



(12) **Offenlegungsschrift**

(21) Aktenzeichen: **10 2020 110 559.2**

(22) Anmeldetag: **17.04.2020**

(43) Offenlegungstag: **18.02.2021**

(51) Int Cl.: **G06F 13/42 (2006.01)**

**G06F 13/38 (2006.01)**

**G06F 13/20 (2006.01)**

(30) Unionspriorität:  
**10-2019-0099851 14.08.2019 KR**  
**16/811,117 06.03.2020 US**

(71) Anmelder:  
**Samsung Electronics Co., Ltd., Suwon-si,  
Gyeonggi-do, KR**

(74) Vertreter:  
**KUHNEN & WACKER Patent- und  
Rechtsanwaltsbüro PartG mbB, 85354 Freising,  
DE**

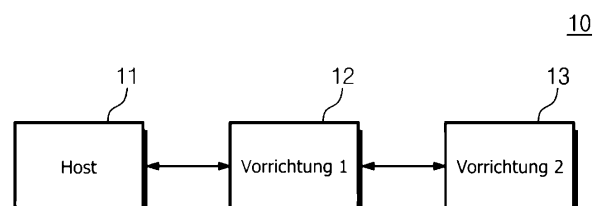
(72) Erfinder:  
**Kim, Dong-Uk, Suwon-si, Gyeonggi-do, KR; Ko,  
Yohan, Suwon-si, Gyeonggi-do, KR; Jo, Insoon,  
Suwon-si, Gyeonggi-do, KR**

Prüfungsantrag gemäß § 44 PatG ist gestellt.

**Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.**

(54) Bezeichnung: **Rechensystem zur Reduzierung von Latenz zwischen seriell verbundenen elektronischen Vorrichtungen**

(57) Zusammenfassung: Ein Rechensystem umfasst einen Host, eine erste elektronische Vorrichtung, die mit dem Host verbunden ist, und eine zweite elektronische Vorrichtung, die mit dem Host durch die erste elektronische Vorrichtung kommuniziert. Die erste elektronische Vorrichtung fordert einen in einer Übermittlungswarteschlange des Hosts geschriebenen Befehl an basierend auf einem Doorbell, übertragen von dem Host, speichert den von dem Host übertragenen Befehl, fordert in einem Datenpuffer des Hosts gespeicherte Schreibdaten an und speichert die von dem Host übertragenen Schreibdaten des Datenpuffers.



**Beschreibung**

## Querverweise auf verwandte Anmeldungen

**[0001]** Diese Anmeldung beansprucht die Priorität der koreanischen Patentanmeldung KR 10-2019-0099851 nach 35 U.S.C. § 119, eingereicht am 14. August 2019 beim koreanischen Patentamt, deren Offenbarungsgehalt hierin vollständig durch Bezugnahme aufgenommen ist.

## Hintergrund

## Gebiet

**[0002]** Ausführungsformen des erfinderischen Konzepts, das vorliegend beschrieben wird, betreffen ein Rechensystem und insbesondere ein Rechensystem zur Verringerung einer Latenz zwischen in Reihe geschalteten bzw. seriell verbundenen elektronischen Vorrichtungen.

## Beschreibung des Stands der Technik

**[0003]** In einem Rechensystem kann eine Mehrzahl von elektronischen Vorrichtungen verbunden sein, um miteinander zu kommunizieren. Die Mehrzahl von elektronischen Vorrichtungen kann in Reihe oder sequenziell bzw. aufeinanderfolgend geschaltet sein. Die in Reihe geschalteten elektronischen Vorrichtungen können mit einem Host des Rechensystems kommunizieren.

**[0004]** Eine elektronische Vorrichtung entsprechend einer Endpunktvorrichtung oder einem Endgerät bzw. Endvorrichtung von der Mehrzahl von elektronischen Vorrichtungen kann mit dem Host durch mindestens eine der in Reihe geschalteten elektronischen Vorrichtungen kommunizieren. Eine Kommunikationsgeschwindigkeit zwischen der Endpunktvorrichtung und dem Host kann aufgrund einer Latenz der zwischengeschalteten mindestens einen oder mehreren elektronischen Vorrichtungen, die zwischen der Endpunktvorrichtung und dem Host angeordnet sind, langsam werden. Insbesondere kann sich in der Konfiguration, in der die Endpunktvorrichtung eine elektronische Vorrichtung mit hoher Geschwindigkeit ist, die Kommunikationsgeschwindigkeit zwischen der elektronischen Vorrichtung mit hoher Geschwindigkeit und dem Host über die zwischengeschalteten elektronischen Vorrichtungen wesentlich verringern.

## Kurzfassung

**[0005]** Ausführungsformen des erfinderischen Konzepts schaffen ein Rechensystem zur Reduzierung einer Latenz zwischen elektronischen Vorrichtungen, die in Reihe geschaltet sind.

**[0006]** Gemäß einer Ausführungsform wird ein Rechensystem geschaffen umfassend einen Host, eine erste elektronische Vorrichtung, die mit dem Host gekoppelt ist, und eine zweite elektronische Vorrichtung, die mit der ersten elektronischen Vorrichtung gekoppelt ist, wobei die zweite elektronische Vorrichtung derart eingerichtet ist, dass sie mit dem Host durch die erste elektronische Vorrichtung kommuniziert. Die erste elektronische Vorrichtung ist eingerichtet, den Host aufzufordern, einen Schreibbefehl zu übertragen, der in eine Übermittlungswarteschlange des Hosts geschrieben wurde, basierend auf einem Doorbell, der von dem Host empfangen wird, den Host aufzufordern, Schreibdaten des Schreibbefehls, die in einem Datenpuffer des Hosts gespeichert sind, zu übertragen, und die von dem Host empfangenen Schreibdaten zu speichern.

**[0007]** Gemäß einer Ausführungsform wird ein Rechensystem geschaffen umfassend einen Host, eine erste elektronische Vorrichtung, die mit dem Host gekoppelt ist, und eine zweite elektronische Vorrichtung, die mit der ersten elektronischen Vorrichtung gekoppelt ist, wobei die zweite elektronische Vorrichtung derart eingerichtet ist, dass sie mit dem Host durch die erste elektronische Vorrichtung kommuniziert. Die erste elektronische Vorrichtung ist eingerichtet, einen Schreibbefehl für die zweite elektronische Vorrichtung von dem Host zu empfangen und den Schreibbefehl an die zweite elektronische Vorrichtung zu übertragen, einen Doorbell, der mit einer Übermittlungswarteschlange assoziiert ist, in der der Schreibbefehl geschrieben ist, von dem Host zu empfangen, den Doorbell an die zweite elektronische Vorrichtung zu übertragen, Schreibdaten des in einem Datenpuffer des Hosts gespeicherten Schreibbefehls anzufordern, und die von dem Host empfangenen Schreibdaten zu speichern.

**[0008]** Gemäß einer Ausführungsform wird ein Rechensystem geschaffen umfassend einen Host, eine erste elektronische Vorrichtung, die mit dem Host gekoppelt ist, welche einen Übermittlungswarteschlangen-Controller-Memory-Buffer (CMB) und einen Schreib-CMB umfasst, sowie eine zweite elektronische Vorrichtung, die mit der ersten elektronischen Vorrichtung gekoppelt ist, wobei die zweite elektronische Vorrichtung eingerichtet ist, mit dem Host durch die erste elektronische Vorrichtung zu kommunizieren. Die erste elektronische Vorrichtung ist eingerichtet, von dem Host einen Schreibbefehl, der in eine Übermittlungswarteschlange des Hosts geschrieben wurde, zu empfangen, und den Schreibbefehl in den Übermittlungswarteschlangen-Controller-Memory-Buffer zu speichern, von dem Host Schreibdaten des in einem Datenpuffer des Hosts gespeicherten Schreibbefehls zu empfangen, die Schreibdaten in dem Schreib-CMB zu speichern, einen Doorbell zu empfangen, der von dem Host bezüglich der Übermittlungswarteschlange übertragen wird, und den

Doorbell an die zweite elektronische Vorrichtung zu übertragen.

#### Figurenliste

**[0009]** Die obige sowie weitere Aufgaben und Merkmale der hier besprochenen erfinderischen Konzepte werden durch eine detaillierte Beschreibung von Ausführungsformen derselben unter Bezugnahme auf die beiliegenden Zeichnungen verständlich.

**Fig. 1** stellt ein Blockdiagramm eines Rechensystems gemäß einer Ausführungsform dar;

**Fig. 2** stellt ein detailliertes Blockschaltbild des Rechensystems aus **Fig. 1** dar;

**Fig. 3** stellt ein Blockschaltbild des Rechensystems aus **Fig. 1** dar;

**Fig. 4** stellt ein Zeitablaufdiagramm des Rechensystems aus **Fig. 3** dar;

**Fig. 5** stellt ein Blockschaltbild des Rechensystems aus **Fig. 1** dar;

**Fig. 6** stellt ein Zeitablaufdiagramm des Rechensystems aus **Fig. 5** dar;

**Fig. 7** stellt ein Verfahren zum Betreiben des Rechensystems aus **Fig. 5** dar;

**Fig. 8** stellt ein Blockschaltbild des Rechensystems aus **Fig. 1** dar;

**Fig. 9** stellt ein Zeitablaufdiagramm des Rechensystems aus **Fig. 8** dar;

**Fig. 10** stellt ein Verfahren zum Betreiben eines Rechensystems aus **Fig. 8** dar;

**Fig. 11** stellt ein Blockschaltbild des Rechensystems aus **Fig. 1** dar;

**Fig. 12** stellt ein Zeitablaufdiagramm eines Rechensystems aus **Fig. 11** dar;

**Fig. 13** stellt ein Verfahren zum Betreiben des Rechensystems aus **Fig. 11** dar;

**Fig. 14** stellt ein Blockschaltbild des Rechensystems aus **Fig. 1** dar;

**Fig. 15** stellt ein Zeitablaufdiagramm eines Rechensystems aus **Fig. 14** dar;

**Fig. 16** stellt ein Blockschaltbild des Rechensystems aus **Fig. 1** dar;

**Fig. 17** stellt ein Zeitablaufdiagramm des Rechensystems aus **Fig. 16** dar;

**Fig. 18** stellt ein Verfahren zum Betreiben des Rechensystems aus **Fig. 16** dar;

**Fig. 19** stellt ein Blockdiagramm einer Speichervorrichtung gemäß einer Ausführungsform dar; und

**Fig. 20** stellt ein Blockdiagramm einer Rechenvorrichtung gemäß einer Ausführungsform dar.

#### Detaillierte Beschreibung

**[0010]** **Fig. 1** stellt ein Blockdiagramm eines Rechensystems gemäß einer Ausführungsform dar.

**[0011]** Wie in **Fig. 1** dargestellt, kann ein Rechensystem **10** einen Host **11**, eine erste elektronische Vorrichtung **12** und eine zweite elektronische Vorrichtung **13** umfassen. Der Host **11**, die erste elektronische Vorrichtung **11** und die zweite elektronische Vorrichtung **12** des Rechensystems **10** können beliebige elektronische Vorrichtungen wie beispielsweise ein Desktop-Computer, ein Laptop-Computer, ein Tablet-Computer, eine Videospielekonsole, eine Workstation, ein Server, eine Rechenvorrichtung und ein elektrisches Fahrzeug oder ein Mainboard einer elektronischen Vorrichtung sein.

**[0012]** Der Host **11** kann mit einem Ein-Chip-System (SoC), einer anwendungsspezifischen integrierten Schaltung (ASIC) oder einem Field Programmable Gate Array (FPGA) implementiert werden. Die Steuerschaltung des Hosts **11** kann auch einen Universal-Prozessor, eine zentrale Verarbeitungseinheit, einen Spezial-Prozessor oder einen Anwendungsprozessor umfassen. Der Host **11** kann als Prozessor selbst implementiert werden, oder der Host **11** kann eine elektronische Vorrichtung oder ein System sein, das einen Prozessor umfasst. Der Host **11** kann übergreifend eine Kommunikation bzw. Nachrichten des Rechensystems **10** steuern, einschließlich Nachrichten zwischen dem Host **11**, der ersten elektronischen Vorrichtung **12** und der zweiten elektronischen Vorrichtung **13**.

**[0013]** Die erste elektronische Vorrichtung **12** kann mit dem Host **11** direkt verbunden oder in Reihe geschaltet sein (oder sequenziell). Die zweite elektronische Vorrichtung **13** kann mit der ersten elektronischen Vorrichtung **12** direkt verbunden oder in Reihe geschaltet sein. So können der Host **11**, die erste elektronische Vorrichtung **12** und die zweite elektronische Vorrichtung **13** sequenziell geschaltet sein. Mit anderen Worten, die zweite elektronische Vorrichtung **13** kann dem Host **11** durch die erste elektronische Vorrichtung **12** verbunden sein. Zum Beispiel kann der Host **11** direkt mit der ersten elektronischen Vorrichtung **12** kommunizieren und kann mit der zweiten elektronischen Vorrichtung **13** durch die erste elektronische Vorrichtung **12** kommunizieren. Entsprechend kann die zweite elektronische Vorrichtung **13** einer Endpunktvorrichtung entsprechen, und die erste elektronische Vorrichtung **12** kann eine Zwi-

schenvorrichtung sein, die die elektronische Endvorrichtung **13** mit dem Host **11** verbindet.

**[0014]** Anders als das in **Fig. 1** dargestellte Beispiel, können zusätzliche elektronische Vorrichtungen ferner in dem Rechensystem **10** an einem beliebigen Punkt zwischen dem Host **11** und der ersten elektronischen Vorrichtung **12** und zwischen der ersten elektronischen Vorrichtung **12** und der zweiten elektronischen Vorrichtung **13** verbunden sein. Selbstverständlich kann eine zusätzliche elektronische Vorrichtung mit ausschließlich der zweiten elektronischen Vorrichtung **13** verbunden sein, sodass die zusätzliche elektronische Vorrichtung die Endpunktvorrichtung des Rechensystems **10** sein kann.

**[0015]** Die erste elektronische Vorrichtung **12** und die zweite elektronische Vorrichtung **13** können derart implementiert sein, dass sie identische elektronische Vorrichtungen oder unterschiedliche elektronische Vorrichtungen sind. In einem anderen Beispiel kann die erste elektronische Vorrichtung **12** einem Switch bzw. einer Schaltvorrichtung entsprechen, welche die zweite elektronische Vorrichtung **13** mit dem Host **11** verbindet, und die zweite elektronische Vorrichtung **13** kann einer Endpunktvorrichtung entsprechen, die sich von der Schaltvorrichtung unterscheidet.

**[0016]** **Fig. 2** stellt ein detailliertes Blockschaltbild des Rechensystems aus **Fig. 1** dar. Ein Rechensystem **100**, das dem Rechensystem **10** aus **Fig. 1** entsprechen kann, kann einen Prozessor **110**, einen Root-Complex **120**, einen Speicher **130** und elektronische Vorrichtungen **141**, **142**, **151** bis **154** und **161** bis **163** umfassen. Hier kann eine elektronische Vorrichtung auch als „Eingabe/Ausgabe(I/O)-Vorrichtung“ bezeichnet werden, und die elektronischen Vorrichtungen **141**, **142**, **151** bis **154** und **161** bis **163** können der ersten elektronischen Vorrichtung **12** und der zweiten elektronischen Vorrichtung **13**, dargestellt in **Fig. 1**, entsprechen. Der Prozessor **110**, der Root-Complex **120** und der Speicher **130** können Komponenten des in **Fig. 1** dargestellten Hosts **11** sein.

**[0017]** Der Prozessor **110** kann verschiedene Arten arithmetischer Operationen oder Logikoperationen durchführen. Zum Beispiel kann der Prozessor **110** mindestens einen oder mehrere Kerne (z. B. einen homogenen Mehrkern bzw. Multi-Core oder einen heterogenen Mehrkern) umfassen, die Operationen und einen internen Cache-Speicher steuern. Der Prozessor **110** kann einen aus dem Speicher **130** geladenen Programmcode, eine Software, ein Anwendungsprogramm usw. ausführen.

**[0018]** Der Root-Complex **120** kann bei der Kommunikation zwischen dem Prozessor **110**, dem Speicher **130** und den elektronischen Vorrichtungen **141**, **142**,

**151** bis **154** und **161** bis **163** vermitteln. Zum Beispiel kann der Root-Complex **120** eine Kommunikationszeit, eine Kommunikationssequenz, eine Kommunikationsumgebung usw. zwischen dem Prozessor **110**, dem Speicher **130** und den elektronischen Vorrichtungen **141**, **142**, **151** bis **154** und **161** bis **163** verwalten. Der Root-Complex **120** kann in einer Hardwarekonfiguration, einer Softwarekonfiguration oder einer Kombination aus Hardware und Software implementiert werden, und kann auf einem Mainboard des Rechensystems **100** implementiert werden. Der Root-Complex **120** kann ein Root sein, der den Prozessor **110** und den Speicher **130** mit einer I/O-Hierarchie, die die elektronischen Vorrichtungen **141**, **142**, **151** bis **154** und **161** bis **163** umfasst, kommunikationsfähig verbindet. Der Root-Complex **120** kann einen oder mehrere Downstream-Ports DP umfassen. Die elektronischen Vorrichtungen **141** und **142** können mit dem Downstream-Port DP des Root-Complex **120** verbunden sein. Die elektronischen Vorrichtungen **151** bis **154** können mit dem Downstream-Port DP des Root-Complex **120** verbunden sein. Und die elektronischen Vorrichtungen **161** bis **163** können mit dem Downstream-Port DP des Root-Complex **120** verbunden sein. Die Anzahl an Downstream-Ports DP ist nicht auf die in **Fig. 2** dargestellte Konfiguration beschränkt. Die Anzahl an elektronischen Vorrichtungen, die jeweils mit den Downstream-Ports DP verbunden sind, kann eins oder mehr betragen.

**[0019]** In einer Ausführungsform kann die Kommunikation zwischen dem Root-Complex **120** und den elektronischen Vorrichtungen **141**, **142**, **151** bis **154** und **161** bis **163** entsprechend verschiedener Kommunikationsschnittstellenprotokolle durchgeführt werden, wie beispielsweise einem Peripheral-Component-Interconnect-Express (PCIe)-Protokoll, einem Mobile-PCIe(M-PCIe)-Protokoll, einem Nonvolatile-Memory-Express(NVMe)-Protokoll, einem Universal-Serial-Bus(USB)-Protokoll, einem Small-Computer-System-Interface(UCSI)-Protokoll, einem Advanced-Technology-Attachment (ATA)-Protokoll, Parallel-ATA(PATA), Serial-ATA (SATA), einem Serial-Attached-SCSI(SAS)-Protokoll, einem Integrated-Drive-Electronics(IDE)-Protokoll, einem Universal-Flash-Storage(UFS)-Protokoll und einem Firewire-Protokoll.

**[0020]** Der Speicher **130** kann Daten speichern, die für einen Betrieb des Rechensystems **100** verwendet werden. Der Speicher **130** kann Daten speichern, die von dem Prozessor **110** verarbeitet wurden oder zu verarbeiten sind. Zum Beispiel kann der Speicher **130** einen flüchtigen Speicher umfassen, wie beispielsweise einen statischen Direktzugriffsspeicher (SRAM) oder einen dynamischen Direktzugriffsspeicher (DRAM), oder einen nichtflüchtigen Speicher. Ein Anwendungsprogramm, ein Dateisystem oder ein Vorrichtungstreiber, der von dem Prozessor **110** aus-

fürbar ist, kann in den Speicher **130** geladen werden. Eine Programm- oder Software-Schicht, die in den Speicher **130** geladen wird, kann gesteuert durch den Prozessor **110** ausgeführt werden, und die Information, die in den Speicher **130** geladen wird, ist nicht auf das in **Fig. 2** dargestellte Beispiel beschränkt. Der Speicher **130** kann einen Host Memory Buffer (HMB) umfassen. Ein Abschnitt des gesamten Speicherbereichs des Speichers **130** kann dem Host Memory Buffer HMB zugewiesen sein.

**[0021]** Der Prozessor **110** kann einem Prozessor des Hosts **11** aus **Fig. 1** entsprechen. Der Prozessor **110** und der Root-Complex **120** können Komponenten des Hosts **11** aus **Fig. 1** entsprechen. Und der Prozessor **110**, der Root-Complex **120** und der Speicher **130** können alle Komponenten des Hosts **11** aus **Fig. 1** entsprechen. Der Prozessor **110**, der Root-Complex **120** und der Speicher **130** können als Ein-Chip-System (SoC) implementiert sein, um den Host **11** zu bilden. Alternativ können der Prozessor **110**, der Root-Complex **120** und der Speicher **130** mit einem oder mehreren getrennten Komponenten implementiert sein, um den Host **11** zu bilden.

**[0022]** Jede der elektronischen Vorrichtungen **142**, **152**, **154** und **163** kann als Endpunktvorrichtung eingerichtet sein. Jede der elektronischen Vorrichtungen **142**, **152**, **154** und **163** kann einen Endpunkt-Port bzw. -Anschluss EP umfassen. Jede der verbleibenden elektronischen Vorrichtungen **141**, **151** bis **153**, **161** und **162** kann einer Zwischenvorrichtung entsprechen. Die Zwischenvorrichtung kann mit einer Endpunktvorrichtung, einer anderen Zwischenvorrichtung oder dem Root-Complex **120** verbunden sein. Jede der elektronischen Vorrichtungen **141**, **153**, **161** und **162** kann einen Upstream-Port UP und einen Downstream-Port DP umfassen. Zum Beispiel kann der Upstream-Port UP der elektronischen Vorrichtungen **141**, **153**, **161** und **162** auf einer vorgeschalteten Seite der elektronischen Vorrichtungen **141**, **153**, **161** und **162** in Richtung des Root-Complex **120** angeordnet sein. Der Downstream-Port DP der elektronischen Vorrichtungen **141**, **153**, **161** und **162** kann auf einer nachgeschalteten Seite der elektronischen Vorrichtungen **141**, **153**, **161** und **162** in Richtung des Endpunkts angeordnet sein. Der Endpunkt-Port EP einer Endpunktvorrichtung kann mit einem Downstream-Port DP einer Zwischenvorrichtung oder dem Root-Complex **120** verbunden sein. Der Endpunkt-Port EP kann auch als „Upstream-Port UP“ bezeichnet werden. In der Konfiguration aus **Fig. 2** kann eine der elektronischen Vorrichtungen **141**, **151** bis **153**, **161** und **162** der ersten elektronischen Vorrichtung **12** aus **Fig. 1** entsprechen, und eine der elektronischen Vorrichtungen **142**, **152**, **154** und **163** kann der zweiten elektronischen Vorrichtung **13** aus **Fig. 1** entsprechen.

**[0023]** In einer Ausführungsform können die elektronischen Vorrichtungen **141**, **151** bis **153**, **161** und **162**, die der ersten elektronischen Vorrichtung **12** aus **Fig. 1** entsprechen, ein PCIe-Schalter, eine PCIe-Vorrichtung, eine NVMe-Vorrichtung, eine Speichervorrichtung oder eine Solid State Drive (SSD) sein. Die elektronischen Vorrichtungen **142**, **152**, **154** und **163**, die den zweiten elektronischen Vorrichtungen **13** aus **Fig. 1** entsprechen, können ebenfalls ein PCIe-Schalter, eine PCIe-Vorrichtung, ein NVMe-Schalter, eine NVMe-Vorrichtung, eine Speichervorrichtung oder eine SSD sein. Wie oben beschrieben, können ferner beliebige andere Endpunktvorrichtungen, die mit den elektronischen Vorrichtungen **142**, **152**, **154** und **163** verbunden sind, in dem Rechen-system **100** umfasst sein.

**[0024]** **Fig. 3** stellt ein Blockschaltbild des Rechen-systems aus **Fig. 1** dar. **Fig. 4** stellt ein Zeitablaufdiagramm des Rechen-systems aus **Fig. 3** dar.

**[0025]** In **Fig. 3** kann ein Rechen-system **200** einen Host **210**, einen Schalter **220** und eine Speichervorrichtung **230** umfassen. Zum Beispiel können das Rechen-system **200**, der Host **210**, der Schalter **220** und die Speichervorrichtung **230** jeweils dem Rechen-system **100**, dem Host **11**, der ersten elektronischen Vorrichtung **12** und der zweiten elektronischen Vorrichtung **13** aus **Fig. 1** entsprechen.

**[0026]** Der Host **210** kann eine Übermittlungswarteschlange (SQ) **211**, eine Beendigungswarteschlange (CQ) **212** und einen Datenpuffer **213** umfassen. Die Übermittlungswarteschlange **211**, die Beendigungswarteschlange **212** und der Datenpuffer **213** können in dem Host Memory Buffer HMB aus **Fig. 2** implementiert sein oder können Abschnitte des Host Memory Buffer HMB belegen. Der Host **210** kann Eingabe-/Ausgabe-(I/O)-Vorgänge für die Speichervorrichtung **230** durch den Schalter **220** basierend auf der Übermittlungswarteschlange **211** und der Beendigungswarteschlange **212** durchführen.

**[0027]** Der Schalter **220** kann zwischen dem Host **210** und der Speichervorrichtung **230** angeordnet sein und kann elektrische Signale von dem Host **210** (oder der Speichervorrichtung **230**) an die Speichervorrichtung **230** (oder den Host **210**) übertragen. So fungiert der Schalter **220** als Zwischenvorrichtung zwischen dem Host **210** und der Speichervorrichtung **230**. Die Speichervorrichtung **230** kann Befehle von dem Host **210** empfangen und kann die empfangenen Befehle verarbeiten, und kann das verarbeitete Ergebnis an den Host **210** über den Schalter **220** übertragen. Die Speichervorrichtung **230** kann einen nichtflüchtigen Speicher **239** sowie einen Controller **231** zur Steuerung des nichtflüchtigen Speichers **239** umfassen. Der nichtflüchtige Speicher **239** kann einen NAND-Flash-Speicher, einen NOR-Flash-Speicher, einen Phasenänderungs-RAM (PRAM), einen

magnetischen RAM (MRAM), einen resistiven RAM (RRAM), einen ferroelektrischen RAM (FeRAM) usw. umfassen.

**[0028]** Der Host **210** kann einen Befehl für die Speichervorrichtung **230** in einen Eintrag (oder einen Slot) der Übermittlungswarteschlange **211** eingeben oder schreiben, und kann einen Tail Pointer der Übermittlungswarteschlange **211**, das heißt, einen Doorbell für die Übermittlungswarteschlange **211** aktualisieren. Zum Beispiel kann der Doorbell eine Adresse umfassen, die die Übermittlungswarteschlange **211** angibt. Das Paar der Übermittlungswarteschlange **211** und der Beendigungswarteschlange **212** kann für jeden Kern des Prozessors **110** vorgesehen werden, und die Anzahl an Paaren kann eins oder mehr betragen. Bezüglich **Fig. 3** und **Fig. 4** kann der Host **210** einen Doorbell für die Übermittlungswarteschlange **211** an die Speichervorrichtung **230** durch den Schalter **220** absenden oder übertragen (①). Wenn Schalterlatenzen des Schalters **220** ablaufen, nachdem der Doorbell von dem Host **210** übertragen wurde (①), kann die Speichervorrichtung **230** den Doorbell von dem Schalter **220** empfangen (①). Eine Latenz kann als „Zeit“ bezeichnet werden. Die (Gesamt-)Latenz des Schalters **220** kann zum Beispiel in eine hostseitige Schalterlatenz und eine speichervorrichtungsseitige Schalterlatenz unterteilt werden. Im Detail kann der Schalter **220** eine oder mehrere Übertragungsschaltungen umfassen, um einen Doorbell zu übertragen, sowie einen Übertragungspfad, der physische Pfade in dem Schalter **220** umfasst. Eine Zeit, die aufgewendet wird, damit der Doorbell einen Abschnitt des Übertragungspfads des Schalters **220** passiert, kann zum Beispiel der hostseitigen Schalterlatenz entsprechen, und eine Zeit, die aufgewendet wird, damit der Doorbell den verbleibenden Abschnitt des Übertragungspfads des Schalters **220** passiert, kann der vorrichtungsseitigen Schalterlatenz entsprechen.

**[0029]** Der Host **210** kann ein Doorbell-Register (SQTDBL) **232** des Controllers **231** der Speichervorrichtung **230** aktualisieren, indem er den Doorbell überträgt (①). Die Speichervorrichtung **230** kann den in dem Doorbell-Register **232** gespeicherten Doorbell auslesen und kann identifizieren (überprüfen), dass ein Befehl eingegeben wurde oder von dem Host **210** in die Übermittlungswarteschlange **211** eingegeben oder geschrieben wurde. Die Speichervorrichtung **230** kann den Befehl der Übermittlungswarteschlange **211** des Hosts **210** basierend auf dem Doorbell abrufen oder auslesen ((2), (3)). Zum Beispiel kann die Speichervorrichtung **230** den Befehl der Übermittlungswarteschlange **211** von dem Hosts **210** anfordern und kann die Anforderung an den Host **210** durch den Schalter **220** übertragen (②). Dann kann die Speichervorrichtung **230** den Befehl der Übermittlungswarteschlange **211** durch den Schalter **220** von dem Host **210** auslesen oder empfangen (

(3)). Wenn Schalterlatenzen ablaufen, nachdem die Anforderung bezüglich des Befehls von der Speichervorrichtung **230** an den Schalter **220** übertragen wurde (②), kann die Anforderung bei dem Host **210** ankommen (②). Außerdem kann, wenn Schalterlatenzen ablaufen, nachdem die Anfrage von dem Host **210** an den Schalter **220** übertragen wurde ((3)), der Befehl der Übermittlungswarteschlange **211** bei der Speichervorrichtung **230** ankommen (③). Wie bei der Übertragung des Doorbells, können Schalterlatenzen in jeweils den Fällen auftreten, in denen eine Anforderung bezüglich eines Befehls übertragen wird (②) und ein Befehl übertragen wird (③).

**[0030]** Der Controller **231** der Speichervorrichtung **230** kann eine Speicherdirektzugriff-Engine **233** (Direct Memory Access, DMA) umfassen, die eingerichtet ist, eine Anforderung oder Daten an den Schalter **220** oder den Host **210** zu übertragen. Die DMA-Engine **233** kann einen Befehl von dem Schalter **220** empfangen und kann den empfangenen Befehl in einem Übermittlungswarteschlangenpuffer **234** des Controllers **231** speichern.

**[0031]** Die Speichervorrichtung **230** kann den Befehl, der in dem Übermittlungswarteschlangenpuffer **234** gespeichert ist, verarbeiten. Zum Beispiel kann der Befehl ein Schreibbefehl sein. Die Speichervorrichtung **230** kann den Befehl decodieren und kann in dem Datenpuffer **213** des Hosts **210** gespeicherte Schreibdaten basierend auf dem Befehl ((4), (5)) lesen. Zum Beispiel kann die Speichervorrichtung **230** die Schreibdaten des Datenpuffers **213** von dem Host **210** anfordern und kann die Anforderung an den Host **210** durch den Schalter **220** übertragen (④). Dann kann die Speichervorrichtung **230** die Schreibdaten des Datenpuffers **213** durch den Schalter **220** von dem Host **210** empfangen (⑤). Wenn Schalterlatenzen ablaufen, nachdem die Anforderung bezüglich der Schreibdaten von der Speichervorrichtung **230** an den Schalter **220** übertragen wurde (④), kann die Anforderung bei dem Host **210** ankommen (④). Auch können, wenn Schalterlatenzen ablaufen, nachdem die Schreibdaten von dem Datenpuffer **213** an den Schalter **220** übertragen wurden (⑤), die Schreibdaten an der Speichervorrichtung **230** ankommen (⑤). Wie bei der Übertragung des Doorbells, können Schalterlatenzen jeweils in den Fällen auftreten, in denen eine Anforderung bezüglich Schreibdaten übertragen wird (④) und in denen die Schreibdaten übertragen werden (⑤).

**[0032]** Die DMA-Engine **233** kann Schreibdaten anfordern, kann die Schreibdaten empfangen, und kann die empfangenen Schreibdaten in einem Schreibpuffer **235** des Controllers **231** speichern. Die obigen Vorgänge wurden in dem Kontext beschrieben, in dem Schreibdaten angefordert werden und übertragen werden, unter der Annahme, dass ein Befehl ein Schreibbefehl ist; allerdings wird die Übertragung

von Schreibdaten möglicherweise nicht durchgeführt, wenn der Befehl nicht der Schreibbefehl ist (z. B. ein Lesebefehl ist).

**[0033]** Die Speichervorrichtung **230** kann einen Befehl des Hosts **210** verarbeiten und kann eine Abschlussinformation bezüglich des Befehls an den Host **210** durch den Schalter **220** übertragen (⊙). Zum Beispiel kann die Abschlussinformation umfassen, ob ein Befehl erfolgreich oder nicht erfolgreich abgeschlossen wurde, ein Ergebnis der Verarbeitung des Befehls usw. Wenn Schalterlatenzen ablaufen, nachdem die Abschlussinformation von der Speichervorrichtung **230** an den Schalter **220** übertragen wurde (⊙), kann die Abschlussinformation bei dem Host **210** ankommen (⊙). Die Abschlussinformation kann in der Beendigungswarteschlange **212** des Hosts **210** gespeichert oder darin geschrieben werden. Wenn Schalterlatenzen ablaufen, nachdem die Abschlussinformation von der Speichervorrichtung **230** an den Schalter **220** übertragen wurde (⊙), kann die Abschlussinformation bei dem Host **210** ankommen (⊙). Wie bei der Übertragung des Doorbells, können Schalterlatenzen selbst in dem Fall auftreten, in dem eine Abschlussinformation übertragen wird (⊙).

**[0034]** Vorliegend beschriebene Ausführungsformen können eine Mehrzahl von Rechensystemen **300** bis **700** und **1000** schaffen, um eine Zeit zu reduzieren, die in Anspruch genommen wird, um einen von dem Host **210** erzeugten (oder ausgegebenen) Befehl zu verarbeiten, oder eine Zeit, die in Anspruch genommen wird, damit ein Befehl oder Schreibdaten, die von dem Host **210** erzeugt wurden, bei einer Endpunktvorrichtung (d. h. der Speichervorrichtung **230**) ankommen, unter Berücksichtigung der Schalterlatenzen aufgrund des Schalters **220** zwischen dem Host **210** und der Speichervorrichtung **230**.

**[0035]** Fig. 5 stellt ein Blockschaltbild des Rechensystems aus Fig. 1 dar. Fig. 6 stellt ein Zeitablaufdiagramm des Rechensystems aus Fig. 5 dar. Fig. 7 stellt ein Verfahren zum Betreiben des Rechensystems aus Fig. 5 dar.

**[0036]** Wie in Fig. 5 gezeigt, kann ein Rechensystem **300** einen Host **310**, einen Schalter **320** und eine Speichervorrichtung **330** umfassen. Der Host **310** kann eine Übermittlungswarteschlange **311**, eine Beendigungswarteschlange **312** und einen Datenpuffer **313** umfassen. Die Komponenten **311** bis **313** des Hosts **310** können jeweils den Komponenten **211** bis **213** des Hosts **210** ähnlich sein. Im Vergleich zu dem Schalter **220** aus Fig. 3 kann der Schalter **320** aus Fig. 5 einen Doorbell-Parser **321** umfassen, einen Befehls-Parser **322**, einen Übermittlungswarteschlangen-Zwischenpuffer **323** und einen Schreibzwischenpuffer **324**. Die Speichervorrichtung **330** kann einen Controller **331** und einen nichtflüchtigen Speicher **339** umfassen. Der Controller **331** kann ein Doorbell-Register **332**, eine DMA-Engine **333**, einen Übermittlungswarteschlangenpuffer **334** und einen Schreibpuffer **335** umfassen. Die Komponenten **331** bis **335** und **339** der Speichervorrichtung **330** aus Fig. 5 können jeweils den Komponenten **231** bis **235** und **239** der Speichervorrichtung **230** aus Fig. 3 ähnlich sein. Auf überflüssige Beschreibungen von Komponenten, die oben beschrieben wurden, wird verzichtet und die nachfolgende Beschreibung wird sich auf einen Unterschied zwischen dem Rechensystem **300** aus Fig. 5 und dem Rechensystem **200** aus Fig. 3 konzentrieren.

**[0037]** Es wird auf Fig. 5, Fig. 6 und Fig. 7 zusammen Bezug genommen; in Vorgang **S303** kann der Host **310** einen Befehl für die Speichervorrichtung **330** in die Übermittlungswarteschlange **311** schreiben oder dort speichern. In Vorgang **S306** kann der Host **310** einen Doorbell für die Übermittlungswarteschlange **311** an den Schalter **320** übertragen (⊕) und der Schalter **320** kann den Doorbell (⊕) empfangen. In Vorgang **S309** kann der Schalter **320** den Doorbell an die Speichervorrichtung **330** übertragen (weiterleiten, weitergeben, erneut übertragen usw.) (⊕). In Vorgang **S313** kann der Schalter **320** den Doorbell lesen oder analysieren (einem Parsing unterziehen). Der Doorbell-Parser **321** des Schalters **320** kann den Doorbell-Parser lesen oder analysieren. In Vorgang **S316** kann der Doorbell-Parser **321**, basierend auf dem Doorbell, den Befehl der Übermittlungswarteschlange **311** von dem Host **310** anfordern und kann die Anforderung an den Host **310** übertragen (⊕). Zum Beispiel kann der Doorbell-Parser **321** auf die Übermittlungswarteschlange **311** von einer oder mehreren Übermittlungswarteschlangen des Hosts **310** zugreifen, die von einer Adresse des Doorbells angegeben wird (dieser entspricht). In Vorgang **S319** kann der Controller **331** der Speichervorrichtung **330** den Doorbell des Doorbell-Registers **332** lesen oder analysieren.

**[0038]** Da der Schalter **320** den Doorbell in Vorgang **S313** nach der Übertragung des Doorbells an die Speichervorrichtung **330** in Vorgang **S309** analysieren kann, können der Schalter **320** und die Speichervorrichtung **330** den Doorbell gleichzeitig analysieren. Alternativ kann nach Empfang des Doorbells von dem Host **310** in Vorgang **S306** der Schalter **320** damit beginnen, den Doorbell in Vorgang **S313** sofort zu analysieren, bevor er den Doorbell in Vorgang **S309** an die Speichervorrichtung **330** überträgt. Diesbezüglich kann der Schalter **320** eine Kopie des Doorbells anfertigen, um den Doorbell sowohl zu analysieren als auch an die Speichervorrichtung **330** zu übertragen, oder einen beliebigen anderen Vorgang durchführen, um eine Analyse bzw. ein Parsing des Doorbells abzuschließen, bevor das Parsing des Doorbells durch die Speichervorrichtung **330** abgeschlossen ist. Folglich kann Latenz aufgrund der

Anwesenheit des Schalters **320** zwischen dem Host **310** und der Speichervorrichtung **330** hinsichtlich der Kommunikation dazwischen reduziert werden.

**[0039]** In Vorgang **S323** kann die Speichervorrichtung **330** den Befehl der Übermittlungswarteschlange **311** von dem Hosts **310** anfordern und kann die Anforderung an den Schalter **320** übertragen (②). Zum Beispiel sind Vorgang **S313** und Vorgang **S316** jeweils Vorgang **S319** und Vorgang **S323** ähnlich. In Vorgang **S326** kann der Schalter **320** den Befehl der Übermittlungswarteschlange **311** von dem Host **310** empfangen (③). Der Empfang des Befehls der Übermittlungswarteschlange **311** von dem Host **310** durch den Schalter **320** kann eine Reaktion auf die Befehlsanforderung des Schalters **320** in Vorgang **S316** sein, und dadurch kann Latenz der Befehlsanforderung der Speichervorrichtung **330** in Vorgang **S323** vorbeugend vermieden werden. Der Übermittlungswarteschlangen-Zwischenpuffer **323** des Schalters **320** kann den empfangenen Befehl speichern. In Vorgang **S329** kann der Schalter **320** den Befehl des Übermittlungswarteschlangen-Zwischenpuffers **323** an die Speichervorrichtung **330** ansprechend auf die Anforderung in Vorgang **S323** übertragen (③). In einem Fall, in dem der Befehl des Übermittlungswarteschlangen-Zwischenpuffers **323** von dem Schalter **320** in Vorgang **S326** empfangen wird, bevor die Befehlsanforderung von der Speichervorrichtung **330** in Vorgang **S323** empfangen wird, kann der Schalter **320** dann den Befehl des Übermittlungswarteschlangen-Zwischenpuffers **323** puffern, bis die Befehlsanforderung von der Speichervorrichtung **330** empfangen wird. Die DMA-Engine **333** des Controllers **331** kann den Befehl empfangen und kann den empfangenen Befehl in dem Übermittlungswarteschlangenpuffer **334** speichern.

**[0040]** Der Doorbell-Parser **321** des Schalters **320** kann basierend auf dem Doorbell den Befehl anfordern, der in der Übermittlungswarteschlange **311** des Hosts **310** geschrieben ist, während mindestens eines Abschnittes einer Zeitspanne des Analysierens des empfangenen Doorbells und des Übertragens des empfangenen Doorbells an die Speichervorrichtung **330**. Der Schalter **320** kann von dem Host **310** den in der Übermittlungswarteschlange **311** des Hosts **310** geschriebenen Befehl während mindestens eines Abschnittes einer Zeitspanne des Übertragens des Doorbells an die Speichervorrichtung **330** und des Empfangens einer Anforderung bezüglich eines Befehls von der Speichervorrichtung **330** empfangen.

**[0041]** Eine Reihenfolge der Vorgänge **S309** bis **S326** ist nicht auf das in **Fig. 7** dargestellte Beispiel beschränkt. Zum Beispiel können der Vorgang **S309** und Vorgang **S313** gleichzeitig durchgeführt werden. Zum Beispiel kann der Vorgang **S326** vor Vorgang **S323** durchgeführt werden. In jedem Fall kann der

Schalter **320** basierend auf dem Doorbell den Befehl anfordern, der in die Übermittlungswarteschlange **311** des Hosts **310** geschrieben wurde, zum Beispiel gleichzeitig mit der Übertragung des empfangenen Doorbells an die Speichervorrichtung **330**. Wie in **Fig. 6** schraffiert dargestellt, kann mindestens ein Abschnitt einer Latenz, die erforderlich ist, damit die Anforderung des Vorgangs **S316**, die von dem Doorbell-Parser **321** übertragen wird, bei dem Host **310** ankommt, mit einer Latenz überlappen, die erforderlich ist, damit der Doorbell des Vorgangs **S309**, der von dem Schalter **320** übertragen wird, bei der Speichervorrichtung **330** ankommt, oder mit einer Latenz, die erforderlich ist, damit die Anforderung des Vorgangs **S323**, die von der Speichervorrichtung **330** übertragen wird, bei dem Schalter **320** ankommt. Wie in **Fig. 6** schraffiert dargestellt, kann mindestens ein Abschnitt einer Latenz, die erforderlich ist, damit der Befehl des Vorgangs **S326**, der von dem Host **310** übertragen wird, bei dem Schalter **320** ankommt, mit einer Latenz überlappen, die erforderlich ist, damit der Doorbell des Vorgangs **S309**, der von dem Schalter **320** übertragen wird, bei der Speichervorrichtung **330** ankommt, oder mit einer Latenz, die erforderlich ist, damit die Anforderung des Vorgangs **S323**, die von der Speichervorrichtung **330** übertragen wird, bei dem Schalter **320** ankommt. Im Vergleich zu dem Schalter **220**, kann der Schalter **320** die Latenz, die erforderlich ist, um den Befehl entsprechend dem Doorbell anzufordern, verschleiern, indem er den Doorbell-Parser **321** verwendet.

**[0042]** In Vorgang **S333** kann der Schalter **320** bestimmen, ob der in Vorgang **S326** empfangene Vorgang ein Schreibbefehl ist. Wenn der in Vorgang **S326** empfangene Schreibbefehl der Schreibbefehl ist (Y), kann der Schalter **320** den Befehl in Vorgang **S336** analysieren bzw. einem Parsing unterziehen, um eine Physical Region Page (PRP) oder eine Scatter Gather List (SGL) zu erhalten. Hier kann die PRP oder die SGL eine Adresse sein, die in dem in Vorgang **S326** empfangenen Befehl umfasst ist, und die Adresse kann einen bestimmten Datenspeicherbereich (Ort) des Datenpuffers **313** oder einen bestimmten Datenspeicherbereich der Speichervorrichtung **330** angeben. In Vorgang **S339** kann der Schalter **320** Schreibdaten des Datenpuffers **313** des Hosts **310** anfordern und kann die Anforderung an den Host **310** übertragen (③). Zum Beispiel kann der Befehls-Parser **322** des Schalters **320** Vorgänge **S333** bis **S339** durchführen. In Vorgang **S343** kann die Speichervorrichtung **330** bestimmen, ob der in Vorgang **S329** empfangene Vorgang ein Schreibbefehl ist. Wenn der in Vorgang **S329** empfangene Schreibbefehl der Schreibbefehl ist (Y), kann die Speichervorrichtung **330** den Befehl in Vorgang **S346** analysieren bzw. einem Parsing unterziehen, um die PRP oder die SGL zu empfangen. In Vorgang **S349** kann die Speichervorrichtung **330** die Schreibdaten des Datenpuffers **313** des Hosts **310** anfordern und



kann die Anforderung an den Switch **320** übertragen (④). Die DMA-Engine **333** kann die Schreibdaten anfordern. In Vorgang **S353** kann der Schalter **320** die Schreibdaten des Datenpuffers **313** von dem Host **310** empfangen (⑤). Der Schreibzwischenpuffer **324** des Schalters **320** kann die empfangenen Schreibdaten speichern. In Vorgang **S356** kann der Schalter **320** die Schreibdaten, die in dem Schreibzwischenpuffer **324** gespeichert sind, ansprechend auf die in Vorgang **S349** empfangene Anforderung an die Speichervorrichtung **330** übertragen (⑤). Die DMA-Engine **333** kann Schreibdaten empfangen und kann die empfangenen Schreibdaten in dem Schreibpuffer **335** speichern. Zum Beispiel können Vorgang **S333**, Vorgang **S336**, Vorgang **S339** und Vorgang **S353** jeweils Vorgang **S343**, Vorgang **S346**, Vorgang **S349** und Vorgang **S356** ähnlich sein.

**[0043]** Wie auf ähnliche Art und Weise oben bezüglich des Doorbells beschrieben, kann der Befehls-Parser **322** des Schalters **320** die in dem Datenpuffer **313** des Hosts **310** gespeicherten Schreibdaten basierend auf dem Befehl oder der in dem Befehl umfassten Adresse vor oder während mindestens eines Abschnitts einer Zeitspanne des Parsings des empfangenen Befehls und des Übertragens des empfangenen Befehls an die Speichervorrichtung **330** anfordern. Der Schalter **320** kann von dem Host **310** die in dem Datenpuffer **313** des Hosts **310** gespeicherten Schreibdaten vor oder während mindestens eines Abschnitts einer Zeitspanne der Übertragung des Befehls an die Speichervorrichtung **330** und des Empfangens einer Anforderung bezüglich der Schreibdaten von der Speichervorrichtung **330** empfangen. Als Folge einer vorbeugenden Verarbeitung des Befehls, kann Latenz aufgrund der Anwesenheit des Schalters **320** zwischen dem Host **310** und der Speichervorrichtung **330** bezüglich der Kommunikation dazwischen reduziert werden.

**[0044]** Eine Reihenfolge der Vorgänge **S329** bis **S356** ist nicht auf das in **Fig. 7** dargestellte Beispiel beschränkt. Zum Beispiel können Vorgang **S329** und Vorgang **S333** gleichzeitig durchgeführt werden. Zum Beispiel kann Vorgang **S353** vor Vorgang **S349** durchgeführt werden. In jedem Fall kann der Schalter **320** basierend auf dem Schreibbefehl den in dem Datenpuffer **313** des Hosts **310** gespeicherten Befehl anfordern, zum Beispiel gleichzeitig mit der Übertragung des empfangenen Schreibbefehls an die Speichervorrichtung **330**. Wie in **Fig. 6** schraffiert dargestellt, kann mindestens ein Abschnitt einer Latenz, die erforderlich ist, damit die Anforderung des Vorgangs **S339**, die von dem Befehls-Parser **322** übertragen wird, bei dem Host **310** ankommt, mit einer Latenz überlappen, die erforderlich ist, damit der Schreibbefehl des Vorgangs **S329**, der von dem Schalter **320** übertragen wird, bei der Speichervorrichtung **330** ankommt, oder mit einer Latenz, die erforderlich ist, damit die Anforderung des Vorgangs **S349**,

die von der Speichervorrichtung **330** übertragen wird, bei dem Schalter **320** ankommt. Wie in **Fig. 6** schraffiert dargestellt, kann mindestens ein Abschnitt einer Latenz, die erforderlich ist, damit die Schreibdaten des Vorgangs **S353**, die von dem Host **310** übertragen werden, bei dem Schalter **320** ankommen, mit einer Latenz überlappen, die erforderlich ist, damit der Schreibbefehl des Vorgangs **S329**, der von dem Schalter **320** übertragen wird, bei der Speichervorrichtung **330** ankommt, oder mit einer Latenz, die erforderlich ist, damit die Anforderung des Vorgangs **S349**, die von der Speichervorrichtung **330** übertragen wird, bei dem Schalter **320** ankommt. Im Vergleich zu dem Schalter **220**, kann der Schalter **320** die Latenz, die erforderlich ist, um die Schreibdaten entsprechend dem Schreibbefehl anzufordern, verschleiern, indem er den Befehls-Parser **322** verwendet.

**[0045]** In Vorgang **S343**, wenn der in Vorgang **S329** empfangene Befehl nicht der Schreibbefehl ist (N), oder nach Vorgang **S356**, in Vorgang **S359**, kann die Speichervorrichtung **330** den Befehl verarbeiten. Zum Beispiel wenn der in Vorgang **S329** empfangene Befehl der Schreibbefehl ist, kann der Controller **331** die Schreibdaten in dem nichtflüchtigen Speicher **339** speichern. Vorgang **S343**, Vorgang **S346**, Vorgang **S349** und Vorgang **S356** können in Vorgang **S359** umfasst sein. Zum Beispiel wenn der in Vorgang **S329** empfangene Befehl ein Lesebefehl ist, kann der Controller **331** die in dem nichtflüchtigen Speicher **339** gespeicherten Lesedaten an den Schalter **320** übertragen. Der Schalter **320** kann die Lesedaten empfangen und kann die empfangenen Lesedaten an den Host **310** übertragen. Zum Beispiel können die Lesedaten in dem Datenpuffer **313** des Hosts **310** gespeichert werden.

**[0046]** In Vorgang **S363** kann die Speichervorrichtung **330** eine Abschlussinformation bezüglich des Befehls an den Schalter **320** übertragen (⑥), und der Schalter **320** kann die Abschlussinformation empfangen (⑥). Wenn der in Vorgang **S326** empfangene Befehl nicht der Schreibbefehl ist (N), oder nach Vorgang **S356**, in Vorgang **S366**, kann der Schalter **320** die Abschlussinformation an den Host **310** übertragen (⑥). Die Abschlussinformation kann in der Beendigungswarteschlange **312** des Hosts **310** gespeichert oder darin geschrieben werden.

**[0047]** In einer Ausführungsform können die Komponenten **321** bis **324** des Schalters **320** in einer Hardwarekonfiguration, in einer Softwarekonfiguration oder in einer Kombination aus Hardware und Software implementiert werden. In dem Fall, in dem Hardware verwendet wird, können die Komponenten **321** bis **324** auf verschiedene Art und Weise implementiert werden unter Verwendung eines Registers, eines Latches, eines Flip-Flops, einer Logikschaltung oder einem Logik-Gate usw. Der Übermittlungswar-

teschlangen-Zwischenpuffer **323** und der Schreibzwischenpuffer **324** können zugeteilten Bereichen auf einem On-Chip-Speicher, der in dem Schalter **320** umfasst ist, entsprechen. In einer Ausführungsform können die Komponenten **332** bis **335** des Controllers **331** in einer Hardwarekonfiguration, in einer Softwarekonfiguration oder in einer Kombination aus Hardware und Software implementiert werden.

**[0048]** In einer Ausführungsform greift der Host **310** möglicherweise nicht direkt auf den Übermittlungswarteschlangen-Zwischenpuffer **323** und den Schreibzwischenpuffer **324** des Schalters **320** sowie den Übermittlungswarteschlangenpuffer **334** und den Schreibpuffer **335** des Controllers **331** der Speichervorrichtung **330** zu. Der Host **310** kann direkt auf das Doorbell-Register **332** des Controllers **331** der Speichervorrichtung **330** durch den Schalter **320** ohne Anforderung von dem Schalter **320** oder der Speichervorrichtung **330** zugreifen. Wenn der Host **310** den Doorbell des Doorbell-Registers **332** aktualisiert, indem er Vorgang **S309** bis Vorgang **S356** aus **Fig. 7** durchführt, können der Schalter **320** und die Speichervorrichtung **330** einen Befehl erhalten, der in die Übermittlungswarteschlange **311** des Hosts **310** geschrieben wurde, oder Schreibdaten, die in dem Datenpuffer **313** gespeichert sind, und können den Befehl oder die Schreibdaten in den obigen Komponenten **323**, **324**, **334** und **335** speichern.

**[0049]** **Fig. 8** stellt ein Blockschaltbild des Rechensystems aus **Fig. 1** dar. **Fig. 9** stellt ein Zeitablaufdiagramm eines Rechensystems aus **Fig. 8** dar. **Fig. 10** stellt ein Verfahren zum Betreiben des Rechensystems aus **Fig. 8** dar. Auf überflüssige Beschreibungen von Komponenten, die oben beschrieben wurden, wird verzichtet und die nachfolgende Beschreibung wird sich auf einen Unterschied zwischen einem Rechensystem **400** aus **Fig. 8** und dem Rechensystem **300** aus **Fig. 5** konzentrieren.

**[0050]** Wie in **Fig. 8** gezeigt, kann das Rechensystem **400** einen Host **410**, einen Schalter **420** und eine Speichervorrichtung **430** umfassen. Der Host **410** kann eine Übermittlungswarteschlange **411**, eine Beendigungswarteschlange **412** und einen Datenpuffer **413** umfassen. Die Komponenten **411** bis **413** des Hosts **410** können jeweils den Komponenten **311** bis **313** des Hosts **310** ähnlich sein. Der Schalter **420** kann einen Befehls-Parser **422** und einen Schreibzwischenpuffer **424** umfassen. Die Komponenten **422** und **424** des Schalters **420** können jeweils den Komponenten **322** und **324** des Schalters **320** ähnlich sein.

**[0051]** Die Speichervorrichtung **430** kann einen Controller **431** und einen nichtflüchtigen Speichers **439** umfassen. Der Controller **431** kann ein Doorbell-Register **432**, eine DMA-Engine **433**, einen Übermittlungswarteschlangen-Controller-Memory-Buffer (SQ

CMB) **434** und einen Schreibpuffer **435** umfassen. Die Komponenten **431**, **432**, **433**, **435** und **439** der Speichervorrichtung **430** können jeweils den Komponenten **331**, **332**, **333**, **335** und **339** der Speichervorrichtung **330** ähnlich sein. Allerdings kann sich der Übermittlungswarteschlangen-Controller-Memory-Buffer **434** von dem Übermittlungswarteschlangenpuffer **334** unterscheiden. Der Host **410** kann einen Befehl direkt in den Übermittlungswarteschlangen-Controller-Memory-Buffer **434** schreiben oder darin speichern. Genauer gesagt kann der Host **410** den Befehl in die Übermittlungswarteschlange **411** schreiben und kann den in die Übermittlungswarteschlange **411** geschriebenen Befehl direkt in den Übermittlungswarteschlangen-Controller-Memory-Buffer **434** schreiben, ohne eine Anforderung des Schalters **420** oder der Speichervorrichtung **430**. Zum Beispiel kann die Übermittlungswarteschlange **311** aus **Fig. 5** auf den Übermittlungswarteschlangen-Controller-Memory-Buffer **434** als Übermittlungswarteschlange platziert werden und kann gleichzeitig auf den Host Memory Buffer HMB als Übermittlungswarteschlange **411** platziert werden. Der gleiche Befehl kann in jeder der Übermittlungswarteschlangen auf der Übermittlungswarteschlange **411** und dem Übermittlungswarteschlangen-Controller-Memory-Buffer **434** gespeichert sein. Die Größe der Übermittlungswarteschlange **411** kann kleiner sein als eine Größe der Übermittlungswarteschlange **311**, und die Größe der Übermittlungswarteschlange auf dem Übermittlungswarteschlangen-Controller-Memory-Buffer **434** kann mit der Größe der Übermittlungswarteschlange **311** identisch sein. Der Host **310** dagegen wird möglicherweise nicht den Befehl der Übermittlungswarteschlange **311** direkt in die Übermittlungswarteschlangenpuffer **334** schreiben oder darin speichern. Stattdessen wird der Befehl möglicherweise erst in dem Übermittlungswarteschlangenpuffer **334** gespeichert, nachdem der Controller **331** Vorgang **S309**, Vorgang **S319**, Vorgang **S323** und Vorgang **S329** durchgeführt hat. Da der Controller **431** den Übermittlungswarteschlangen-Controller-Memory-Buffer **434** umfasst oder unterstützt, auf den der Host **410** direkt zugreifen kann, umfasst der Schalter **420** möglicherweise nicht den Doorbell-Parser **321** und den Übermittlungswarteschlangen-Zwischenpuffer **323**.

**[0052]** Es wird auf **Fig. 8**, **Fig. 9** und **Fig. 10** zusammen Bezug genommen; der Host **410** kann einen Befehl für die Speichervorrichtung **430** in der Übermittlungswarteschlange **411** speichern. In Vorgang **S403** kann der Host **410** den in der Übermittlungswarteschlange **411** gespeicherten Befehl an den Schalter **420** übertragen (①), und der Schalter **420** kann den Befehl empfangen (①). In Vorgang **S406** kann der Schalter **420** den Befehl an die Speichervorrichtung **430** übertragen (①). In Vorgang **S409**, der auf den Vorgang **S403** folgen kann, kann der Host **410** einen Doorbell für die Übermittlungswarteschlange,

die auf dem Übermittlungswarteschlangen-Controller-Memory-Buffer **434** platziert ist, an den Schalter **420** übertragen (②), und der Schalter **420** kann den Doorbell empfangen (②). In Vorgang **S413** kann der Schalter **420** den Doorbell an die Speichervorrichtung **430** übertragen (②). Eine Latenz, die erforderlich ist, damit die Speichervorrichtung **430** einen Befehl und einen Doorbell empfängt, kann kürzer sein als eine Latenz, die erforderlich ist, damit die Speichervorrichtung **230/330** sowohl den Befehl als auch den Doorbell empfängt.

**[0053]** Vorgang **S416** bis Vorgang **S449** in **Fig. 10** sind Vorgang **S333** bis Vorgang **S366** in **Fig. 7** ähnlich. Der Schalter **420** kann Vorgang **S416** bis Vorgang **S423** durchführen, kann Schreibdaten des Datenpuffers **413** anfordern und kann eine Anforderung an den Host **410** übertragen (②). Die Speichervorrichtung **430** kann Vorgang **S426** bis Vorgang **S433** durchführen, kann Schreibdaten von dem Datenpuffer **413** anfordern, und kann eine Anforderung an den Switch **420** übertragen (③). Der Schalter **420** kann den Vorgang **S436** durchführen und kann die Schreibdaten des Datenpuffers **413** von dem Host **410** empfangen (④). Der Schalter **420** kann Vorgang **S436** und Vorgang **S439** durchführen, kann Schreibdaten in dem Schreibzwischenpuffer **424** speichern, und kann die Schreibdaten, die in dem Schreibzwischenpuffer **424** gespeichert sind, an die Speichervorrichtung **430** übertragen (④). Die Speichervorrichtung **430** kann Vorgang **S439** bis Vorgang **S446** durchführen und kann eine Abschlussinformation bezüglich des Befehls an den Schalter **420** übertragen (⑤). In Vorgang **S449** kann der Schalter **420** die Abschlussinformation an den Host **410** übertragen (⑤).

**[0054]** **Fig. 11** stellt ein Blockschaltbild des Rechensystems aus **Fig. 1** dar. **Fig. 12** stellt ein Zeitablaufdiagramm eines Rechensystems aus **Fig. 11** dar. **Fig. 13** stellt ein Verfahren zum Betreiben des Rechensystems aus **Fig. 11** dar. Auf überflüssige Beschreibungen von Komponenten, die oben beschrieben wurden, wird verzichtet und die nachfolgende Beschreibung wird sich auf einen Unterschied zwischen einem Rechensystem **500** aus **Fig. 11** und dem Rechensystem **300** aus **Fig. 5** konzentrieren.

**[0055]** Wie in **Fig. 11** gezeigt, kann das Rechensystem **500** einen Host **510**, einen Schalter **520** und eine Speichervorrichtung **530** umfassen. Der Host **510** kann eine Übermittlungswarteschlange **511**, eine Beendigungswarteschlange **512** und einen Datenpuffer **513** umfassen. Die Komponenten **511** bis **513** des Hosts **510** aus **Fig. 11** können jeweils den Komponenten **311** bis **313** des Hosts **310** aus **Fig. 5** ähnlich sein. Der Schalter **520** kann einen Doorbell-Parser **521** und einen Übermittlungswarteschlangen-Zwischenpuffer **523** umfassen. Die Komponenten **521** und **523** des Schalters **520** aus **Fig. 11**

können jeweils den Komponenten **321** und **323** des Schalters **320** aus **Fig. 5** ähnlich sein.

**[0056]** Die Speichervorrichtung **530** kann einen Controller **531** und einen nichtflüchtigen Speichers **539** umfassen. Der Controller **531** kann ein Doorbell-Register **532**, eine DMA-Engine **533**, einen Übermittlungswarteschlangenpuffer **534** und einen Schreib-Controller-Memory-Buffer (Schreib-CMB) **535** umfassen. Die Komponenten **531**, **532**, **533**, **534** und **539** der Speichervorrichtung **530** aus **Fig. 11** können jeweils den Komponenten **331**, **332**, **333**, **334** und **339** der Speichervorrichtung **330** aus **Fig. 5** ähnlich sein. Allerdings kann sich der Schreib-Controller-Memory-Buffer **535** von dem Schreibpuffer **335** unterscheiden. Der Host **510** kann die Schreibdaten direkt in den Schreib-Controller-Memory-Buffer **535** speichern. Genauer gesagt kann der Host **510** die Schreibdaten in den Datenpuffer **513** speichern und kann die Schreibdaten, die in dem Datenpuffer **513** gespeichert sind, direkt in den Schreib-Controller-Memory-Buffer **535** durch den Schalter **520** speichern, ohne eine Anforderung des Schalters **520** oder der Speichervorrichtung **530**. Zum Beispiel kann der Datenpuffer **513** aus **Fig. 5** auf dem Schreib-Controller-Memory-Buffer **535** als Datenpuffer platziert werden und kann gleichzeitig auf dem Host Memory Buffer HMB als Datenpuffer **513** platziert werden. Der gleiche Befehl kann in allen der Datenpuffer auf dem Datenpuffer **513** und dem Schreib-Controller-Memory-Buffer **535** gespeichert werden. Die Größe des Datenpuffers **513** kann kleiner sein als die Größe des Datenpuffers **313**, und die Größe des Datenpuffers auf dem Schreib-Controller-Memory-Buffer **535** kann mit der Größe des Datenpuffers **313** identisch sein. Dagegen speichert der Host **310** die Schreibdaten des Datenpuffers **313** möglicherweise nicht direkt in den Schreibpuffer **335**. Stattdessen wird der Befehl möglicherweise erst in dem Schreibpuffer **335** gespeichert, nachdem der Controller **331** Vorgang **S343**, Vorgang **S346**, Vorgang **S349** und Vorgang **S356** durchgeführt hat. Da der Controller **531** den Schreib-Controller-Memory-Buffer **535** umfasst, auf den der Host **510** direkt zugreifen kann, umfasst der Schalter **520** möglicherweise nicht den Befehls-Parser **322** und den Schreibzwischenpuffer **324**.

**[0057]** Es wird auf **Fig. 11**, **Fig. 12** und **Fig. 13** gemeinsam Bezug genommen; Vorgang **S503**, Vorgang **S506** und Vorgang **S509** können jeweils Vorgang **S303**, Vorgang **S353** und Vorgang **S356** ähnlich sein. In Vorgang **S503** kann der Host **510** einen Befehl für die Speichervorrichtung **530** in der Übermittlungswarteschlange **511** speichern. Wenn der in Vorgang **S503** empfangene Befehl der Schreibbefehl ist, kann der Host **510**, in Vorgang **S506** die Schreibdaten an den Schalter **520** übertragen (①). In Vorgang **S509** kann der Schalter **520** die Schreibdaten an die Speichervorrichtung **530** übertragen (①). Eine Latenz, die erforderlich ist, damit die Speichervor-

richtung **530** die Schreibdaten empfängt, kann kürzer sein als eine Latenz, die erforderlich ist, damit die Speichervorrichtung **230/330/430** die Schreibdaten empfängt. Wenn der Befehl nicht der Schreibbefehl ist, können Vorgang **S506** und Vorgang **S509** ausgelassen werden.

**[0058]** Vorgang **S513** bis Vorgang **S536** sind Vorgang **S306** bis Vorgang **S329** ähnlich. Der Host **510** kann Vorgang **S513** durchführen und kann einen Doorbell für die Übermittlungswarteschlange **511** an den Schalter **520** übertragen (②). Der Schalter **520** kann Vorgang **S513** bis Vorgang **S523** durchführen, kann einen Doorbell an die Speichervorrichtung **530** übertragen (②), kann den Befehl der Übermittlungswarteschlange **511** anfordern und kann die Anforderung an den Host **510** übertragen (②). Die Speichervorrichtung **530** kann Vorgang **S516**, Vorgang **S526** und Vorgang **S529** durchführen, kann den Befehl der Übermittlungswarteschlange **511** anfordern und kann die Anforderung an den Schalter **520** übertragen (③). Der Schalter **520** kann Vorgang **S533** und Vorgang **S536** durchführen, kann den Befehl der Übermittlungswarteschlange **511** von dem Host **510** empfangen (④) und kann den Befehl, der in dem Übermittlungswarteschlangen-Zwischenpuffer **523** gespeichert ist, an die Speichervorrichtung **530** übertragen (④). Die Speichervorrichtung **530** kann Vorgang **S536** durchführen und kann den Befehl (④) empfangen. Die Speichervorrichtung **530** kann Vorgang **S539** und Vorgang **S543** durchführen und kann eine Abschlussinformation bezüglich des Befehls an den Schalter **520** übertragen (⑤). In Vorgang **S546** kann der Schalter **520** die Abschlussinformation an den Host **510** übertragen (⑤).

**[0059]** Fig. 14 stellt ein Blockschaltbild des Rechensystems aus Fig. 1 dar. Fig. 15 stellt ein Zeitablaufdiagramm des Rechensystems aus Fig. 14 dar. Auf überflüssige Beschreibungen von Komponenten, die oben beschrieben wurden, wird verzichtet und die nachfolgende Beschreibung wird sich auf einen Unterschied zwischen einem Rechensystem **600** aus Fig. 14 und den Rechensystemen **300** aus Fig. 5, **400** aus Fig. 8 und **500** aus Fig. 11 konzentrieren.

**[0060]** Das Rechensystem **600** kann einen Host **610**, einen Schalter **620** und eine Speichervorrichtung **630** umfassen. Der Host **610** kann eine Übermittlungswarteschlange **611**, eine Beendigungswarteschlange **612** und einen Datenpuffer **613** umfassen. Die Komponenten **611**, **612** und **613** des Hosts **610** in Fig. 14 können jeweils den Komponenten **411** und **412** des Hosts **410** aus Fig. 8 und der Komponente **513** des Hosts **510** aus Fig. 11 ähnlich sein. Der Schalter **620** aus Fig. 11 kann dem Schalter **220** aus Fig. 3 ähnlich sein.

**[0061]** Die Speichervorrichtung **630** kann einen Controller **631** und einen nichtflüchtigen Speicher **639**

umfassen. Der Controller **631** kann ein Doorbell-Register **632** umfassen, einen Übermittlungswarteschlangen-Controller-Memory-Buffer **634** und einen Schreib-Controller-Memory-Buffer **635**. Die Komponenten **631**, **632**, **634**, **635** und **639** der Speichervorrichtung **630** aus Fig. 11 können jeweils den Komponenten **331** und **332** der Speichervorrichtung **330** aus Fig. 5, der Komponente **434** der Speichervorrichtung **430** aus Fig. 8, der Komponente **535** der Speichervorrichtung **530** aus Fig. 11 und der Komponente **339** der Speichervorrichtung **330** aus Fig. 5 ähnlich sein. Obwohl in den Zeichnungen nicht dargestellt, kann der Controller **631** ferner eine DMA-Engine umfassen. Da der Controller **631** den Übermittlungswarteschlangen-Controller-Memory-Buffer **634** und den Schreib-Controller-Memory-Buffer **635** umfasst, auf den durch den Host **610** direkt zugegriffen werden kann, umfasst der Schalter **620** möglicherweise nicht die Komponenten **321** bis **324** des Schalters **320**.

**[0062]** Es wird auf Fig. 14 und Fig. 15 zusammen Bezug genommen; der Host **610** kann einen Befehl für die Speichervorrichtung **630** in der Übermittlungswarteschlange **611** speichern. Der Host **610** kann den Befehl der Übermittlungswarteschlange **611** an den Schalter **620** übertragen (①). Der Schalter **620** kann den Befehl an die Speichervorrichtung **630** übertragen (①). Wenn der Befehl ein Schreibbefehl ist, kann der Host **610** die Schreibdaten an den Schalter **620** übertragen (②). Der Schalter **620** kann die Schreibdaten an die Speichervorrichtung **630** übertragen (②). Wenn der Befehl nicht der Schreibbefehl ist, kann die Übertragung der Schreibdaten ausgelassen werden. Wie in Fig. 14 und Fig. 15 dargestellt, können die Schreibdaten vor dem Befehl übertragen werden, oder sowohl die Schreibdaten als auch der Befehl können gleichzeitig übertragen werden. Nachdem der Befehl und die Schreibdaten an den Schalter **620** übertragen wurden, kann der Host **610** einen Doorbell für die Übermittlungswarteschlange **611** an den Schalter **620** übertragen (③). Der Schalter **620** kann den Doorbell an die Speichervorrichtung **630** übertragen ((3)). Eine Latenz, die erforderlich ist, damit die Speichervorrichtung **630** einen Befehl, Schreibdaten und einen Doorbell empfängt, kann kürzer sein als eine Latenz, die erforderlich ist, damit die Speichervorrichtung **230/330/530** den Befehl, die Schreibdaten und den Doorbell allesamt empfängt. Die Speichervorrichtung **630** kann einen Befehl verarbeiten. Die Speichervorrichtung **630** kann eine Abschlussinformation bezüglich des Befehls an den Schalter **620** übertragen (④). Der Schalter **620** kann die Abschlussinformation an den Host **610** übertragen (④).

**[0063]** Fig. 16 stellt ein Blockschaltbild des Rechensystems aus Fig. 1 dar. Fig. 17 stellt ein Zeitablaufdiagramm des Rechensystems aus Fig. 16 dar. Fig. 18 stellt ein Verfahren zum Betreiben des Rechensystems aus Fig. 16 dar. Auf überflüssige Be-

schreibungen von Komponenten, die oben beschrieben wurden, wird verzichtet und die nachfolgende Beschreibung wird sich auf einen Unterschied zwischen einem Rechensystem **700** aus **Fig. 16** und den Rechensystemen **300** aus **Fig. 5** und **600** aus **Fig. 14** konzentrieren.

**[0064]** Das Rechensystem **700** kann einen Host **710**, einen Schalter **720** und eine Speichervorrichtung **730** umfassen. Der Host **710** kann eine Übermittlungswarteschlange **711**, eine Beendigungswarteschlange **712** und einen Datenpuffer **713** umfassen. Die Komponenten **711** bis **713** des Hosts **710** in **Fig. 16** können jeweils den Komponenten **611** bis **613** des Hosts **610** aus **Fig. 14** ähnlich sein. Die Speichervorrichtung **730** kann einen Controller **731** und einen nichtflüchtigen Speicher **739** umfassen. Der Controller **731** kann ein Doorbell-Register **732**, eine DMA-Engine **733**, einen Übermittlungswarteschlangenpuffer **734** und einen Schreibpuffer **735** umfassen. Die Komponenten **731**, **732**, **733**, **734**, **735** und **739** der Speichervorrichtung **730** aus **Fig. 16** können jeweils den Komponenten **331**, **332**, **333**, **334**, **335** und **339** der Speichervorrichtung **330** aus **Fig. 5** ähnlich sein.

**[0065]** Der Schalter **720** kann einen Übermittlungswarteschlangen-Controller-Memory-Buffer **723** und einen Schreib-Controller-Memory-Buffer **724** umfassen. Ein Vorgang des Übermittlungswarteschlangen-Controller-Memory-Buffers **723** kann dem Vorgang des Übermittlungswarteschlangen-Controller-Memory-Buffers **434** ähnlich sein. Der Host **710** kann einen Befehl direkt in den Übermittlungswarteschlangen-Controller-Memory-Buffer **723** des Schalters **720** schreiben oder darin speichern. Im Gegensatz dazu kann der Host **710** den Befehl möglicherweise nicht direkt in den Übermittlungswarteschlangenpuffer **734** des Controllers **731** der Speichervorrichtung **730** schreiben oder darin speichern. Ein Vorgang des Schreib-Controller-Memory-Buffers **724** kann dem Vorgang des Schreib-Controller-Memory-Buffers **535** ähnlich sein. Der Host **710** kann die Schreibdaten direkt in den Schreib-Controller-Memory-Buffer **724** des Schalters **720** speichern. Im Gegensatz dazu kann der Host **710** die Schreibdaten möglicherweise nicht direkt in den Schreibpuffer **735** des Controllers **731** der Speichervorrichtung **730** speichern.

**[0066]** Es wird auf **Fig. 15**, **Fig. 16** und **Fig. 17** zusammen Bezug genommen; in Vorgang **S703** kann der Host **710** einen Befehl der Übermittlungswarteschlange **711** an den Schalter **720** übertragen (①). Der Übermittlungswarteschlangen-Controller-Memory-Buffer **723** kann den in Vorgang **S703** empfangenen Befehl speichern. Wenn der Befehl ein Schreibbefehl ist, kann der Host **710** in Vorgang **S706** die Schreibdaten des Datenpuffers **713** an den Schalter **720** übertragen (②). Der Schreib-Controller-Memory-Buffer **724** kann die in Vorgang **S706** empfangenen Schreibdaten speichern. Anders als bei dem

in **Fig. 16** bis **Fig. 18** dargestellten Beispiel, können die Schreibdaten vor dem Befehl übertragen werden, oder sowohl die Schreibdaten als auch der Befehl können gleichzeitig übertragen werden. Nach Vorgang **S703** und Vorgang **S706** kann der Host **710** in Vorgang **S709** einen Doorbell für die Übermittlungswarteschlange **711** an den Schalter **720** übertragen (③). Vorgang **S713** kann Vorgang **S309** ähnlich sein. Der Schalter **720** kann Vorgang **S713** durchführen und kann den Doorbell an die Speichervorrichtung **730** übertragen (④).

**[0067]** Vorgang **S716**, Vorgang **S719**, Vorgang **S723**, Vorgang **S726**, Vorgang **S729**, Vorgang **S733**, Vorgang **S736**, Vorgang **S739**, Vorgang **S743** und Vorgang **S746** aus **Fig. 18** sind jeweils Vorgang **S319**, Vorgang **S323**, Vorgang **S329**, Vorgang **S343**, Vorgang **S346**, Vorgang **S349**, Vorgang **S356**, Vorgang **S359**, Vorgang **S363** und Vorgang **S366** aus **Fig. 7** ähnlich. Die Speichervorrichtung **730** kann Vorgang **S716** und Vorgang **S719** durchführen, kann den Befehl der Übermittlungswarteschlange **711** anfordern und kann die Anforderung an den Schalter **720** übertragen (⑤). Der Schalter **720** kann Vorgang **S723** durchführen und kann den Befehl des Übermittlungswarteschlangen-Controller-Memory-Buffers **723** an die Speichervorrichtung **730** übertragen (⑥). Die Speichervorrichtung **730** kann Vorgang **S726**, Vorgang **S729** und Vorgang **S733** durchführen, kann die Schreibdaten von dem Datenpuffer **713** anfordern, und kann die Anforderung an den Schalter **720** übertragen (⑦). Der Schalter **720** kann Vorgang **S736** durchführen und kann die Schreibdaten des Schreib-Controller-Memory-Buffers **724** an die Speichervorrichtung **730** übertragen (⑧). Die Speichervorrichtung **730** kann Vorgang **S739** und Vorgang **S743** durchführen und kann die Abschlussinformation bezüglich des Befehls an den Schalter **720** übertragen (⑨). In Vorgang **S746** kann der Schalter **720** die Abschlussinformation an den Host **710** übertragen (⑩).

**[0068]** **Fig. 19** stellt ein Blockdiagramm einer Speichervorrichtung gemäß einer Ausführungsform dar.

**[0069]** Die Speichervorrichtung **830** aus **Fig. 19** wird unter Bezugnahme auf **Fig. 3** bis **Fig. 18** beschrieben. Die Speichervorrichtung **830** kann eine der Speichervorrichtungen **230** bis **730** sein. Die Speichervorrichtung **830** kann einen Controller **831**, einen Speicherpuffer **838** und nichtflüchtige Speicher **839** umfassen.

**[0070]** Der Controller **831** kann mit einem SoC, einer ASIC oder einer FPGA usw. implementiert sein. Der Controller **831** kann einen Prozessor **831-1**, einen On-Chip-Speicher **831-2**, eine nichtflüchtige Speicherinterfaceschaltung **831-3**, eine externe Interfaceschaltung **831-4**, eine DMA-Engine **831-5** und einen Puffer-Interfaceschaltkreis **831-6** umfassen. Der

Prozessor **831-1** kann die Komponenten **831-2** bis **831-6** des Controllers **831** steuern. Der Prozessor **831-1** kann mindestens einen oder mehrere Kerne (z. B. einen homogenen Mehrkern bzw. Multi-Core oder einen heterogenen Mehrkern) und einen internen Cache-Speicher umfassen. Der Prozessor **831-1** kann einen Programmcode, eine Software, ein Anwendungsprogramm usw. ausführen, die auf den On-Chip-Speicher **831-2** oder in den Speicherpuffer **838** geladen wurden.

**[0071]** Der On-Chip-Speicher **831-2** kann ein Latch, ein Register, einen SRAM, einen DRAM, einen dynamischen Thyristor-Direktzugriffsspeicher (TRAM), einen Tightly Coupled Memory (TCM), usw. umfassen. Ein Flash-Translation-Layer (FTL) kann in den On-Chip-Speicher **831-2** geladen werden. Die FTL kann eine Zuordnung bzw. Mapping zwischen einer logischen Adresse, die von außen bereitgestellt wird, und einer physischen Adresse der nichtflüchtigen Speicher **839** verwalten. Die FTL kann ferner einen Speicherbereinigungsvorgang bzw. eine Garbage Collection und dergleichen, sowie den Adresszuordnungsvorgang durchführen.

**[0072]** Die nichtflüchtige Speicherinterfaceschaltung **831-3** kann mit den nichtflüchtigen Speichern **839** entsprechend einem Interfaceprotokoll wie beispielsweise Toggle Double Data Rate (DDR) kommunizieren. Die nichtflüchtige Speicherinterfaceschaltung **831-3** kann mit einem oder mehreren nichtflüchtigen Speichern **839** durch einen Kanal CHI kommunizieren, und kann mit einem oder mehreren nichtflüchtigen Speichern **839** durch einen Kanal CHn kommunizieren (wobei n eine ganze Zahl größer gleich 2 ist). Die Anzahl an Kanälen CHI bis CHn zwischen dem Controller **831** und den nichtflüchtigen Speichern **839** kann eins oder mehr betragen, die Anzahl an nichtflüchtigen Speichern, die einem Kanal zugeordnet sind, kann eins oder mehr betragen, und jeder der nichtflüchtigen Speicher **839** kann einer der nichtflüchtigen Speicher **239** bis **739** sein, die unter Bezugnahme auf **Fig. 3** bis **Fig. 18** beschrieben wurden. Gesteuert durch den Prozessor **831-1** kann die nichtflüchtige Speicherinterfaceschaltung **831-3** die Schreibdaten, die von außerhalb übertragen wurden (z. B. dem Host **11**, dem Root-Complex **120**, den elektronischen Vorrichtungen **141**, **151**, **153**, **161** und **162** und den Schaltern **220** bis **720**; siehe **Fig. 2** bis **Fig. 18**), an die nichtflüchtigen Speicher **839** übertragen, und die Schreibdaten können in den nichtflüchtigen Speichern **839** gespeichert werden. Die nichtflüchtige Speicherinterfaceschaltung **831-3** kann, gesteuert durch den Prozessor **831-1**, die von den nichtflüchtigen Speichern **839** übertragenen Lesedaten empfangen.

**[0073]** Der externe Interfaceschaltkreis **831-4** kann mit außen (z. B. dem Host **11**, dem Root-Complex **120**, den elektronischen Vorrichtungen **141**,

**151**, **153**, **161** und **162**, und den Schaltern **220** bis **720**: siehe **Fig. 2** bis **Fig. 18**) entsprechend einem Kommunikationsschnittstellenprotokoll kommunizieren. Zum Beispiel kann das Interfaceprotokoll USB, SCSI, PCIe, M-PCIe, NVMe, ATA, PATA, SATA, SAS, IDE, UFS, Firewire usw. sein.

**[0074]** Gesteuert durch den Prozessor **831-1** kann die DMA-Engine **831-3** direkt auf verschiedene Speicherpuffer (z. B. **130**, HMB, **211**, **213**, **323**, **324**, **424**, **523**, **723** und **724**) der Vorrichtungen zugreifen (z. B. Host **11**, Root-Complex **120**, elektronische Vorrichtungen **141**, **151**, **153**, **161** und **162**, und die Schalter **220** bis **720**: siehe **Fig. 2** bis **Fig. 18**). Zum Beispiel kann die DMA-Engine **831-5** auf eine der obigen Speicherpuffer zugreifen, kann einen Befehl empfangen, kann Schreibdaten empfangen und kann Lesedaten der Speichervorrichtung **830** übertragen. Die DMA-Engine **831-5** kann auf verschiedene Speicherpuffer **234** bis **734** zugreifen und **235** bis **735** der Speichervorrichtungen **230** bis **830**, den On-Chip-Speicher **831-2** und den Speicherpuffer **838** zugreifen und kann mit diesen Daten austauschen.

**[0075]** Der Puffer-Interfaceschaltkreis **831-6** kann mit dem Speicherpuffer **838** entsprechend einem Interfaceprotokoll wie beispielsweise einem DDR-Standard kommunizieren. Der Puffer-Interfaceschaltkreis **831-6** kann Daten mit dem Speicherpuffer **838** gesteuert durch den Prozessor **831-1** austauschen. Der Speicherpuffer **838** kann einen Latch, ein Register, einen SRAM, einen DRAM, einen TRAM, einen TCM usw. umfassen. Zum Beispiel kann der Speicherpuffer **838** außerhalb des Controllers **831** ausgebildet sein, oder er kann in der Speichervorrichtung **830** platziert sein. In einem anderen Beispiel ist der Speicherpuffer **838** möglicherweise nicht in der Speichervorrichtung **830** umfasst. In der Konfiguration, in der der Speicherpuffer **838** in der Speichervorrichtung **830** umfasst ist, kann der Prozessor **831-1** als Cache-Speicher den Speicherpuffer **838** sowie den On-Chip-Speicher **831-2** verwenden.

**[0076]** In einer Ausführungsform kann der Controller **831** die nachfolgenden Vorgänge, die mit den unter Bezugnahme auf **Fig. 5** bis **Fig. 18** beschriebenen Speichervorrichtungen **330** bis **730** assoziiert sind, durchführen: **S309**, **S319**, **S323**, **S329**, **S343**, **S346**, **S349**, **S356**, **S359**, **S363**, **S406**, **S413**, **S426**, **S429**, **S433**, **S439**, **S443**, **S446**, **S509**, **S516**, **S526**, **S529**, **S536**, **S539**, **S543**, **S713**, **S716**, **S719**, **S723**, **S726**, **S729**, **S733**, **S736**, **S739** und **S743**. In einer anderen Ausführungsform kann ein Vorgang der Speichervorrichtung **830**, welche keine Endpunktvorrichtung sondern eine Zwischenvorrichtung ist, den Vorgängen der Schalter **320** bis **720** ähnlich sein, die unter Bezugnahme auf **Fig. 5** bis **Fig. 18** beschrieben wurden. In diesem Fall kann der Controller **831** ferner die Komponenten **321** bis **324**, **422**, **424**, **521**, **523**, **723** und **724** der Schalter **320** bis **720** umfassen. Der

Controller **831** kann die nachfolgenden Vorgänge, die mit den unter Bezugnahme auf **Fig. 5** bis **Fig. 18** beschriebenen Schaltern **320** bis **720** assoziiert sind, durchführen: **306**, **S309**, **S313**, **S316**, **S323**, **S326**, **S329**, **S333**, **S336**, **S339**, **S349**, **S353**, **S356**, **S363**, **S366**, **S403**, **S406**, **S409**, **S413**, **S416**, **S419**, **S423**, **S433**, **S436**, **S439**, **S446**, **S449**, **S506**, **S509**, **S513**, **S516**, **S519**, **S523**, **S529**, **S533**, **S536**, **S543**, **S546**, **S703**, **S706**, **S709**, **S713**, **S719**, **S723**, **S733**, **S736**, **S743** und **S746**.

**[0077]** **Fig. 20** stellt ein Blockdiagramm einer Rechenvorrichtung gemäß einer Ausführungsform dar. In den Rechensystemen **10** und **100** bis **700**, die unter Bezugnahme auf **Fig. 1** bis **Fig. 19** beschrieben wurden, können verschiedene Ausführungsformen des erfinderischen Konzepts bei einer Rechenvorrichtung **1000** angewandt werden. Die Rechenvorrichtung **1000** kann einen Hauptprozessor **1100**, einen Speicher **1200**, eine Nutzerschnittstelle **1300**, eine Speichervorrichtung **1400**, einen Kommunikationsblock **1500** und einen Grafikprozessor **1600** umfassen. Zum Beispiel kann die Rechenvorrichtung **1000** als „mobile Vorrichtung“ bezeichnet werden.

**[0078]** Der Hauptprozessor **1100** kann Gesamtvorgänge der Rechenvorrichtung **1000** steuern. Der Hauptprozessor **1100** kann eingerichtet sein, verschiedene Arten arithmetischer Operationen oder logischer Operationen zu verarbeiten. Der Hauptprozessor **1100** kann mit einer dedizierten Logikschaltung, einer FPGA, einer ASIC, einem SoC usw. implementiert werden, welcher einen oder mehrere Prozessorkerne umfasst. Der Hauptprozessor **1100** kann mit einer zentralen Verarbeitungseinheit, einem Mikroprozessor, einem Universal-Prozessor, einem Spezial-Prozessor oder einem Anwendungsprozessor implementiert werden. Zum Beispiel können der Host **11** und **210** bis **710** und der Prozessor **110** jeweils dem Hauptprozessor **1100** entsprechen.

**[0079]** Der Speicher **1200** kann vorübergehend Daten speichern, die für einen Vorgang der Rechenvorrichtung **1000** verwendet werden. Der Speicher **1200** kann Daten speichern, die von dem Hauptprozessor **1100** verarbeitet wurden oder von diesem noch verarbeitet werden. Der Speicher **130** kann zum Beispiel dem Speicher **1200** entsprechen.

**[0080]** Die Benutzeroberfläche **1300** kann eine Kommunikationsmediation zwischen einem Nutzer und der Rechenvorrichtung **1000** gesteuert durch den Hauptprozessor **1100** durchführen. Zum Beispiel kann die Benutzeroberfläche **1300** Eingaben von einer Tastatur, einer Maus, einem Tastenfeld, einer Schaltfläche, einem Touch-Panel, einem Touchscreen, einem Touchpad, einem Touchball, einer Kamera, einem Gyroskopsensor, einem Vibrationssensor usw. verarbeiten. Zudem kann die Benutzeroberfläche **1300** Ausgaben verarbeiten, die einer Anzei-

gevorrichtung, einem Lautsprecher, einem Elektromotor usw. bereitgestellt werden sollen.

**[0081]** Die Speichervorrichtung **1400** kann ein Speichermedium umfassen, das fähig ist, Daten unabhängig davon zu speichern, ob eine Leistung zugeführt wird. Zum Beispiel kann die Speichervorrichtung **1400** eine der elektronischen Vorrichtungen **12**, **13**, **141**, **142**, **151** bis **154** und **161** bis **163** sein, die Schalter **220** bis **720** oder der Speichervorrichtungen **230** bis **730** sein, die unter Bezugnahme auf **Fig. 1** bis **Fig. 19** beschrieben wurden. Die Speichervorrichtung **1400** kann eine Zwischenvorrichtung sein; ferner können eine andere Zwischenvorrichtung und eine andere Endpunktvorrichtung, die mit der Speichervorrichtung **1400** verbunden ist, in der Rechenvorrichtung **1000** umfasst sein.

**[0082]** Der Kommunikationsblock **1500** kann mit einer externen Vorrichtung/System der Rechenvorrichtung **1000** gesteuert durch den Hauptprozessor **1100** kommunizieren. Der Kommunikationsblock **1500** kann zum Beispiel mit der externen Vorrichtung/System der Rechenvorrichtung **1000** basierend auf mindestens einem verschiedener verdrahteter Kommunikationsprotokolle kommunizieren, wie beispielsweise Ethernet, Transfer Control Protocol/Internet Protocol (TCP/IP), einem Universal Serial Bus (USB) und Firewire und/oder mindestens einem verschiedener drahtloser Kommunikationsprotokolle wie beispielsweise Long Term Evolution (LTE), Worldwide Interoperability For Microwave Access (WiMax), Global System For Mobile Communications (GSM), Code Division Multiple Access (CDMA), Bluetooth, Near Field Communication (NFC), Wireless-Fidelity (Wi-Fi) und Radio Frequency Identification (RFID).

**[0083]** Der Grafikprozessor **1600** kann eine Graphics Processing Unit (GPU) sein, und kann eine Mehrzahl von Prozessorkernen (z. B. Grafikprozessoreinheiten) umfassen. Die Prozessorkerne, die in dem Grafikprozessor **1600** umfasst sind, können Grafikdaten schnell parallel verarbeiten. Zum Beispiel kann der Grafikprozessor **1600** verschiedene Grafikvorgänge, wie beispielsweise Pixelshader, Super-Sampling und Farbraumtransformation unter Verwendung der Prozessorkerne verarbeiten.

**[0084]** Der Hauptprozessor **1100**, der Speicher **1200**, die Benutzeroberfläche **1300**, die Speichervorrichtung **1400**, der Kommunikationsblock **1500** und der Grafikprozessor **1600** können jeweils mit einer Vorrichtung einer Schaltungsebene, einer Chipebene und/oder einer Package-Ebene implementiert werden, sodass sie in der Rechenvorrichtung **1000** angebracht ist bzw. sind. Alternativ können der Hauptprozessor **1100**, der Speicher **1200**, die Benutzeroberfläche **1300**, die Speichervorrichtung **1400**, der Kommunikationsblock **1500** und der Grafikprozessor **1600** jeweils mit einer unabhängigen elektronischen Vorrich-

tung implementiert werden, sodass sie innerhalb der Rechenvorrichtung **1000** zusammengebaut sind. Die angebrachten oder zusammengebauten Komponenten können miteinander über einen Bus **1700** verbunden sein.

**[0085]** Der Bus **1700** kann einen Kommunikationspfad mit Drähten zwischen den Komponenten der Rechenvorrichtung **1000** vorsehen. Die Komponenten der Rechenvorrichtung **1000** können Daten miteinander basierend auf einem Busformat des Buses **1700** austauschen. Das Busformat kann zum Beispiel eines oder mehrere verschiedener Protokolle wie beispielsweise PCIe, NVMe, SCSI, ATA, PATA, SATA, SAS und UFS umfassen.

**[0086]** Gemäß einer Ausführungsform des erfinderischen Konzepts kann eine Kommunikationsgeschwindigkeit zwischen einer Endpunktvorrichtung und einem Host verbessert werden, indem ein Puffer einer elektronischen Vorrichtung verwendet wird, die zwischen der Endpunktvorrichtung und dem Host platziert ist.

**[0087]** Während das erfinderische Konzept unter Bezugnahme auf Ausführungsformen desselben beschrieben wurde, wird es für einen Fachmann ersichtlich sein, dass verschiedene Veränderungen und Abwandlungen daran vorgenommen werden können, ohne von dem Geist und dem Umfang des erfinderischen Konzepts wie in den nachfolgenden Ansprüchen dargelegt abzuweichen.



**ZITATE ENTHALTEN IN DER BESCHREIBUNG**

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

**Zitierte Patentliteratur**

- KR 1020190099851 [0001]

**Patentansprüche**

1. Rechensystem aufweisend:

einen Host;

eine erste elektronische Vorrichtung, die mit dem Host gekoppelt ist, und

eine zweite elektronische Vorrichtung, die mit der ersten elektronischen Vorrichtung gekoppelt ist, wobei die zweite elektronische Vorrichtung eingerichtet ist, mit dem Host durch die erste elektronische Vorrichtung zu kommunizieren,

wobei die erste elektronische Vorrichtung eingerichtet ist:

den Host aufzufordern, einen Schreibbefehl zu übertragen, der in eine Übermittlungswarteschlange des Hosts geschrieben ist, basierend auf einem Doorbell, der von dem Host empfangen wird;

den von dem Host empfangenen Schreibbefehl zu speichern;

den Host aufzufordern, Schreibdaten des Schreibbefehls, die in einem Datenpuffer des Hosts gespeichert sind, zu übertragen; und

die von dem Host empfangenen Schreibdaten zu speichern.

2. Rechensystem nach Anspruch 1, wobei die erste elektronische Vorrichtung ein Peripheral-Component-Interconnect-express(PCIe)-Schalter ist, und wobei die zweite elektronische Vorrichtung eine Non-Volatile-Memory-express(NVMe)-Vorrichtung ist.

3. Rechensystem nach Anspruch 1, wobei die erste elektronische Vorrichtung aufweist:

einen Übermittlungswarteschlangen-Zwischenpuffer, der eingerichtet ist, den Schreibbefehl zu speichern; und

einen Schreibzwischenpuffer, der eingerichtet ist, die Schreibdaten zu speichern.

4. Rechensystem nach Anspruch 3, wobei der Host nicht direkt auf den Übermittlungswarteschlangen-Zwischenpuffer und den Schreibzwischenpuffer der ersten elektronischen Vorrichtung zugreift.

5. Rechensystem nach Anspruch 3, wobei die erste elektronische Vorrichtung ferner eingerichtet ist:

den in dem Übermittlungswarteschlangen-Zwischenpuffer gespeicherten Schreibbefehl ansprechend auf eine Anforderung der zweiten elektronischen Vorrichtung, die den Doorbell empfängt, an die zweite elektronische Vorrichtung zu übertragen; und die Schreibdaten, die in dem Schreibzwischenpuffer gespeichert sind, an die zweite elektronische Vorrichtung ansprechend auf eine Anforderung der zweiten elektronischen Vorrichtung, die den Schreibbefehl empfängt, zu übertragen.

6. Rechensystem nach Anspruch 5, wobei die erste elektronische Vorrichtung ferner eingerichtet ist:

Abschlussinformation des Schreibbefehls von der zweiten elektronischen Vorrichtung zu empfangen, die den Schreibbefehl empfängt, und die Abschlussinformation an den Host zu übertragen.

7. Rechensystem nach Anspruch 3, wobei die erste elektronische Vorrichtung ferner aufweist:

einen Doorbell-Parser, der eingerichtet ist, den Schreibbefehl anzufordern, der in der Übermittlungswarteschlange des Hosts gespeichert ist während mindestens eines Abschnittes des Analysierens des Doorbells und des Übertragens des Doorbells an die zweite elektronische Vorrichtung.

8. Rechensystem nach Anspruch 7, wobei der Doorbell-Parser ferner eingerichtet ist, auf die Übermittlungswarteschlange des Hosts, der durch eine Adresse des Doorbells angegeben wird, zuzugreifen.

9. Rechensystem nach Anspruch 3, wobei die erste elektronische Vorrichtung ferner aufweist:

einen Befehls-Parser, der eingerichtet ist, die Schreibdaten anzufordern, die in dem Datenpuffer des Hosts gespeichert sind, während mindestens ein Teil des Schreibbefehls analysiert wird und der Schreibbefehl an die zweite elektronische Vorrichtung übertragen wird.

10. Rechensystem nach Anspruch 1, wobei die erste elektronische Vorrichtung ferner eingerichtet ist:

den Host aufzufordern, einen Lesebefehl, der in der Übermittlungswarteschlange des Hosts geschrieben ist, zu übertragen;

den von dem Host empfangenen Lesebefehl zu speichern;

Lesedaten des Lesebefehls zu empfangen, der von der zweiten elektronischen Vorrichtung übertragen wird, die den Lesebefehl empfängt; und die Lesedaten an den Host zu übertragen.

11. Rechensystem aufweisend:

einen Host;

eine erste elektronische Vorrichtung, die mit dem Host gekoppelt ist; und

eine zweite elektronische Vorrichtung, die mit der ersten elektronischen Vorrichtung gekoppelt ist, wobei die zweite elektronische Vorrichtung eingerichtet ist, mit dem Host durch die erste elektronische Vorrichtung zu kommunizieren,

wobei die erste elektronische Vorrichtung eingerichtet ist:

einen Schreibbefehl für die zweite elektronische Vorrichtung von dem Host zu empfangen und den Schreibbefehl an die zweite elektronische Vorrichtung zu übertragen;

einen Doorbell, der mit einer Übermittlungswarteschlange assoziiert ist, in der der Schreibbefehl geschrieben ist, von dem Host zu empfangen;

den Doorbell an die zweite elektronische Vorrichtung zu übertragen;  
Schreibdaten des in einem Datenpuffer des Hosts gespeicherten Schreibbefehls anzufordern; und  
die von dem Host empfangenen Schreibdaten zu speichern.

12. Rechensystem nach Anspruch 11, wobei die zweite elektronische Vorrichtung aufweist:  
einen Übermittlungswarteschlangen-Controller-Memory-Buffer (CMB), der eingerichtet ist, den Schreibbefehl zu speichern, der von dem Host durch die erste elektronische Vorrichtung übertragen wurde; und  
ein Doorbell-Register, das eingerichtet ist, den von dem Host durch die erste elektronische Vorrichtung übertragenen Doorbell zu speichern.

13. Rechensystem nach Anspruch 12, wobei der Host eingerichtet ist, direkt auf das Doorbell-Register und den Übermittlungswarteschlangen-CMB der zweiten elektronischen Vorrichtung ohne eine Anforderung der zweiten elektronischen Vorrichtung zuzugreifen.

14. Rechensystem nach Anspruch 11, wobei die erste elektronische Vorrichtung aufweist:  
einen Befehls-Parser, der eingerichtet ist, die Schreibdaten anzufordern, die in dem Datenpuffer des Hosts gespeichert sind, während mindestens ein Teil des Schreibbefehls analysiert wird und der Schreibbefehl an die zweite elektronische Vorrichtung übertragen wird; und  
einen Schreibzwischenpuffer, der eingerichtet ist, die Schreibdaten zu speichern.

15. Rechensystem nach Anspruch 14, wobei die erste elektronische Vorrichtung ferner eingerichtet ist, die Schreibdaten, die in dem Schreibzwischenpuffer gespeichert sind, an die zweite elektronische Vorrichtung ansprechend auf eine Anforderung der zweiten elektronischen Vorrichtung, die den Schreibbefehl empfängt, zu übertragen.

16. Rechensystem nach Anspruch 15, wobei die erste elektronische Vorrichtung ferner eingerichtet ist:  
Abschlussinformation des Schreibbefehls von der zweiten elektronischen Vorrichtung zu empfangen, die den Schreibbefehl empfängt, und die Abschlussinformation an den Host zu übertragen.

17. Rechensystem nach Anspruch 11, wobei der Host eingerichtet ist, den Doorbell an die erste elektronische Vorrichtung zu übertragen, nachdem der Schreibbefehl an die erste elektronische Vorrichtung übertragen wurde.

18. Rechensystem aufweisend:  
einen Host;

eine erste elektronische Vorrichtung, die mit dem Host gekoppelt ist, wobei die erste elektronische Vorrichtung aufweist:

einen Übermittlungswarteschlangen-Controller-Memory-Buffer (CMB); und  
einen Schreib-CMB; und

eine zweite elektronische Vorrichtung, die mit der ersten elektronischen Vorrichtung gekoppelt ist, wobei die zweite elektronische Vorrichtung eingerichtet ist, mit dem Host durch die erste elektronische Vorrichtung zu kommunizieren,

wobei die erste elektronische Vorrichtung eingerichtet ist:

einen Schreibbefehl von dem Host zu empfangen, der in eine Übermittlungswarteschlange des Hosts geschrieben wurde, und den Schreibbefehl in dem Übermittlungswarteschlangen-CMB zu speichern;

von dem Host Schreibdaten des Schreibbefehls zu empfangen, der in einem Datenpuffer des Hosts gespeichert ist;

die Schreibdaten in den Schreib-CMB zu speichern;  
einen Doorbell, der von dem Host bezüglich der Übermittlungswarteschlange übertragen wird, zu empfangen; und

den Doorbell an die zweite elektronische Vorrichtung zu übertragen.

19. Rechensystem nach Anspruch 18, wobei die erste elektronische Vorrichtung ferner eingerichtet ist:

den in dem Übermittlungswarteschlangen-CMB gespeicherten Schreibbefehl ansprechend auf eine Anforderung der zweiten elektronischen Vorrichtung, die den Doorbell empfängt, an die zweite elektronische Vorrichtung zu übertragen; und

die Schreibdaten, die in dem Schreib-CMB gespeichert sind, an die zweite elektronische Vorrichtung ansprechend auf eine Anforderung der zweiten elektronischen Vorrichtung, die den Schreibbefehl empfängt, zu übertragen.

20. Rechensystem nach Anspruch 19, wobei die erste elektronische Vorrichtung ferner eingerichtet ist:

Abschlussinformation des Schreibbefehls von der zweiten elektronischen Vorrichtung zu empfangen, die den Schreibbefehl empfängt; und  
die Abschlussinformation an den Host zu übertragen.

Es folgen 20 Seiten Zeichnungen

Anhängende Zeichnungen

FIG. 1

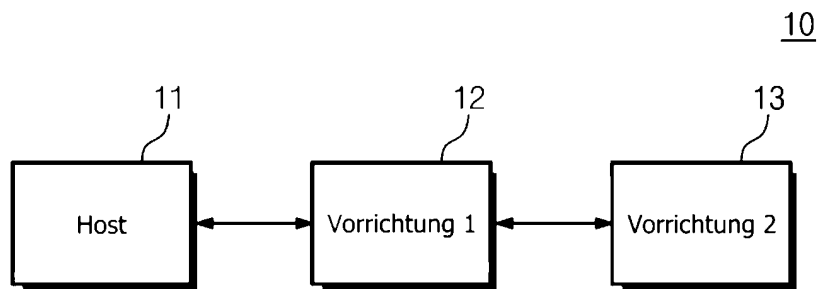


FIG. 2

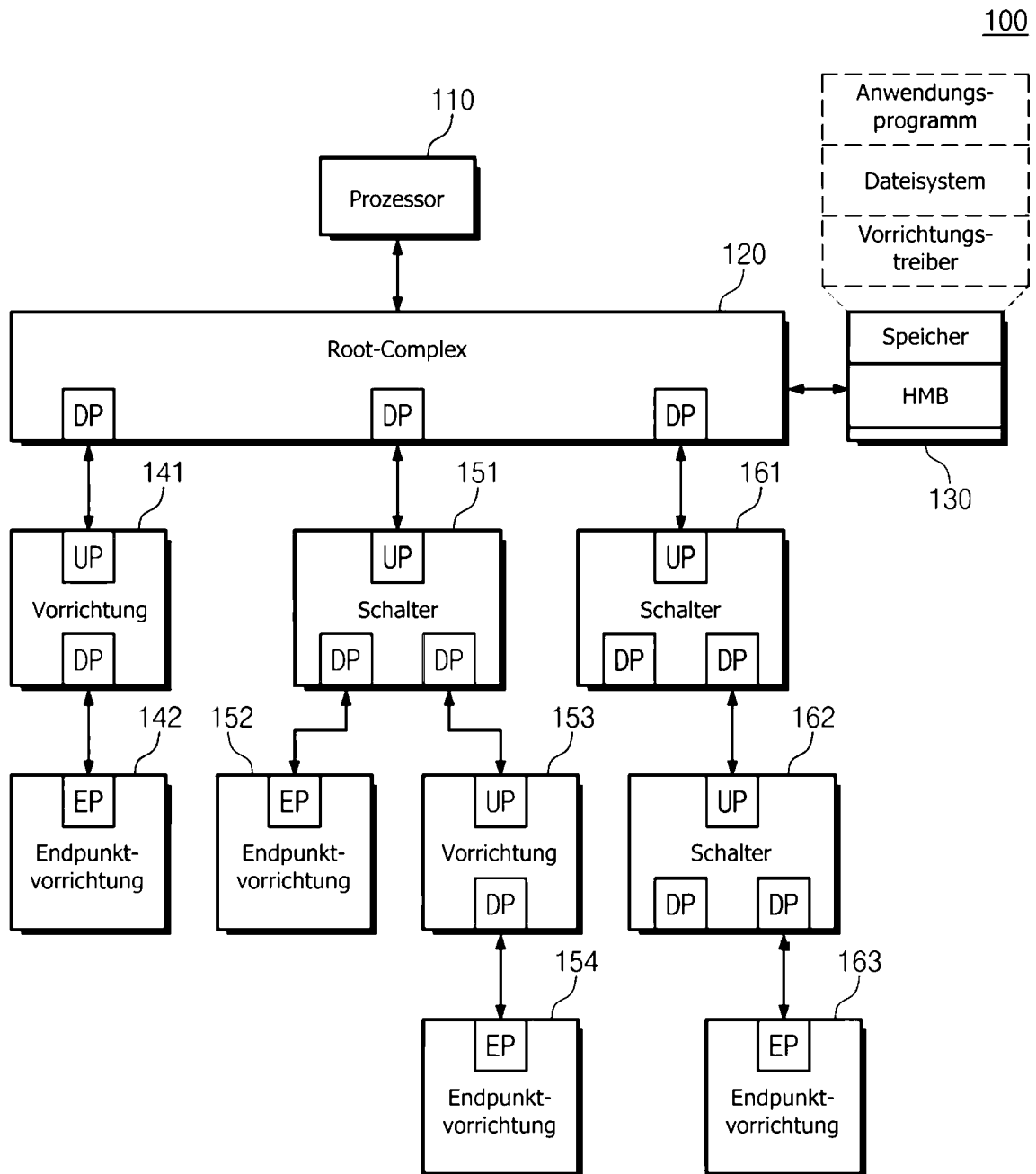


FIG. 3

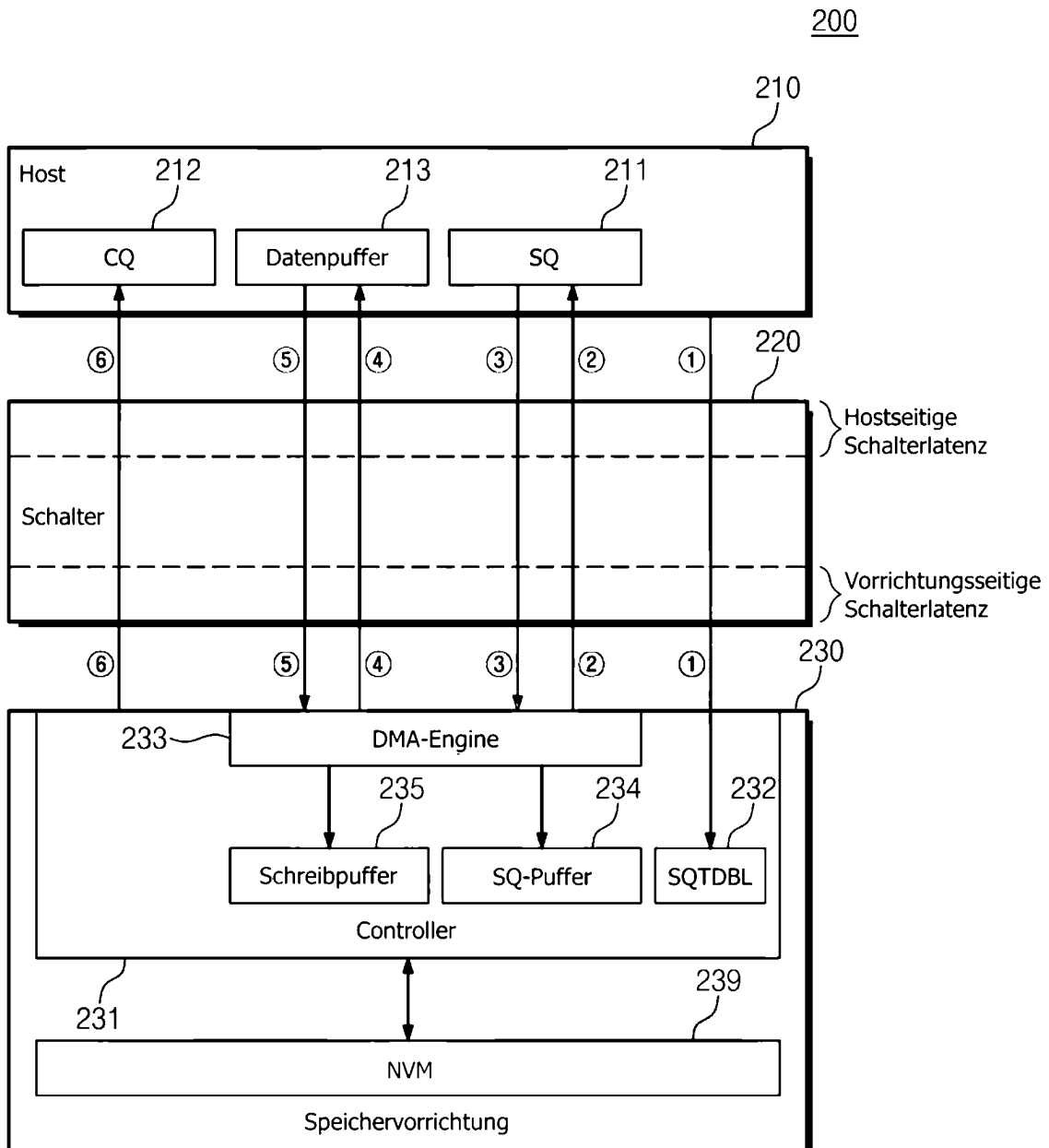


FIG. 4

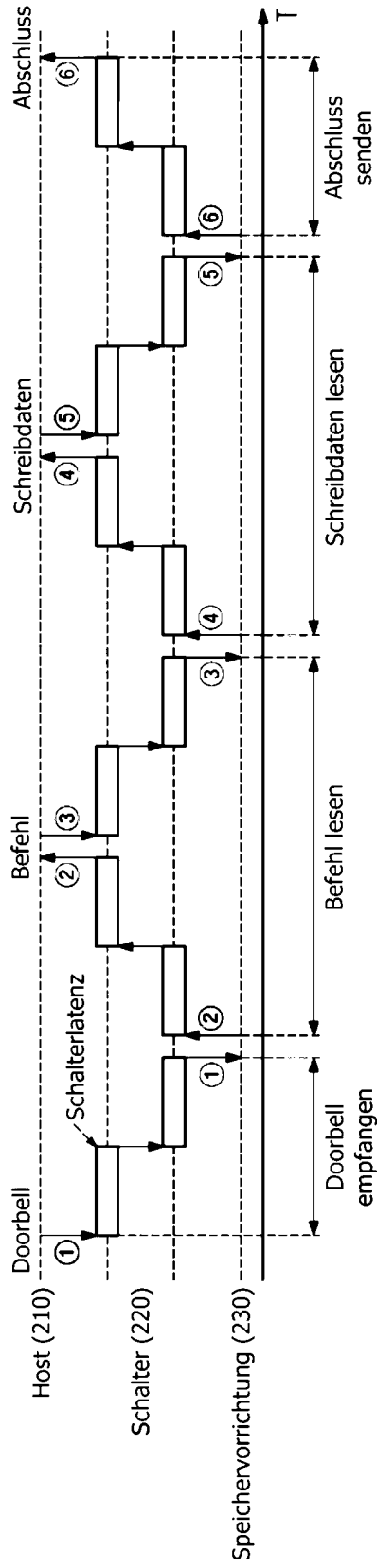


FIG. 5

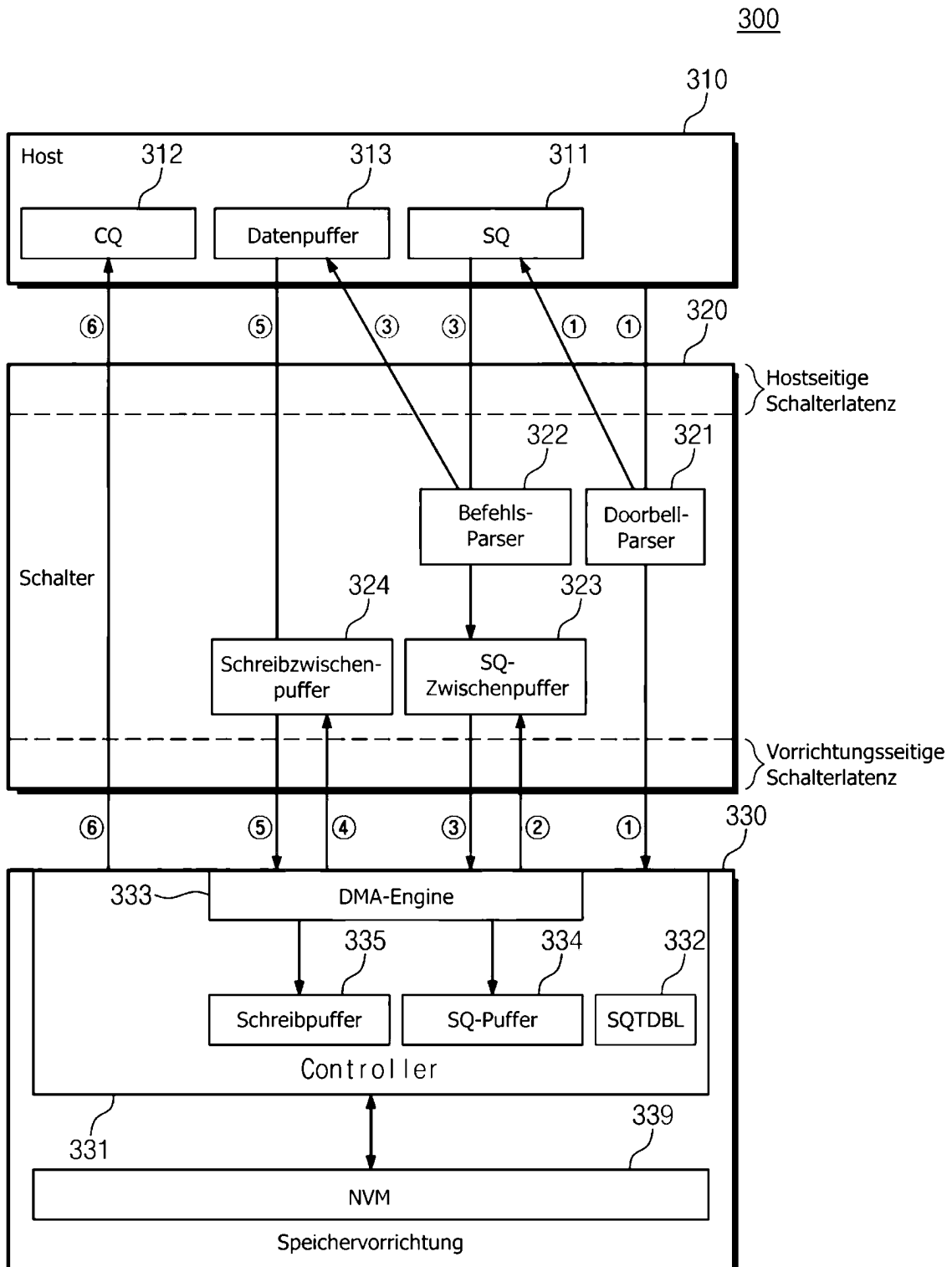




FIG. 6

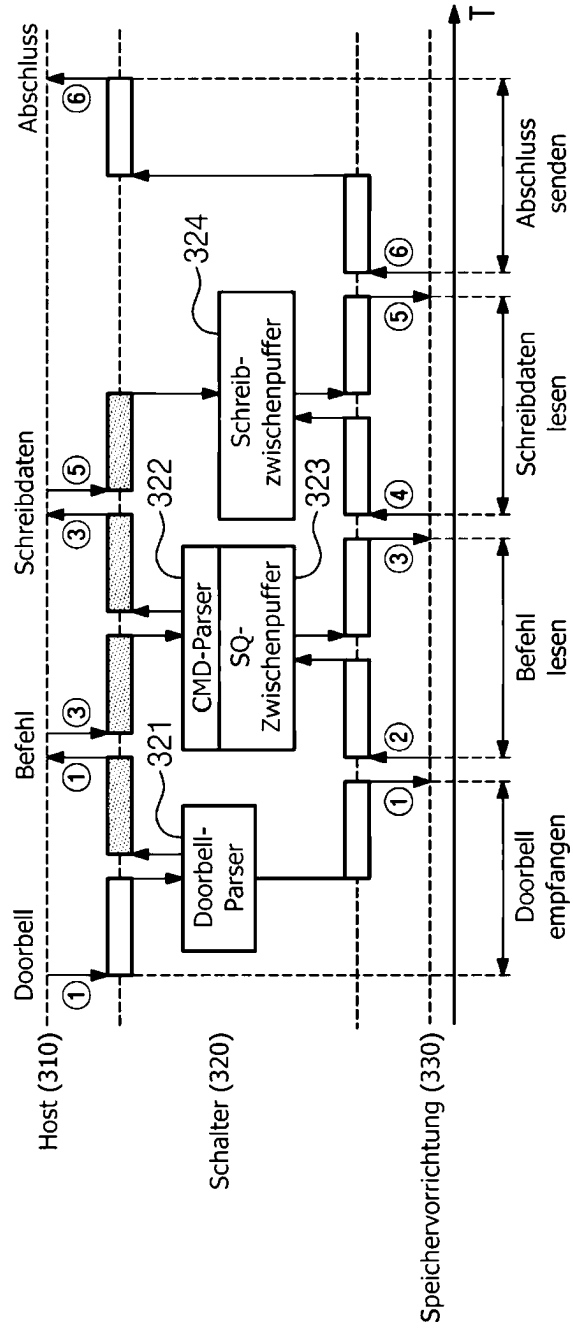




FIG. 8

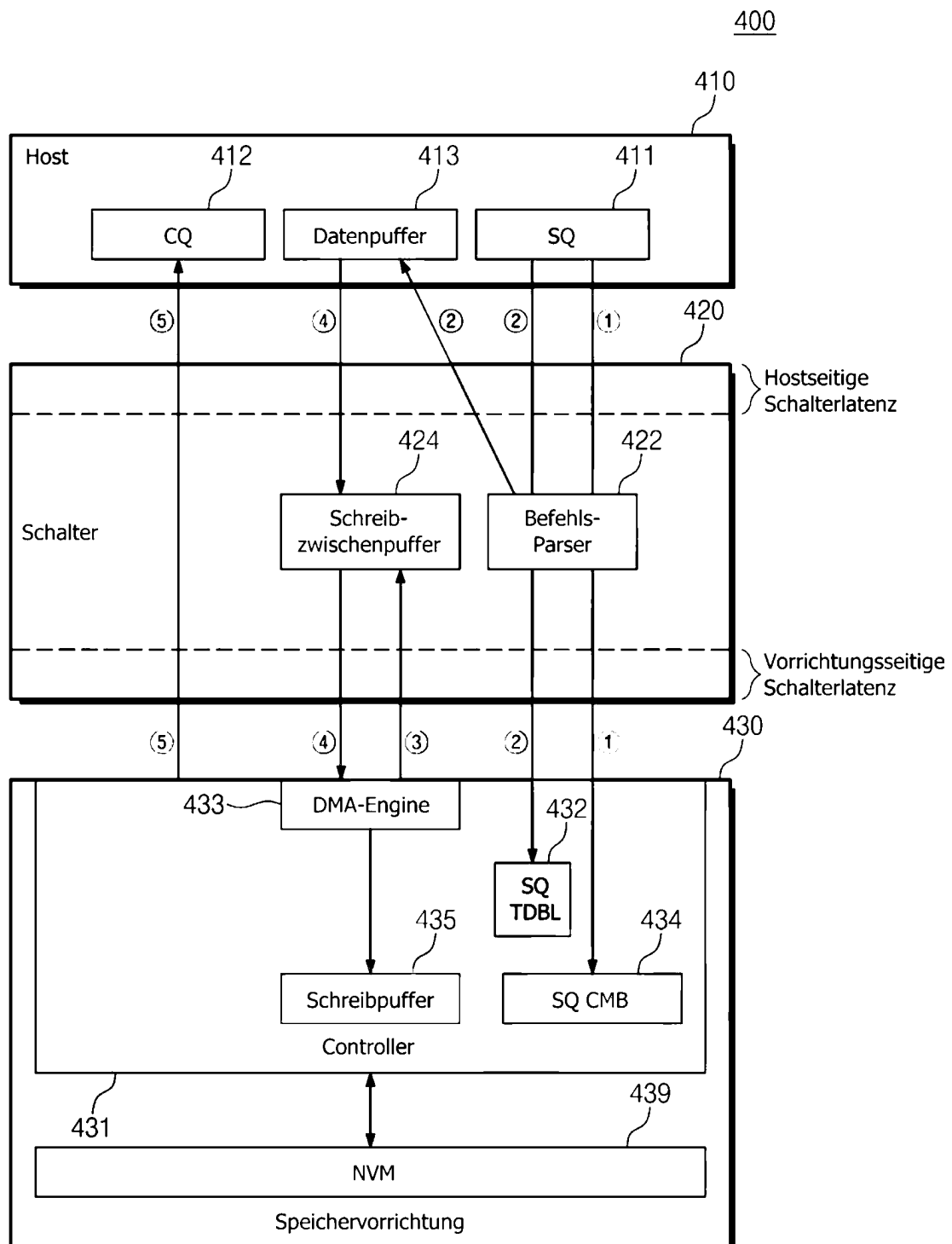


FIG. 9

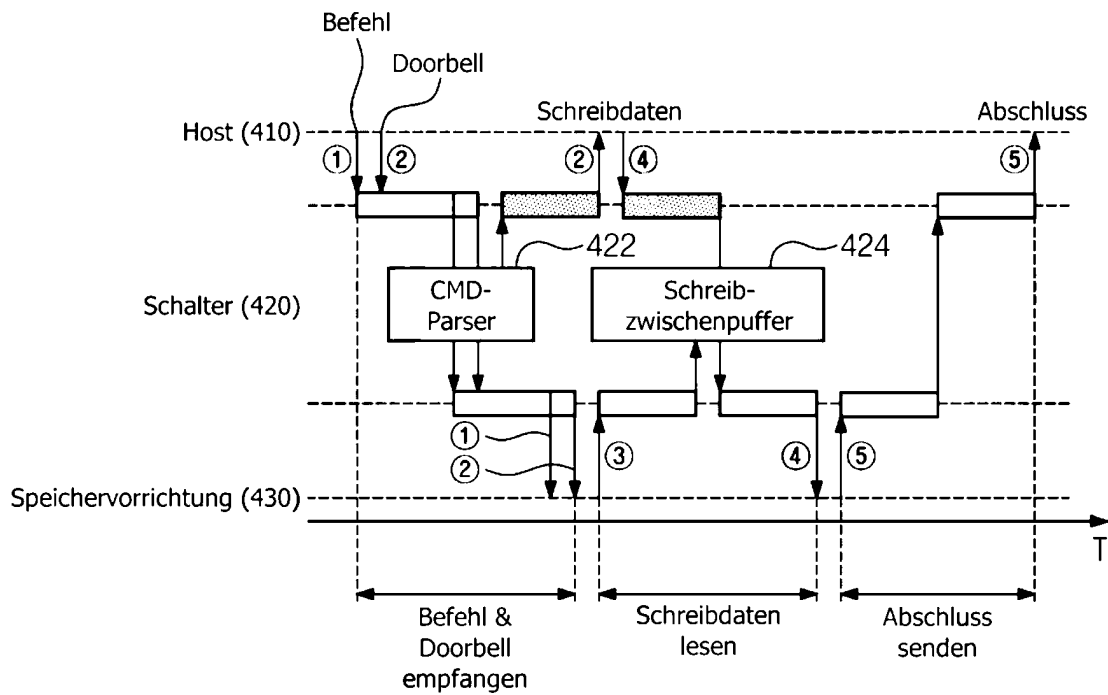


FIG. 10

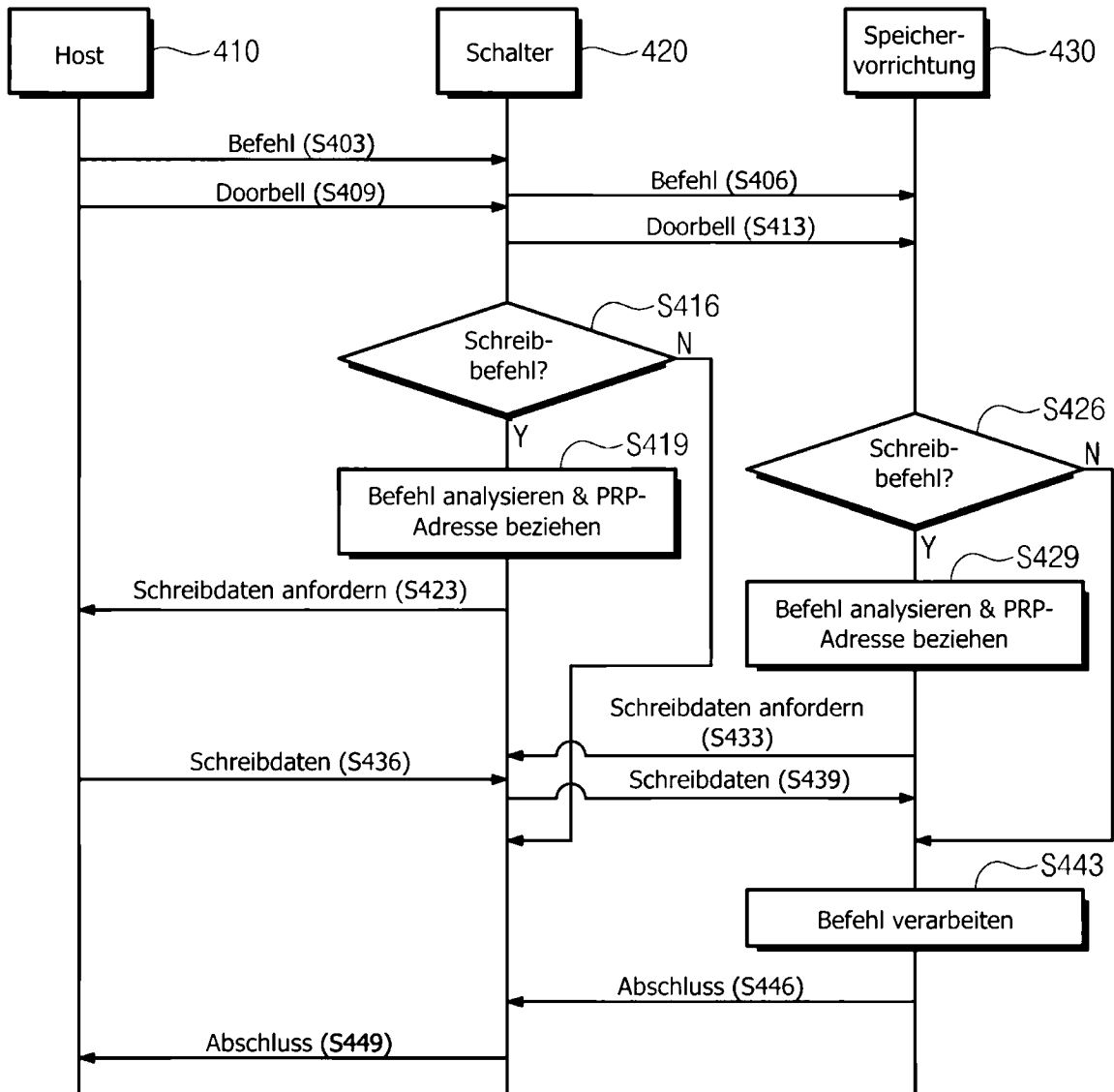


FIG. 11

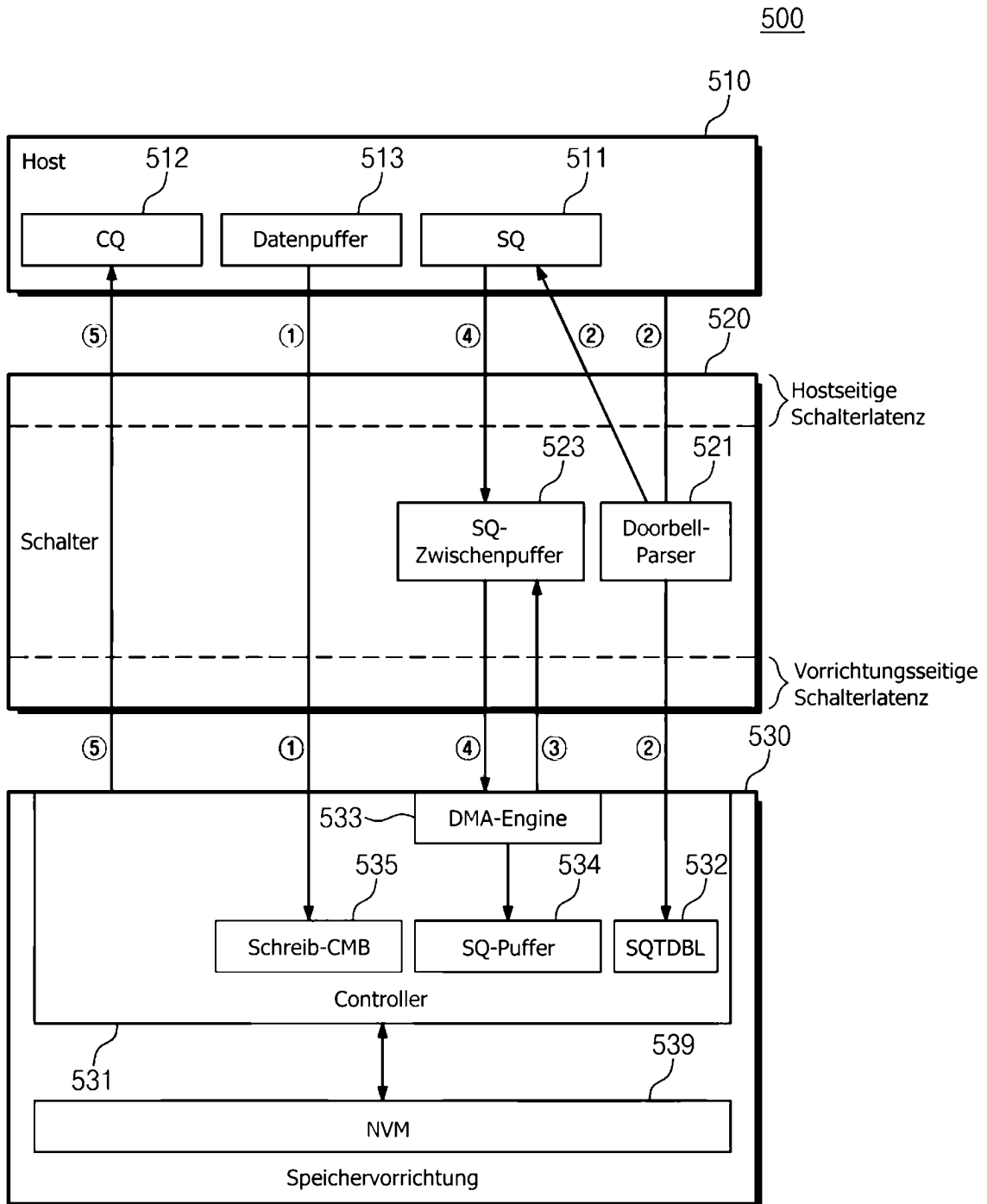


FIG. 12

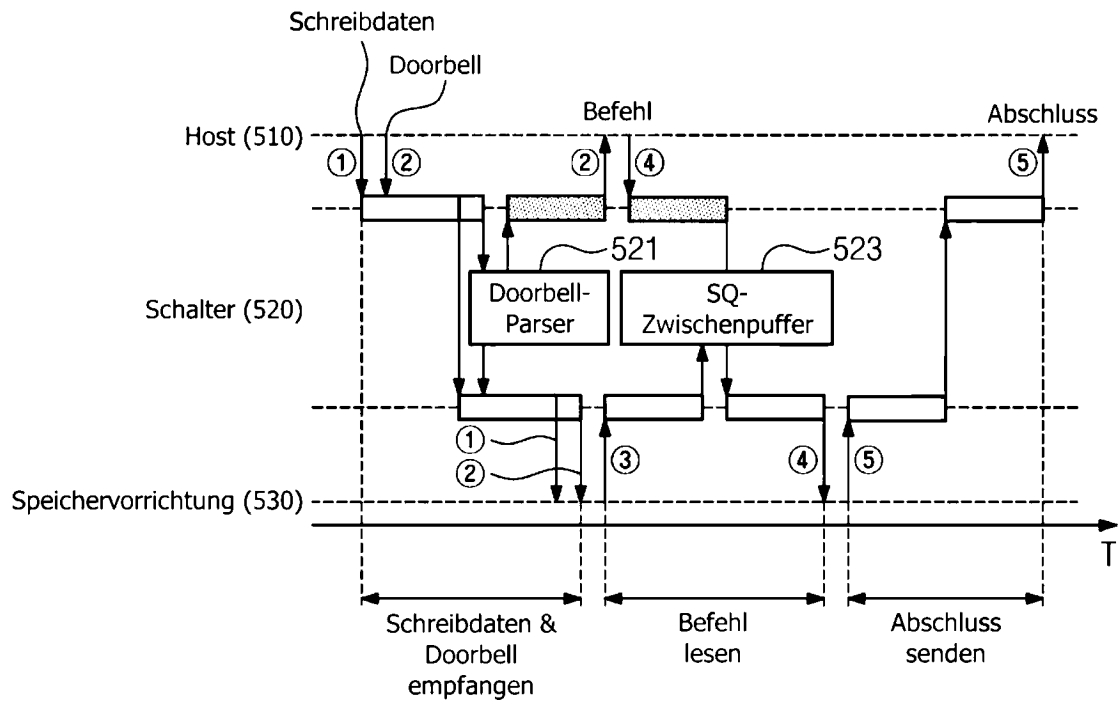


FIG. 13

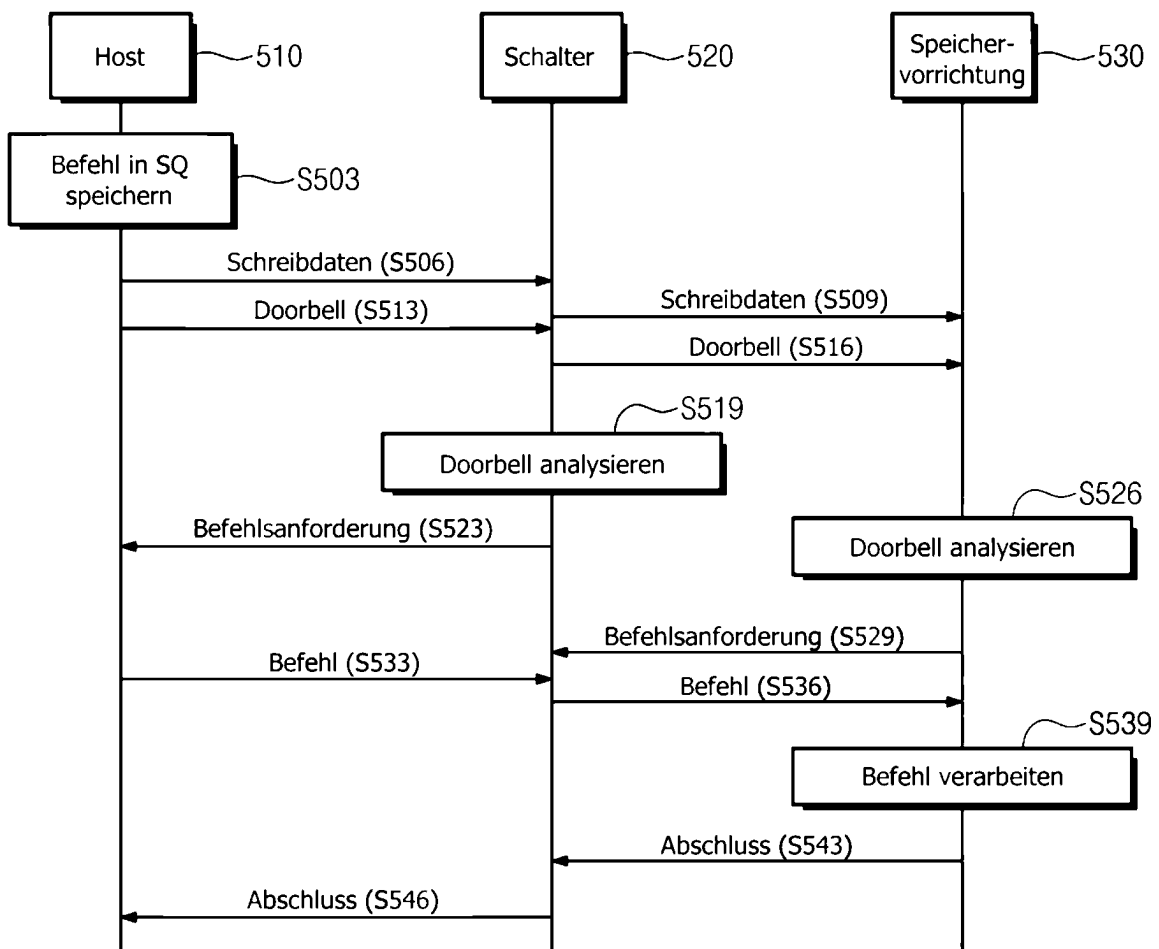




FIG. 14

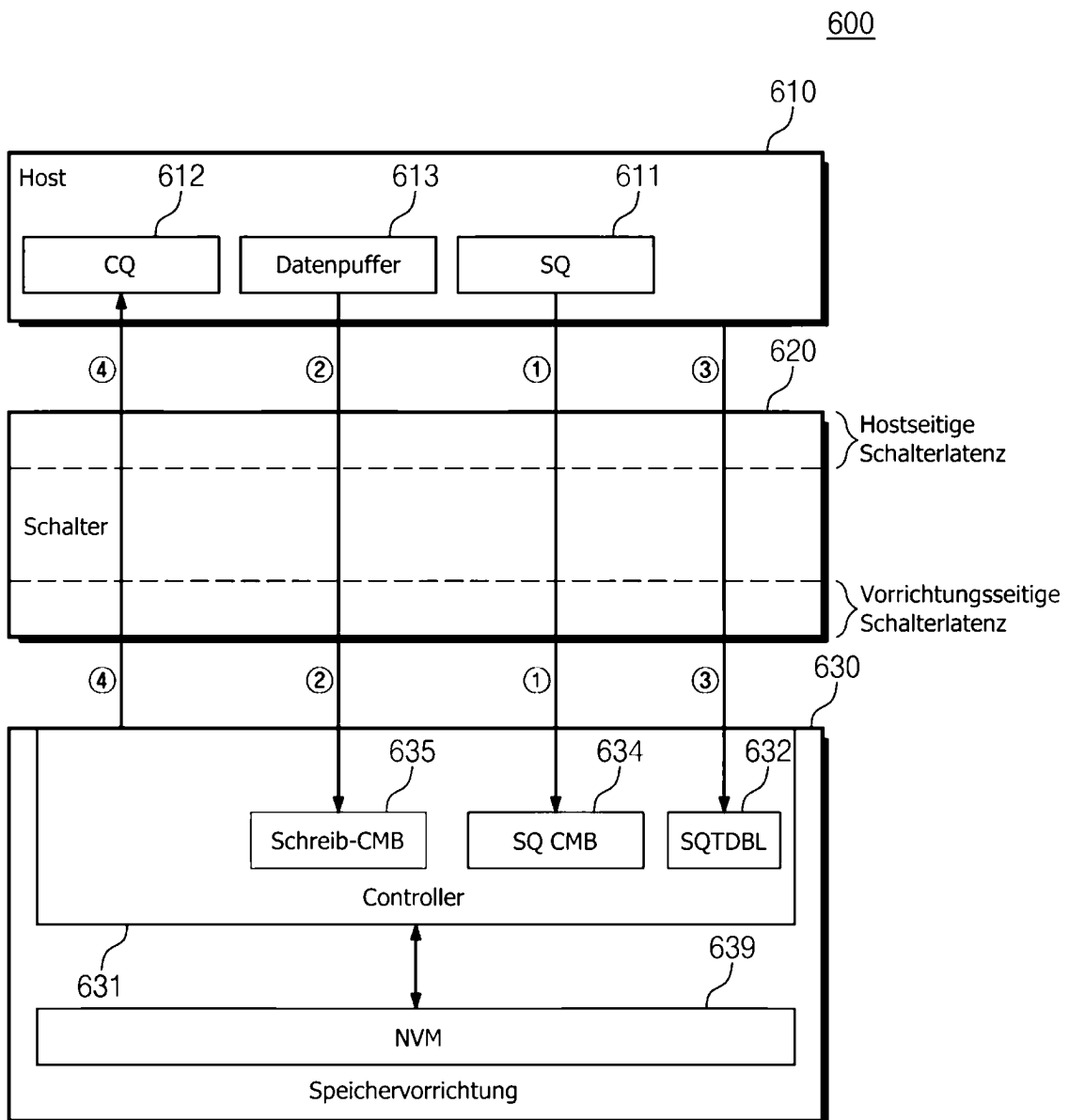


FIG. 15

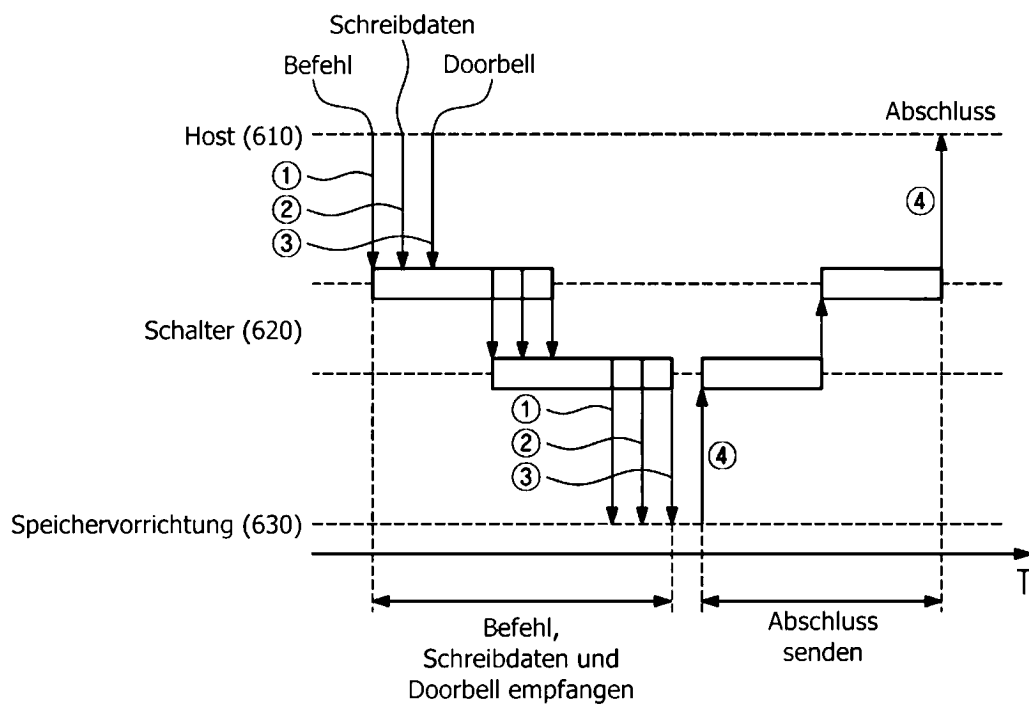


FIG. 16

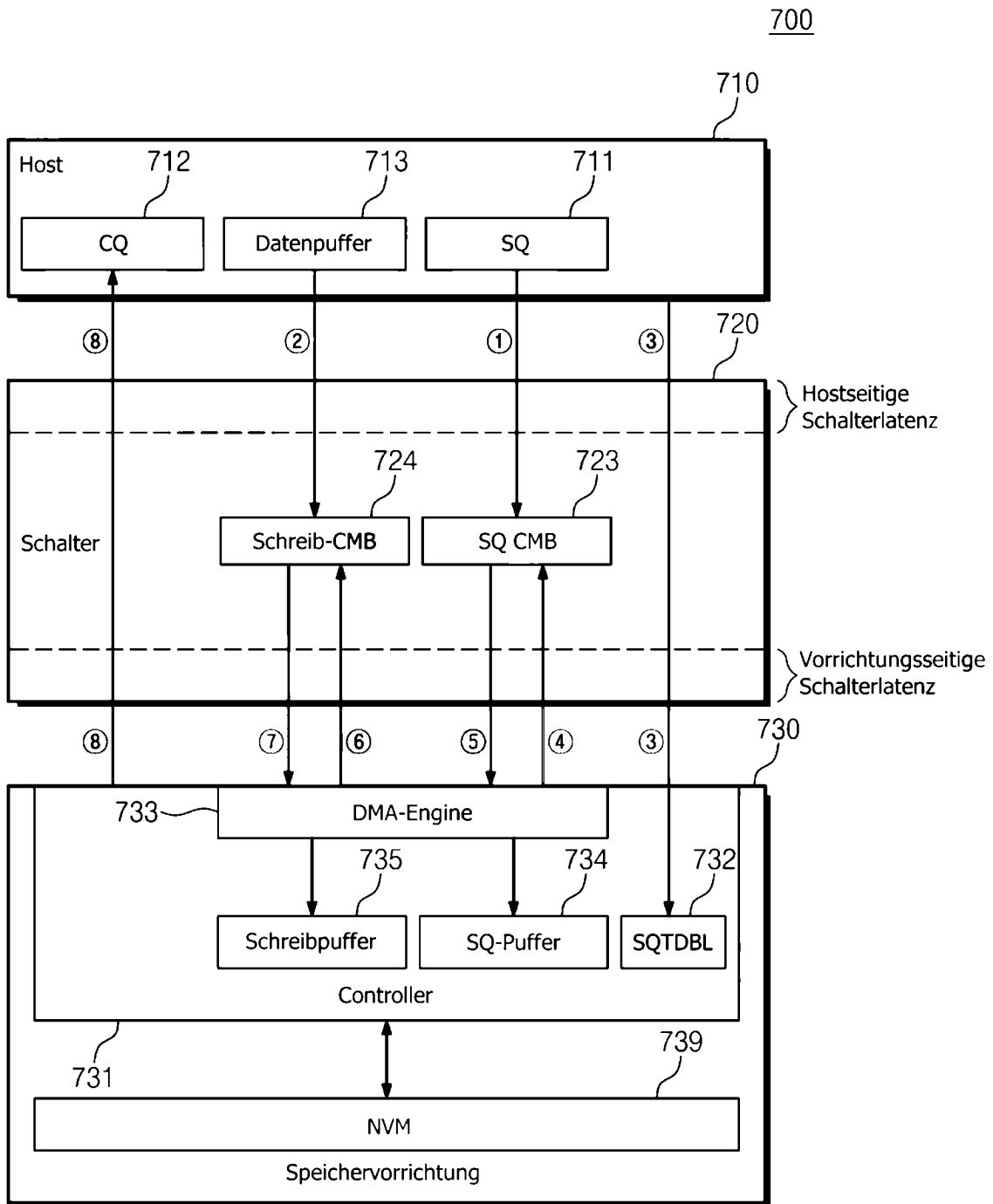


FIG. 17

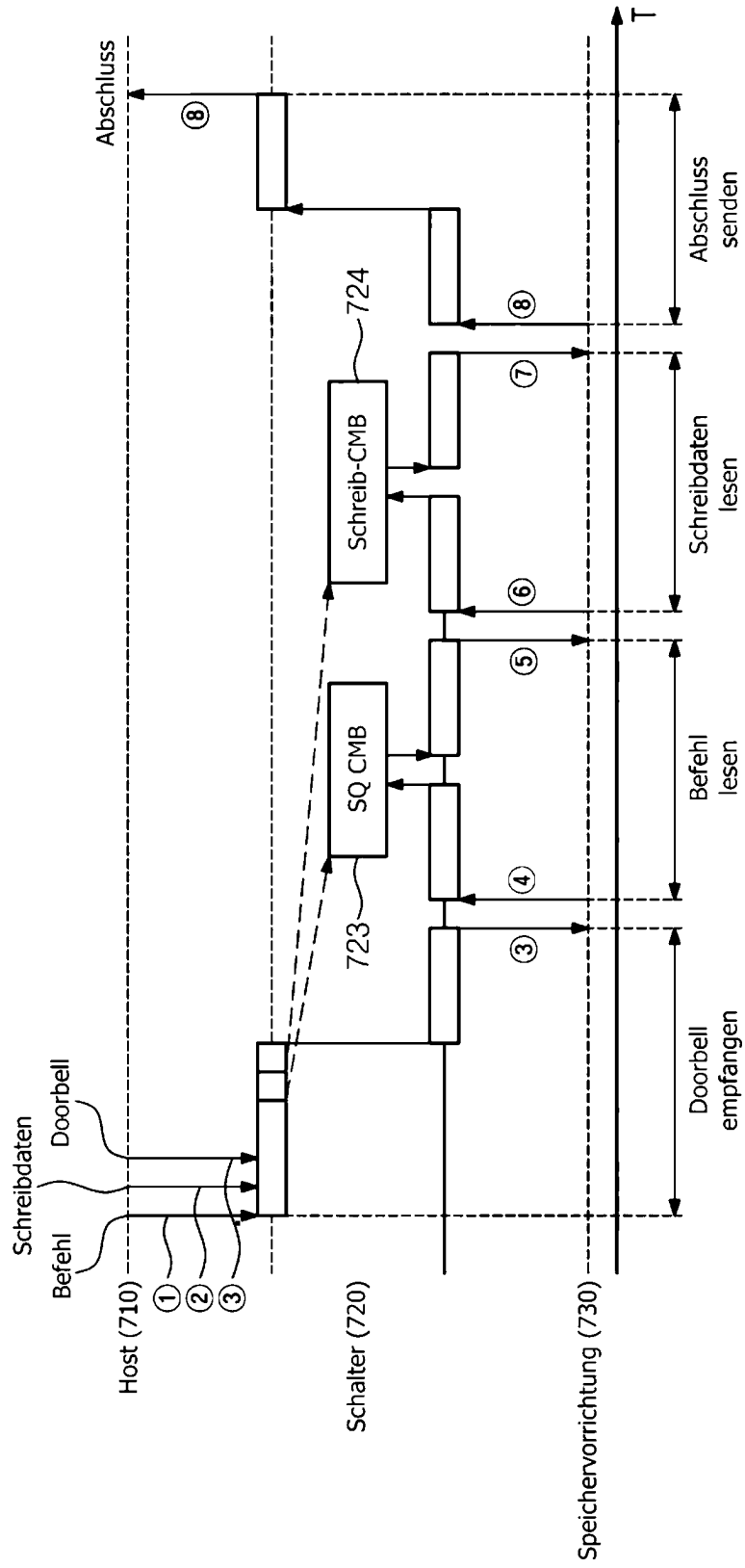


FIG. 18

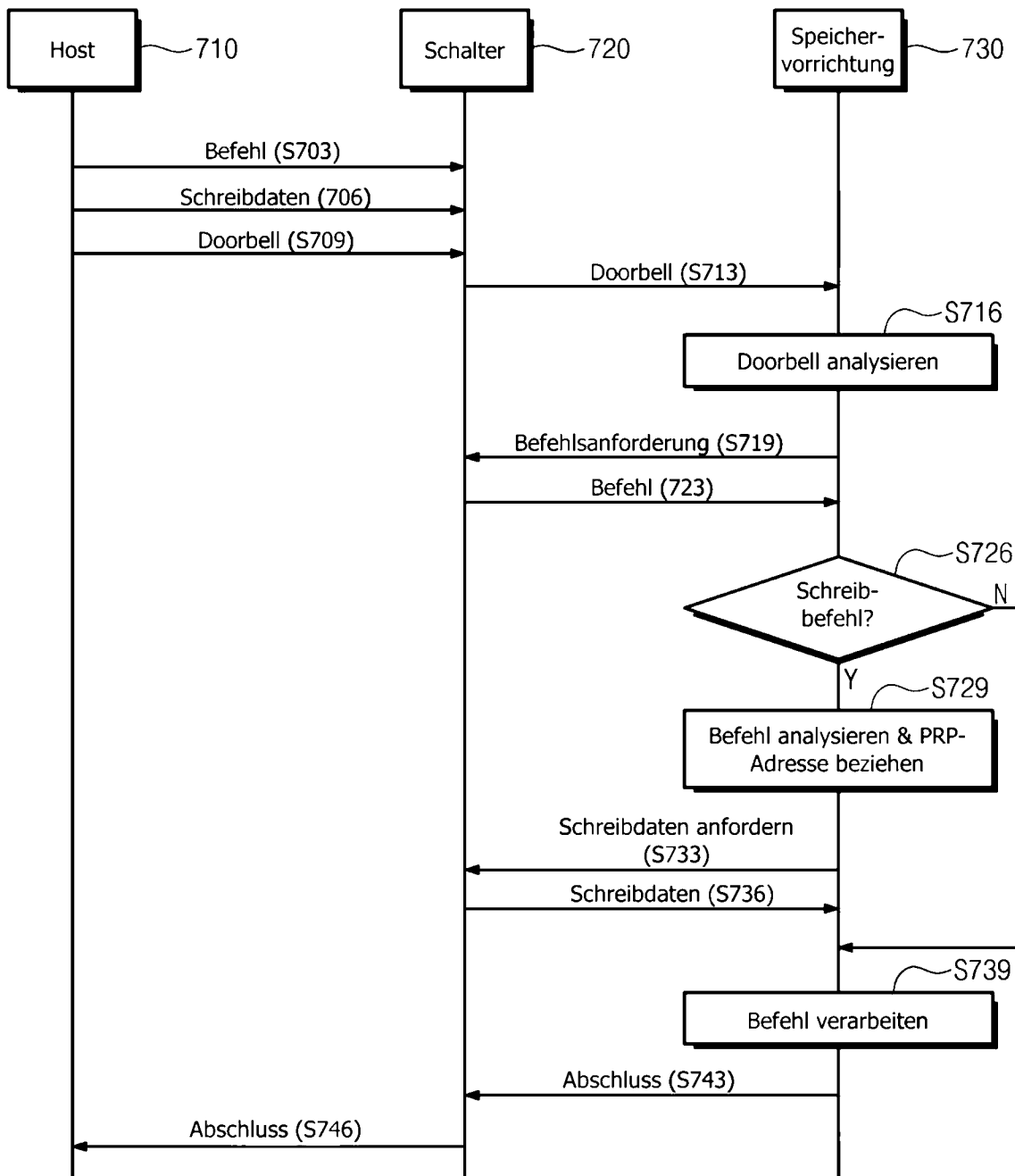


FIG. 19

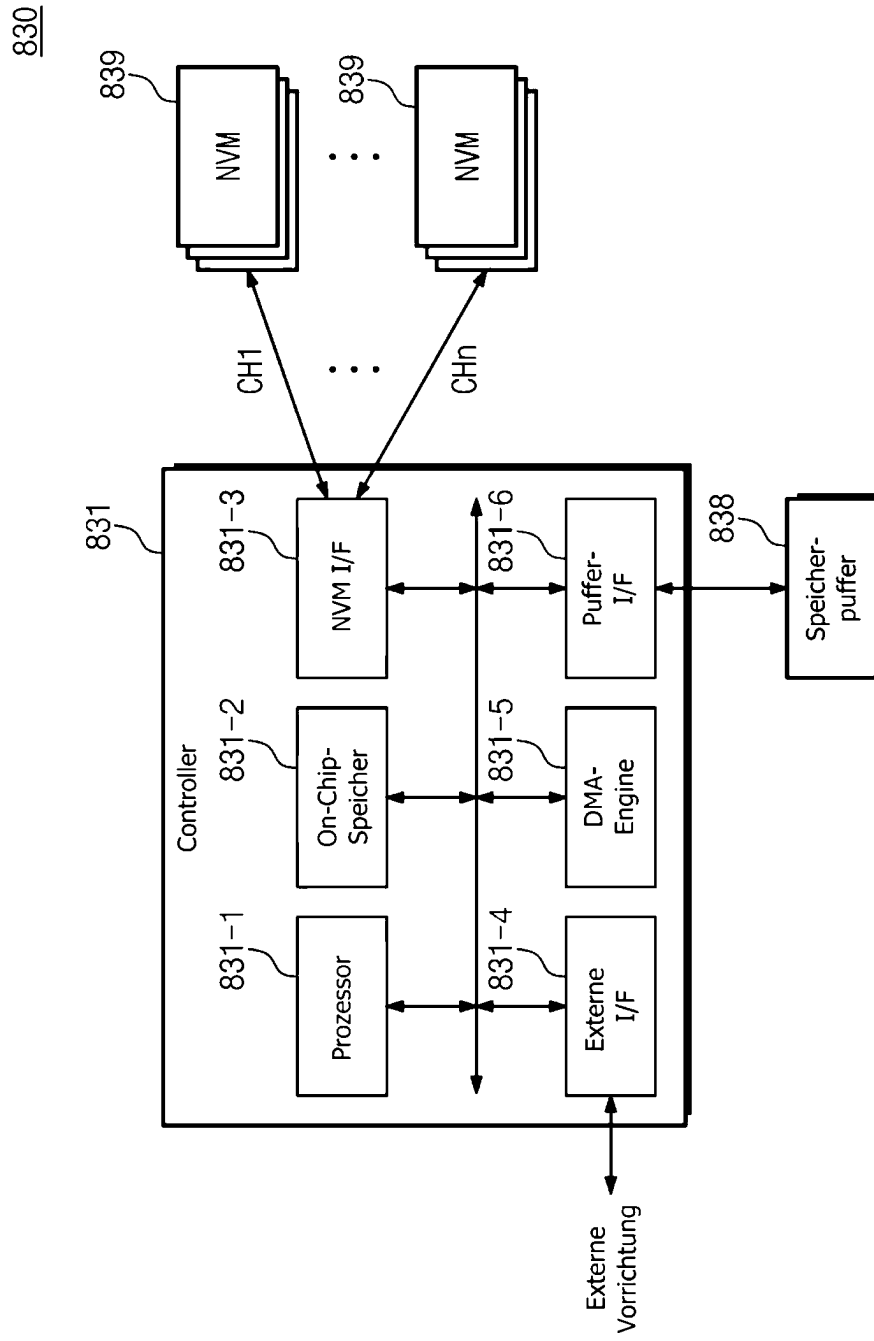


FIG. 20

