | 12.04.2000 | US |

(84) Benannte Vertragsstaaten:

**AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT,
LI, LU, MC, NL, PT, SE**

(73) Patentinhaber:

Qlogic Corp., Aliso Viejo, Calif., US

(72) Erfinder:

**DENNIN, W., William, Lake Forest, CA 92630, US;
WHITE, C., Theodore, Rancho Santa Margarita, CA
92688, US**

(74) Vertreter:

**Dr. Weber, Dipl.-Phys. Seiffert, Dr. Lieke, 65183
Wiesbaden**

(54) Bezeichnung: **ANORDNUNGEN UND VERFAHREN FÜR EINE PLATTENSTEUERUNGSSPEICHER ARCHITEKTUR**

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

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

Beschreibung

Gebiet der Erfindung

[0001] Die vorliegende Erfindung bezieht sich allgemein auf Plattensteuerungen bzw. Festplattensteuerungen und genauer gesagt auf Plattensteuerungen, die eine Speicherarchitektur mit mehreren Anschlüssen haben.

Hintergrund der Erfindung

[0002] Konventionelle Computersysteme umfassen typischerweise mehrere funktionelle Einheiten. Diese funktionellen Einheiten umfassen z. B. eine zentrale Verarbeitungseinheit (CPU), einen Hauptspeicher, Eingabe-/Ausgabeeinrichtungen und magnetische Plattenlaufwerke. In konventionellen Systemen ist der Hauptspeicher mit der CPU eng über einen Systembus oder einen lokalen Speicherbus verbunden. Der Hauptspeicher wird verwendet, um für die CPU zum Zeitpunkt der Ausführung einen schnellen Zugriff auf Daten- oder Programminformation bereitzustellen, die in dem Hauptspeicher gespeichert ist. Typischerweise besteht der Speicher aus Speicherschaltkreisen mit wahlfreiem Zugriff (Random Access Memory – RAM). Ein Computersystem einschließlich der Kombination aus CPU und Hauptspeicher wird oft als ein Host-System bezeichnet.

[0003] Der Hauptspeicher ist typischerweise kleiner als das magnetische Plattenlaufwerk. Der Hauptspeicher ist üblicherweise flüchtig, während das magnetische Festplattenlaufwerk nichtflüchtig ist. Daher wird Information mit Programmierdaten oft auf dem magnetischen Plattenlaufwerk gespeichert und wird in den Hauptspeicher gelesen, wenn es benötigt wird. Im Gegensatz zu dem Hauptspeicher, der eng mit der CPU verbunden ist, ist das magnetische Plattenlaufwerk üblicherweise weiter entfernt bzw. lose mit der CPU verbunden. Da das magnetische Plattenlaufwerk nur entfernt bzw. indirekt mit der CPU verbunden ist, ist der Zugriff auf das magnetische Plattenlaufwerk typischerweise langsamer und komplizierter als der Zugriff auf den Hauptspeicher. Eine Plattensteuerung verbindet typischerweise das Host-System mit dem magnetischen Plattenlaufwerk und behandelt die komplexen Details in der Verbindung bzw. Schnittstellenbildung zwischen dem magnetischen Plattenlaufwerk und dem Host-System. Kommunikation zwischen dem Host-System und der Plattensteuerung wird üblicherweise bereitgestellt unter Verwendung einer aus einer Vielfalt standardmäßiger I/O(Eingabe/Ausgabe)-Busschnittstellen.

[0004] Der Aufbau des magnetischen Festplattenlaufwerks besteht üblicherweise aus einer oder mehreren magnetischen Platten. Jede Platte hat typischerweise eine Anzahl konzentrischer Ringe oder Spuren, auf welchen Daten gespeichert werden. Die Spuren selbst sind weiter aufgeteilt in Sektoren, welche die kleinste zugriffsfähige Dateneinheit repräsentieren. Auf einen Sektor wird zugegriffen durch Anordnen eines Magnetkopfes oberhalb der entsprechenden Spur. Der erste Sektor einer Spur wird typischerweise durch einen Indeximpuls identifiziert. Der Beginn jedes weiteren Spursektors wird identifiziert mit einem Sektorimpuls. Das Laufwerk wartet, bis der gewünschte Sektor sich unter den Kopf gedreht hat und fährt dann fort mit dem Lesen oder Schreiben der Daten. Auf die Daten wird seriell zugegriffen, und zwar jeweils bitweise. Üblicherweise hat jede Platte in dem Plattenlaufwerk ihren eigenen Lese-/Schreibkopf.

[0005] Das Plattenlaufwerk ist mit der Plattensteuerung verbunden. Die Plattensteuerung führt zahlreiche Funktionen aus, wie z. B. das Umwandeln digitaler Daten in analoge Kopfsignale, das Umwandeln analoger Kopfsignale in digitale Daten, die Plattenformatierung, Fehlerüberprüfung, Zuordnung logischer zu physikalischen Adressen und Datenpufferung. Beispielsweise formatiert das Plattenlaufwerk typischerweise die Daten von dem Laufwerk. Die Daten von dem Laufwerk sind seriell angeordnet und die Plattensteuerung wandelt die seriellen Daten in eine parallele Anordnung um.

[0006] Die Datenpuffertfunktion wird verwendet beim Übertragen von Daten zwischen dem Host und dem Massenspeicher. Die Datenpufferung ist erforderlich, da die Geschwindigkeit, mit welcher das Plattenlaufwerk Daten von dem Host zuführen oder aufnehmen kann, verschieden ist von der Geschwindigkeit, mit welcher der Host in entsprechender Weise die Daten lesen oder zuführen kann. Daher enthält die Plattensteuerung üblicherweise einen Pufferspeicher, der temporär Daten speichert, die gelesen oder auf die Festplatte geschrieben werden sollen, um die Daten mit der Geschwindigkeit des I/O-Busses zu synchronisieren, mit welcher es verbunden ist. Der Puffer entkoppelt also die Rate bzw. Geschwindigkeit, mit welchem Daten zwischen dem Plattenlaufwerk und dem Host ausgetauscht werden und der Geschwindigkeit, mit welcher Daten geschrieben und von dem Laufwerk gelesen werden. Die Pufferfunktion ist insbesondere deshalb notwendig, weil jeder Sektor als Ganzes gelesen oder beschrieben wird.

[0007] Zusätzlich zur Gewährung des Zugriffs sowohl auf den I/O-Bus als auch auf das Plattenlaufwerk wird auf den Pufferspeicher oftmals durch einen lokalen Prozessor auf der Plattensteuerung zugegriffen. Auf dem Pufferspeicher der Plattensteuerung kann also durch viele funktionelle Teile der Plattensteuerung und durch den Host zugegriffen werden. In konventionellen Systemen ist die Konkurrenz um den Zugriff auf den Pufferspeicher oftmals ein Engpaß, der in starkem Maße die Datenübertragungsgeschwindigkeiten zwischen dem Pufferspeicher und den funktionellen Einheiten ebenso wie zu dem Plattenlaufwerk und dem Host beschränkt.

[0008] Ein weiterer Engpaß für die Leistungsfähigkeit, die man in konventionellen Plattensteuerungen findet, liegt in dem lokalen Prozessor. Der lokale Prozessor verwaltet typischerweise einen I/O-Bus-Schnittstellenschaltkreis, den Pufferspeicher, den Plattenformatierer und betreibt auch die Verwaltung der Plattensteuerung als Ganzes. Weiterhin erfordern konventionelle Systeme, daß der lokale Prozessor Datenübertragungen durch den Pufferspeicher auf einem niedrigen Niveau verwaltet. Typischerweise sind Vielzweckprozessoren (Standardprozessoren) nicht richtig angepaßt, um derartige anwendungsspezifische Funktionen in effizienter Weise zu handhaben bzw. auszuführen.

[0009] Zusätzlich nutzen konventionelle Plattensteuerungen ihre Prozessoren nicht vollständig aus. Wenn beispielsweise während der Übertragung von Daten von der Platte auf den I/O-Bus einer oder mehrere Prozessoren der Plattensteuerung verwendet werden, überschreitet die Übertragung die Fähigkeit der Festplatte, Daten zuzuführen und die Prozessoren verschwenden Zeit, indem sie pausieren, bis die Festplatte nachgekommen ist.

[0010] Die WO 9814861 beschreibt ein Verfahren zum Betreiben mehrfacher Kontexte innerhalb einer Zieleinrichtung, die das Speichern jedes Befehls innerhalb seines eigenen Kontextes innerhalb der Steuerung der Zieleinrichtung umfaßt (beispielsweise eine eingebettete Steuerung in dem SCSI-Plattenlaufwerk), um alle eindeutigen Parameter, die für die Verarbeitung jedes Befehls erforderlich sind, zu halten. Die Verwendung von mehreren Kontexten ermöglicht das Verarbeiten mehrerer Befehle innerhalb der Zieleinrichtung, um die Bandbreite eines Host-Kanals besser auszunutzen und die Latenzzeit beim Bedienen aufgereihter Befehle oder Initiatordienste zu reduzieren.

Zusammenfassung der Erfindung

[0011] Verschiedene entsprechende Aspekte und Merkmale der Erfindung sind in den Ansprüchen dargelegt. Ausführungsformen der vorliegenden Erfindung beziehen sich allgemein auf effiziente und flexible Verfahren und Systeme zum Puffern und Zugreifen auf Daten, die in Massenspeichereinrichtungen geschrieben oder von diesen gelesen werden.

[0012] In einer Ausführungsform enthält die Plattensteuerung einen Datenpuffer, der verwendet wird, um Daten, die zwischen einer Massenspeichereinrichtung und einem I/O-Bus übertragen werden, zu puffern. In einer anderen Ausführungsform enthält der Puffer einen Speicher mit mehreren Anschlüssen. Der Speicher mit Mehrfachanschlüssen ist mit einer Mehrzahl von Kanälen verbunden, wie beispielsweise einem Plattenkanal und einem I/O-Kanal, ebenso mit einem oder mehreren Prozessoren.

[0013] In einer Ausführungsform kann der Speicher mit mehreren Anschlüssen verwendet werden sowohl als Speicher mit wahlfreiem Zugriff (RAM) und als First-In-First-Out (FIFO)-Speicher. Wenn er als FIFO-Speicher verwendet wird, kann der Speicher mit mehreren Anschlüssen verwendet werden, um Daten zwischen dem Plattenkanal und dem I/O-Kanal zu puffern. Der Speicher mit mehreren Anschlüssen kann verwendet werden für die Kontextumschaltung, wobei Registerdaten von einem oder mehreren Prozessoren in den Speicher oder aus diesem übergeben werden. Das wirksame Übergeben bzw. Austauschen von Registern, welches durch eine Ausführungsform der vorliegenden Erfindung bereitgestellt wird, ermöglicht in vorteilhafter Weise, daß der Prozessor schnell zwischen Aufgaben umschaltet. Dies ist sehr zweckmäßig, wenn eine erste Aufgabe sich in einem Wartezustand befindet, da anstelle des Vergeudens von Prozessorzeit durch Pausieren, während er darauf wartet, daß der Wartezustand endet, kann der Prozessor schnell zu einer anderen Aufgabe umschalten und kann dann, nachdem der Wartezustand für die erste Aufgabe beendet ist, zu der ersten Aufgabe zurückschalten.

[0014] Zusätzlich ermöglicht in einer Ausführungsform die Eigenschaft des wahlfreien Zugriffs, daß ausgewählte Daten, die in einem Speicher mit mehreren Anschlüssen gespeichert sind, schnell wiedergewonnen werden. Wenn beispielsweise der Speicher verwendet wird, um einen Frame von einem Faserkanal und zugehörige CRC-Daten zu speichern, können die CRC-Daten sofort wiedergewonnen werden, ohne daß man den gesamten Frame auslesen muß.

[0015] In einer Ausführungsform ist der Speicher mit mehreren Anschlüssen ein DMA-Speicher (Direct Memory Access – Direktzugriffsspeicher). In einer beispielhaften Ausführungsform umfassen die Anschlüsse des Speichers mit mehreren Anschlüssen einen Anschluß für wahlfreien Zugriff, einen FIFO-Zugriffsanschluß, einen Registerzugriffsanschluß und/oder einen DMA-Anschluß einer Puffersteuerung. Der Anschluß für wahlfreien Zugriff ist mit einem Schnittstellenbus eines Mikroprozessors verbunden, der seinerseits mit einem oder mehreren Prozessoren verbunden ist, wie z. B. einem Mikroprozessor und einem Mikrocontroller, ebenso wie der FIFO-Zugriffsanschluß. Der Registerzugriffsanschluß kann auch mit einem oder mehreren Prozessoren, wie z. B. einem Mikrocontroller verbunden sein. Der DMA-Anschluß der Puffersteuerung ist mit einem CRC (zyklischer Redundanzcode)-Überprüfer und mit einem Pufferspeicher verbunden.

[0016] In einer Ausführungsform enthält der Speicher mit mehreren Anschlüssen 64 Bytes und kann einen vollständigen Befehl eines Faserkanalprotokolls erhalten. Weiterhin kann eine Zustandsmaschine, die zu dem Speicher mit mehreren Anschlüssen gehört, mehrere (verschiedene) Befehle ausführen. In einer anderen Ausführungsform sind einer oder mehrere der Befehle so ausgestaltet, daß sie in effizienter Weise I/O-Pakete oder Frames verwalten. In noch einer anderen Ausführungsform umfassen die Befehle der Zustandsmaschine einen oder mehrere der folgenden:

einen Heranholbefehl für ein aktuelles FCP (Faserkanalprotokoll), einen Aktualisierungsbefehl für den aktuellen FCP-Zeiger, einen FIFO-Ladebefehl, einen FIFO-Entladebefehl, einen Befehl zum Laden des FIFO mit CRC, einen Befehl zum Entladen des FIFO mit CRC, einen Befehl zum Schreiben in den Pufferspeicher und einen Befehl zum Lesen aus dem Pufferspeicher. In einer Ausführungsform wird ein Befehl bereitgestellt, der die Übertragung eines kompletten Faserkanalbefehls in den FIFO ohne weiteren Eingriff durch den Mikroprozessor oder den Mikrocontroller bewirkt. Eine Ausführungsform stellt einen direkten Frame-Zugriff auf Faserkanalframes bereit. In einer anderen Ausführungsform können die Befehle angehalten und/oder unterbrochen werden.

Kurze Beschreibung der Figuren

[0017] [Fig. 1](#) ist eine Darstellung einer Umgebung, welche eine Ausführungsform der vorliegenden Erfindung verwendet.

[0018] [Fig. 2](#) ist eine Darstellung der Anschlüsse einer Ausführungsform einer Puffersteuerung.

[0019] [Fig. 3](#) ist eine genauere Darstellung der Puffersteuerung nach [Fig. 2](#).

[0020] [Fig. 4](#) veranschaulicht eine Ausführungsform einer Speichereinheit.

[0021] [Fig. 5A–D](#) veranschaulichen eine Ausführungsform einer Zustandsmaschine, die mit der Puffersteuerung verwendet wird.

[0022] [Fig. 6](#) ist eine Darstellung einer Ausführungsform der Datenübertragung.

[0023] [Fig. 7](#) ist eine Darstellung eines Faserkanalsteuerblocks.

[0024] [Fig. 8](#) ist eine Darstellung einer ersten beispielhaften Datenübertragung.

[0025] [Fig. 9](#) ist eine Darstellung einer zweiten beispielhaften Datenübertragung.

[0026] [Fig. 10](#) ist eine Darstellung einer dritten beispielhaften Datenübertragung.

[0027] [Fig. 11](#) ist eine Darstellung einer vierten beispielhaften Datenübertragung.

[0028] [Fig. 12](#) ist eine Darstellung einer fünften beispielhaften Datenübertragung.

[0029] [Fig. 13](#) ist eine Darstellung einer sechsten beispielhaften Datenübertragung.

[0030] [Fig. 14](#) ist eine Darstellung einer siebten beispielhaften Datenübertragung.

[0031] [Fig. 15](#) ist eine Darstellung einer achten beispielhaften Datenübertragung.

[0032] Die vorliegende Erfindung bezieht sich allgemein auf effiziente und flexible Verfahren und ein System zum Puffern von Daten, die in eine Massenspeichereinrichtung geschrieben oder aus dieser gelesen werden.

[0033] Wie in [Fig. 1](#) dargestellt, wird in einer Ausführungsform eines elektronischen Systems **100** eine Plattensteuerung **102** verwendet, um eine Schnittstelle von einem Host-System **104** zu einer Massenspeichereinrichtung **106** zu bilden, wie z. B. an einem magnetischen Plattenlaufwerk oder einem optischen Plattenlaufwerk, und zwar über einen Plattenanschluß **136**. Die dargestellte Plattensteuerung **102** enthält einen Mikroprozessor **108**, eine eingebettete Steuerung **110**, einen Pufferspeicher **112** und eine externe DMA (Direktzugriffsspeicher)-Einrichtung **114**. Die Plattensteuerung ist mit dem Host-System **104** über einen I/O-Bus **116**, wie z. B. einen Faserkanalbus, mit doppelten Schleifen, verbunden. Die eingebettete Steuerung **110** umfaßt eine Mikroprozessorschnittstelle **118**, die mit dem Mikroprozessor **108** verbunden ist. Ein Mikroprozessorbuss **130** verbindet die Mikroprozessorschnittstellenlogik **118** mit der externen DMA-Schnittstelle **134**, einem Mikrocontroller **120**, einer Puffersteuerung **122** und einer Faserkanalschnittstelle **124**, die in der eingebetteten Steuerung **110** angeordnet ist. Die externe DMA-Schnittstelle **134** ist ihrerseits mit der externen DMA-Einrichtung **134** und der Puffersteuerung **122** verbunden. Die Puffersteuerung **122** ist außerdem mit dem Pufferspeicher **112** und mit der Faserkanalschnittstelle **124** verbunden. Die Faserkanalschnittstelle **124** ist mit den Faserkanalschleifen **116** über einen Faserkanalsende- und Empfangsschaltkreis innerhalb der Schnittstelle **124** verbunden. Die Faserkanalschleifen **116** sind auch mit einem oder mehreren Hosts **104** verbunden. Die Faserkanalschnittstelle **124** kann verwendet werden, um sowohl über das Faserkanalprotokoll als auch über das SCSI-Protokoll zu kommunizieren.

[0034] Eine Datenstromsteuerung **126** ist mit der externen DMA-Schnittstelle **134**, der Puffersteuerung **122** und der Faserkanalschnittstelle **124** verbunden. Ein ECC-Maschinenmodul **132** ist über den Datenbus **126** mit der Plattenformatierung **128**, der Puffersteuerung **122** und der Faserkanalschnittstelle **124** verbunden.

[0035] Die Faserkanalschnittstelle **124** implementiert das Faserkanalprotokoll. In einer Ausführungsform implementiert die Faserkanalschnittstelle **124** das FC-AL (faserkanalvermittelte Schleife)-Protokoll einschließlich der FC-1 und FC-2-Ebenen der Faserkanalsignalgebungsschnittstelle. Die Faserkanalschnittstelle **124** enthält einen Encoder und einen Decodierer sowie einen Frame-Puffer für die Verwaltung bzw. Handhabung von Taktabweichungen (Clock Skew) und eine Handhabungsmaschine für das Faserkanalprotokoll. Die Faserkanalschnittstelle **124** erklärt Datenzüge, die von den Faserkanalschleifen empfangen wurden, als gültig und leitet sie zu geeigneten Bereichen in dem Frame-Puffer. Der Übertragungspfad der Faserkanalschnittstelle **122** sendet Frames von dem Frame-Puffer an den Faserkanal. Die Faserkanalschnittstelle handhabt automatisch die Frame-Begrenzungen und die Frame-Steuerung.

[0036] In einer Ausführungsform stellt der Plattenformatierer **128** eine Schnittstelle für die Festplatte **106** bereit. Der Plattenformatierer **128** empfängt Steuerinformationsparameter von dem lokalen Prozessor **108**. Der Plattenformatierer **128** führt dann die angeforderte Aktion durch. Der Mikroprozessor kann auch andere Aufgaben handhaben, wie z. B. eine Signalverarbeitung für die Servosteuerung des Plattenkopfes und die SCSI-Protokollverwaltung, die verwendet werden kann, um Daten über den Faserkanal zu kommunizieren. Der Mikrocontroller **120** kann verwendet werden, um andere Aufgaben auszuführen, einschließlich untergeordneter Aufgaben, wie z. B. Aufgaben der Faserkanalübertragungsverwaltung und entlastet dadurch den Prozessor **108**. In einer Ausführungsform wird daher die Arbeitsbelastung der Steuerung in vorteilhafter Weise auf mehrere Prozessoren aufgeteilt, was eine effizientere Ausführung der Aufgaben erlaubt.

[0037] In einer Ausführungsform ist der Mikrocontroller **120** eine 8-Bit-Maschine, die entweder 8 Bit- oder 16 Bit-DMA-Übertragungen unterstützen kann. Zusätzlich hat der Mikrocontroller **120** Register und eine interne ALU, einschließlich eines Sammlers bzw. Akkumulators, der für mathematische Operationen verwendet wird.

[0038] In einer Ausführungsform ist die Puffersteuerung **122** eine DMA-Steuerung mit mehreren Kanälen, die in eine synchrone DRAM (SDRAM)-Steuerung integriert ist. In einer anderen Ausführungsform unterstützt die Puffersteuerung **122** Puffergrößen bis herauf zu 4 Megabytes. In einer weiteren Ausführungsform werden aber auch Größen von mehr oder weniger als 4 Megabytes unterstützt. Die Puffersteuerung **122** stellt eine Hochgeschwindigkeitsinitialisierung und -verifizierung für den Pufferspeicher **112** bereit und reduziert die Energie bei der Initialisierung. Die Puffersteuerung **122** stellt auch Steuerungen für den Pufferspeicher **112** bereit, die in einer Ausführungsform unter Verwendung eines synchronen DRAM (SDRAM) implementiert ist. Andere Speicherarchitekturen, wie z. B. asynchrone RAM oder statische RAM, können jedoch ebenso verwendet werden.

[0039] Die Puffersteuerung wird durch eine vierkanalige Hochgeschwindigkeits-DMA-Steuerung, die im Burst-Betrieb arbeitet, bereitgestellt. Die Puffersteuerung **122** stellt eine Schnittstelle zwischen einer Vielfalt von Modulen und Schnittstellen bereit, einschließlich beispielsweise des Puffer- oder Haltespeichers **112**, des Plattenkanals, eines ECC-Kanals, des Faserkanals, des Mikroprozessors **108** und des Mikrocontrollers **120**. Die Puffersteuerung **122** reguliert und puffert die Datenbewegung in den SDRAM-Pufferspeicher **112** und aus diesem heraus. Jeder DMA-Kanal unterstützt DMA mehrerer Blocks von Daten im Burst-Betrieb, was Übertragungen mit großer Bandbreite ermöglicht. Jeder DMA-Kanal hat zugehörige Steuer-, Konfigurations- und Pufferspeicheradressregister. In einer Ausführungsform wird der Pufferspeicher **112** durch andere Einheiten gemeinsam verwendet, die mit Zugriffswechseln oder Zugriffsanteilen versehen sind. Innerhalb des Anteils gibt eine gegebene Datenquelle Daten im Burst-Betrieb aus, bis die Übertragung abgeschlossen ist oder bis die zugehörige anteilige Zeit abgelaufen ist. Viele der DMA-Kanäle sind jedoch mit Modulen verbunden, die mit einem anderen Takt oder mit einer anderen Geschwindigkeit als die Puffersteuerung **122** laufen. Wie unten erläutert, wird eine Speichereinheit, die als FIFO (First-In-First-Out)-Speicher konfigurierbar ist, verwendet, um Kanaldaten zu synchronisieren und zu puffern.

[0040] Die Puffersteuerung **122** stellt auch eine Mikroprozessoradressdecodierung, eine Prioritätsvermittlung für die Puffennessource, eine Block-CRC (BCRC)-Überprüfung und eine automatische DRAM-Erneuerungssteuerung bereit. In einer Ausführungsform umfaßt ein CRC-Berechner/Überprüfer einen Akkumulator mit Rückkopplung.

[0041] Die Datenstromsteuerung **126** reduziert die Datenübertragungszeit, indem sie in einer Ausführungsform automatisch den Datenstrom zwischen der Festplatte **106** und den Faserkanalanschlüssen überwacht und steuert. Die automatische Steuerung des Datenstroms zwischen diesen Kanälen reduziert die Anzahl von Interrupts, die bei einer typischen Datenübertragung von Platte zu Faserkanal auftreten. Zusätzlich hilft die Datenstromsteuerung **126** automatisch dabei, Pufferüberlauf- und Leerlaufzustände zu verhindern, während der Plattenformatierer oder der Faserkanal DMA-Kanal zeitweilig ausgesetzt oder angehalten werden, bevor der Puffer voll oder leer wird.

[0042] Beispielhafte Datenstromoperationen zwischen dem Host **104** und der Platte **106** werden nunmehr beschrieben. Eine typische Schreiboperation verläuft folgendermaßen. Der Host **104** sendet einen Schreibbefehl über den Faserkanalbus **116** an die Plattensteuerung **102**. Der Befehl durchläuft die Faserkanalschnittstelle **124** und wird dann über die Bussteuerung **124** in den Pufferspeicher **112** geladen. Der Hostpfad zu der Plattensteuerung wird Kanal **1** genannt. Wenn der Schreibbefehl in den Pufferspeicher **112** geschrieben worden ist, liest der Mikroprozessor **108** den Befehl aus dem Pufferspeicher **112** und stellt die Plattensteuerregister in Vorbereitung der befohlenen Schreiboperation in passender Weise ein. Die Plattensteuerregister enthalten Register, die zu der Puffersteuerung **122**, dem Plattenformatierer **128** und der Datenstromsteuerung **126** gehören. Wenn die Einstellung abgeschlossen ist, benachrichtigt der Mikroprozessor **108** den Host **104**, daß die Plattensteuerung **102** bereit ist, die Schreibdaten zu empfangen. Der Host **104** löst dann eine DMA (direkte Speicherzugriffs)-Übertragung in den Pufferspeicher **112** aus. Während des Pufferladevorgangs berechnet das CH1-Modul **310** kontinuierlich ein CRC auf der Basis der Schreibdaten. Das berechnete CRC wird in dem Pufferspeicher **112** mit den zugehörigen Schreibdaten gespeichert. Die Datenstromsteuerung **126** überwacht, wann eine gegebene Anzahl von Sektoren in dem Pufferspeicher **112** gespeichert ist und löst die Übertragung aus dem Pufferspeicher **112** durch die Puffersteuerung **122** an die Plattenformatierung **128** aus, und von dort schreibt der Plattenformatierer **128** über den Schreibkanal die Daten auf die Platte **106**. Wenn die Daten aus dem Pufferspeicher **112** gelesen und auf die Platte **106** geschrieben werden, wird ein zugehöriger ECC-Code berechnet und am Ende der Sektordaten angehängt. Wenn der Schreibvorgang abgeschlossen ist, erzeugt der Mikroprozessor **108** einen Interrupt, der über die Faserkanalschnittstelle an den Host **104** übertragen wird und dadurch den Host **104** informiert, daß der Schreibvorgang abgeschlossen ist.

[0043] Ein Lesevorgang wird in ähnlicher Weise wie der Schreibvorgang, jedoch in umgekehrter Weise ausgeführt. Der Host **104** sendet einen Lesebefehl an die Plattensteuerung **102**, die ihrerseits den Lesebefehl in dem Pufferspeicher **112** speichert. Der Mikroprozessor **108** liest dann den Befehl aus dem Pufferspeicher **112** und initialisiert in entsprechender Weise die verschiedenen Funktionsblöcke der Plattensteuerung **102**, um das Lesen durchzuführen. Die Daten werden von der Platte **106** gelesen und durchlaufen den Plattenformatierer **128** zu der Puffersteuerung **122** und von der Puffersteuerung **122** zu dem Pufferspeicher **112**. Sowohl die Daten als auch der zugehörige CRC werden gemeinsam in dem Pufferspeicher **112** gespeichert. Das ECC-Modul **132** überwacht die Datenübertragung und berechnet den ECC-Code durchgehend während des Datenspeichervorgangs. Nach Abschluß der ECC-Berechnung stellt das ECC-Modul **132** fest, welche Fehler bei dem Lesen der Daten aufgetreten sind und korrigiert diese Fehler entsprechend in dem Pufferspeicher **112**. Das CH1-Modul **310** liest Daten aus dem Pufferspeicher **112** und überprüft dann den CRC des resultierenden, ge-

speicherten Sektors aus dem Pufferspeicher **112** und stellt sicher, daß keine Fehler vorliegen. Wenn festgestellt wurde, daß keine Fehler in den Daten verblieben sind, werden die Daten von dem Pufferspeicher **112** durch die Puffersteuerung **122** und nach außen an den Host übertragen. Der CRC-Code wird während dieses Vorgangs abgestreift, so daß nur die Daten an den Host **104** gesendet werden.

[0044] [Fig. 2](#) veranschaulicht eine Ausführungsform der Schnittstellen zu der Puffersteuerung. Die Puffersteuerung ist mit einer Mikroprozessorschnittstelle **202**, einer Mikroprozessorschnittstelle **204**, einer ECC-Schnittstelle **206**, einer Plattenformatierschnittstelle **208**, einer Host-Schnittstelle **212** und einer Schnittstelle **210** eines SDRAM-Pufferspeichers verbunden. Die Puffersteuerung verwendet entsprechende FIFOs, um die zwischen den verschiedenen Schnittstellen übertragenen Daten zu puffern.

[0045] [Fig. 3](#) ist eine genauere Darstellung einer Ausführungsform der Puffersteuerung **122**. Die Puffersteuerung **122** enthält eine Speichereinheit **302**, die als FIFO mit 4 Anschlüssen konfigurierbar ist. Beispielsweise kann der MP-FIFO **302** eine Schnittstelle für wahlfreien oder direkten Zugriff, eine Schnittstelle für einen FIFO oder indirekten Zugriff, eine Schnittstelle für den Registerzugriff und eine Schnittstelle für den DMA der Puffersteuerung haben. Auf den FIFO **302** kann durch viele Clients und andere Einrichtungen zugegriffen werden und er wirkt damit als eine Umschaltstation, was es erlaubt, daß Daten zeitweise zwischengespeichert werden, während sie von einem Ort zu einem anderen geleitet werden. In der dargestellten Ausführungsform wird die DMA-Schnittstelle verwendet, um Zugriff auf den Pufferspeicher **112** zu gewähren. Die Registerzugriffsschnittstelle, wobei der gesamte FIFO oder ein Teil desselben als interne Registerdatei durch den Mikrocontroller **120** adressierbar ist, stellt einen von mehreren Zugriffspfaden zu dem Mikrocontroller **120** bereit. Die Schnittstelle für indirekten Zugriff, wobei der gesamte FIFO oder ein Teil desselben unter Verwendung von FIFO-Zeigern adressierbar ist, stellt noch einen weiteren Zugriffspfad bereit. Die Schnittstelle für direkten Zugriff, wobei der gesamte FIFO oder ein Teil desselben als Speicher mit wahlfreiem Zugriff adressierbar ist und Speicher entsprechend abgebildet bzw. zugeordnet ist, stellt noch einen weiteren Zugriffspfad bereit.

[0046] In einer Ausführungsform liegt der gesamte FIFO **302** oder ein Teil desselben innerhalb des Adreßraumes eines oder mehrerer Prozessoren, wie z. B. des Mikroprozessors **108** und/oder des Mikrocontrollers **120**. In einer weiteren Ausführungsform kann der FIFO **302** mehr oder weniger Anschlüsse haben. Indem mehrere Typen von FIFO-Zugriffsanschlüssen bereitgestellt werden, werden die Flexibilität, Effizienz und der Durchsatz von Datenübertragungen durch den FIFO **302** in hohem Maße verbessert. Der FIFO **302** bietet weiterhin dem Mikroprozessor **108** Zugriff auf Plattenformattabellen, Daten, Programmcodes und Faserkanalbefehlsblöcke, die in dem FIFO **302** oder in dem Puffer **112** gespeichert sind. Beispielsweise kann der FIFO **302** verwendet werden, um in dem Pufferspeicher **112** gespeicherte Programme zu überlagern. Der FIFO **302** kann also auch als MP (Mikroprozessor)-FIFO oder als MP-DMA-FIFO bezeichnet werden.

[0047] In einer Ausführungsform enthält der MP-FIFO **302** 64 Bytes an Speicherraum. Wie unten erläutert, bietet dies in vorteilhafter Weise genug Speicherplatz um einen vollständigen Faserkanalbefehl zu halten, zusätzlich zu einem zugehörigen CRC und einem Pufferlängenfeld. Das Format des Faserkanalbefehls ist in [Fig. 7](#) veranschaulicht. Der FIFO **302** kann verwendet werden, um in einem Burst Daten in den Pufferspeicher **112** und aus diesem heraus zu übertragen. Demnach können Latenzzeiten der Datenübertragung in hohem Maße reduziert werden. Auf den Anschluß für wahlfreien Zugriff kann über den Mikroprozessorbus und den Mikrocontrollerbus **130** zugegriffen werden, die jeweils eine Verbindung zu dem Mikroprozessor **108** bzw. dem Mikrocontroller **120** herstellen. Zusätzlich kann der Mikrocontroller **120** auf den MP-FIFO **302** über den Registerzugriffsanschluß zugreifen. Der DMA-Anschluß der Puffersteuerung stellt eine Schnittstelle zu dem SDRAM-Pufferspeicher **112** ebenso wie zu dem CRC-Überprüfer und Generator bereit. Verschiedene funktionelle Blöcke oder Module der Puffersteuerung sind über einen internen Datenbus der Puffersteuerung miteinander verbunden. Demnach kann in einer Ausführungsform auf den FIFO **302** durch direktes Adressieren als Speicher mit wahlfreiem Zugriff zugegriffen werden. Zusätzlich kann auf den MP-FIFO **302** als First-In-First-Out-Speicher zugegriffen werden, wobei Daten über ein FIFO-Zugriffsregister oder einen entsprechenden Anschluß in den FIFO **302** geladen und aus diesem herausgeholt werden.

[0048] [Fig. 4](#) veranschaulicht noch genauer eine Ausführungsform des MP-FIFO **302** mit mehreren Zugriffskanälen. In einer Ausführungsform speichert der MP-FIFO bis zu 64 Bytes, die in einem 16×32 -Array **402** angeordnet sind. Das FIFO-Array **402** kann also sechzehn 32-Bit-Worte halten. In einer Ausführungsform ist das FIFO-Array **402** durch Verwendung synchroner, statischer RAM-Elemente mit 2 Anschlüssen implementiert. Einer der beiden RAM-Anschlüsse ist ein Leseanschluß und der andere RAM-Anschluß ist ein Schreibanschluß. Jeder der Lese- und Schreibanschlüsse kann getrennt und unabhängig gesteuert werden. Auf die Lese- und Schreibanschlüsse kann über eine oder mehrere Schnittstellen oder Zugriffsanschlüsse zugegriffen werden. In einer Ausführungsform kann auf das FIFO-Array **402** als interne Registerdatei durch einen oder

mehrere Prozessoren, wie z. B. den Mikrocontroller **120** zugegriffen werden. Demnach kann auf das FIFO-Array **402** durch den Mikrocontroller **120** mit nur geringer Zusatzlast an Zeit oder Verwaltung schnell zugegriffen werden. Dies ist ein Unterschied zu konventionellen Plattensteuerungen, welche langsamer auf Daten von einem externen Speicher zugreifen.

[0049] Wie in [Fig. 4](#) dargestellt, enthält der MP-FIFO in einer Ausführungsform ein Übertragungsadreßregister **404** des Pufferspeichers, einen Übertragungszähler **406** des Pufferspeichers, ein MP-FIFO-Befehlsregister **408**, einen FIFO-Schreibzeiger **410**, einen FIFO-Lesezeiger **412**, einen Zähler **414** der FIFO-Ausnutzung, das FIFO-Zugriffsregister **416**, Adreßregister **420**, **422** für das Heranholen von FCP-Befehlen und eine Zustandsmaschine **418**. Die vorerwähnten Register und Zähler werden verwendet, um Datenübertragungen zu und von dem MP-FIFO **302** zu steuern. Das FIFO-Array **402** kann eine direkte Speicherabbildung bzw. Zuordnung auf eine interne Registerdatei des Mikrocontrollers haben. In einer Ausführungsform kann auf die Register, Zähler und das FIFO-Array **402** durch den Mikroprozessor **108** und/oder den Mikrocontroller **120** als Speicher mit wahlfreiem Zugriff zugegriffen werden. Wie unten erläutert, wird dadurch eine hohe Flexibilität beim Zugreifen auf Faserframes und beim Lokalisieren zugehöriger CRC-Daten, die zufällig im FIFO angeordnet sind, bereitgestellt.

[0050] In einer Ausführungsform wird eine Semaphore verwendet, um Konflikte zwischen mehreren Einrichtungen zu vermeiden, wie z. B. dem Mikroprozessor **108** und dem Mikrocontroller **120**, welche versuchen, auf die FIFO-Ressourcen zuzugreifen. Demnach wird in einer Ausführungsform ein Register verwendet, um ein Semaphorengewährungsbit zu speichern. Bevor eine anfordernde Einrichtung versucht, den FIFO **302** zu verwenden oder FIFO-Information, wie z. B. Pointer, Zähler, Zugriffsregister und dergleichen zu modifizieren, versucht sie zunächst, die Semaphore zu beschaffen.

[0051] Das MP-FIFO-Befehlsregister **408** wird verwendet, um Übertragungen zu oder von dem Pufferspeicher **112** von oder zu dem MP-FIFO **302** auszulösen. Eine Vielfalt von FIFO-Befehlen wird verwendet, um Datenübertragungen zwischen dem FIFO **302** und dem Pufferspeicher **112** zu steuern. Die Übertragungsbefehle können im allgemeinen in 4 Gruppen kategorisiert werden. Diese Befehlsgruppen umfassen Lade- und Entladebefehle, Lese- und Schreibbefehle, Steuerbefehle und automatische Befehle. Einige der Befehle ermöglichen einen direkten Frame-Zugriff auf Faserkanalframes. Die Lade- und Entladebefehle der Datenübertragung behandeln den FIFO **302** als einen First-In-First-Out-Speicher. Die Übertragungsbefehle für das Laden und Entladen von Daten bewirken das Laden oder Entladen des FIFOs **302** unter Verwendung des MP-FIFO-Schreibzeigers **410**, des MP-FIFO-Lesezeigers **412** und des MP-FIFO-Ausnutzungszählers **414**. In einer Ausführungsform sind einer oder mehrere Übertragungsbefehle so ausgestaltet, daß sie Übertragungen handhaben, welche die FIFO-Größe übersteigen.

[0052] Die Übertragungsbefehle zum Lesen und Schreiben von Daten behandeln den FIFO als einen Speicher mit wahlfreiem Zugriff. In einer Ausführungsform lesen und schreiben die Lese- und Schreibbefehle Daten mit dem Beginn an der Position 0 in dem FIFO **302**. In einer anderen Ausführungsform können die Datenübertragungen bei anderen FIFO-Speicheradressen beginnen.

[0053] Die Steuerbefehle, wie z. B. der Befehl zum Bereinigen bzw. Löschen des FIFOs, können verwendet werden, um den FIFO freizumachen, indem die zugehörigen Zähler und Register freigemacht werden. Zusätzlich können einer oder mehrere der Steuerbefehle verwendet werden, um irgendwelche Befehle, die aktuell in Verarbeitung sind, zu unterbrechen oder anzuhalten. Die automatischen Befehle, wie z. B. der Befehl zum Heranholen von FCP und der Befehl zum Aktualisieren des aktuellen FCP-Befehlszeigers automatisieren in effektiver Weise das Heranholen oder das Übertragen von Befehlen, wie z. B. Befehlsframes eines Faserkanals, von dem Pufferspeicher **112** und dergleichen zu dem FIFO **302**. Wie zuvor erläutert, werden in einer Ausführungsform die FIFO-Befehle durch eine Zustandsmaschine ausgeführt.

[0054] Die FIFO-Lese- und Schreibzeiger **412**, **410** werden jeweils verwendet, um die aktuellen Lese- und Schreibpositionen zu verfolgen, wenn auf den FIFO **302** in der First-In-First-Out-Weise zugegriffen wird. Der Schreibzeiger **410** steuert die Position oder Adresse eines aktuellen Schreibvorganges. Der Schreibzeiger **410** enthält einen Byte-Versatz relativ zu der Startadresse des FIFO-Arrays **402**. In einer Ausführungsform wird der Schreibzeiger **410** nach einem Einschalt-Reset oder nach Empfang eines Reset-Befehles eines MP-DMA-FIFO auf 0 initialisiert. In ähnlicher Weise steuert der Lesezeiger **412** die Position oder Adresse eines aktuellen FIFO-Lesevorgangs. Der Lesezeiger **412** enthält einen Byte-Versatz relativ zu der Startadresse des FIFO-Arrays **402**. In einer Ausführungsform wird der Lesezeiger **412** nach einem Einschalt-Reset oder nach Empfang eines Reset-Befehles eines MP-DMA-FIFO auf 0 initialisiert.

[0055] Der FIFO-Ausnutzungszähler **414** wird verwendet, um die Zahl bzw. den Wert der FIFO-Ausnutzung zu verfolgen. Beispielsweise kann der FIFO-Ausnutzungszähler **414** den Betrag bzw. die Menge an verwendeten FIFO-Speicherplätzen verfolgen und damit die Anzahl von FIFO-Positionen, die nicht verwendet bzw. ausgenutzt werden. In einer Ausführungsform kann der Ausnutzungszähler **414** gelesen werden oder es kann in ihn geschrieben werden als ein Register, und zwar durch den lokalen Mikroprozessor **108** und/oder Mikrocontroller **120**. Eine lesbare Zahl bzw. -Zählung stellt die Anzahl von Bytes bereit, die in dem FIFO-Array gespeichert sind. In einer Ausführungsform wird die Anzahl von Bytes auf die nächstliegende Wortgrenze abgerundet. Die FIFO-Zahl bzw. -Zählung wird schrittweise herabgesetzt, während Daten von dem FIFO-Array **402** gelesen werden. Demnach zählt ein 4-Bytes-Lesevorgang über das Zugriffsregister **416** die Zahl um 4 herab, wenn das letzte der 4 Bytes gelesen worden ist. In ähnlicher Weise zählt das Lesen von zwei 16-Bit-Halbwörtern von dem FIFO-Array **402** die FIFO-Zahl um 4 herab, wenn das letzte 16-Bit-Halbwort gelesen worden ist. Zusätzlich wird die FIFO-Zahl um 4 herabgezählt, wenn ein 32-Bit-Wort aus dem FIFO-Array **402** gelesen wird. Die von dem Array **402** gelesenen Daten können in einem anderen Speicher, wie z. B. dem Pufferspeicher **412**, gespeichert werden.

[0056] Das MP-FIFO-Zugriffsregister **416** wird durch den Mikroprozessor **108** verwendet, um Daten in dem FIFO-Array **402** zu lesen bzw. in dieses zu schreiben. Demnach werden Daten, die in das Zugriffsregister **416** geschrieben werden, ihrerseits an die FIFO-Position geschrieben, welche durch den Schreibzeiger **410** angezeigt wird. Der Schreibzeiger **410** wird typischerweise im Anschluß an einen solchen Schreibvorgang schrittweise heraufgesetzt. In ähnlicher Weise wird, wenn Daten über das Zugriffsregister **416** gelesen werden, der FIFO-Lesezähler heraufgesetzt. In einer Ausführungsform kann eine variierende Anzahl von Bytes über das Zugriffsregister **416** gelesen oder geschrieben werden. Demnach können beispielsweise entweder 1 Byte oder 2 Bytes in einer Operation, welche das Zugriffsregister **416** verwendet, gelesen oder geschrieben werden. SDRAM-Zugriff auf das FIFO-Array **402** wird für die Zustandsmaschine **418** gewährt, wie nachstehend noch beschrieben wird.

[0057] Das Übertragungsregister **404** des MP-FIFO-Pufferspeichers wird verwendet, wenn Daten zu oder von dem Pufferspeicher **112** übertragen werden. Das Übertragungsadreßregister **404** des Pufferspeichers wird mit der geeigneten Pufferspeicheradresse geladen, bevor eine Übertragung stattfindet. Der Übertragungszähler **406** des Pufferspeichers wird verwendet, um die Anzahl von Bytes zu verfolgen, die an den Pufferspeicher **112** übertragen werden. Der Übertragungszähler **406** wird auf eine Zahl gesetzt, welche der Menge an Daten, wie z. B. der Anzahl von Bytes, entspricht, die übertragen werden sollen. In einer Ausführungsform ist die Byte-Zahl ein Vielfaches von 4. In vorteilhafter Weise kann die Byte-Zahl die Größe des FIFO-Arrays **402** übersteigen. In einer Ausführungsform kann der Übertragungszähler so eingestellt werden, daß er bis zu 64 K Bytes überträgt. In einer anderen Ausführungsform kann der Übertragungszähler **406** auf eine Zahl gesetzt werden, die größer als 64 K ist. Demnach sind in einer Ausführungsform der vorliegenden Erfindung die Blockübertragungen nicht auf die FIFO-Größe beschränkt. Wenn das FIFO-Array **402** während eines Übertragens von Daten an den FIFO voll wird, wird die Übertragung ausgesetzt, bis die Daten aus dem FIFO-Array **402** ausgelesen werden. In ähnlicher Weise wird, wenn das FIFO-Array **402** während einer Übertragung voll wird, die Übertragung ausgesetzt, bis Daten in das FIFO-Array **402** geladen sind.

[0058] In einer Ausführungsform kann eine Einrichtung, wie z. B. der Mikroprozessor **108** oder der Mikrocontroller **120** über das MP-FIFO-Zugriffsregister **416** auf den FIFO **302** zugreifen, während die FIFO-Befehle ausgeführt werden. Demnach können beispielsweise der Mikroprozessor oder Mikrocontroller aus dem FIFO **302** lesen, während der FIFO **302** gefüllt wird. In ähnlicher Weise können der Mikroprozessor oder der Mikrocontroller **120** in den FIFO **302** schreiben, während der FIFO gelesen oder geleert wird.

[0059] Die Betriebsweise des MP-FIFO **302** als ein First-In-First-Out-Speicher wird nunmehr erläutert. In einer Ausführungsform bewirkt ein Schreiben in das FIFO-Zugriffsregister **416**, das die Daten an dem Offset-Punkt gespeichert werden, der in den Schreibzeiger geladen ist. Der Schreibzeiger wird im Kreislauf schrittweise heraufgesetzt, und zwar jedes Mal, wenn Daten in das FIFO-Zugriffsregister **416** geschrieben werden. Ein Unterbrechungssignal wird erzeugt, wenn ein Schreibvorgang auf das Zugriffsregister **416** versucht wird, während die von dem Ausnutzungszähler erfaßte Zahl gleich der FIFO-Größe ist. Dies stellt sicher, daß die FIFO-Daten nicht überschrieben werden. Das Unterbrechungssignal bewirkt, daß die Einrichtung, welche in den FIFO **302** schreibt, wie z. B. der Mikroprozessor **108** oder der Mikrocontroller **120**, anhält, bis Speicherstellen verfügbar gemacht worden sind. Die Speicherstellen können verfügbar gemacht werden durch Ausführen eines Lesevorgangs von dem Zugriffsregister, welches dementsprechend die Ausnutzungszahl herabsetzen würde. Die Unterbrechungsfunktion ermöglicht, daß Datenübertragungen zu und durch den FIFO **302** größer sind als die FIFO-Größe. Wenn sich der FIFO **302** während eines Schreibe- oder Lesevorgangs füllt oder leert, wird der Prozeß einfach unterbrochen anstatt daß er abgeschlossen wird. Demnach stellt die Unterbre-

chungsfunktion sicher, daß während Datenübertragungen Überlaufzustände nicht auftreten.

[0060] Zusätzlich können die FIFO-Speicherpositionen nach Ausführung eines FIFO-Freigabebefehls verfügbar gemacht werden, welcher den Ausnutzungszähler löschen oder auf 0 setzen würde.

[0061] Das Lesen des FIFO-Zugriffsregisters **416** bewirkt, daß die Daten, die an dem Lesezeigerversatz angeordnet sind, gelesen werden, und setzt den Lesezeiger in umlaufender Weise herauf. Die Wortzählung der Ausnutzung wird jedes Mal herabgesetzt, wenn ein Lesevorgang ausgeführt wird. Das Unterbrechungssignal wird erzeugt, wenn ein Lesevorgang von dem FIFO-Zugriffsregister versucht wird, während die FIFO-Ausnutzungszahl 0 ist. Das Unterbrechungssignal bewirkt, daß die Einrichtung, welche den FIFO **302** liest, wie z. B. der Mikroprozessor **108** oder der Mikrocontroller **120** oder der DMA-Kanal, anhalten, bis eine oder mehrere FIFO-Speicherpositionen gefüllt oder geladen sind. Die Speicherstellen können durch Ausführen eines Schreibvorgangs von dem Zugriffsregister geladen werden, welches dadurch die Ausnutzungszahl heraufsetzen würde. Demnach stellt die Unterbrechungsfunktion sicher, daß während Datenübertragungen keine Leerlaufzustände auftreten.

[0062] Wie zuvor bereits erläutert, kann in einer Ausführungsform auf das MP-FIFO **402** auch direkt und mit wahlfreiem Zugriff zugegriffen werden, und zwar durch den Mikroprozessor **108** oder den Mikrocontroller **120**. Die statischen RAM-Elemente, die verwendet werden, um das MP-FIFO-Array **402** zu bilden, ermöglichen, daß das MP-FIFO **302** als ein 64-Byte-Speicher mit wahlfreiem Zugriff gelesen wird. Das Array **402** kann sowohl auf den Adreßraum des Mikroprozessors als auch auf den Adreßraum des Mikrocontrollers abgebildet bzw. diesem zugeordnet werden. In einer Ausführungsform beeinflußt das Lesen oder Schreiben des MP-FIFO **302** als ein direkt adressierbares Speicherarray mit wahlfreiem Zugriff die MP-FIFO-Register oder -Zähler nicht, die zu der Betriebsweise des First-In-First-Out-Typs gehören.

[0063] Der MP-FIFO **302** kann auch verwendet werden, um während Kontextumschaltvorgängen interne Registerinformation des Mikroprozessors oder Mikrocontrollers zu speichern. Der Mikroprozessor **108** oder Mikrocontroller **120** kann also schnell Registerdateidaten mit dem FIFO-Array **402** austauschen. Die effiziente Registerübergabe, welche durch eine Ausführungsform der vorliegenden Erfindung bereitgestellt wird, ermöglicht in vorteilhafter Weise, daß der Prozessor schnell zwischen Aufgaben umschaltet.

[0064] Die Kontextumschaltung ermöglicht es, daß Plattensteuerungsaufgaben in effizienterer Weise ausgeführt werden. Wenn beispielsweise während Daten von der Platte **106** an die Faserkanalschnittstelle **124** unter Verwendung eines oder mehrerer Plattensteuerprozessoren, wie z. B. des Mikrocontrollers **120**, übertragen werden, die Übertragung die Fähigkeit der Platte, Daten zuzuführen, übersteigt, wäre es für den Mikrocontroller effizienter, während dieser Zeitdauer irgendeine andere Aufgabe auszuführen anstatt in verschwenderischer Weise eine Unterbrechung einzulegen, während die Platte **160** aufholt. Beispielsweise wäre es vorteilhaft, CACHEDATEN an einen anderen Host zu senden, während darauf gewartet wird, daß zusätzliche Plattendaten noch anfallen. Demnach ermöglicht eine Ausführungsform der vorliegenden Erfindung, daß der Kontext einer Operation in dem FIFO **302** aufbewahrt wird, während eine andere Aufgabe ausgeführt wird. Die Kontextumschaltung kann gemäß einer Ausführungsform der vorliegenden Erfindung besonders schnell ausgeführt werden, da auf das FIFO-Array **402** als interne Registerdatei des Mikrocontrollers **120** zugegriffen werden kann. Der Mikrocontroller **120** kann daher Kontextinformation schnell in das FIFO-Array **402** und aus diesem heraus austauschen. Der Kontext muß also nicht in einem langsameren Sekundärspeicher gespeichert werden, wie z. B. dem Pufferspeicher **112**. Abhängig von der Größe des FIFO-Arrays **402** und der Menge an Kontextdaten kann das FIFO-Array **402** auch verwendet werden, um gleichzeitig mehrere Kontexte zu speichern.

[0065] Die Kontextinformation kann relevante Information enthalten, die verwendet wird, um eine Aufgabe, wie z. B. Framezahlen, Sequenzzahlen, Übertragungszahlen und dergleichen zu verfolgen. Wenn demnach in einer Ausführungsform eine Übertragung keine verfügbaren Daten mehr hat, kann der Übertragungsvorgang ausgesetzt werden und der Kontext der Operation kann in dem FIFO **402** gespeichert werden. Abhängig von der Größe der CACHEDATEN, kann der Kontext mehrmals verändert werden.

[0066] Der gespeicherte Kontext kann unter Verwendung einer Vielfalt von Techniken wiederhergestellt werden. Beispielsweise kann der gespeicherte Kontext wiederhergestellt werden, wenn zusätzliche Daten für die ausgesetzte Übertragung verfügbar sind. Der gespeicherte Kontext kann einen existierenden Kontext überschreiben, wenn dieser nicht mehr benötigt wird. Unter Verwendung einer anderen Technik kann ein Kontext wieder in den FIFO **402** zurückgegeben werden und der ersetzte Kontext kann aus dem FIFO **402** gleichzeitig ausgegeben bzw. ausgetauscht werden.

[0067] Gemäß [Fig. 3](#) werden eine SDRAM-Steuerung **304** und ein Vermittlermodul **306** beim Zugreifen auf den Pufferspeicher **112** verwendet. Der Vermittler **306** wählt einen Anforderer aus, wenn mehrere Anforderungen aktiv sind, und stellt einen Zugriffsanteil bzw. ein Zugriffsfenster bereit. Der Vermittler **306** gewährt dann dem Anforderer einen Zugriff, wenn er an der Reihe ist. Die SDRAM-Steuerung **304** fährt dann fort, Daten zwischen dem Pufferspeicher **112** und dem Datenbus der Puffersteuerung zu verschieben.

[0068] Wie zuvor bereits erläutert, sind spezielle Befehle vorgesehen, um den Zugriff auf den in dem MP-FIFO **302** gespeicherte Daten zu erleichtern. In einer Ausführungsform nimmt die FIFO-Zustandsmaschine **418** Befehle, die in der Tabelle 2 unten aufgelistet sind, an und führt sie aus.

[0069] In einer Ausführungsform wird eine Übertragung nach dem Schreiben eines Befehls in das Befehlsregister **408** ausgelöst. Ein Aktivbit eines MP-DMA-FIFO-Befehls wird in ein Statutsregister gesetzt, um anzuzeigen, daß ein FIFO-Befehl gerade in Bearbeitung ist. Die nachstehende Tabelle 1 veranschaulicht eine Ausführungsform der Bitzuordnung für das Befehlsregister **408**.

| Bitposition | Name |
|-------------|---------------------------------|
| 15-8 | reserviert |
| 7 | nach Abschluß Interrupt |
| 6 | wähle FCP-CMD 2 aus |
| 5 | tausche Halbwort in Worten aus |
| 4 | tausche Bytes in Halbworten aus |
| 3-0 | Befehl |

Tabelle 1

[0070] Gemäß Tabelle 1 sind die Bits 15 bis 8 des Befehlsregisters **408** reserviert. Wenn das LBA (logische Blockadresse)-Formatbit (8) gesetzt ist, so wird ein CRC-Hardwareakkumulator bzw. -sammler initialisiert, wobei das invertierte LBA den Keim darstellt. Ansonsten verwendet der CRC-Sammler einfach den Keim („die Saat“), welcher in einer Ausführungsform nur aus Einsen besteht. In dieser Ausführungsform ermöglicht die Verwendung eines speziell dafür ausgelegten CRC-Sammlers in Hardware, daß der CRC ohne diese Aufgabe in die Prozessoren herunterzuladen, berechnet wird.

[0071] Befehle, die in das Befehlsregister **408** geladen sind, können bewirken, daß ein Interruptsignal erzeugt wird. In einer Ausführungsform wird ein Interrupt erzeugt, wenn ein Befehl abgeschlossen ist, und das Bit (7) für den Interrupt nach Abschluß wird in dem Befehlsregister **408** gesetzt. Ein Befehlsaktivbit wird in dem Statutsregister gesetzt, wenn ein Befehl noch aktiv ist. Zusätzlich kann ein Bit für den Befehlsabschluß in dem Statutsregister gesetzt werden, wenn ein Befehl abgeschlossen ist. In einer Ausführungsform kann der Interrupt zu einem Prozessor, wie z. B. dem Mikrocontroller **120** oder dem Mikroprozessor **108**, gehen. Der empfangende Prozessor kann den Interrupt maskieren oder den Interrupt an einen anderen Prozessor oder an den Host **104** weiterleiten. Eine Beschreibung der Bits 6–4 wird später gegeben. Die nachstehende Tabelle 2 veranschaulicht die Vorgänge, welche durch die Bits 3–0 des Befehlsregisters **408** definiert werden.

| Betrieb |
|---|
| Lösche FIFO |
| Stoppe FIFO |
| Lade FIFO |
| Entlade FIFO |
| Lade FIFO mit BCRC-Überprüfung |
| Entlade FIFO mit BCRC-Erzeugung |
| Lese aus Pufferspeicher |
| Schreibe in Pufferspeicher |
| Hole FCP-Befehl heran |
| Aktualisiere den Zeiger des aktuellen FCP-Befehls |

Tabelle 2

[0072] Der Befehl für das Löschen des FIFO setzt oder löscht die FIFO-Lese- und Schreibzeiger **412**, **410** auf 0. In einer Ausführungsform bleibt der Inhalt des FIFO-Arrays **402** durch den FIFO-Löschbefehl unverändert. In einer Ausführungsform kann der FIFO-Löschbefehl ausgegeben werden, während andere Befehle aktiv sind.

[0073] Der „Stoppe FIFO“-Befehl stoppt einen aktiven FIFO-Befehl und setzt die FIFO-Zustandsmaschine **418** zurück. In einer Ausführungsform kann der „Stoppe FIFO“-Befehl ausgegeben werden, während andere Befehle aktiv sind. In einer Ausführungsform bleiben die Inhalte der Zeiger **410**, **412** und des FIFO-Arrays **402** aufgrund des „Stoppe FIFO“-Befehls unverändert.

[0074] Der „Lade FIFO“-Befehl überträgt Daten von dem Pufferspeicher **112** in den MP-FIFO **302**. Die Daten werden in das FIFO-Array **402** geladen, wobei an der Position begonnen wird, welche durch den MP-FIFO-Schreibzeiger **410** definiert wird. Die Anzahl von Bytes, die übertragen werden sollen, wird in dem Zähler **406** für die Pufferspeicherübertragung eingestellt. Die Startadresse des Pufferspeichers für die Übertragung wird in dem Übertragungsadressregister **404** des Puffers eingestellt. In vorteilhafter Weise kann die Anzahl von zu übertragenden Bytes die Größe des FIFO-Arrays **402** übersteigen. Demnach können Übertragungen irgendeine gewünschte Größe haben. In einer Ausführungsform ist der Übertragungszähler so bemessen, daß er Übertragungen von einer Größe bis zu 4 K Bytes verwaltet, im Gegensatz der FIFO-Array-Größe von 64 Bytes. Der Übertragungszähler kann jedoch auch so bemessen sein, daß er größere oder kleinere Übertragungen handhabt, je nach dem was erwünscht ist.

[0075] In einer Ausführungsform können als Ergebnis eines Ladebefehls 4 K Bytes übertragen werden. Der Vorgang „Lade FIFO“ wird ausgesetzt, bis der FIFO **302** voll ist und die Übertragung noch nicht abgeschlossen ist. Ein Bit, daß der MP-DMA-FIFO voll ist, wird in dem Statusregister gesetzt, um anzuzeigen, daß das FIFO-Array **402** voll ist. Die Übertragung kann automatisch wieder aufgenommen werden, wenn sich der FIFO **302** zu leeren beginnt und wenn das Bit dafür, daß der MP-DMA-FIFO voll ist, gelöscht bzw. herabgesetzt ist.

[0076] [Fig. 8](#) veranschaulicht ein Beispiel eines Befehlsvorgangs „Lade FIFO“. Das FIFO-Array **402** enthalte „nicht zu beachtende“ Information, die durch „xx“ bezeichnet ist. Der Schreibzeiger (WP) **410** und der Lesezeiger (RP) **412** sind auf 0 gesetzt und zeigen dadurch auf die physikalische Anfangsadresse des FIFO-Arrays **402**. Wenn der Lesezeiger **412** auf dieselbe Adresse wie der Schreibzeiger **410** zeigt, ist der FIFO **402** entweder leer oder voll, je nach der Ausnutzungszahl. Die FI-FO-Wortzahl, welche anzeigt, wie viele FIFO-Wortpositionen verwendet worden sind, steht auf „0“, was anzeigt, daß der FIFO **402** leer ist. Der Übertragungszähler **406** des Pufferspeichers ist mit „32“ geladen, was anzeigt, daß 32 Bytes von dem Pufferspeicher **112** in das FIFO-Array **402** übertragen werden sollen. Aus Gründen der Klarheit ist in dem vorliegenden Beispiel der Pufferspeicher **112** mit sequentiellen Anzahlen geladen, auch wenn in einer Ausführungsform der eigentliche Pufferspeicherinhalt den Betrieb des Befehls „Entlade FIFO“ nicht beeinflusst.

[0077] Wenn der „Entlade FIFO“-Befehl von dem Befehlsregister **408** empfangen worden ist, findet die Übertragung von 32 Bytes von dem FIFO-Array **402** in den Pufferspeicher **112** statt. Die Daten werden, beginnend an der durch den Lesezeiger **412** angegebenen Position, in diesem Beispiel „0“, aus dem FIFO **302** ausgele-

sen. Der Lesezeiger **412** wird schrittweise um 4 heraufgesetzt, jedes Mal, wenn ein Wort übertragen und in den Pufferspeicher **112** geladen wird. Dieser Vorgang wiederholt sich, bis die Anzahl von Bytes, die durch die Übertragungszahl angegeben wird, in diesem Fall 32, übertragen ist. Wie in [Fig. 9](#) dargestellt, enthält der Pufferspeicher **112**, sobald der Befehl „Entlade FIFO“ abgeschlossen ist, die übertragenen Daten, der Schreibzeiger **410** bleibt bei 32, der Lesezeiger **412** ist auf 32 herabgesetzt und die FIFO-Wortzahl ist auf 0, was anzeigt, daß der FIFO leer ist. In einer Ausführungsform sind, auch wenn das FIFO-Array **402** leer ist, die Daten nicht aus dem FIFO-Array **402** gelöscht worden.

[0078] Der Befehl „Lade FIFO mit BCRC (Block-CRC)-Überprüfung“ überträgt Daten von dem Pufferspeicher **112** an den MP-FIFO **302**, während eine CRC-Überprüfung ausgeführt wird. Dieser Befehl ist besonders zweckmäßig für das Übertragen von Plattensektoren mit zugehörigen Sektor-CRCs. Die Daten werden in das FIFO-Array **402** geladen, mit dem Beginn einer Position, welche durch den MP-FIFO-Schreibzeiger **410** definiert wird. Die Anzahl von zu übertragenden Bytes wird in dem Übertragungszähler **406** des Pufferspeichers eingestellt. Die Startadresse der Quelle des Pufferspeichers für die Übertragung wird in dem Übertragungsadreßregister **404** des Puffers eingestellt. Die Anzahl von zu übertragenden Bytes kann die Größe des FIFO-Arrays **402** überschreiten. Die Block-CRC, die in einer Ausführungsform das letzte zu übertragende Wort ist, wird überprüft. Wenn die CRC-Überprüfung anzeigt, daß die Block-CRC gültig ist, so wird in dem Statusregister ein Bit gesetzt, welches „MP-DMA-FIFO-CRC-gut“ anzeigt. Der Vorgang „Lade FIFO mit BCRC-Überprüfung“ wird ausgesetzt oder unterbrochen, wenn der FIFO **302** voll ist und die Übertragung noch nicht abgeschlossen ist. Die Übertragung kann automatisch wieder aufgenommen werden, wenn sich der FIFO **302** zu leeren beginnt.

[0079] Das Unterbrechungsmerkmal ist besonders vorteilhaft für die Übertragung großer Datenblöcke, deren Größe die des FIFO **302** übersteigt, mit einem zugehörigen Fehlererfassungs- und/oder Korrekturcode, aus dem Pufferspeicher **112**. Der Fehlererfassungscode kann beispielsweise ein CRC sein. Unter Verwendung konventioneller Techniken kann die Übertragung von Datenblöcken, die größer als das FIFO-Array **402** sind, in nachteiliger Weise ausgeführt werden, indem der große Block wie mehrere kleine Blöcke behandelt wird, deren Größe gleich der des FIFO-Arrays **402** ist. Demnach würde der große Block in einer Serie kleinerer, getrennter Übertragungen übertragen werden. Diese konventionelle Technik kann jedoch die Berechnung der CRC, die für den gesamten Block gilt, schwierig machen oder in der Ausführung langsam machen, indem ein erneutes Impfen des CRC-Sammlers zu Beginn jeder Übertragung mit FIFO-Größe erforderlich ist. Im Gegensatz dazu ermöglicht eine Ausführungsform der vorliegenden Erfindung, daß der gesamte Block als ein Block behandelt wird, wobei der CRC über den gesamten Durchlauf der Übertragung akkumuliert wird, ohne daß eine Neuimpfung oder neue Keimlegung erforderlich ist.

[0080] [Fig. 10](#) veranschaulicht ein Beispiel einer Befehlsoperation des FIFO-Ladens mit BCRC-Überprüfung. Das FIFO-Array **402** enthalte nicht beachtenswerte Information, die mit „xx“ bezeichnet ist. Der Schreibzeiger (WP) **410** und der Lesezeiger (RP) **412** sind auf „0“ gesetzt und zeigen damit auf die physikalische Anfangsadresse des FIFO-Arrays **402**. Die Wortzahl der FIFO-Ausnutzung beträgt „0“, was anzeigt, daß das FIFO-Array **402** leer ist. Der Übertragungszähler **406** des Pufferspeichers ist mit „32 + 4“ oder 36 geladen, was 36 Bytes anzeigt, einschließlich 32 Datenbytes und 4 BCRC-Bytes, die von dem Pufferspeicher **112** in das FIFO-Array **402** übertragen werden sollen. Aus Gründen der Klarheit ist in dem vorliegenden Beispiel der Pufferspeicher **112** mit sequentiellen Zahlen geladen, auch wenn in einer Ausführungsform die tatsächlichen Pufferspeichereinhalte den Betrieb des Befehls des FIFO-Ladens mit BCRC-Überprüfung nicht beeinflussen.

[0081] Wenn der Befehl des FIFO-Ladens mit BCRC-Überprüfung von dem Befehlsregister **408** empfangen wird, findet die Übertragung von 32 Datenbytes und 4 BCRC-Bytes aus dem Pufferspeicher **112** in das FIFO-Array **402** statt. Die Pufferspeicherdaten und der BCRC werden in den FIFO **302** geschrieben, wobei an der Position begonnen wird, die durch den Schreibzeiger **410** angezeigt wird, in diesem Beispiel „0“. Der Schreibzeiger **410** wird schrittweise heraufgesetzt, während jeweils ein Byte übertragen wird und in das FIFO-Array **402** geladen wird. Dieser Vorgang wiederholt sich, bis die Anzahl von Bytes, welche durch die Transferzahl angegeben wird, in diesem Fall 36, übertragen worden ist. Der BCRC wird während der Übertragung überprüft, und wenn die CRC-Überprüfung anzeigt, daß der CRC gültig ist, so wird das Bit „MP-DMA-FIFO-CRC-gut“ in dem Statusregister gesetzt. Wie in [Fig. 10](#) gezeigt wird, enthält, wenn der Befehl des FIFO-Ladens mit BCRC-Überprüfung abgeschlossen ist, das FIFO-Array **402** die übertragenen Daten, der Schreibzeiger **410** ist auf 36 heraufgesetzt, der Lesezeiger **410** verbleibt bei 0 und die FIFO-Wortzahl steht bei 9. In einer Ausführungsform bleiben die übertragenen Daten auch in dem Pufferspeicher **112**.

[0082] Der Befehl des FIFO-Entladens mit BCRC-Erzeugung überträgt Daten von dem MP-FIFO **302** in den Pufferspeicher **112**, während ein Block-CRC erzeugt wird. Die Übertragung beginnt an der Position, welche durch den MP-FIFO-Lesezeiger **412** definiert wird. Die Anzahl von zu übertragenden Bytes wird in dem Über-

tragungszähler **406** des Pufferspeichers eingestellt. Der Startpunkt der Zieladresse für die Übertragung in den Pufferspeicher wird in dem Übertragungsadreßregister **404** des Puffers eingestellt. Das letzte zu übertragende Wort ist ein Dummy-Wort (Ersatzwort) und wird durch einen Block-CRC ersetzt, der durch die CRC-Maschine **132** berechnet wird. Die Anzahl von zu übertragenden Bytes darf die Größe des FIFO-Arrays **402** übersteigen. Die Operation des FIFO-Entladens mit BCRC-Erzeugung wird ausgesetzt oder unterbrochen, wenn der FIFO **302** leer ist und die Übertragung noch nicht abgeschlossen ist. Die Übertragung kann automatisch wieder aufgenommen werden, wenn der FIFO **302** sich zu füllen beginnt.

[0083] [Fig. 11](#) veranschaulicht ein Beispiel eines Befehlsvorgangs des FIFO-Entladens mit BCRC-Erzeugung. Der Pufferspeicher **112** enthalte nicht beachtenswerte Information, die mit „xx“ bezeichnet wird. Aus Gründen der Klarheit ist in dem vorliegenden Beispiel das FIFO-Array **402** mit 32 Bytes sequentieller Zahlen geladen. Zusätzlich enthalten die 4 Bytes, die auf die Daten folgen, ein Dummy-BCRC. D.h., der BCRC für die 32 Datenbytes an den FIFO-Adressen 0–31 ist noch nicht berechnet worden, jedoch für Zwecke der Verwaltung des FIFO wird der Schreibzeiger **410** auf 36 gesetzt, als ob die Adreßpositionen 32 bis 35 einen echten BCRC enthalten würden. Der Lesezeiger (RP) **412** wird auf „0“ gesetzt. Die Wortzahl der FIFO-Ausnutzung, die anzeigt, wie viele Wortpositionen des FIFO verwendet werden, liegt bei „9“, was anzeigt, daß sich 9 Worte in dem FIFO Array **402** befinden. Der Übertragungszähler **406** des Pufferspeichers wird mit „32 + 4“ oder 36 geladen, was anzeigt, daß 36 Bytes von dem FIFO-Array **402** in den Pufferspeicher **112** übertragen werden sollen.

[0084] Wenn der Befehl für das Entladen des FIFO mit BCRC-Erzeugung von dem Befehlsregister **408** empfangen wird, findet die Übertragung von 32 Bytes von dem FIFO-Array **402** in den Pufferspeicher **112** statt. Die Daten werden aus dem FIFO **302** gelesen, beginnend an der Position, die durch den Lesezeiger **412** angezeigt wird, in diesem Beispiel „0“. Der Lesezeiger **412** wird heraufgesetzt, während je ein Wort übertragen und in den Pufferspeicher **112** geladen wird. Der BCRC wird während dieser Übertragung kontinuierlich berechnet. Dieser Vorgang wiederholt sich, bis die Anzahl von durch die Übertragungszahl angegebenen Bytes, abzüglich der 4 Bytes für die BCRC-Daten, übertragen worden ist. In diesem Beispiel werden 32 Bytes von dem Array **402** übertragen. Die letzten 4 Bytes, die an den Pufferspeicher **112** übertragen werden, enthalten die BCRC-Daten, die anstelle durch das FIFO-Array **402** durch den CRC-Sammler geliefert werden. Wie in [Fig. 11](#) dargestellt, enthält der Pufferspeicher **112**, wenn der Befehl des FIFO-Entladens mit BCRC-Erzeugung abgeschlossen ist, die übertragenen Daten mit dem zugehörigen BCRC. Der Schreibzeiger **410** bleibt bei 36, der Lesezeiger **412** ist auf 36 heraufgesetzt worden und die Wortzahl des FIFOs liegt bei 0, was anzeigt, daß der FIFO leer ist. In einer Ausführungsform sind, auch wenn das FIFO-Array **402** leer ist, die Daten nicht aus dem FIFO-Array **402** gelöscht worden.

[0085] Der Befehl des Lesens aus dem Pufferspeicher überträgt 64 Bytes an Daten von dem Pufferspeicher **112** in den MP-FIFO **302**. Die Daten werden in das FIFO-Array **402** geladen, beginnend am Anfang oder bei der Nulladresse des FIFO-Arrays **402**. Der Anfang der Quelladresse des Pufferspeichers für die Übertragung wird in dem Übertragungsadreßregister **402** des Puffers eingestellt. In einer Ausführungsform werden der Lesezeiger **412** und der Übertragungszähler **406** unter Verwendung von Standardwerten automatisch initialisiert. Anstatt daß man also einen längeren Einstellvorgang durchlaufen muß, initialisiert ein Befehl sehr schnell die Zähler. Zusätzlich wird in einer Ausführungsform keine CRC-Überprüfung ausgeführt. Der Befehl für das Lesen aus dem Pufferspeicher, welcher automatisch 64 Bytes zu Beginn des FIFO-Arrays **302** lädt, erlaubt also, daß aufeinanderfolgende Seiten schnell geladen werden, ohne daß man die zusätzliche Zeit benötigt, die mit dem Verwalten der FIFO-Zeiger verknüpft ist. Dieser Befehl ermöglicht es einem Prozessor, wie z. B. dem Mikroprozessor **108** oder dem Mikrocontroller **120**, sehr schnell FIFO-Beladungen mit Intervallen von 64 Bytes auszuführen, um die in dem Puffer gespeicherten Daten zu untersuchen. In einer Ausführungsform wird während dieser Übertragung kein CRC ausgeführt.

[0086] [Fig. 13](#) veranschaulicht ein Beispiel einer Befehlsoperation des Lesens aus einem Pufferspeicher. Das FIFO-Array **402** enthalte nicht beachtenswerte Information, die mit „xx“ bezeichnet ist. Der Schreibzeiger (WP) **410**, der Lesezeiger (RP) **412**, der Übertragungszähler **406** und die Wortzahl, die auf dem Ausnutzungszähler **414** beruht, werden automatisch mit Standardwerten initialisiert. Aus Gründen der Klarheit wird in dem vorliegenden Beispiel der Pufferspeicher **112** mit aufeinanderfolgenden Zahlen geladen, auch wenn in einer Ausführungsform die tatsächlichen Pufferspeicherinhalte die Arbeitsweise des Befehls des Lesens aus dem Pufferspeicher nicht beeinflussen.

[0087] Wenn der Befehl des Lesens aus dem Pufferspeicher durch das Befehlsregister **408** empfangen worden ist, findet die automatische Übertragung von 64 Bytes aus dem Pufferspeicher **112** in das FIFO-Array **402** statt. Die Pufferspeicherdaten werden in das FIFO-Array **402** geschrieben, wobei an der ersten Speicherposi-

tion des FIFO begonnen wird. Alle 64 Bytes werden übertragen. Wie in [Fig. 13](#) dargestellt, enthält das FIFO-Array **402**, wenn der Befehl des Lesens aus dem Pufferspeicher abgeschlossen ist, die übertragenen Daten. In einer Ausführungsform verbleiben die übertragenen Daten auch in dem Pufferspeicher **112**.

[0088] Der Befehl des Schreibens in den Pufferspeicher überträgt den gesamten 64 Byte-Inhalt des FIFO-Arrays **402** in den Pufferspeicher **112**. Die Daten werden in das FIFO-Array **402** geladen, in dem am Anfang oder bei der Nulladresse des FIFO-Arrays **402** begonnen wird. Die anfängliche Zieladresse des Pufferspeichers für die Übertragung wird in dem Übertragungsadreßregister **404** des Puffers eingestellt. In einer Ausführungsform verwendet der Befehl des Schreibens in dem Pufferspeicher den Schreibzeiger **410**, den Lesezeiger **412** oder den Übertragungszähler **406** nicht. Der Befehl des Schreibens aus dem Pufferspeicher, welcher automatisch 64 Bytes vom Anfang des FIFO-Arrays **302** entläßt, ermöglicht also, daß aufeinanderfolgende Seiten schnell an den Pufferspeicher **112** übertragen werden, ohne daß man irgendwelche zusätzliche Zeit benötigt, die mit der Verwaltung der FIFO-Zeiger verknüpft ist. In einer Ausführungsform wird während dieser Übertragung kein CRC ausgeführt. [Fig. 14](#) zeigt ein Beispiel eines Befehlsvorgangs des Schreibens in den Pufferspeicher. Der Pufferspeicher **112** enthalte nicht beachtenswerte Information, welche mit „xx“ bezeichnet ist. Der Schreibzeiger (WP) **410**, der Lesezeiger (RP) **412**, und der Übertragungszähler **406** und der Ausnutzungszähler **414** werden automatisch mit Standardwerten initialisiert. Aus Gründen der Klarheit wird in dem vorliegenden Beispiel das FIFO-Array **402** mit aufeinanderfolgenden Nummern bzw. Zahlen geladen, auch wenn in einer Ausführungsform der eigentliche Inhalt die Betriebsweise des Befehls des Schreibens in den Pufferspeicher nicht beeinflusst.

[0089] Wenn der Befehl des Schreibens in den Pufferspeicher von dem Befehlsregister **408** empfangen wird, findet die automatische Übertragung von 64 Bytes aus dem FIFO-Array **402** in den Pufferspeicher **112** statt. Die 64 Bytes an FIFO-Daten werden automatisch in den Pufferspeicher **112** geschrieben, beginnend mit der Adresse, die in dem Übertragungsadreßregister **404** des Puffers gespeichert ist. Wie in [Fig. 14](#) dargestellt, enthält der Pufferspeicher **112**, wenn der Befehl des Schreibens in den Pufferspeicher abgeschlossen ist, die übertragenen Daten, und der Schreibzeiger **410** und der Lesezeiger **412** sind unverändert. In einer Ausführungsform verbleiben die übertragenen Daten auch in dem FIFO-Array **402**.

[0090] Das Bit (6) für die Auswahl des FCP (Faserkanalprotokoll)-CMD 2 wird in Verbindung mit dem Befehl für das Heranholen des aktuellen FCP und dem Befehl für das Aktualisieren des aktuellen FCP-Befehlszeigers verwendet. Wenn der Befehl für das Heranholen des aktuellen FCP ausgeführt wird, wird der aktuelle FCP-Befehl von dem Befehlspeicher in dem Pufferspeicher **112** an das MP-FIFO-Array **402** übertragen. In einer Ausführungsform wird diese Übertragung automatisch ausgeführt, sobald der FIFO-Befehl in das Befehlsregister **408** geschrieben worden ist, ohne daß der Prozessor eingreift.

[0091] Wie unten noch genauer erläutert wird, führt die Zustandsmaschine **418** den Befehl des Heranholens des aktuellen FCP aus, indem eine Sequenz des Löschsens bzw. Freimachens des FIFO, des Löschsens der Zeiger und des Ausnutzungszählers ausgeführt wird, und dann die Heranholadresse für den FCP-Befehl in das Übertragungsadreßregister **404** des MP-FIFO-Pufferspeichers kopiert wird. Das Bit 6 für die Auswahl des FCP-CMD 2 legt fest, welche der beiden Heranholadressen, die Heranholadresse FCP-Befehl 1 oder die Heranholadresse FCP-Befehl 2 verwendet wird, um den aktuellen FCP-Befehl heranzuholen. Die Bereitstellung der beiden Heranholadressen, nämlich der Heranholadresse FCP-Befehl 1 oder der Heranholadresse FCP-Befehl 2, ermöglicht in vorteilhafter Weise, daß zwei Befehle in dem Pufferspeicher **112** vorhanden sind. Die Heranholadresse FCP-Befehl 1 und die Heranholadresse FCP-Befehl 2 werden gemeinsam in dem Heranholadreßregister **420** für den FCP-Befehl 1 und dem Heranholadreßregister **422** für den Heranholbefehl 2 gespeichert. Die Register **420**, **422** enthalten die entsprechende Frametabellenbasis des FCP-Befehls, den Framezeiger der Heranholadresse des FCP-Befehls und den Framebeginn des 64 Byte-Befehls.

[0092] Die Zustandsmaschine **418** setzt dann den Übertragungszähler **406** des MP-FIFO-Pufferspeichers auf 60, welches diejenige Größe in Bytes eines FCP-Befehls ist, welcher den zugehörigen CRC und das Pufferlängenfeld ausschließt. Das Pufferlängenfeld wird, beginnend am oberen Ende oder der Position der Nulladresse des FIFO-Arrays **402** in den MP-FIFO **302** geladen. Demnach kann das Pufferlängenfeld schnell und automatisch für die Verwendung durch Verarbeitungselemente, wie z. B. den Mikrocontroller **120**, lokalisiert werden. In einer Ausführungsform befindet sich das Pufferlängenfeld an einer festen Position oder in einem festen Abstand relativ zu den Kopfzeilen bzw. Führungsdaten des FCP-Befehls. Das Pufferlängenfeld kann auch verwendet werden, um den CRC zu lokalisieren, der überprüft wird, um festzustellen, ob der Frame beschädigt ist. In einer Ausführungsform wird der FCP-Befehlsframe in dem Pufferspeicher **112** gespeichert, wobei der CRC oberhalb des Pufferlängenfeldes angeordnet wird, wie es in [Fig. 7](#) dargestellt ist. Ein Befehl des FIFO-Ladens mit BCRC-Überprüfung wird dann automatisch ausgeführt. In einer Ausführungsform wird das

Bit für die Unterbrechung nach Abschluß ebenso gesetzt.

[0093] Indem man also die Zustandsmaschine automatisch die komplexe Aufgabe des Lesens und Überprüfens eines FCP-Befehls ausführen läßt, ohne weiteren Eingriff durch den Mikroprozessor oder Mikrocontroller, wird die Verarbeitungslast für den Mikroprozessor **108** und den Mikrocontroller **120** in hohem Maße reduziert. Zusätzlich wird das Lesen des FCP-Befehls wesentlich schneller und effizienter ausgeführt.

[0094] Der Befehl für das Aktualisieren des aktuellen FCP-Befehlszeigers führt die Schritte aus, die benötigt werden, um den Heranholadreßzähler oder das Register für den aktuellen Befehl weiter zu bewegen. Als erstes wird der Heranholadreßzeiger um die Befehlsgröße, die hier 64 ist, heraufgesetzt. Das Bit (6) für die Auswahl des FCP-CMD 2 legt fest, welche der Heranholadressen FCP-Befehl 1 oder FCP-Befehl 2 verwenden. Der Interrupt bzw. die Unterbrechung für den Abschluß des MP-FIFO-Befehls wird dann erzeugt. Dann wird die Befehlszahl herabgesetzt.

[0095] [Fig. 12](#) veranschaulicht ein Beispiel einer Befehlsoperation eines Heranholens des aktuellen FCP. Das FIFO-Array **402** enthalte nicht-beachtenswerte Information, welche mit „xx“ bezeichnet wird. Der Schreibzeiger (WP) **410**, der Lesezeiger (RP) **412** und der Wortzähler sind ebenfalls „nicht beachtenswert“. Der Pufferspeicher **112** wird mit einem FCP-Befehlsframe geladen, welcher 6 Worte für die Framekopfzeilen, 8 Worte für die Framenutzlast, ein Wort für den FCP-Befehls-CRC und ein Wort für das Pufferlängenfeld enthält, was insgesamt 64 Bytes ausmacht.

[0096] Wenn der Befehl für das Heranholen des aktuellen FCP durch das Befehlsregister **408** empfangen worden ist, findet die automatische Übertragung von 64 Bytes aus dem Pufferspeicher **112** in das FIFO-Array **402** statt. Die Übertragung erfolgt von der Pufferspeicherstelle, welche durch die Heranholadresse des entsprechenden FCP-Befehls angezeigt wird. Als erstes wird das Pufferlängenfeld automatisch zu Beginn oder in der obersten Adreßposition des FIFO-Arrays **402** geladen. Demnach ist das Pufferlängenfeld das erste Wort, welches später aus dem FIFO-Array **402** entladen wird. Dann wird die Kopfzeile (Header) des FCP-Befehlsframes übertragen, gefolgt von der Framenutzlast und dem CRC, der überprüft wird. Wie in [Fig. 12](#) dargestellt, enthält das FIFO-Array **402**, wenn der Befehl des Heranholens des aktuellen FCP abgeschlossen ist, die übertragenen Daten mit dem Pufferlängenfeld an der Spitze. Der Schreibzeiger **410** und der Lesezeiger **412** bleiben bei 0 und die FIFO-Wortzahl steht bei 64, was anzeigt, daß das FIFO-Array **402** voll ist.

[0097] [Fig. 15](#) veranschaulicht eine Ausführungsform eines Kontextumschaltvorgangs. Dieses Beispiel veranschaulicht das Austauschen oder Übertragen eines Kontextes, der zuvor in dem FIFO-Array **402** gespeichert war, in eine beispielhafte Registerdatei, während der Kontext aus der Registerdatei in das FIFO-Array **402** übertragen wird. Die Registerdatei kann eine aus irgendeiner Anzahl von Registerdateien sein und kann verwendet werden, um eine Vielfalt von Daten oder Kontexten zu enthalten, wie z. B. eine FCP-Kopfzeile, den Zustand des Übertragungszählers, usw. Im vorliegenden Beispiel enthält das FIFO-Array **402** einen ersten Kontext, der in den ersten 28 Bytes des Arrays gespeichert ist. Der übrige Teil des Array **402** enthalte nicht-beachtenswerte Information, welche mit „xx“ bezeichnet wird. Der Schreibzeiger (WP) **410** steht bei 28, d.h. der Schreibzeiger **410** bewirkt, daß die nächsten Daten, welche durch das FIFO-Array **402** empfangen werden, in die Speicherposition geschrieben werden, welche auf den ersten Kontext folgt. Der Lesezeiger (RP) **412** ist auf „0“ gesetzt, und zeigt demnach auf die physikalische Anfangsadresse des FIFO-Arrays **402**. Die Wortzahl der FIFO-Ausnutzung, welche anzeigt, wie viele Wortpositionen des FIFO verwendet wurden, steht auf „7“. Der Sammler des Mikrocontrollers (MC) ist auf „14“ eingestellt. Der Übertragungszähler **406** des Pufferspeichers ist „nicht von Interesse“, da bei diesem Vorgang Daten nicht zu oder von dem Pufferspeicher **112** übertragen werden. Die Registerdatei **1502** enthält einen zweiten Kontext, der 28 Bytes umfaßt. Während in diesem Beispiel die ersten und zweiten Kontexte dieselbe Größe haben, könnten sie auch ebenso gut unterschiedliche Größen haben. Aus Gründen der Klarheit wird in dem vorliegenden Beispiel das FIFO-Array **402** mit aufeinanderfolgenden Zahlen 00-1B geladen, während die Registerdatei **1502** mit aufeinanderfolgenden Nummern bzw. Zahlen 20-3B geladen wird.

[0098] Ein Befehl für den MC-Austausch und das FIFO-Laden wird verwendet, um eine Kontextumschaltung bzw. einen Austausch durchzuführen. Wenn ein Befehl für den MC-Austausch und das Laden des FIFO ausgegeben wird, wird der Kontext, der in dem FIFO-Array **402** gespeichert ist, gegen den Kontext in der Registerdatei **1502** ausgetauscht. In dem dargestellten Beispiel wird der erste Kontext ausgegeben, wird jedoch ebenfalls nicht aus dem FIFO-Array **402** gelöscht. Falls gewünscht, kann jedoch der erste Kontext in dem FIFO-Array **402** während eines Austauschvorgangs überschrieben werden, während die ausgetauschte Kopie des ersten Kontextes in der Registerdatei **1502** gespeichert wird. Wenn beispielsweise der Kontext, der eingebracht wurde, größer war als die nicht-benutzten FIFO-Speicherstellen, kann der gesamte erste Kontext oder

ein Teil desselben, der in dem FIFO-Array **402** gespeichert ist, überschrieben werden. Demnach wird in einer Ausführungsform, wenn der Kontext, der aus der Registerdatei hereingeholt wird, 64 Bytes umfaßt, der erste Kontext in dem FIFO-Array **402** vollständig überschrieben.

[0099] Wie in [Fig. 15](#) dargestellt, wird in diesem Beispiel der von der Registerdatei **1502** in das FIFO-Array **402** übertragene Kontext unmittelbar hinter dem ersten Kontext gespeichert. Die Registerdatei **1502** enthält nunmehr den ersten Kontext. Der Schreibezeiger (WP) **410** ist auf 56 weiterbewegt worden, was den zusätzlichen Daten entspricht, die in das FIFO-Array **402** geschrieben wurden. Der Lesezeiger (RP) **412** ist auf 28 weitergegangen, was die Tatsache widerspiegelt, daß 28 Bytes aus dem Array **402** gelesen worden sind. Die FIFO-Wortzahl bleibt bei 7, da dieselbe Anzahl von Worten aus dem Array **402** gelesen wurden, wie während des Austauschs in das Array **402** hineingeschrieben wurden.

[0100] Eine Ausführungsform der vorliegenden Erfindung führt optional auch eine Datenübersetzung bzw. -verschiebung aus, wenn Daten zwischen dem MP-FIFO **302** und dem Pufferspeicher **112** verschoben werden. Wie in [Fig. 6](#) veranschaulicht, können zwei 16-Bit-Halbworte innerhalb eines 32-Bit-Wortes ausgetauscht werden und 2 Bytes innerhalb jedes Halbwortes können ausgetauscht werden. Bit 5 des Befehlsregisters **408** wird verwendet, um zu bewirken, daß die Halbworte innerhalb eines Wortes ausgetauscht werden. Bit 4 wird verwendet, um zu bewirken, daß Bytes innerhalb von Halbworten ausgetauscht werden. Diese Austauschfunktionen ermöglichen in vorteilhafter Weise, daß Daten, die in einem Format mit dem höchsten Wert am Ende (Big-Endian) in ein Format mit dem niedrigsten Wert am Ende (Little-Endian) umgewandelt werden. In ähnlicher Weise ermöglichen diese Austauschfunktionen in vorteilhafter Weise auch, daß Daten, die in dem Format mit niedrigwertigem Ende in ein Format mit höchstwertigem Ende umgewandelt werden. Beispielsweise verwendet ein Faserkanal üblicherweise das Format mit niedrigwertigem Ende, während die Prozessoren **106**, **120** möglicherweise das Format mit höchstwertigem Ende benutzen mögen. Zusätzlich ermöglicht die Austauschfunktion, daß Wörter, die in den FIFO **302** oder aus diesem heraus übertragen werden, an unterschiedlichen Rändern bzw. Grenzen neu ausgerichtet werden.

[0101] Zusätzliche Befehlsinformation einschließlich der Zugriffsrichtung auf den Pufferspeicher, der Übertragungszahlwerte, der Quelle der Pufferspeicheradressen und der Kommentare wird durch die nachstehende Tabelle 3 wiedergegeben.

| Befehl | Zugriffsrichtung des Pufferspeichers | Übertragungszahl | Pufferspeicheradresse | Kommentare |
|-------------------------------|--------------------------------------|------------------|-----------------------|--|
| Kein Betrieb | keine | 0 | N/A | |
| Lösche FIFO | keine | 0 | N/A | Löschen des Lese-/Schreibzeigers und Ausnutzungszählers des FIFO |
| Stoppe FIFO | keine | 0 | N/A | Zwingen der Zustandsmaschine in den Leerlaufzustand |
| Lade FIFO | Lesen | XFER CNT REG | BUFFER, XFER ADDR | |
| Entlade FIFO | Schreiben | XFER CNT REG | BUFFER XFER ADDR | |
| Lade FIFO mit BCRC | Lesen | XFER CNT REG | BUFFER XFER ADDR | Überprüfen des CRC beim Laden des FIFO |
| Entlade FIFO mit BCRC | Schreiben | XFER CNT REG | BUFFER XFER ADDR | Erzeugen von CRC beim Entladen des FIFO |
| Lese | Schreiben | 64 | BUFER XFER ADDR | |
| Schreibe | Schreiben | 64 | BUFFER XFER ADDR | |
| Befehl für Heranholen des FCP | Lesen | 64 | FCT FETCH ADDR PTR | Überprüfen von CRC |
| Aktualisieren der FCP-Zeiger | keine | 0 | N/A | Aktualisieren von FCP-Adresse und -Zahl und Interrupt nach Abschluß des Löschbefehls |

Tabelle 3

[0102] In einer Ausführungsform können eine oder mehrere der oben aufgelisteten Befehle ausgegeben werden, während ein weiterer Befehl ausgeführt wird. Beispielsweise werden in einer Ausführungsform die „kein Betrieb“ (NOP)-, „Lösche FIFO“- und „Stoppe FIFO“-Befehle ausgegeben, während andere Befehle ausgeführt werden.

[0103] Die Zustandsmaschine **418** steuert die Speicherzugriffsressourcen der Steuerung, einschließlich der MP-FIFO-Zähler, des Übertragungszählers **406** des MP-FIFO-Pufferspeichers, die Heranholadresszähler des Faserkanalprotokolls (FCP), die CRC-Sammler und den Zähler für die Framelänge. Die Zustandsmaschine **418** kann diese Ressourcen initialisieren, Betriebsarten einstellen und wirkt als der SDRAM-Anforderer.

[0104] Die [Fig. 5A–D](#) veranschaulichen eine Ausführungsform der MP-FIFO-Zustandsmaschine **418**. Beginnend in einem Leerlaufzustand **514** geht die Zustandsmaschine **418** nach Empfang eines Befehls für das Entladen des FIFOs in den Zustand **502** weiter, woraufhin der Vorgang des Entladens des FIFO **302** beginnt. Wenn der FIFO **302** nicht leer ist, geht die Zustandsmaschine **418** weiter zu Zustand **504**. Der FIFO **302** wird dann kontinuierlich entladen, bis nur ein Wort in dem FIFO-Array **402** verbleibt und die Übertragungszahl gleich 1 ist. Die Zustandsmaschine **418** geht dann weiter zu einem Zustand **506**, in welchem das letzte FIFO-Wort

entladen wird. Demnach wird, wenn sowohl das Bestätigungs (ACK)-Signal von dem MP-FIFO-Vermittler als auch das Datenbestätigungs (DACK)-Signal von der SDRAM-Steuerung aktiv sind, das letzte Datenelement in dem FIFO-Array **402** übertragen. Die Übertragungszahl wird auf null gesetzt und die Zustandsmaschine **418** kehrt in den Leerlaufzustand **514** zurück.

[0105] Wenn während des Leerlaufzustandes **514** ein Befehl des Entladens des FIFO mit CRC empfangen wird, so geht die Zustandsmaschine **418** weiter in einen Zustand **508**. Der Befehl für das Entladen des FIFOs mit CRC wird verwendet, um Daten von dem FIFO-Array **402** in den Pufferspeicher **112** zu schreiben, während ein zugehöriger CRC erzeugt wird. In einer Ausführungsform ist der CRC das letzte Wort, welches in den Pufferspeicher **112** geschrieben wird. Wenn das FIFO-Array **402** leer ist, verbleibt die Zustandsmaschine **418** in dem Zustand **508** des Entladens des FIFOs. Wenn jedoch das FIFO-Array **402** nicht leer ist, geht die Zustandsmaschine **418** weiter in einen Zustand **510** des Entladens des FIFO. Die CRC-Berechnung wird ausgeführt, während das FIFO-Array **402** entladen und in den SDRAM-Pufferspeicher geschrieben wird. Wenn die Übertragungszahl gleich 1 ist, geht die Zustandsmaschine **418** weiter in einen Zustand **512** des Entladens des FIFO. Wenn die CRC-Berechnung abgeschlossen ist, werden die CRC-Daten an den SDRAM-Pufferspeicher **112** übertragen und die Übertragungszahl wird auf 0 gesetzt. Die Zustandsmaschine **418** geht dann weiter zurück in den Leerlaufzustand **514**.

[0106] Wenn die Zustandsmaschine **418** einen Befehl für das Laden des FIFO empfängt, so geht die Zustandsmaschine **418** von dem Leerlaufzustand **514** in einen Zustand **516** des FIFO-Ladens. Der Befehl für das Laden des FIFO wird verwendet, um Daten aus dem Pufferspeicher in das FIFO-Array **402** zu laden. Wenn die Zustandsmaschine **418** nicht voll ist, so geht die Zustandsmaschine **418** weiter in Schritt **518**, und zwar nach Empfang eines MP-FIFO-Bestätigungssignals von dem Vermittler. Die Zustandsmaschine **418** bewirkt dann, daß Daten von dem Pufferspeicher **112** in das FIFO-Array **402** übertragen werden, bis die Übertragungszahl gleich 1 ist, und der MP-FIFO **302** erzeugt ein „fast voll“-Signal. Wenn die Übertragungszahl gleich 1 ist und das FIFO-Array **402** fast voll ist, was bedeutet, daß nur noch Platz für lediglich ein weiteres Datenwort ist, so geht die Zustandsmaschine **418** weiter in einen Zustand **520** des FIFO-Ladens. Wenn der Übertragungszähler gleich 1 ist und der MP-FIFO **302** ein „voll“-Signal erzeugt, so wird nach Empfang eines MP-FIFO-Bestätigungssignals von dem Vermittler und eines DACK-Signals von der SDRAM-Steuerung das letzte Wort in das FIFO-Array **402** geladen und die Zustandsmaschine **418** geht dann zurück in den Leerlaufzustand **514**.

[0107] Nach Empfang eines Befehls des FIFO-Ladens mit CRC geht die Zustandsmaschine **418** in einen Zustand **522** des FIFO-Ladens mit CRC. Der Befehl des FIFO-Ladens mit CRC wird verwendet, um Daten aus dem Pufferspeicher **112** zu lesen, während der CRC überprüft wird. In einer Ausführungsform ist der CRC das letzte zu übertragende Wort. In dem Zustand **522** wird, wenn der MP-FIFO **302** nicht voll ist, das MP-FIFO-Lesesignal aktiviert, ebenso wie auch ein MP-FIFO-CRC-Freigabesignal. Nach Empfang eines MP-FIFO-Bestätigungssignals geht die Zustandsmaschine **418** weiter in einen Zustand **524** des FIFO-Ladens mit CRC. Die Zustandsmaschine **418** fährt fort, um Daten aus dem SDRAM-Pufferspeicher **112** in das FIFO-Array **402** zu laden, bis ein „fast voll“-Signal ausgegeben wird und die Übertragungszahl größer als 1 ist. Während dieser Zeitdauer wird der CRC berechnet. Die Zustandsmaschine **418** geht dann weiter in einen Zustand **526** für das FIFO-Laden mit CRC. Die Zustandsmaschine **418** bewirkt dann, daß der CRC ist das FIFO-Array **402** geladen wird. Die Übertragungszahl wird auf 1 gesetzt und die Zustandsmaschine **418** geht zurück in den Leerlaufzustand **514**.

[0108] Nach Empfang eines Befehls des Schreibens in den Pufferspeicher geht die Zustandsmaschine **418** aus dem Leerlaufzustand **514** in den Schreibzustand **528**. Wie zuvor beschrieben, wird der Befehl für das Schreiben in den Pufferspeicher verwendet, um Daten aus dem FIFO-Array **402** in den Pufferspeicher **112** zu schreiben. Wenn das FIFO-Array **402** nicht leer ist, geht die Zustandsmaschine **418** weiter in einen Zustand **530**, nach Empfang eines Bestätigungssignals von dem MP-FIFO-Vermittler. Das FIFO-Array **402** wird dann entladen und die entladenen Daten werden in den Pufferspeicher **112** geschrieben. Der FIFO-Entladevorgang wird fortgesetzt, bis nur ein Wort in dem FIFO-Speicher verbleibt und die Übertragungszahl gleich 1 ist. Die Zustandsmaschine **418** geht dann weiter in einen Zustand **532**, in welchem das letzte FIFO-Wort entladen wird. Wenn dann sowohl das Bestätigungs (ACK)-Signal von dem MP-FIFO-Vermittler als auch das Datenbestätigungs (DACK)-Signal von der SDRAM-Steuerung aktiv sind, wird das letzte Datenelement in dem FIFO übertragen. Die Übertragungszahl wird auf 0 gesetzt und die Zustandsmaschine **418** kehrt in den Leerlaufzustand **514** zurück.

[0109] Nach Empfang eines Befehls für das Aktualisieren der FCP-Befehle geht die Zustandsmaschine **418** aus dem Leerlaufzustand **514** in den Zustand **528** für FCMD-PTR. Ein Befehlssignal für das Zeigerheraufsetzen wird für einen Takt aktiviert, was die FCP-Adresse und -Zahl aktualisiert. Die Zustandsmaschine **418** geht

dann weiter zum Zustand **536** und löscht den Interrupt für den Befehlsabschluß. Die Zustandsmaschine **418** kehrt dann in den Leerlaufzustand **514** zurück.

[0110] Nach Empfang eines Befehls für das Lesen des Pufferspeichers geht die Zustandsmaschine **418** aus dem Leerlaufzustand **514** in einen Lesezustand **538** und gibt ein SDRAM-Lesesignal aus. Wenn der FIFO nicht voll ist, so geht die Zustandsmaschine **418** weiter in einen Zustand **540**, und zwar nach Empfang eines Bestätigungssignals von dem MP-FIFO-Vermittler. Die Daten werden dann aus dem Pufferspeicher **112** in das FIFO-Array **420** geladen. Der Leseprozeß wird fortgesetzt, bis der FIFO-Speicher fast voll ist und die Übertragungszahl gleich 1 ist. Die Zustandsmaschine **418** geht dann weiter in einen Zustand **542**, in welchem nur noch ein weiteres Wort aus dem Pufferspeicher **112** in den FIFO gelesen wird. Wenn dann sowohl das Bestätigungs (ACK)-Signal von dem MP-FIFO-Vermittler als auch das Datenbestätigungs (DACK)-Signal aus der SDRAM-Steuerung aktiv sind, so wird die Übertragungszahl auf 0 gesetzt und die Zustandsmaschine **418** kehrt in den Leerlaufzustand **514** zurück.

[0111] Nach Empfang eines Befehls für das FCP-Heranziehen geht die Zustandsmaschine **418** aus dem Leerlaufzustand **514** in einen FCP-Befehlszustand **544**. Die Zustandsmaschine setzt in geeigneter Weise die FCP_CMD, RD_SIG, LD_CMD_LEN_ADR, CLR_FIFO, LD_XFER_CNT_64, und REQ_SIG-Signale, um das Erhalten der Paketlängendaten, die am Ende des Paketes angeordnet sind, vorzubereiten. Demnach wird die Übertragungsadresse mit der Framelängenadresse geladen. In einer Ausführungsform enthält die Framelängenadresse eine Basisadresse, die zu einer Segmentgröße hinzugefügt ist. Weitergehend in einen Zustand **546** werden die Framelängendaten gewonnen. Wenn sowohl das Bestätigungs (ACK)-Signal von dem MP-FIFO-Vermittler als auch das Datenbestätigungs (DACK)-Signal von der SDRAM-Steuerung aktiv sind, wird die Framelänge in dem FIFO-Array **402** eingeordnet. Unter Fortschreiten in einen Zustand **548** wird der CRC-Keim zurückgesetzt und die Übertragungsadresse wird auf die Startadresse oder Basisadresse des FCP-Befehlsframes eingestellt. Wenn das Bestätigungssignal abfällt, geht die Zustandsmaschine weiter in einen Zustand **550**. Die Übertragung des Paketes aus dem Pufferspeicher **112** beginnt und die CRC-Sammlung beginnt. Die Übertragung wird fortgesetzt, bis die Übertragungszahl gleich 1 ist. Dann geht die Zustandsmaschine **418** weiter in einen Zustand **552** und das letzte Wort wird übertragen. Der Übertragungszähler ist nunmehr gleich 0 und die Zustandsmaschine geht in den Leerlaufzustand **514**.

[0112] Während gewisse bevorzugte Ausführungsformen der Erfindung beschrieben worden sind, sind diese Ausführungsformen nur als Beispiel präsentiert worden und sie sollen nicht den Schutzzumfang der Erfindung einschränken. Dementsprechend sollte der Geist und Schutzzumfang der vorliegenden Erfindung nur entsprechend den folgenden Ansprüchen definiert werden.

Patentansprüche

1. Steuerung (Controller) für eine Massenspeichereinrichtung um eine Schnittstelle zwischen einer Massenspeichereinrichtung (**106**) über einen Bus (**116**) zu einem Computer (**104**) zu bilden, wobei die Steuerung (**102**) der Massenspeichereinrichtung einen Pufferspeicher (**112**) aufweist, der so ausgestaltet ist, daß er Daten von der Massenspeichereinrichtung empfängt und Daten von dem Bus empfängt, wobei die Steuerung der Massenspeichereinrichtung aufweist:

zumindest einen Prozessor (**108**, **120**), und

einen Speicherschaltkreis (**302**), der mit dem Pufferspeicher und mit dem Prozessor verbunden ist, wobei der Speicherschaltkreis so ausgestaltet ist, daß er während zumindest einer ersten Übertragung von Daten zwischen dem Speicherschaltkreis und dem Pufferspeicher als ein First-In-First-Out-Speicher (FIFO) fungiert, wobei der Speicherschaltkreis weiterhin so ausgelegt ist, daß er während zumindest einer zweiten Übertragung von Daten zwischen dem Speicherschaltkreis und dem zumindest einen Prozessor als ein Speicher mit wahlfreiem Zugriff (RAM) arbeitet.

2. Steuerung für eine Massenspeichereinrichtung nach Anspruch 1, wobei zumindest eine der Übertragungen, nämlich der ersten Übertragung oder der zweiten Übertragung, angehalten wird, wenn der Speicherschaltkreis (**302**) voll ist.

3. Steuerung für eine Massenspeichereinrichtung nach Anspruch 1, wobei der Speicherschaltkreis (**302**) so ausgestaltet ist, daß er Daten schubweise (als Burst) in den Pufferspeicher (**112**) verschiebt.

4. Steuerung für eine Massenspeichereinrichtung nach Anspruch 1, wobei der Speicherschaltkreis (**302**) automatisch neu geladen wird, während Daten schub- bzw. stoßweise aus dem Speicherschaltkreis in den Pufferspeicher (**112**) verschoben werden.

5. Steuerung einer Massenspeichereinrichtung nach Anspruch 1, wobei der Speicherschaltkreis (302) mit einer Mehrzahl von Registerdateien einer Plattenlaufwerksteuerung verbunden ist.
6. Steuerung für eine Massenspeichereinrichtung nach Anspruch 1, wobei der Speicherschaltkreis (302) eine Speicherkapazität hat und so ausgelegt ist, daß er einen Übertragungsbefehl akzeptiert, der eine Übertragungsgröße angibt, die größer ist als die Speicherkapazität des Speicherschaltkreises.
7. Steuerung für eine Massenspeichereinrichtung nach Anspruch 1, wobei der Bus (116) eine I/O-Schnittstelle ist und wobei die I/O-Schnittstelle eine Schnittstelle zu einem Faserkanal ist.
8. Steuerung für eine Massenspeichereinrichtung nach Anspruch 1, welche weiterhin eine Zustandsmaschine bzw. einen endlichen Automaten (418) aufweist, der mit dem zumindest einen Prozessor (108, 120), mit dem Pufferspeicher (112) und mit dem Speicherschaltkreis (302) verbunden ist, wobei die Zustandsmaschine bzw. der endliche Automat so ausgestaltet ist, daß er bewirkt, daß ein Steuerblock des Faserkanals von dem Pufferspeicher in den Speicherschaltkreis übertragen wird, und daß die Überprüfung eines Fehlererfassungs-codes veranlaßt wird.
9. Steuerung für eine Massenspeichereinrichtung nach Anspruch 1, wobei die Massenspeichereinrichtung (106) ein magnetisches Plattenlaufwerk ist.
10. Steuerung für eine Massenspeichereinrichtung nach Anspruch 1, wobei der Bus (116) eine I/O-Schnittstelle aufweist, die Steuerung (102) für die Massenspeichereinrichtung weiterhin eine Schnittstelle (136) eines Plattenlaufwerks aufweist, die mit einem Plattenlaufwerk (106) verbunden ist, und eine I/O-Schnittstelle (124) aufweist, die mit dem Bus verbunden ist, der Pufferspeicher (112) so angeschlossen ist, daß er Daten zu der Plattenlaufwerkschnittstelle und von dieser sowie zu und von der I/O-Schnittstelle überträgt, und die Steuerung der Massenspeichereinrichtung weiterhin dadurch gekennzeichnet ist, daß:
 - der zumindest eine Prozessor (108, 120) aufweist:
 - einen ersten Prozessor (108), der verwendet wird, um zumindest die Plattenlaufwerkschnittstelle zu steuern, und
 - einen zweiten Prozessor (120), der verwendet wird, um zumindest Aufgaben auf niedriger Ebene auszuführen, und daß
 - der Speicherschaltkreis (302) aufweist:
 - einen DMA-Anschluß (210), der mit dem Pufferspeicher verbunden ist,
 - einen Registerzugangsanschluß (408), der mit dem ersten Prozessor und dem zweiten Prozessor verbunden ist,
 - einen wahlfreien Zugriffsanschluß (404), der mit dem ersten Prozessor und dem zweiten Prozessor verbunden ist, und
 - einen Lesezeiger (412) und einen Schreibzeiger (410), auf welche durch zumindest einen der beiden, nämlich den ersten Prozessor oder den zweiten Prozessor, zugegriffen werden kann.
11. Steuerung einer Massenspeichereinrichtung nach Anspruch 10, welche weiterhin einen Speichernutzungszähler (414) aufweist.
12. Steuerung einer Massenspeichereinrichtung nach Anspruch 10, wobei der Lesezeiger (412) und der Schreibzeiger (410) entweder durch den ersten Prozessor (108) oder durch den zweiten Prozessor (120) geladen werden können.
13. Steuerung einer Massenspeichereinrichtung nach Anspruch 10, welche weiterhin ein Fehlererfassungsmodul (424) aufweist, welches einen zyklischen Redundanzcode (CRC) für in den Pufferspeicher (112) geschriebene Daten verifiziert.
14. Steuerung einer Massenspeichereinrichtung nach Anspruch 1, wobei der Speicherschaltkreis (302) unter Verwendung eines direkten Speicherzugriffs ladbar ist.
15. Steuerung einer Massenspeichereinrichtung nach Anspruch 14, welche weiterhin eine Faserkanalschnittstelle (116) aufweist, die mit dem Pufferspeicher (112) verbunden ist, wobei Daten zwischen dem Pufferspeicher und der Schnittstelle des Faserkanals über den Speicherschaltkreis (302) übertragen werden.

16. Steuerung einer Massenspeichereinrichtung nach Anspruch 14, welche weiterhin einen Steuerschaltkreis (**122**) aufweist, der mit dem Pufferspeicher (**112**) und dem Speicherschaltkreis (**302**) verbunden ist, wobei der Steuerschaltkreis so ausgestaltet ist, daß er einen Steuer- bzw. Kontrollblock des Faserkanals von dem Pufferspeicher zu dem Speicherschaltkreis überträgt.

17. Steuerung einer Massenspeichereinrichtung nach Anspruch 14, wobei der Speicherschaltkreis (**302**) so ausgestaltet ist, daß er einen direkten Frame-Zugriff auf zumindest einen ersten Frame durch den zumindest einen Prozessor (**108**) vorsieht.

18. Steuerung einer Massenspeichereinrichtung nach Anspruch 14, wobei der Speicherschaltkreis (**302**) so bemessen ist, daß er einen Steuerblock des Faserkanals und einen zugehörigen Fehlererfassungscode hält.

19. Steuerung einer Massenspeichereinrichtung nach Anspruch 14, wobei der Speicherschaltkreis zumindest vier Anschlüsse (Ports) hat.

20. Steuerung einer Massenspeichereinrichtung nach Anspruch 14, wobei der Speicherschaltkreis (**302**) eine Registerzugangsschnittstelle (**408**) und eine Schnittstelle (**404**) für direkten Zugriff aufweist.

21. Steuerung einer Massenspeichereinrichtung nach Anspruch 14, welche weiterhin ein Fehlererfassungsmodul (**424**) aufweist, welches einen zyklischen Redundanzcode (CRC) für Daten überprüft, die von dem Pufferspeicher (**112**) an den Speicherschaltkreis (**302**) übertragen werden.

22. Steuerung einer Massenspeichereinrichtung nach Anspruch 14, wobei der Speicherschaltkreis (**302**) weiterhin einen Schreibzeiger (**410**) und einen Lesezeiger (**412**) aufweist, die durch den zumindest einen Prozessor (**108**, **120**) geladen und gelesen werden können.

23. Steuerung einer Massenspeichereinrichtung nach Anspruch 14, welche weiterhin ein Adreßregister (**404**) des Pufferspeichers aufweist.

24. Steuerung einer Massenspeichereinrichtung nach Anspruch 14, wobei der Speicherschaltkreis (**302**) so ausgestaltet ist, daß er zumindest Prozessorprogrammcode aufnimmt.

25. Steuerung einer Massenspeichereinrichtung nach Anspruch 14, wobei der Speicherschaltkreis (**302**) weiterhin ein Register (**420**, **422**) für das Heranholen von Adressen aufweist.

26. Steuerung für eine Massenspeichereinrichtung nach Anspruch 14, wobei:
der Pufferspeicher (**112**) als ein Speicher mit wahlfreiem Zugriff (RAM) ausgestaltet ist und aufweist:
eine Faserkanalschnittstelle (**116**), die mit dem Pufferspeicher verbunden ist, wobei der Pufferspeicherschaltkreis so ausgestaltet ist, daß er einen Faserkanalsteuerblock von der Faserkanalschnittstelle empfängt, und
eine Zustandsmaschine (**418**), die mit dem zumindest einen Prozessor (**108**, **120**), dem Speicherschaltkreis (**302**) und dem Pufferschaltkreis verbunden ist, wobei die Zustandsmaschine so ausgestaltet ist, daß sie einen Befehl von dem zumindest einen Prozessor empfängt und den Faserkanalsteuerblock zwischen dem Pufferspeicher und dem Speicherschaltkreis in Reaktion auf den Befehl überträgt.

27. Steuerung für eine Massenspeichereinrichtung nach Anspruch 14, wobei der Pufferspeicher (**112**) so ausgestaltet ist, daß er einen direkten Frame-Zugriff auf den Frame eines Faserkanals bereitstellt.

28. Steuerung für eine Massenspeichereinrichtung nach Anspruch 1, wobei:
die Steuerung (**102**) der Massenspeichereinrichtung über den Bus (**106**) mit einem Hostcomputer (**104**) verbunden ist,
der Speicherschaltkreis (**302**) so ausgestaltet ist, daß er während zumindest einer ersten Datenübertragung zwischen dem Speicherschaltkreis und dem Pufferspeicher (**112**) als ein First-In-First-Out-Speicher (FIFO) arbeitet, und
der Speicherschaltkreis so ausgelegt ist, daß er während einer zweiten Datenübertragung zwischen dem Speicherschaltkreis und dem zumindest einen Prozessor (**108**) als ein Speicher mit wahlfreiem Zugriff (RAM) arbeitet.

29. Steuerung eines Massenspeichers nach Anspruch 28, welche weiterhin einen Mikrocontroller (**120**) aufweist, der mit dem Speicherschaltkreis (**302**) verbunden ist, wobei der Speicherschaltkreis zumindest zwei

Anschlüsse (Ports) aufweist, um es dem Mikrocontroller und dem zumindest einen Prozessor (**108**) zu ermöglichen, gleichzeitig auf den Speicherschaltkreis zuzugreifen.

30. Steuerung einer Massenspeichereinrichtung nach Anspruch 1, wobei der Pufferspeicher (**112**) so ausgestaltet ist, daß er Kontextinformation hält, der zumindest eine Prozessor (**108**, **120**) einen Prozessor (**108**) einer ersten Festplattensteuerung aufweist, der mit dem Pufferspeicher verbunden ist, wobei der Prozessor der ersten Festplattensteuerung so ausgestaltet ist, daß er Operationen entsprechend der in dem Pufferspeicher gespeicherten Kontextinformation ausführt, wobei der Speicherschaltkreis (**302**) in einer ersten Betriebsart als ein Speicher mit wahlfreiem Zugriff (RAM) arbeitet, und in einer zweiten Betriebsart als ein First-In-First-Out-Speicher (FIFO) arbeitet, wobei der Speicherschaltkreis verwendbar ist, um sowohl wechselnde Kontextinformation als auch Pufferfestplattendaten zu speichern, und weiterhin aufweist einen Austauschschaltkreis, der dafür ausgelegt ist, zumindest einen Teil der Kontextinformation, welche in dem Pufferspeicher abgespeichert ist, mit der wechselnden bzw. alternativen Kontextinformation auszutauschen, die in dem Speicherschaltkreis gespeichert ist, und zwar in Reaktion auf zumindest ein erstes Ereignis.

31. Steuerung für eine Massenspeichereinrichtung nach Anspruch 1, wobei: die Steuerung (**102**) für die Massenspeichereinrichtung mit einem Festplattenlaufwerk (**106**) und einem Hostcomputer (**104**) verbunden ist, die Steuerung der Massenspeichereinrichtung Daten verwaltet bzw. handhabt, welche zwischen dem Festplattenlaufwerk und dem Hostcomputer übertragen werden, und wobei der Pufferspeicher (**112**) so ausgestaltet ist, daß er Kontextinformation hält, der zumindest eine Prozessor (**108**, **120**) ein erster Festplattensteuerungsprozessor (**108**) ist, welcher mit dem Pufferspeicher verbunden ist und welcher so ausgestaltet ist, daß er Operationen entsprechend der Kontextinformation, die in dem Pufferspeicher gespeichert sind, ausführt, der Speicherschaltkreis (**302**) wechselnde bzw. alternative Kontextinformation in einer ersten Betriebsart als ein Speicher mit wahlfreiem Zugriff (RAM) speichert und in einer zweiten Betriebsart Festplattendaten als First-In-First-Out-Speicher speichert, und ein Austauschschaltkreis so ausgeschaltet ist, daß er zumindest einen Teil der in dem Pufferspeicher gespeicherten Kontextinformation gegen die alternative Kontextinformation austauscht, die in dem Speicherschaltkreis gespeichert ist, und zwar in Reaktion auf zumindest ein erstes Ereignis.

32. Verfahren zum Zugreifen auf und zum Speichern von Daten in der Steuerung (**102**) einer Massenspeichereinrichtung, wobei die Steuerung einen Pufferspeicher (**112**), zumindest einen Prozessor (**108**, **120**) und einen Speicherschaltkreis (**302**) hat, wobei das Verfahren die Schritte aufweist: Empfangen von Daten aus der Massenspeichereinrichtung und von einem Bus (**116**) bei dem Pufferspeicher, Übertragen von Daten zwischen dem Speicherschaltkreis und dem Pufferspeicher in einem First-In-First-Out-Betrieb, und Übertragen von Daten zwischen dem Speicherschaltkreis und dem zumindest einen Prozessor in einem Betrieb mit wahlfreiem Zugriff.

33. Verfahren zum Zugreifen auf und Speichern von Daten nach Anspruch 32, wobei der Speicherschaltkreis (**302**) ein Speicherschaltkreis mit mehreren Anschlüssen ist, der auf der Steuerung der Massenspeichereinrichtung angeordnet ist, und wobei das Verfahren die Schritte aufweist: Übertragen von Daten aus dem Pufferspeicher auf den Speicherschaltkreis (**302**) mit mehreren Anschlüssen und Verwendung eines ersten Anschlusses des Speichers mit mehreren Anschlüssen, wobei der erste Anschluß ein First-In-First-Out-Anschluß ist und wobei der Speicher mit mehreren Anschlüssen so konfigurierbar ist, daß er in einem Speicherbetrieb nach dem First-In-First-Out-Prinzip arbeitet und so, daß er in einem Speicherbetrieb mit wahlfreiem Zugriff (RAM-Betrieb) arbeitet, Übertragen von Daten zwischen dem Speicher mit mehreren Anschlüssen und einem ersten Prozessor (**108**) einer Festplattensteuerung unter Verwendung eines zweiten Anschlusses des Speichers mit mehreren Anschlüssen, wobei der zweite Prozessor der Festplattensteuerung so ausgestaltet ist, daß er Aufgaben auf hohem Niveau bzw. auf höherer Ebene ausführt, und Übertragen von Daten zwischen dem Speicher mit mehreren Anschlüssen und einem zweiten Prozessor (**120**) der Festplattensteuerung unter Verwendung eines dritten Anschlusses des Speichers mit mehreren Anschlüssen, wobei der zweite Prozessor der Festplattensteuerung so ausgestaltet ist, daß er Aufgaben auf niedriger Ebene ausführt.

34. Verfahren zum Zugreifen auf und zum Speichern von Daten nach Anspruch 32, welches weiterhin das Modifizieren der in dem Speicher (**302**) mit mehreren Anschlüssen gespeicherten Daten aufweist.

35. Verfahren zum Zugreifen auf und zum Speichern von Daten nach Anspruch 32, welches weiterhin das Lesen der modifizierten Daten aus dem Speicher (302) mit mehreren Anschlüssen aufweist.
36. Verfahren zum Zugreifen auf und zum Speichern von Daten nach Anspruch 32, welches weiterhin das stoßweise Verschieben von Daten (in einem Burst) von dem Speicher (302) mit mehreren Anschlüssen in den Pufferspeicher (112) aufweist.
37. Verfahren zum Zugreifen auf und zum Speichern von Daten nach Anspruch 32, welches weiterhin das Übertragen der Daten von dem Pufferspeicher (112) in den Speicher (302) mit mehreren Anschlüssen aufweist, während ein zu den Daten gehöriger Fehlerertassungscode überprüft wird.
38. Verfahren zum Zugreifen auf und zum Speichern von Daten nach Anspruch 32, welches weiterhin aufweist:
Speichern einer Festplattenformattabelle in dem Speicher (302) mit mehreren Anschlüssen, und
Zugreifen auf die gespeicherte Festplattenformattabelle unter Verwendung des zweiten Anschlusses des Speichers mit mehreren Anschlüssen.
39. Verfahren zum Zugreifen auf und zum Speichern von Daten nach Anspruch 32, welches weiterhin das Zugreifen auf Programmcode aufweist, der in dem Speicher (302) mit mehreren Anschlüssen angeordnet ist.
40. Verfahren zum Zugreifen auf und zum Speichern von Daten nach Anspruch 32, welches weiterhin das Bereitstellen eines direkten Frame-Zugriffs auf einen Faserkanalframe aufweist, der in dem Speicher (302) mit mehreren Anschlüssen gespeichert ist.
41. Verfahren zum Zugreifen auf und zum Speichern von Daten nach Anspruch 32, welches weiterhin die Schritte aufweist:
Übertragen eines Pufferlängenfeldes, welches zu einem Datenpaket aus dem Pufferspeicher (112) gehört, an eine erste Stelle des Speichers (302) mit mehreren Anschlüssen über den First-In-First-Out-Anschluß des Speichers mit mehreren Anschlüssen, wobei zumindest ein Teil des Pufferlängenfeldes Information enthält, die sich auf die Menge von Daten bezieht, die über das Datenpaket übertragen werden,
Übertragen zumindest eines Teils des Datenpakets aus dem Pufferspeicher an zumindest eine zweite Stelle des Speichers mit mehreren Anschlüssen, und
Lesen des Pufferlängenfeldes aus dem Speicher mit mehreren Anschlüssen und dann Lesen des zumindest eines Teils des Datenpakets aus dem Speicher mit mehreren Anschlüssen.
42. Verfahren zum Zugreifen auf und zum Speichern von Daten nach Anspruch 41, wobei zumindest ein Abschnitt bzw. Teil des Pufferlängenfeldes verwendet wird, um die Position eines CRC in dem Datenpaket zu lokalisieren.
43. Verfahren zum Zugreifen auf und zum Speichern von Daten nach Anspruch 32, welches die Schritte aufweist:
Verwenden des Speichers (302) mit mehreren Anschlüssen in einer ersten Betriebsart unter der Verwaltung des ersten Prozessors (108) der Festplattensteuerung, um Daten, welche zwischen einer Festplattenspeichereinrichtung (106), die mit der Festplattensteuerung (102) verbunden ist, und einem Bus (116), der mit der Steuerung der Massenspeichereinrichtung verbunden ist, temporär zu speichern,
Speichern eines ersten Kontextes in dem Pufferspeicher (112),
Verwenden des Speichers mit mehreren Anschlüssen in einer zweiten Betriebsart, um einen zweiten Kontext für eine spätere Verwendung durch den ersten Prozessor der Festplattensteuerung zu speichern,
Austauschen zumindest eines Teils der ersten Kontextinformation, die in dem Pufferspeicher gespeichert ist, gegen zumindest einen Teil der zweiten Kontextinformation, die in dem Speicher mit mehreren Anschlüssen gespeichert ist, und zwar zumindest teilweise und in Reaktion auf ein erstes Ereignis, und
erneutes Speichern des ausgetauschten, zumindest einen Abschnittes der ersten Kontextinformation zurück in den Pufferspeicher, und zwar zumindest teilweise und in Reaktion auf ein zweites Ereignis.
44. Verfahren zum Zugreifen auf und zum Speichern von Daten nach Anspruch 32, welches die Schritte aufweist:
Speichern eines ersten Kontextes in dem Pufferspeicher (112),
Speichern eines zweiten Kontextes in dem Speicher (302) mit mehreren Anschlüssen,
Verwenden des ersten Prozessors (108) der Festplattensteuerung, um einen ersten Übertragungsvorgang, der sich auf den ersten Kontext bezieht, zu verwalten bzw. zu bewirken, wobei der erste Übertragungsvorgang Da-

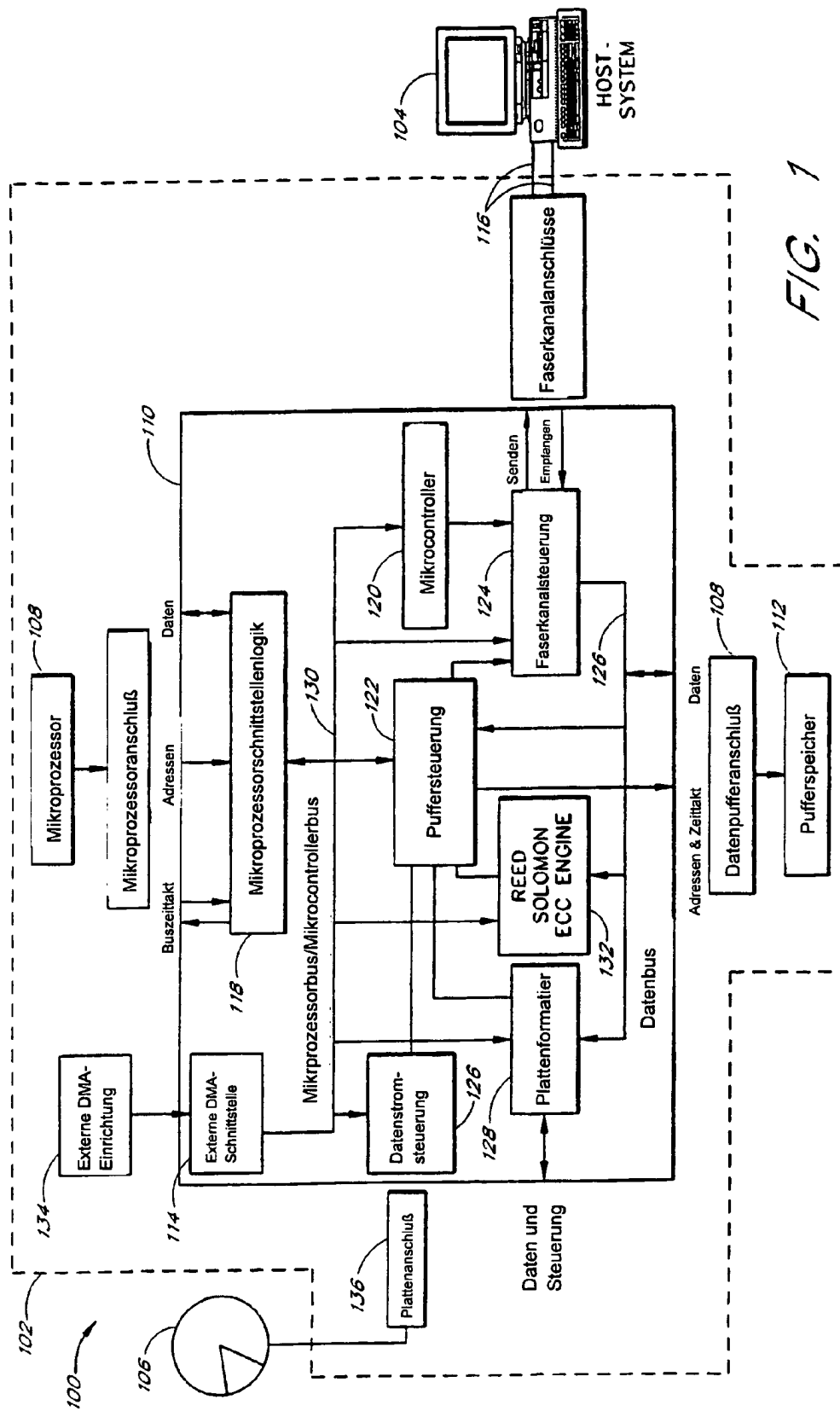
ten von einem Festplattenlaufwerk an eine Kommunikationsverbindung bzw. ein Kommunikationsglied überträgt,
 Feststellen, daß das Festplattenlaufwerk (**106**) keine Daten an die Kommunikationsverbindung (**116**) überträgt, sobald die Kommunikationsverbindung in der Lage ist, Daten aufzunehmen,
 Aussetzen des ersten Übertragungsvorganges,
 Austauschen des ersten Kontextes aus dem Pufferspeicher durch den zweiten Kontext, der in dem Speicher mit mehreren Anschlüssen gespeichert ist,
 Verwenden des ersten Prozessors der Festplattensteuerung, um einen zweiten Übertragungsvorgang, welcher dem zweiten Kontext entspricht, zu verwalten bzw. zu bewirken,
 erneutes Speichern des ersten Kontextes in den Pufferspeicher aus dem Speicher mit mehreren Anschlüssen, nachdem zumindest ein Teil des zweiten Übertragungsvorganges abgeschlossen ist,
 Abschließen bzw. Beenden des ersten Übertragungsvorganges, und
 Betreiben des Speichers mit mehreren Anschlüssen in der Speicherbetriebsart First-In-First-Out, um Festplattendaten zwischen dem Speicher mit mehreren Anschlüssen und dem Pufferspeicher zu übertragen.

45. Computersystem, welches eine Steuerung einer Massenspeichereinrichtung nach einem der Ansprüche 1 bis 31 aufweist.

46. Festplattenlaufwerkssystem, welches aufweist:
 ein Festplattenlaufwerk, und
 eine Steuerung, die mit dem Festplattenlaufwerk verbunden ist, um das Festplattenlaufwerk zu steuern und um Daten zu verwalten, die zwischen dem Festplattenlaufwerk und einem Hostcomputer übertragen werden, wobei die Steuerung aufweist:
 einen ersten Speicher, der so ausgestaltet ist, daß er Kontextinformation hält,
 einen ersten Prozessor der Festplattensteuerung, welcher dafür ausgelegt ist, Vorgänge entsprechend der in dem ersten Speicher gespeicherten Kontextinformation auszuführen, wobei der erste Prozessor der Festplattensteuerung mit dem ersten Speicher verbunden ist,
 einen zweiten Speicher, der so ausgestaltet ist, daß er in einer ersten Betriebsart als ein Speicher mit wahlfreiem Zugriff und in einer zweiten Betriebsart als ein Speicher nach dem First-In-First-Out-Prinzip arbeitet, wobei der zweite Speicher verwendbar ist, um sowohl alternative Kontextinformation zu speichern als auch um Festplattendaten zu puffern, wobei der zweite Speicher mit dem ersten Speicher verbunden ist, und
 einen Austauschschaltkreis, der so ausgelegt ist, daß er zumindest einen Teil der Kontextinformation, welche in dem ersten Speicher gespeichert ist, durch alternative Kontextinformation, welche in dem zweiten Speicher gespeichert ist, austauscht, und zwar in Reaktion auf zumindest ein erstes Ereignis.

Es folgen 18 Blatt Zeichnungen

Anhängende Zeichnungen



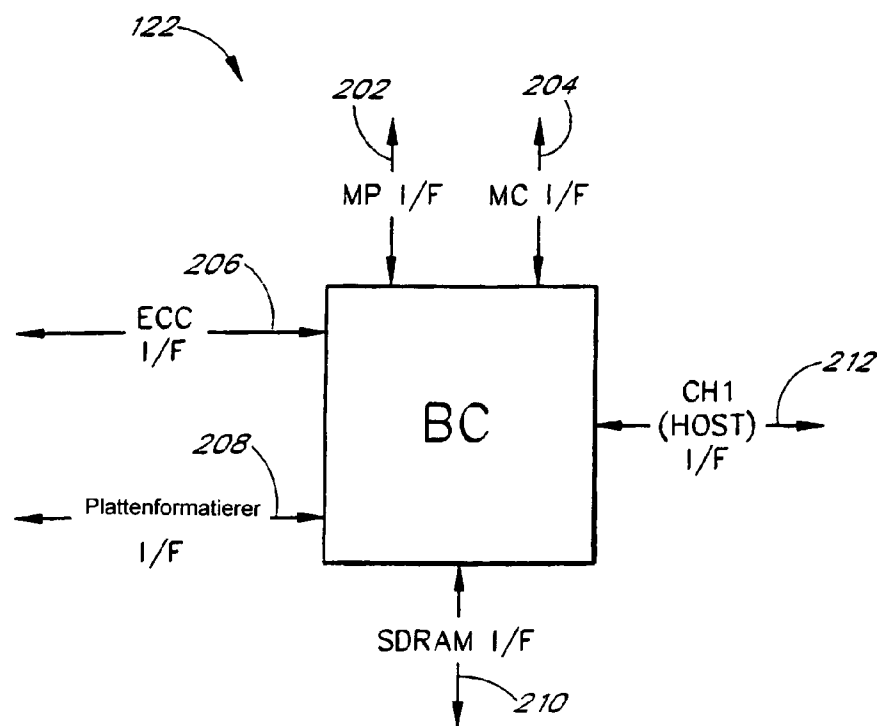


FIG. 2

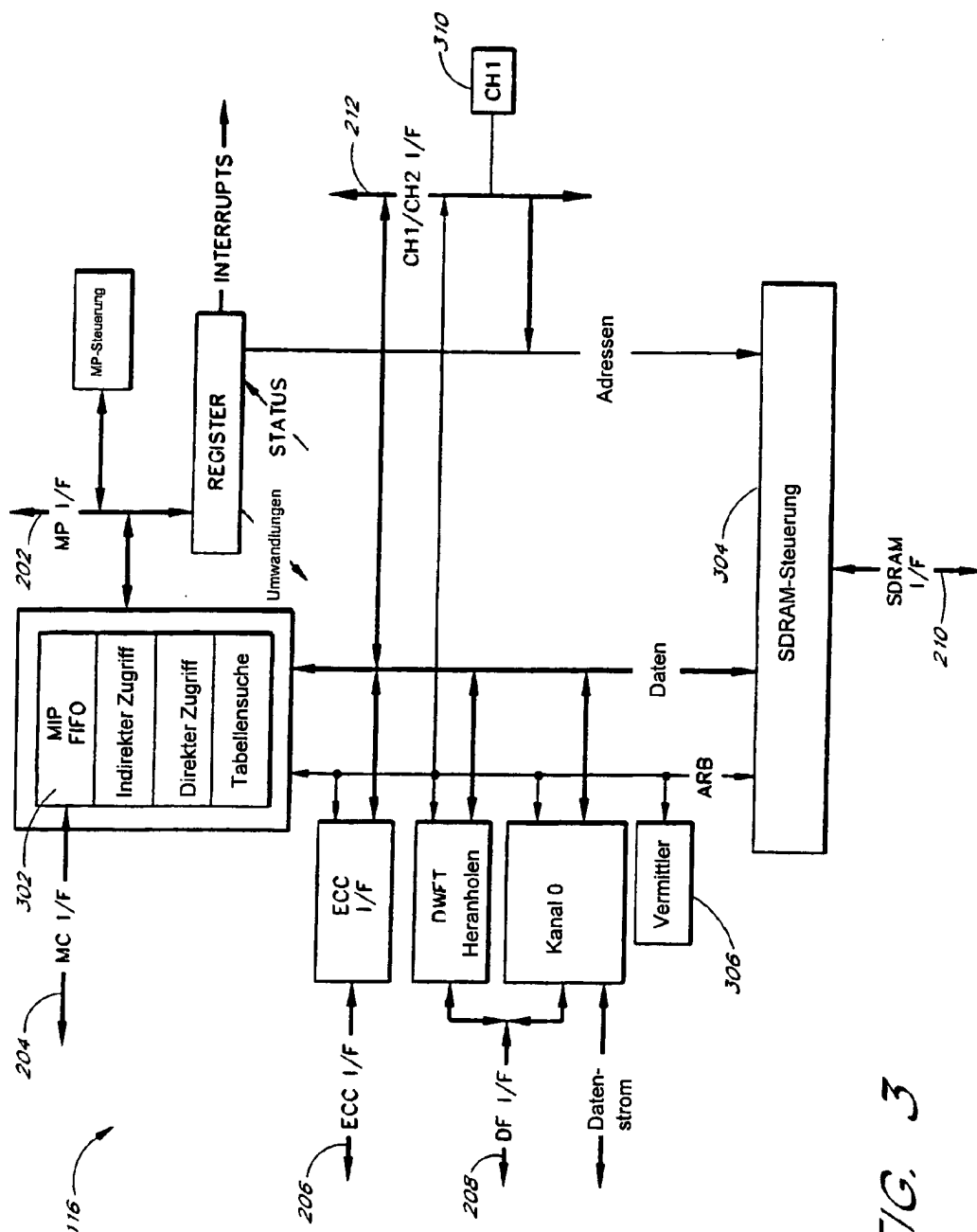


FIG. 3

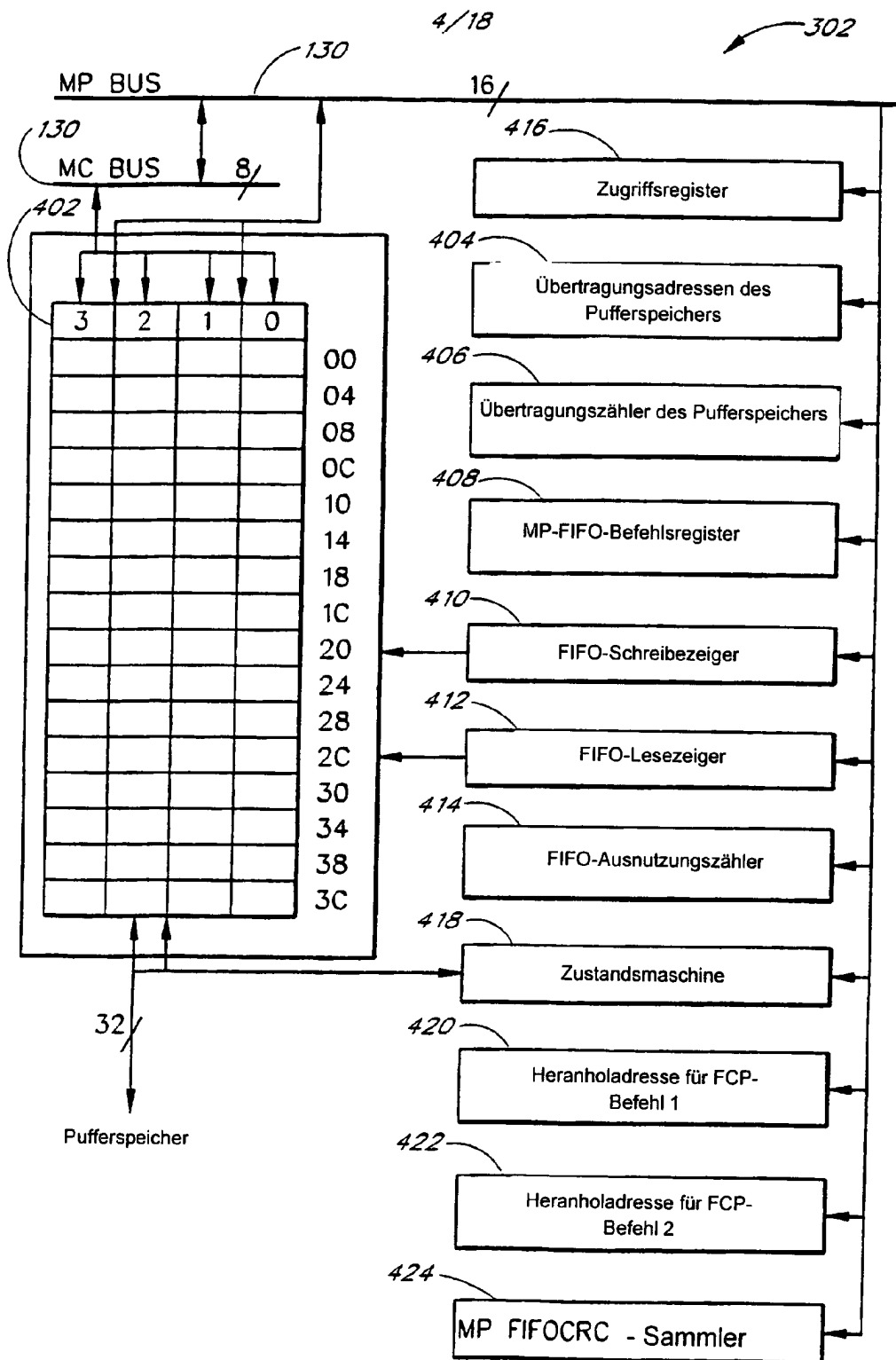


FIG. 4

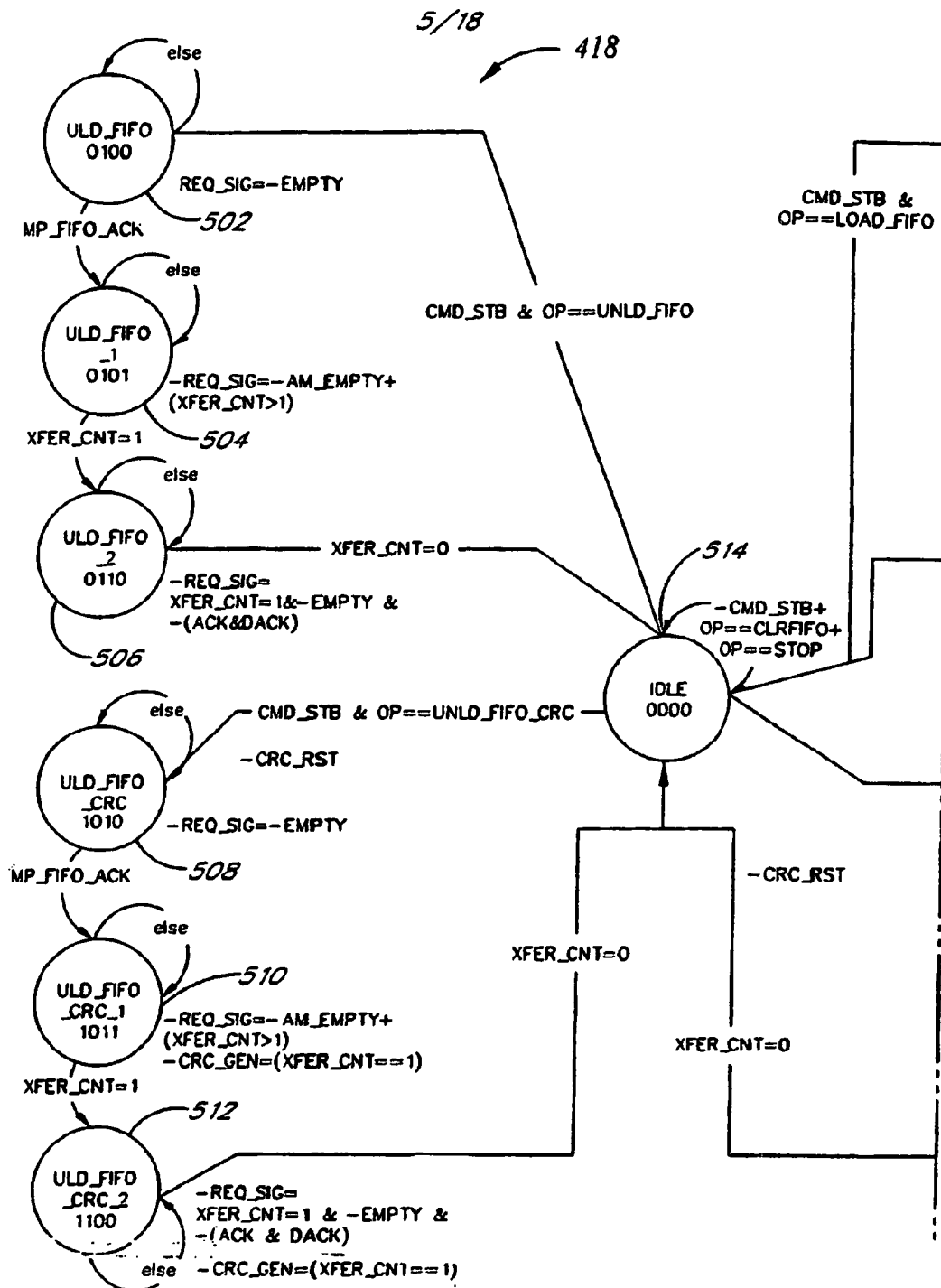
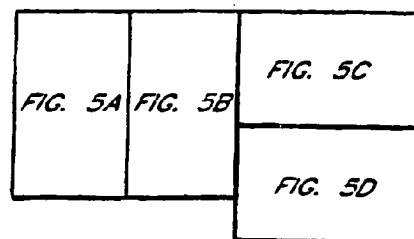


FIG. 5



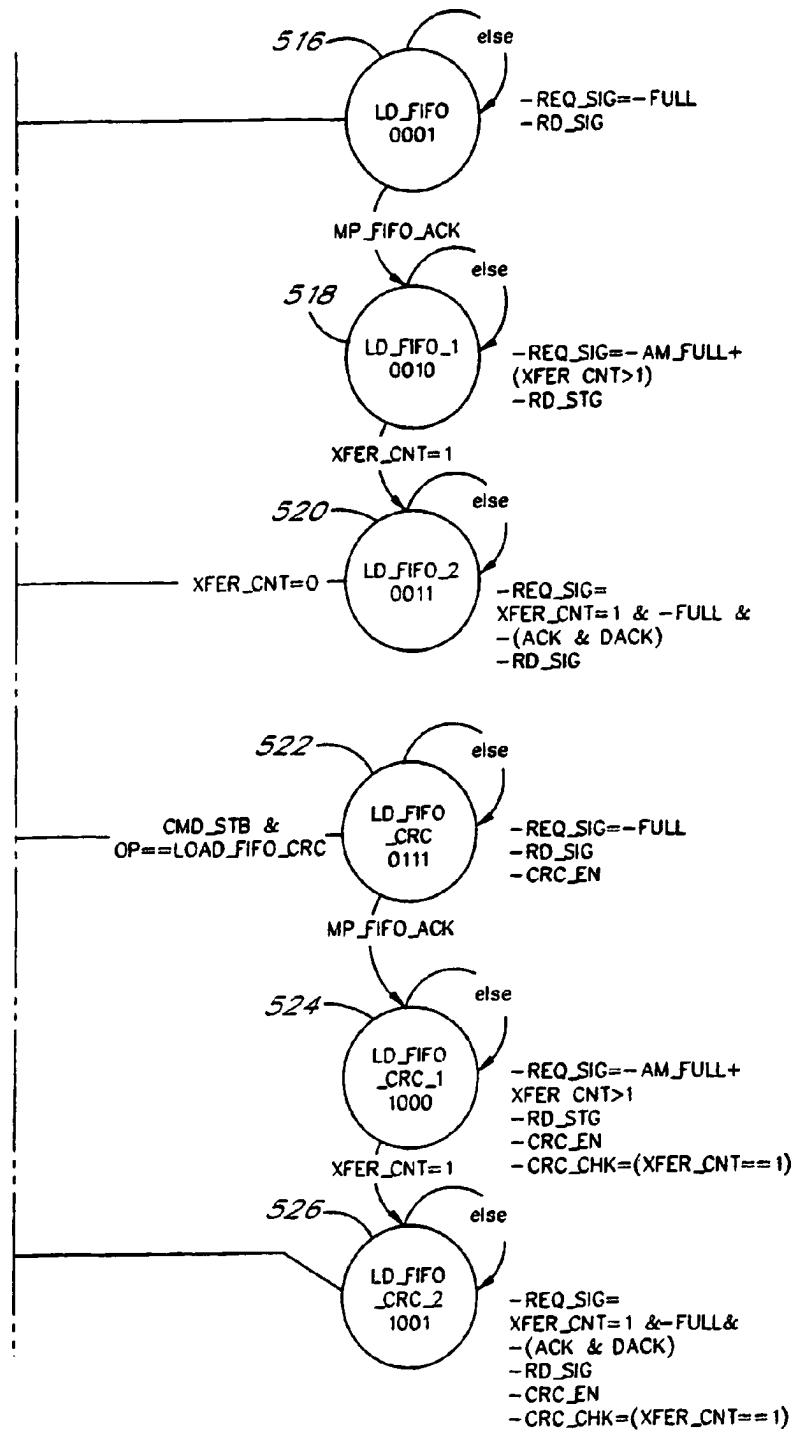
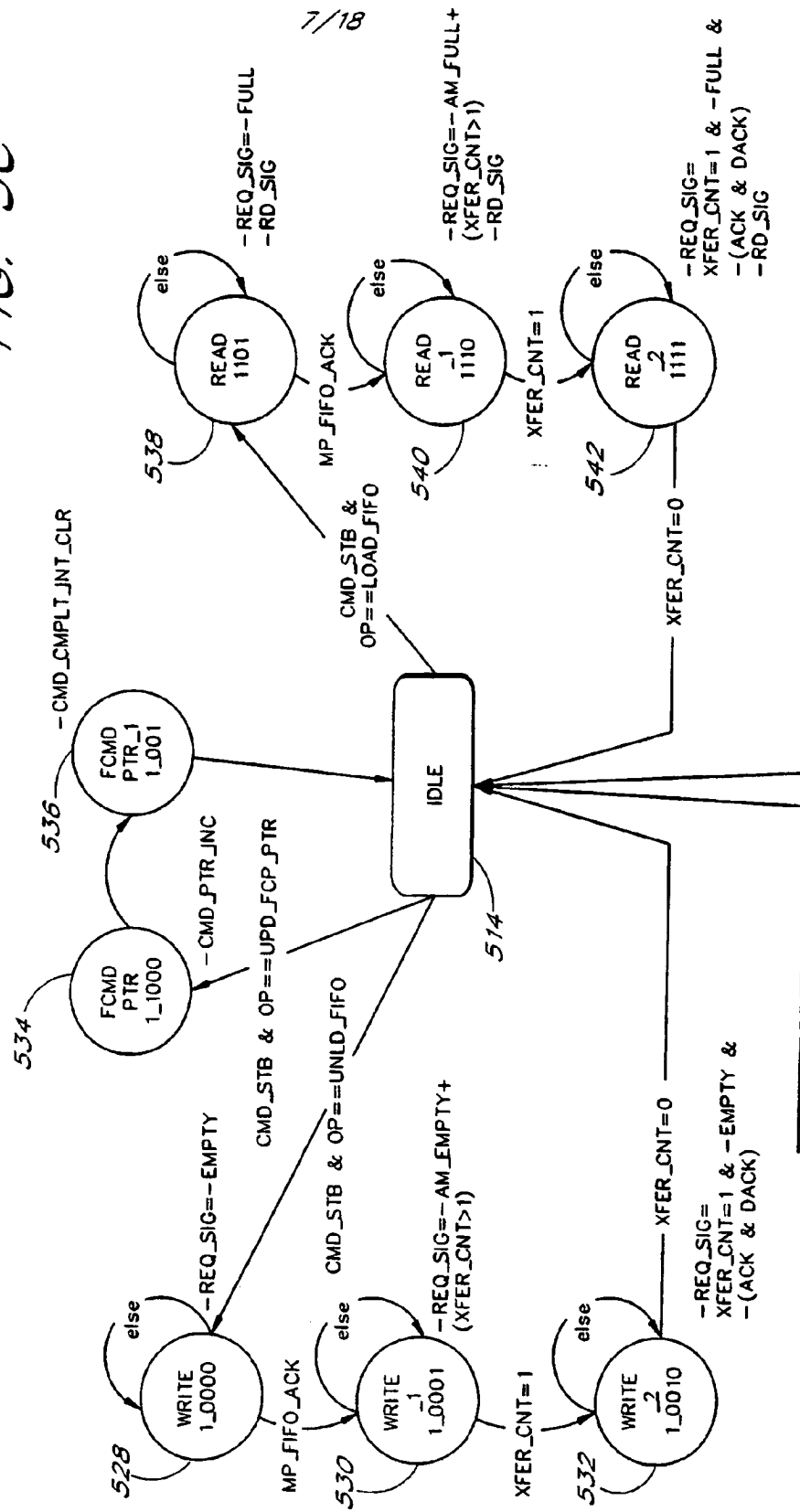


FIG. 5B

FIG. 5C



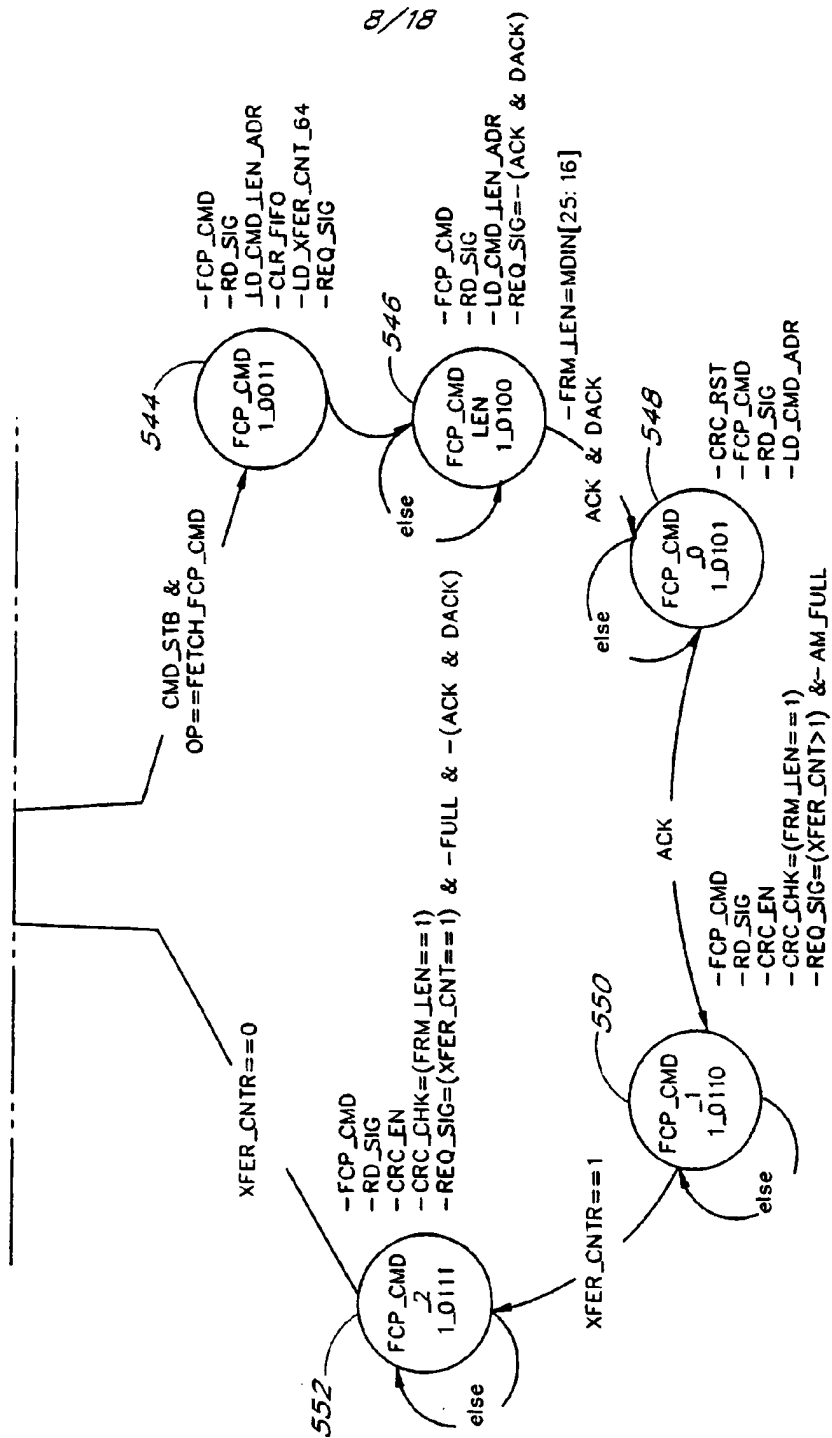


FIG. 5D

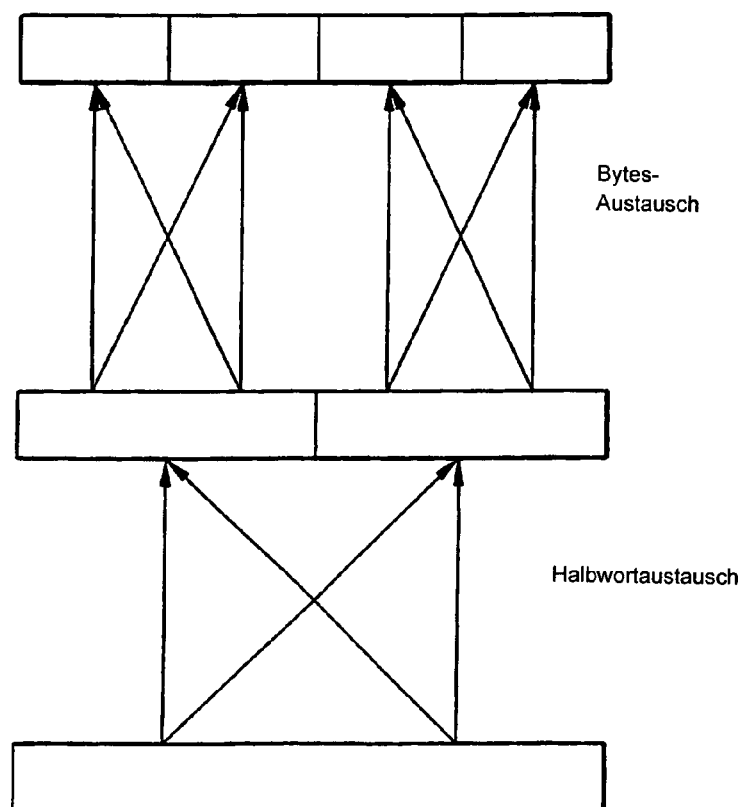


FIG. 6

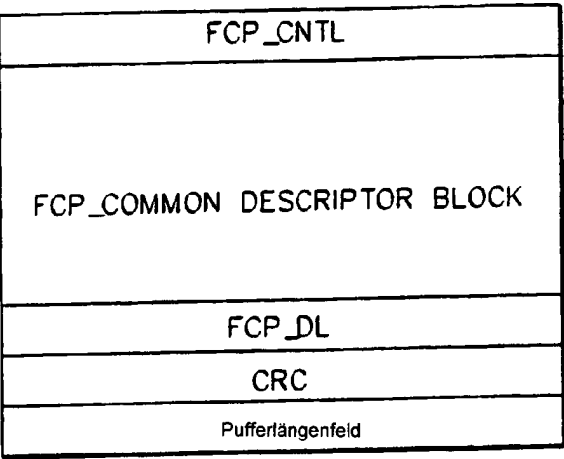
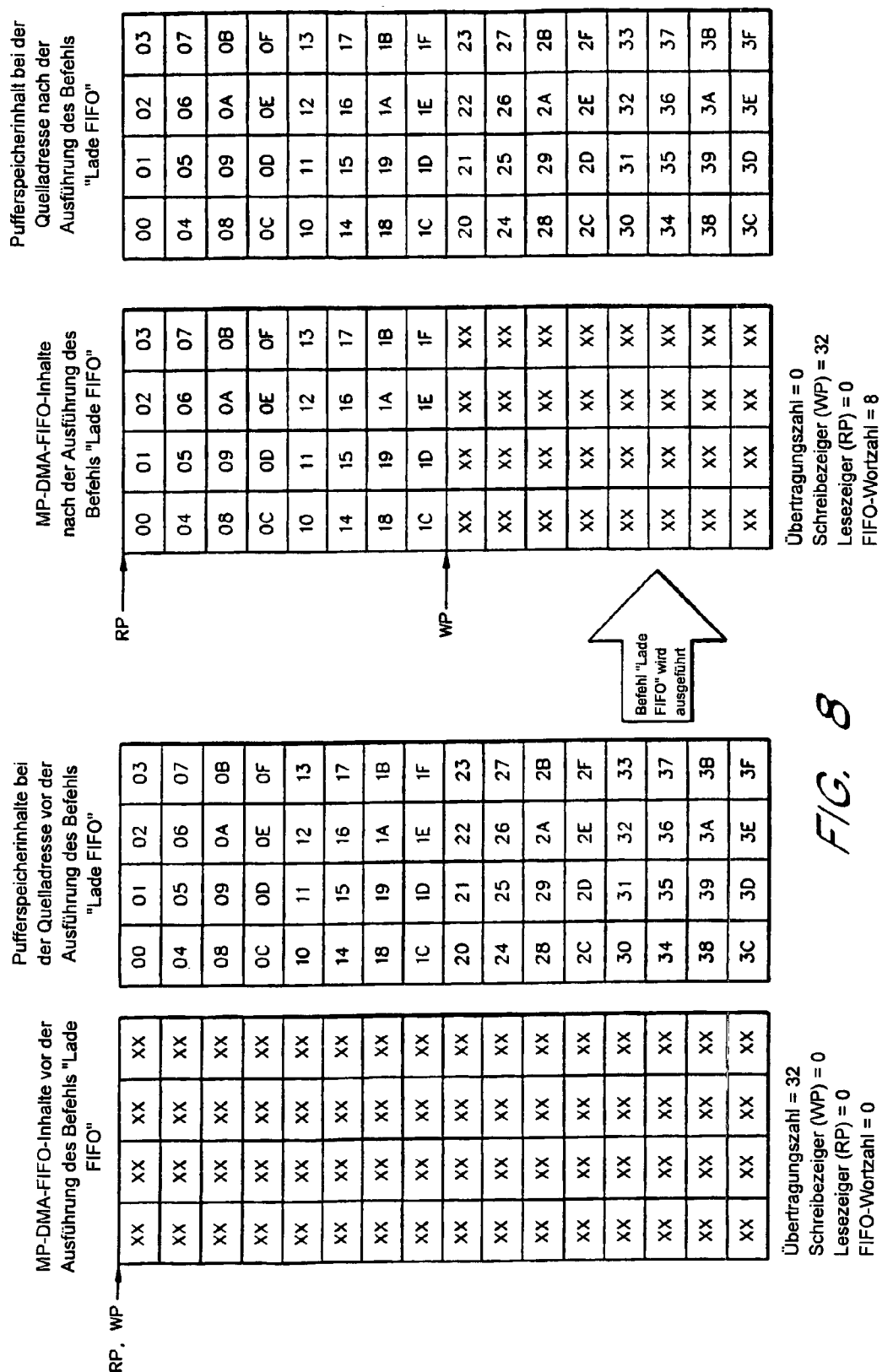
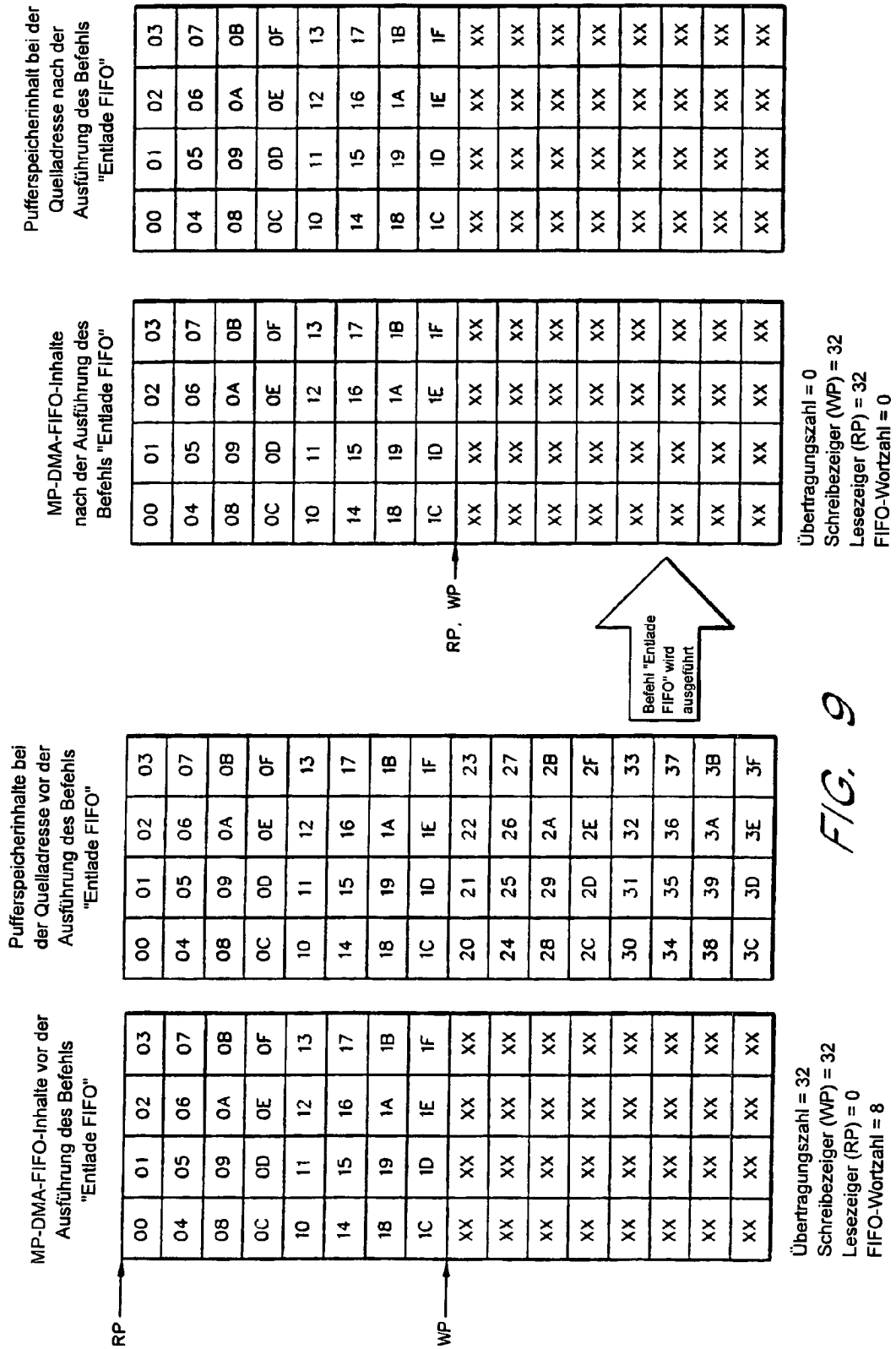
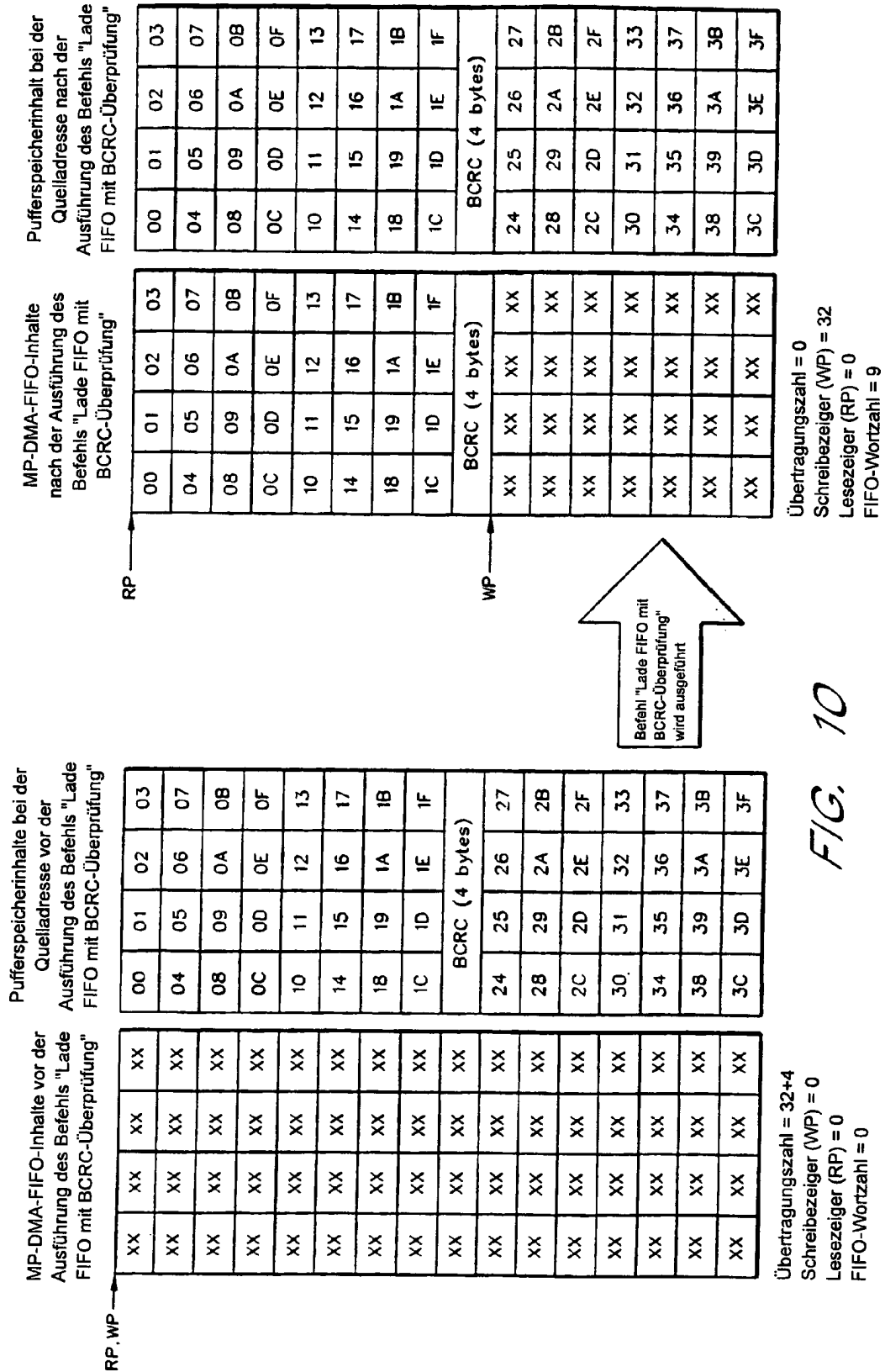
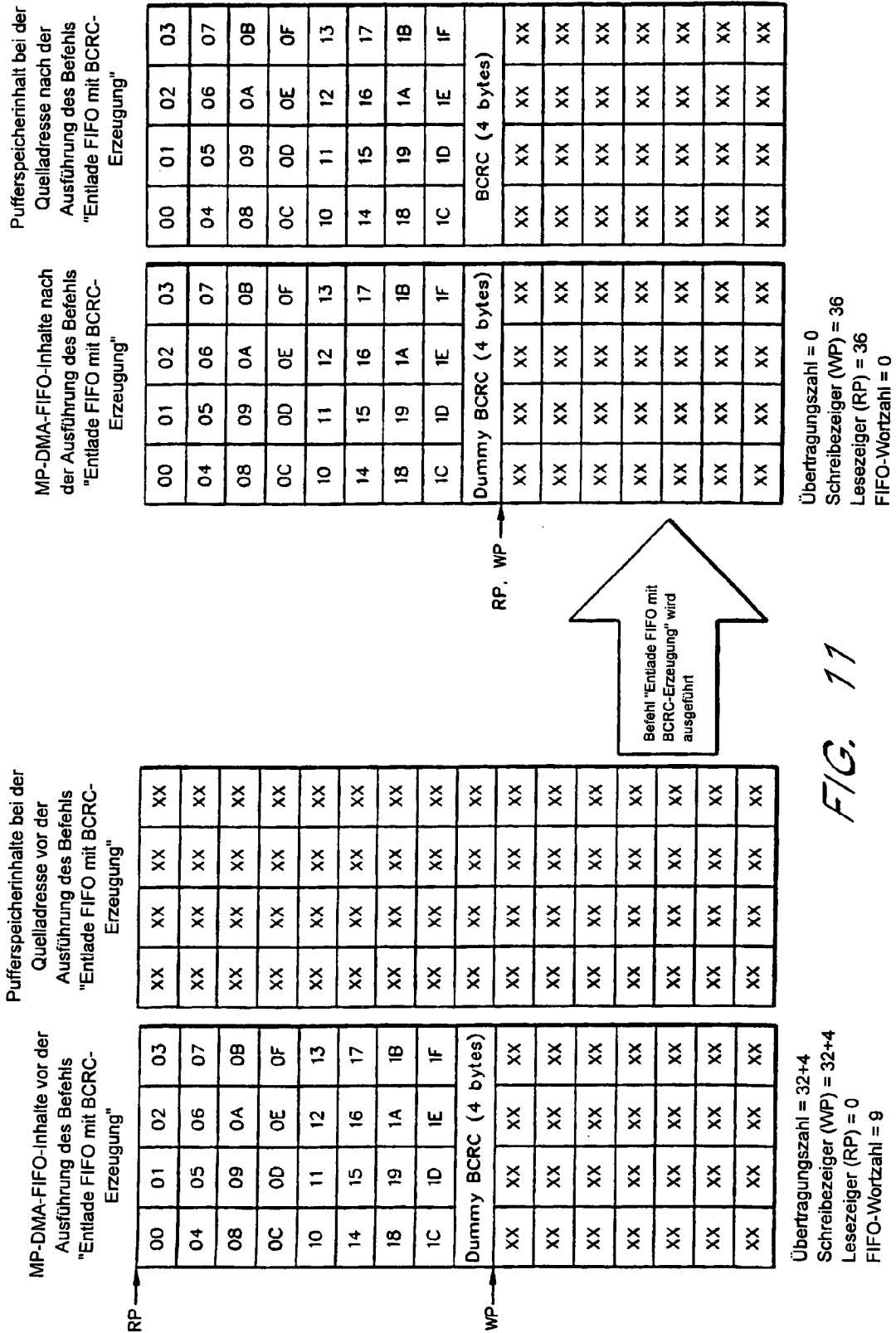


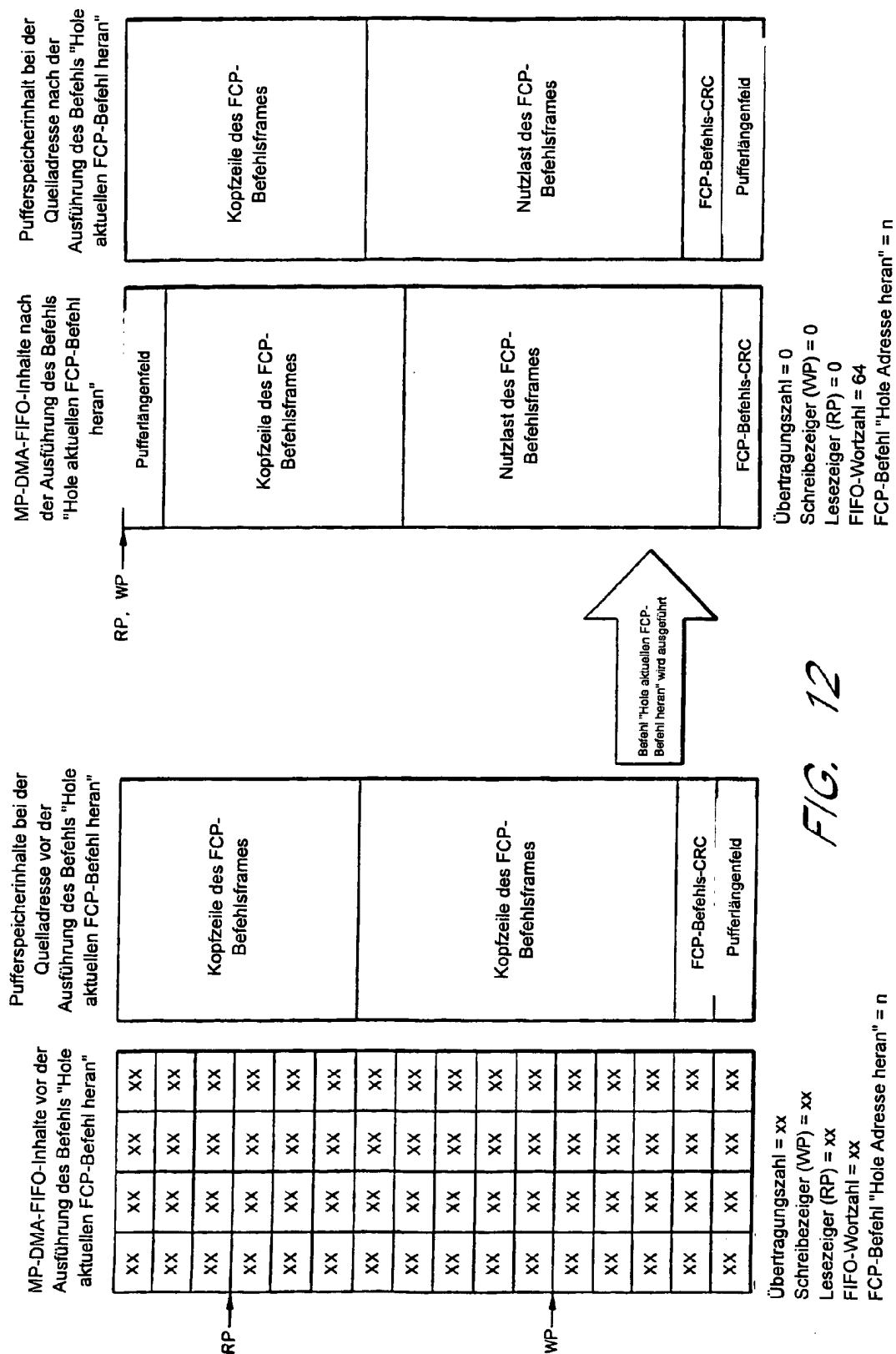
FIG. 7

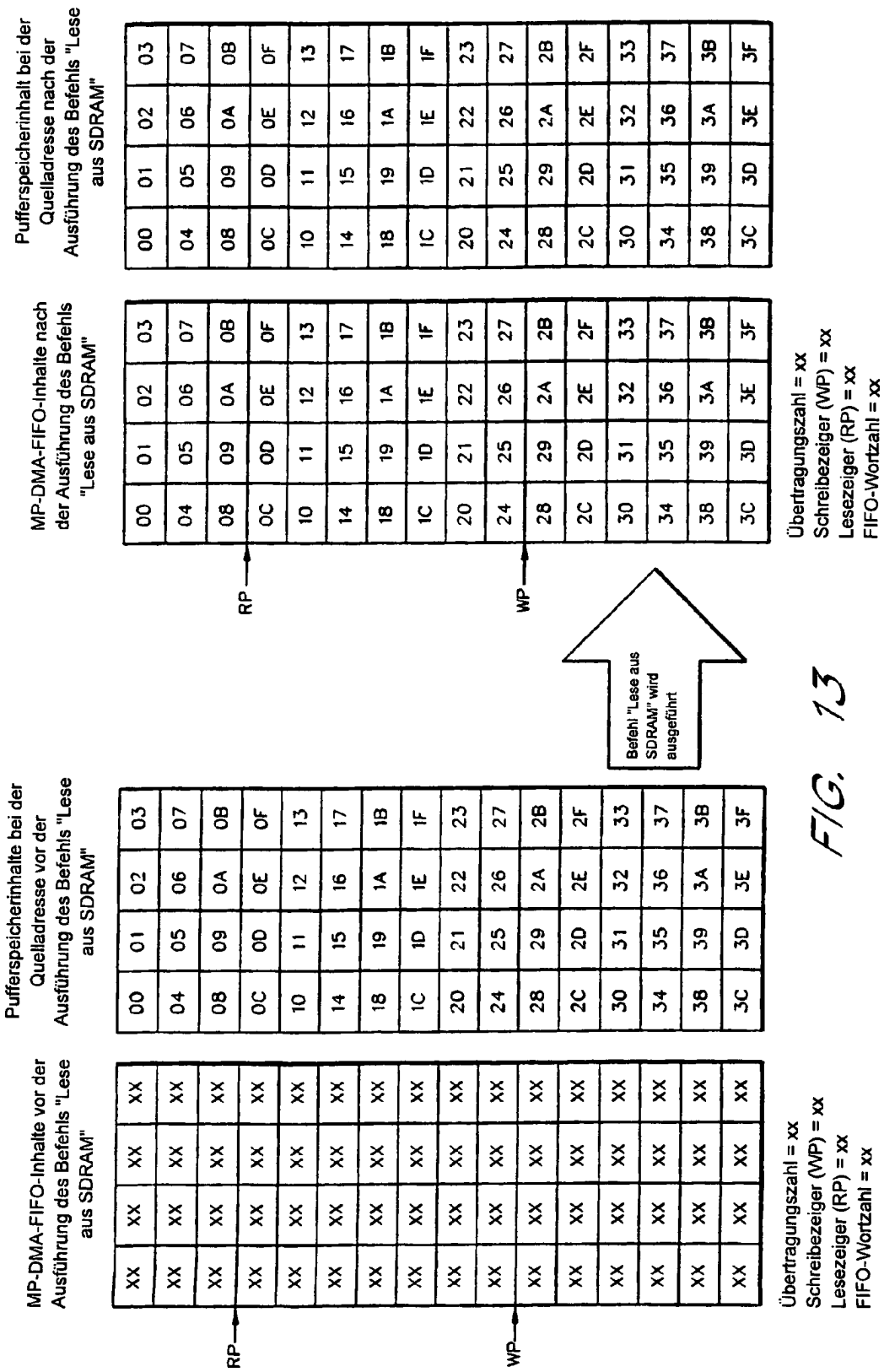












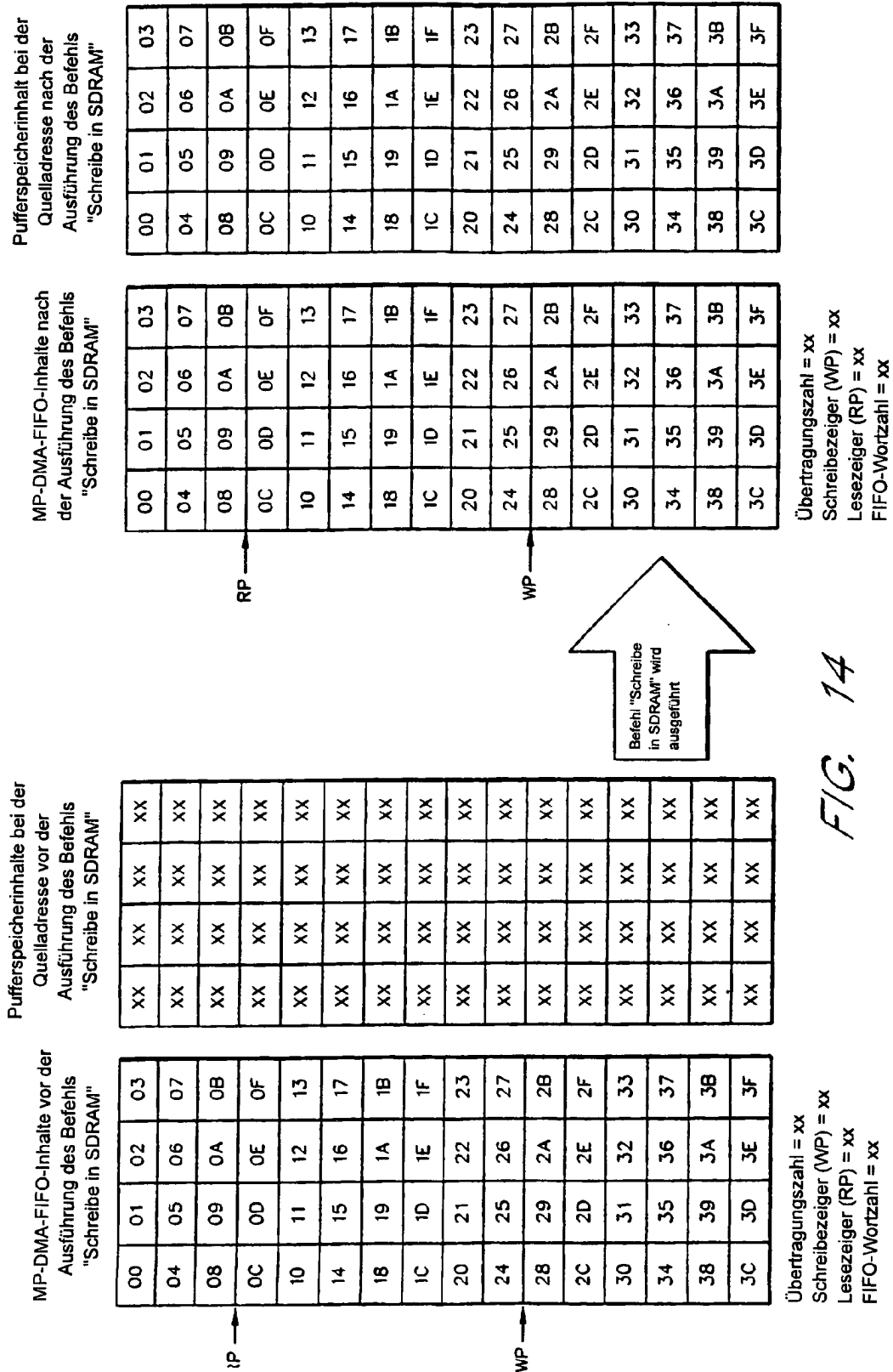


FIG. 14

