



(12) **Veröffentlichung**

der internationalen Anmeldung mit der
 (87) Veröffentlichungs-Nr.: **WO 2014/004213**
 in deutscher Übersetzung (Art. III § 8 Abs. 2 IntPatÜG)
 (21) Deutsches Aktenzeichen: **11 2013 003 248.1**
 (86) PCT-Aktenzeichen: **PCT/US2013/046574**
 (86) PCT-Anmeldetag: **19.06.2013**
 (87) PCT-Veröffentlichungstag: **03.01.2014**
 (43) Veröffentlichungstag der PCT Anmeldung
 in deutscher Übersetzung: **02.04.2015**

(51) Int Cl.: **H04L 29/06 (2006.01)**

(30) Unionspriorität:
13/534,541 **27.06.2012** **US**

(74) Vertreter:
BOEHMERT & BOEHMERT Anwaltspartnerschaft
mbB - Patentanwälte Rechtsanwälte, 28209
Bremen, DE

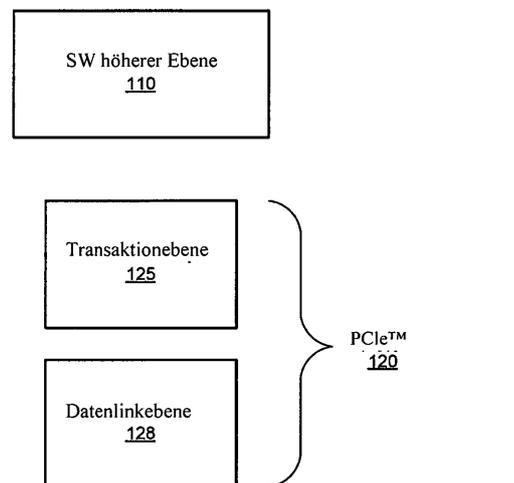
(71) Anmelder:
Intel Corporation, Santa Clara, Calif., US

(72) Erfinder:
Wagh, Mahesh, Portland, Oreg., US

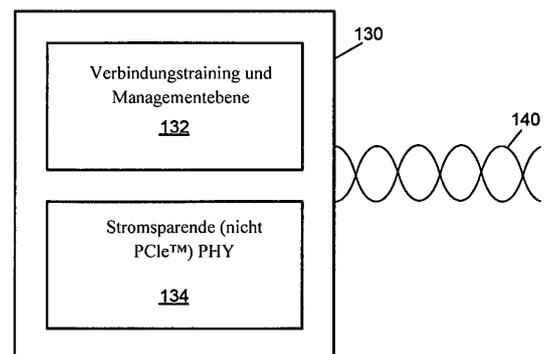
Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Steuern einer physischen Verbindung von einem ersten Protokoll unter Verwendung einer erweiterten Funktionsstruktur eines zweiten Protokolls**

100



(57) Zusammenfassung: In einer Ausführungsform umfasst ein Verfahren den Zugriff auf ein erstes Feld eines ersten Verbindungsfunktions-Registers einer ersten Vorrichtung, die einen Protokollstapel hat, der eine Transaktionsebene und eine Verbindungsebene nach einem ersten Kommunikationsprotokoll und eine physische Ebene des Protokollstapels umfasst, wobei dieser Protokollstapel eine physische Einheit eines zweiten Kommunikationsprotokolls hat, wobei das erste Feld als ein Zeigerwert zu einer Stelle in einem zweiten Verbindungsfunktions-Register der ersten Vorrichtung verwendet wird und Informationen von der Stelle in dem zweiten Verbindungsfunktions-Register verwendet werden, um eine Konfigurationsoperation für eine physische Verbindung durchzuführen, die an das Gerät gekoppelt ist. Weitere Ausführungsformen sind beschrieben und werden beansprucht.



Beschreibung

Technisches Gebiet

[0001] Die Ausführungsformen beziehen sich auf Verbindungstechnologien.

Hintergrund

[0002] Um die Kommunikation zwischen verschiedenen Geräten in einem System bereitzustellen, wird eine Art von Verbindungsmechanismus verwendet. Eine große Vielzahl solcher Verbindungen sind je nach Systemimplementierung möglich. Oft benutzen zwei Geräte ein gemeinsames Kommunikationsprotokoll, um miteinander zu kommunizieren.

[0003] Ein typisches Kommunikationsprotokoll für die Kommunikationen zwischen Geräten in einem Computersystem ist ein Peripheral Component Interconnect Express (PCI Express™ (PCIe™)) Kommunikationsprotokoll in Übereinstimmung mit Verbindungen auf der Basis des PCI Express™ Spezifikation Basisspezifikation Version 3.0 (veröffentlicht am 18. November 2010, im Folgenden mit PCIe™-Spezifikation bezeichnet). Dieses Kommunikationsprotokoll ist ein Beispiel eines Lade-/Speicher-Eingabe-/Ausgabe-(IO-)Verbindungssystems. Die Kommunikation zwischen den Geräten findet in der Regel seriell entsprechend diesem Protokoll mit sehr hohen Geschwindigkeiten statt. Verschiedene Parameter hinsichtlich dieses Protokolls wurden mit der Absicht entwickelt, maximale Leistung ohne Rücksicht auf die Energieeffizienz zu erzielen, da das PCIe™-Kommunikationsprotokoll im Rahmen von Desktop-Computer entwickelt wurde. Als Folge eignen sich viele ihrer Funktionen nicht für niedrigere Stromverbrauchslösungen, die in mobile Systeme integriert werden könnten.

[0004] Neben diesen Leistungsbedenken mit herkömmlichen Lade-/Speicher-Kommunikationsprotokollen sind existierende Verbindungsmanagement-Systeme in der Regel sehr komplex und beinhalten eine große Anzahl von Status, was zu einem langwierigen Prozess für Übergänge zwischen den Status führen kann. Dies liegt zum Teil an existierenden Verbindungsmanagement-Mechanismen, die entwickelt wurden, um mehrere verschiedene Formfaktor-Anforderungen, wie z. B. Stecker, Einarbeitung verschiedener System und so weiter, zu verstehen. Ein solches Beispiel ist das Verbindungsmanagement in Übereinstimmung mit dem PCIe™-Kommunikationsprotokoll.

Kurze Beschreibung der Zeichnungen

[0005] Fig. 1 ist ein Blockdiagramm hoher Ebene eines Protokollstapels für ein Kommunikationsprotokoll in Übereinstimmung mit einer Ausführungsform der vorliegenden Erfindung.

[0006] Fig. 2 ist ein Blockdiagramm eines Systems auf einem Chip (System-on-Chip, SoC) gemäß einer Ausführungsform der vorliegenden Erfindung.

[0007] Fig. 3 ist ein Blockdiagramm einer physischen Einheit gemäß einer weiteren Ausführungsform der vorliegenden Erfindung.

[0008] Fig. 4 ist ein Blockdiagramm, das weitere Details eines Protokollstapels gemäß einer Ausführungsform der vorliegenden Erfindung aufzeigt.

[0009] Fig. 5 ist ein Zustandsdiagramm für eine Verbindungstraining-Statusmaschine, die Teil eines Verbindungsmanagers in Übereinstimmung mit einer Ausführungsform der vorliegenden Erfindung sein kann.

[0010] Fig. 6 ist ein Ablaufdiagramm für die verschiedenen Status eines Seitenbandmechanismus gemäß einer Ausführungsform der vorliegenden Erfindung.

[0011] Fig. 7 ist ein Ablaufdiagramm eines Verfahrens gemäß einer Ausführungsform der vorliegenden Erfindung.

[0012] Fig. 8 ist ein Blockdiagramm von Komponenten in einem Computersystem gemäß einer Ausführungsform der vorliegenden Erfindung.

[0013] Fig. 9 ist ein Blockdiagramm eines Beispielsystems, mit dem Ausführungsformen verwendet werden können.

[0014] Fig. 10 ist ein Ablaufdiagramm, das eine Anfrage nach Rekonfiguration gemäß einer Ausführungsform der vorliegenden Erfindung veranschaulicht.

[0015] Fig. 11 ist ein Zeitverlaufsdiagramm, das eine Anfrage nach Rekonfiguration gemäß einer Ausführungsform der vorliegenden Erfindung veranschaulicht.

[0016] Fig. 12 ist ein Blockdiagramm einer Wiederherstellungs-Unterstatus-Maschine gemäß einer Ausführungsform der vorliegenden Erfindung.

[0017] Fig. 13 ist ein Blockdiagramm einer Funktionsstruktur gemäß einer Ausführungsform der vorliegenden Erfindung.

[0018] Fig. 14 ist ein Ablaufdiagramm eines Verfahrens gemäß einer anderen Ausführungsform der vorliegenden Erfindung.

Ausführliche Beschreibung

[0019] Ausführungsformen können eine Eingabe-/Ausgabe-(IO-)Verbindungstechnologie bieten, die geringen Stromverbrauch und eine Lade-/Speicher-Architektur hat und insbesondere für den Einsatz in mobilen Geräten wie Mobiltelefonen, Smartphones, Tablet-Computer, elektronische Lesegeräte, Ultrabooks™ usw. geeignet ist.

[0020] In verschiedenen Ausführungsformen kann ein Protokollstapel für ein gegebenes Kommunikationsprotokoll mit einer physischen Einheit mit einem anderen Kommunikationsprotokoll oder zumindest mit einer physischen (PHY) Einheit, die von der physischen Einheit für das jeweilige Kommunikationsprotokoll abweicht, verwendet werden. Eine physische Einheit umfasst sowohl eine logische Ebene als auch eine physische oder elektrische Ebene, die die tatsächliche physische Kommunikation von Informationssignalen über eine Verbindung zur Verfügung stellt, wie z. B. eine Verbindung, die zwei unabhängige Halbleiterchips verbindet, wobei diese zwei Halbleiterchips in einem integrierten Schaltungspaket (IC) oder in separaten Paketen, die z. B. über Leiterplatten-Routing, -Tracing usw. gekoppelt sind, sein können. Darüber hinaus kann die physische Einheit Framing/Deframing von Datenpaketen durchführen, Verbindungstraining und Initialisierung durchführen und die Datenpakete für die Lieferung an eine physische Verbindung/den Empfang von einer physischen Verbindung verarbeiten.

[0021] Obwohl verschiedene Implementierungen möglich sind, kann der Protokollstapel in einer Ausführungsform ein auf einem herkömmlichen Personalcomputer (PC) basierendes Kommunikationsprotokoll sein, wie beispielsweise ein Peripheral Component Interconnect Express-(PCI Express™ (PCIe™))Kommunikationsprotokoll in Übereinstimmung mit dem PCI™ Spezifikation Basisspezifikation Version 3.0 (veröffentlicht am 18. November 2010, im Folgenden mit PCIe™-Spezifikation bezeichnet) oder eine weitere Version, die sich auf Protokollerweiterungen bezieht, oder ein anderes solches Protokoll, während die physische Einheit nicht nach dem PCIe™-Kommunikationsprotokoll eingerichtet ist. Diese physische Einheit kann speziell für Zwecke konstruiert sein, um einen stromsparenden Betrieb zu ermöglichen, um die Verwendung eines im wesentlichen unveränderten oberen PCIe™-Protokollstapels mit diesen stromsparenden physischen Schaltungen zu ermöglichen. Auf diese Weise kann die verbreitete bestehende Nutzung des PCIe™-Kommunikationsprotokolls ausgenutzt werden, um dieses in tragbare und andere nicht-PC-basierte Formfaktoren einzubinden, die mit geringem Stromverbrauch betrieben werden. Obwohl der Umfang der vorliegenden Erfindung nicht in dieser Hinsicht beschränkt ist, kann in einer Ausführungsform diese physische Einheit eine physische Einheit sein, die von einer mobilen Plattform angepasst ist, wie z. B. die sogenannte M-PHY gemäß der M-PHY-Spezifikation Version 1.00.00 – 8. Februar 2011 (vom MIPI Board genehmigt am 28. Apr. 2011) der Mobil Industry Processor Interface (MIPI) Alliance (nachfolgend MIPI-Spezifikation), welche eine Gruppe ist, die Standards für Mobile-Computing-Geräte setzt. Aber auch andere physische Einheiten niedriger Leistung können verwendet werden, wie z. B. in Übereinstimmung mit anderen Niederleistungs-Spezifikationen, wie sie z. B. zum Koppeln einzelner Chips innerhalb eines Multi-Chip-Pakets verwendet werden, oder eine speziell angefertigte Niederleistungslösung. In dieser Schrift bedeutet der Begriff „Niederleistung“ oder „stromsparend“ einen Leistungsverbrauch, der geringer ist als bei einem herkömmlichen PC-System, und kann sich auf eine große Vielfalt von mobilen und tragbaren Geräten beziehen. Zum Beispiel kann „Niederleistung“ eine physische Einheit sein, die weniger Strom verbraucht als eine konventionelle physische PCIe™-Einheit.

[0022] Auf diese Weise kann durch Konvergieren eines traditionellen PCIe™-Protokollstapels mit einem anderen Typ physischer Einheiten eine hohe Wiederverwendung von Legacy-Komponenten, die für die PCIe™

entwickelt wurden, für den Einbau in mobile oder sonstige tragbare Plattformen oder Niederleistungsplattformen ausgenutzt werden.

[0023] Ausführungsformen können auch die Erkenntnis ausnutzen, dass existierende Lade-/Speicher-IO-Technologien, insbesondere PCIe™, mit der Absicht entwickelt wurden, maximale Leistung zu erzielen, wobei die Energieeffizienz kein Hauptanliegen war, und die somit nicht für Niederleistungs-Anwendungen geeignet sind. Durch die Kombination von Teilen eines herkömmlichen Lade-/Speicher-Protokollstapels mit einer physischen Einheit mit stromsparenden Design können Ausführungsformen die Leistungsvorteile der PCIe™ bewahren, während eine Best-In-Class-Leistung auf der Gerät- und Plattformebene erzielt wird.

[0024] Als solche können Ausführungsformen mit den allgegenwärtigen PCIe™-Architekturen, die gegenwärtig weitverbreitet verwendet werden, Software-kompatibel sein. Darüber hinaus können Ausführungsformen auch die direkte PHY-Wiederverwendung eines für den Mobilgebrauch designten PHY, wie z. B. M-PHY ermöglichen. Auf diese Weise können geringe Wirk- und Blindleistung mit effizienter Leistung/Bit-Übertragung erzielt werden, wobei dieser Lösungsansatz für elektromagnetische Schnittstellen/Radiofrequenz-Schnittstelle (electromagnetic interface/radio frequency interface, EMI/RFI) geeignet ist, da die PHY bei Taktraten arbeiten kann, die nicht mit den zugeordneten Funkgeräte interferieren (da die harmonischen Schwingungen der Taktfrequenz der PHY nicht mit Funkfrequenzen interferiert, die bei typischen Funklösungen gebräuchlich sind (z. B. 1,8, 1,9, 2,4 Gigahertz (GHz) oder andere solche Funkfrequenzen).

[0025] Ausführungsformen können ferner Architekturverbesserungen bieten, die Folgendes ermöglichen: ein optimiertes Verbindungstraining und einen optimierten Managementmechanismus (link training and management mechanism, LTSSM), optimierte Flusssteuerung und Wiederholungspufferung und optimierte Managementmechanismen; ein Architekturprotokoll zum Ändern des Verbindungsbetriebsmodus; schnelle Hardware-gestützte Gerätestatus-Speicherung und -Wiederherstellung; und einen vereinten Seitenbandmechanismus für das Verbindungsmanagement mit optionaler In-Band-Unterstützung.

[0026] In verschiedenen Ausführungsformen, können die PCIe™-Transaktions- und Datenverbindungsebenen als Teil eines Protokoll-Stacks mit beschränkten Änderungen implementiert werden, um unterschiedliche Verbindungsgeschwindigkeiten und asymmetrische Verbindungen zu berücksichtigen. Zusätzlich kann ein überarbeitetes Verbindungstraining und -management vorgesehen werden, um mehrspurige Kommunikation, asymmetrische Verbindungskonfigurationen, Seitenbandkonsolidierung und dynamische Skalierbarkeit der Bandbreite zu umfassen. Ausführungsformen können ferner Unterstützung für die Überbrückung zwischen existierenden PCIe™-basierten und nicht-PCIe™-basierten Logiken und Schaltungen bieten, wie z. B. M-PHY-Logik und -Schaltungen.

[0027] Dieser auf Ebenen verteilte Ansatz ermöglicht es bestehenden Software-Stapeln (z. B. Betriebssystemen (OS), virtuellen Maschinenmanager und Treibern), nahtlos auf einer anderen physischen Ebene ausgeführt zu werden. Die Auswirkungen auf die Datenverbindung und Transaktionsebene ist minimiert und kann die Änderung von Zeitgebern zur Aktualisierung der Acknowledgment-Frequenz, Replay-Timers usw. umfassen.

[0028] So können Ausführungsformen die Flexibilität, die in PCIe™-Systeme geboten wird, etwas reduzieren, da diese Flexibilität in einigen Fällen gewisse Komplexität in sowohl PCIe™-Systemen als auch anderen Systemen verursachen kann. Dies ist darauf zurückzuführen, dass beide diese Protokolle für ein hohes Maß an Flexibilität sorgen, um Plug-and-Play-Funktionen zu ermöglichen. Stattdessen können Ausführungsformen eine Lösung individuell erstellen, die Flexibilität im Design minimiert, da eine bekannte und feste Konfiguration vorliegt, wenn sie in ein gegebenes System inkorporiert werden, z. B. als ein System auf einem Chip (SoC), das mit einem anderen integrierten Schaltkreis (IC) verbunden ist. Da bei der Implementierung die bestehende Konfiguration genau bekannt ist, da sowohl der SoC als auch das angeschlossene Gerät innerhalb der Plattform fest eingebaut sind, z. B. auf einer Leiterplatte des Systems verlötet sind, wird keine Plug-and-Play-Fähigkeit für diese Geräte benötigt, und damit ist die größere Flexibilität, die in einem PCIe™-Protokoll oder in anderen PC-basierten Kommunikationsprotokollen besteht und die eine nahtlose Einbindung von verschiedenen Geräten in einem System mit Plug-and-Play-Fähigkeit ermöglicht, nicht erforderlich.

[0029] Beispielsweise kann der SoC als Root-Komplex arbeiten, der in einem ersten IC implementiert ist und mit einem zweiten IC gekoppelt ist, wobei dieser IC eine Funklösung sein kann, die eine oder mehrere Vorrichtungen von mehreren drahtlosen Kommunikationsvorrichtungen umfassen kann. Solche Vorrichtungen können von Niederleistungs-Nahbereichskommunikations-Systemen wie z. B. in Übereinstimmung mit einer Bluetooth™-Spezifikation, über lokale drahtlose Kommunikation wie z. B. einem sogenannten WiFi™-System in Übereinstimmung mit einem gegebenen Institute of Electrical and Electronics Engineers (IEEE) 802.11 Standard,

bis zu leistungsstärkeren Funkssysteme wie z. B. einem gegebenen Mobilkommunikationsprotokoll, wie z. B. ein 3G- oder 4G-Kommunikationsprotokoll, reichen.

[0030] Bezugnehmend auf **Fig. 1** wird ein Blockdiagramm hoher Ebene eines Protokollstapels gemäß einer anderen Ausführungsform der vorliegenden Erfindung gezeigt. Wie in **Fig. 1** gezeigt, kann Stapel **100** eine Kombination von Software, Firmware und Hardware in einem Halbleiterbauelement sein, wie beispielsweise ein IC, um die Handhabung von Datenkommunikation zwischen der Halbleitervorrichtung und einer anderen Vorrichtung, die daran gekoppelt ist, bereitzustellen. In der Ausführungsform der **Fig. 1** wird eine Ansicht auf hoher Ebene dargestellt, beginnend mit einer Software höherer Ebene **110**, die verschiedenen Arten von Software sein kann, die auf einer Plattform ausgeführt werden. Diese Software einer höheren Ebene kann die Betriebssystemsoftware (OS), Firmware, Anwendungssoftware und so weiter umfassen. Daten, die über eine Verbindung **140** übertragen werden, die eine physische Verbindung sein kann, die die Halbleitervorrichtung mit einer anderen Komponente verkoppelt, kann durch verschiedene Ebenen eines Protokollstapels laufen, die allgemein in **Fig. 1** gezeigt werden. Wie ersichtlich ist, können Teile dieses Protokollstapels Teil eines herkömmlichen PCIe™-Stapels **120** sein und eine Transaktionsebene **125** sowie eine Datenverbindungsebene **128** umfassen. In der Regel arbeitet die Transaktionsebene **125**, um Transaktionsebenen-Pakete (transaction layer packets, TLP) zu generieren, die durch Zeit getrennte Anfrage- oder Antwort-basierte Pakete sein können, so dass die Verbindung anderen Verkehr übertragen kann, während das Zielgerät Daten für die Antwort sammelt. Die Transaktionsebene verarbeitet ebenso kreditbasierte Flusststeuerung. Bei einer Ausführungsform bietet die Transaktionsebene **125** eine Schnittstelle zwischen der Verarbeitungsschaltung eines Geräts und der Verbindungsstruktur-Architektur wie z. B. der Datenverbindungsebene und der physischen Ebene. In dieser Hinsicht kann ist eine Hauptverantwortung der Transaktionsebene das Zusammenfügen und Zerlegen von Paketen (d. h. von Transaktionsebenen-Paketen, transaction layer packets (TLPs)) sowie die Verarbeitung von kreditbasierter Flusststeuerung.

[0031] Im Gegenzug kann die Datenverbindungsebene **128** TLPs anreihen, die von der Transaktionsebene erzeugt werden, und für eine sichere Lieferung der TLPs zwischen zwei Endpunkten (einschließlich der Durchführung der Fehlerprüfung) und Quittungsverarbeitung sorgen. Somit wirkt Verbindungsebene **128** als eine Zwischenstufe zwischen der Transaktionsebene und der physischen Ebene und bietet einen zuverlässigen Mechanismus für den Austausch von TLPs zwischen zwei Komponenten durch eine Verbindung. Eine Seite der Datenverbindungsebene akzeptiert TLPs, die von der Transaktionsebene zusammengefügt sind, wendet Kennungen an, berechnet einen Fehlererkennungscode und wendet ihn an, z. B. eine zyklische Redundanzprüfung (CRC), und reicht die modifizierten TLPs zur Übertragung über eine physische Ebene zu einem externen Gerät über physische Verbindung weiter.

[0032] Nach der Verarbeitung auf der Datenverbindungsebene **128** können Pakete an eine PHY-Einheit **130** weitergegeben werden. Im allgemeinen kann die PHY-Einheit **130** eine Niederleistungseinheit PHY **134** umfassen, die sowohl logische Ebenen als auch physische (einschließlich elektrische) Unterebenen enthalten kann. In einer Ausführungsform sendet die physische Ebene, durch die PHY-Einheit **130** dargestellt ist, ein Paket physisch an ein externes Gerät. Die Logikebene kann einen Sendeteil, um ausgehende Informationen für die Übertragung vorzubereiten, und einen Empfängerteil einschließen, um empfangene Informationen vor dem Weitergeben an die Verbindungsebene zu identifizieren und vorzubereiten. Dem Sender werden Symbole geliefert, die serialisiert sind und an eine externe Vorrichtung übertragen werden. Der Empfänger wird mit serialisierten Symbolen von einem externen Gerät versorgt und transformiert die empfangenen Signale in einen Bitstrom. Der Bitstrom wird deserialisiert und dem logischen Unterblock bereitgestellt.

[0033] In einer Ausführungsform kann ein Niederleistungsgerät PHY **134**, das entweder speziell entwickelt oder von einer anderen PHY, wie z. B. einer M-PHY, abgeleitet ist, die Verarbeitung der paketierte Daten zur Kommunikation über Verbindung **140** übernehmen. Wie weiter in **Fig. 1** gesehen werden kann, kann eine Verbindungstrainings- und Managementebene **132** (hier auch als Verbindungsmanager bezeichnet) ebenfalls in der PHY-Einheit **130** präsent sein. In verschiedenen Ausführungsformen kann der Verbindungsmanager **132** eine bestimmte Logik umfassen, die von einem anderen Kommunikationsprotokoll, wie beispielsweise einem PCIe™-Protokoll, implementiert wird, sowie spezialisierte Logik zur Handhabung der Schnittstelle zwischen dem herkömmlichen Protokoll, beispielsweise dem PCIe™-Protokollstapel oben, und dem physischen Gerät PHY **134** mit einem anderen Protokoll.

[0034] In der Ausführungsform der **Fig. 1** kann die Verbindung **140** als Differentialpaare von Drähten implementiert sein, die zwei Paare von unidirektionalen Leitungen sein können. In einigen Implementierungen können mehrere Sätze von Differentialpaaren verwendet werden, um die Bandbreite zu erhöhen. Man beachte, dass nach dem PCIe™-Kommunikationsprotokoll die Anzahl der Differentialpaare in jeder Richtung gleich sein

muss. Gemäß verschiedenen Ausführungsformen können jedoch unterschiedliche Anzahlen von Paaren in beiden Richtungen vorgesehen sein, was einen effizienteren und stromsparenderen Leistungsbetrieb erlaubt. Dieser gesamte konvergierte Stapel und die Verbindung **140** können als Mobile Express PCIe™ Interconnect oder Verbindung bezeichnet werden. Obwohl dies auf dieser hohen Ebene in der Ausführungsform von **Fig. 1** gezeigt ist, versteht sich, dass dadurch der Umfang der vorliegenden Erfindung in dieser Hinsicht in keiner Weise eingeschränkt wird. Das heißt, es versteht sich, dass die Ansicht in **Fig. 1** einfach im Hinblick auf die Protokollstapel von der Transaktionsebene durch die physische Ebene zusammen mit Software einer höheren Ebene zu verstehen ist und verschiedene andere Schaltungen eines SoCs oder andere Halbleitervorrichtung, die diesen Stapel enthalten, nicht dargestellt sind.

[0035] Es wird nun auf **Fig. 2** Bezug genommen, die ein Blockdiagramm eines SoC gemäß einer Ausführungsform der vorliegenden Erfindung zeigt. Wie in **Fig. 2** gezeigt wird, kann der SoC **200** ein beliebiger Typ von SoC für die Implementierung in verschiedene Arten von Plattformen sein, angefangen von relativ kleinen tragbaren Niederleistungsgeräten wie Smartphones, Minicomputer (Personal Digital Assistants, PDAs), Tablet-PCs, Netbooks, Ultrabooks™ und so weiter, bis zu fortgeschritteneren SoCs, die in Systemen höherer Ebene implementiert werden können.

[0036] Wie in **Fig. 2** gesehen werden kann, kann das SoC **200** einen oder mehrere Kerne **210₀–210_n** umfassen. In verschiedenen Ausführungsformen ist daher ein Mehrkern-SoC möglich, wobei die Kerne alle homogenen Kerne einer gegebenen Architektur, z. B. ein In-Order- bzw. Out-Of-Order-Prozessor, sein können. Alternativ können es heterogene Kerne sein, z. B. einige relativ kleine Low-Power-Cores, wie z. B. Kerne einer In-Order-Architektur, mit zusätzlichen Kernen, die eine größere und kompliziertere Architektur haben können, wie z. B. eine Out-Of-Order-Architektur. Ein Protokollstapel ermöglicht die Kommunikation von Daten zwischen einem oder mehreren dieser Kerne und anderen Komponenten eines Systems. Wie ersichtlich ist, kann dieser Stapel Software **215** umfassen, wobei dies Software einer höheren Ebene, wie z. B. ein Betriebssystem, Firmware und Anwendungssoftware sein kann, die auf einem oder mehreren der Kerne ausgeführt werden kann. Darüber hinaus enthält der Protokollstapel eine Transaktionsebene **220** und eine Datenverbindungsebene **230**. In verschiedenen Ausführungsformen können diese Transaktions- und Datenverbindungsebenen ein bestimmtes Kommunikationsprotokoll umfassen, wie beispielsweise ein PCIe™-Protokoll. Selbstverständlich können in anderen Ausführungsformen Ebenen mit anderen Protokollstapeln, wie z. B. in Übereinstimmung mit einem universellen seriellen Bus (USB) Protokollstapel vorhanden sein. In einigen Implementierungen können Niederleistungs-PHY-Schaltungen, wie sie hierin beschrieben sind, mit bestehenden alternativen Protokollstapeln multiplext werden.

[0037] Immer noch mit Bezug auf **Fig. 2** kann dieser Protokollstapel wiederum mit einer physischen Einheit **240** gekoppelt sein, die mehrere physische Einheiten für die Kommunikation über mehrere Verbindungen umfassen kann. In einer Ausführungsform kann eine erste physische Einheit **250** eine Niederleistungs-PHY-Einheit sein, die in einer Ausführungsform einem M-PHY gemäß der MIPI-Spezifikation entspricht, um die Kommunikation über eine primäre Verbindung **280** bereitzustellen. Zusätzlich kann eine Seitenband-(SB-)PHY-Einheit **244** präsent sein. In der gezeigten Ausführungsform kann diese Seitenband-PHY-Einheit Kommunikation über eine Seitenbandverbindung **270** erlauben, die ein konsolidiertes Seitenband zur Lieferung bestimmter Seitenbandinformationen sein kann, z. B. Lieferung mit langsameren Datenraten als bei der primären Verbindung **280**, die an die erste PHY **250** gekoppelt ist. In einigen Ausführungsformen können die verschiedenen Ebenen des Protokollstapels eine separate Seitenbandkopplung mit dieser SB-PHY **244** haben, um die Kommunikation entlang dieser Seitenbandverbindung zu ermöglichen.

[0038] Zusätzlich kann PHY-Einheit **240** ferner einen SB-Verbindungsmanager **242** umfassen, der verwendet werden kann, um die SB-PHY **244** zu steuern. Darüber hinaus kann ein Verbindungstrainer und ein Statusmanager **245** präsent sein und kann verwendet werden, um den Protokollstapel, der ein erstes Kommunikationsprotokoll verwendet, an die erste PHY **250**, die ein zweites Kommunikationsprotokoll verwendet, anzupassen, sowie die allgemeine Steuerung der ersten PHY **250** und Verbindung **280** zur Verfügung stellen.

[0039] Wie weiter zu sehen ist, können verschiedene Komponenten in der ersten PHY **250** präsent sein. Genauer gesagt, kann eine Sender- und Empfängerschaltung (nämlich TX **253** und RX **254**) präsent sein. Im allgemeinen kann diese Schaltung verwendet werden, um Serialisierungsoperationen und Deserialisierungsoperationen zusammen mit dem Senden und Empfangen von Daten über die Primärverbindung **280** auszuführen. Ein Savestate-Manager **251** kann präsent sein und kann verwendet werden, um eine Konfiguration und andere Statusinformationen über die erste PHY **250** zu speichern, wenn sie sich in einem Energiesparmodus befindet. Es kann ebenfalls ein Codierer **252** präsent sein, um Leitungscodierung durchzuführen, z. B. gemäß eines 8b/10b-Protokolls.

[0040] Wie weiter in **Fig. 2** gesehen werden kann, kann eine mechanische Schnittstelle **258** präsent sein. Diese mechanische Schnittstelle **258** kann eine gegebene Verbindung sein, die Kommunikation von dem Root-Complex **200** und insbesondere zu/von der ersten PHY **250** über die primäre Verbindung **280** liefert. In verschiedenen Ausführungsformen kann diese mechanische Verbindung mittels Pins der Halbleitervorrichtung montiert sein, wie einer Kugelgitteranordnung (ball grid array, BGA), oder eine andere oberflächenmontierte Anordnung oder eine Durchsteckverbindung sein.

[0041] Zusätzlich zu diesen Haupt-Kommunikationsmechanismen kann eine zusätzliche Kommunikationsschnittstelle eine serielle Niederleistungsschnittstelle (low power serial, LPS) PHY-Einheit **255** sein, die über einen separaten Stapel gekoppelt ist und eine Softwareebene **216**, eine Transaktionsebene **221** und eine Verbindungsebene **231** zwischen den Kernen **210** und einem oder mehreren Off-Chip-Geräten **260a–c** umfasst, die verschiedene Peripheriegeräte mit niedrigen Datenraten sein können, wie z. B. Sensoren, Beschleunigungssensoren, Temperatursensoren, globale Positionierungssystem-(GPS)Schaltungen, Kompassschaltungen, Touch-Screen-Schaltungen, Tastaturschaltungen, Mausschaltungen und so weiter.

[0042] Man beachte, dass in verschiedenen Ausführungsformen sowohl die Seitenbandverbindung **270** als auch die Primärverbindung **280** zwischen SoC **200** und einer anderen Halbleiterkomponente, beispielsweise eine weitere IC, wie z. B. eine Multiband-Funklösung, gekoppelt sein können.

[0043] Während die Darstellung der **Fig. 2** auf einem relativ hohen Niveau gezeigt wird, sind auch hier Variationen möglich. Zum Beispiel ist es möglich, mehrere Niederleistungs-PHYs zur Verfügung zu stellen, um höhere Raten der Datenkommunikation zu ermöglichen, z. B. über mehrere Kanäle, wobei jeder Kanal mit einer unabhängigen PHY verbunden ist. Bezugnehmend auf **Fig. 3** ist ein Blockdiagramm einer physischen Einheit gemäß einer weiteren Ausführungsform der vorliegenden Erfindung gezeigt. Wie in **Fig. 3** gezeigt wird, umfasst die physische Einheit **300** Verbindungstraining und einen Statusmanager **310**. Dieser Statusmanager kann wie oben beschrieben sein und kann eine Logiksammlung sein, um eine Schnittstellenverbindung eines Protokollstapels mit einem ersten Kommunikationsprotokoll zu einer physischen Einheit eines zweiten Protokolls, z. B. eines anderen Kommunikationsprotokolls, zu ermöglichen.

[0044] Wie weiter in **Fig. 3** gesehen werden kann, können Verbindungstraining und Statusmanager **310** in Kommunikation mit mehreren M-PHY **320₀–320_n** sein. Indem mehr als eine solche PHY zur Verfügung gestellt werden, können höhere Datenkommunikationsraten erzielt werden. Man beachte, dass, obwohl jede M-PHY, die in **Fig. 3** gesehen werden kann, eine gewisse Anzahl an Logikelementen enthalten kann, damit ihre einzelnen unabhängigen Kommunikationen durchgeführt werden können, die Gesamtsteuerung der Kommunikation dieser unterschiedlichen M-PHY über Verbindungstraining und Statusmanager **310** erfolgen kann. Man verstehe auch, dass, obwohl in **Fig. 3** eine Vielzahl von M-PHYs gezeigt sind, in anderen Ausführungsformen mehrere PHY-Einheiten eines anderen Typs präsent sein können und weitere vielfache heterogene PHY-Einheiten bereitgestellt werden können. Man beachte, dass jede M-PHY-Einheit entweder als Teil einer besonderen logischen Verbindung oder in Gruppen benutzt werden kann, wobei eine Gruppe einer einzelnen logischen Verbindung zugeordnet ist. Jedes Gerät kann typischerweise eine einzige logische Verbindung benutzen, obwohl in einigen Ausführungsformen ein einzelnes physisches Gerät mehrere logische Verbindungen benutzen kann, z. B. um dedizierte Verbindungsressourcen für die verschiedenen Funktionen einer Multifunktionskomponente bereitzustellen.

[0045] Es wird nun auf **Fig. 4** Bezug genommen, die ein Blockdiagramm ist, das weitere Details eines Protokollstapels gemäß einer Ausführungsform der vorliegenden Erfindung zeigt. Wie in **Fig. 4** gezeigt, umfasst Stapel **400** verschiedene Ebenen, einschließlich einer Transaktionsebene **410**, einer Datenverbindungsebene **420** und einer physischen Ebene **430**. Wie oben beschrieben, können diese verschiedenen Ebenen unter Verwendung von entweder herkömmlichen Transaktions- und Datenverbindungsteilen eines PCIe™-Protokollstapels oder von modifizierten Versionen des Stapels konfiguriert werden, um eine Wechselwirkung zwischen diesen Ebenen des ersten Kommunikationsprotokolls und einer physischen Ebene von einem anderen Kommunikationsprotokoll, welches in der Ausführungsform der **Fig. 4** M-PHYs gemäß der MIPI-Spezifikation sein kann, zu ermöglichen.

[0046] Wie in **Fig. 4** bezüglich einer Senderichtung, in der Informationen von Protokollstapel **400** übertragen werden, gesehen werden kann, werden die eingehenden Informationen an den Protokollstapel, z. B. von anderen Schaltungen eines SoC, wie z. B. einem Kern oder einer anderen Verarbeitungslogik, in einem Sendepaket-Assembler **412** der Transaktionsebene empfangen, die typischerweise Steuer- und Datenpfade kombiniert, um ein TLP zu bilden. Nachdem sie in Sendepaketten zusammengesetzt wurden, welche in verschiedenen Ausführungsformen Datenpakete sein können, die z. B. 1-4096 Bytes (oder eine kleinere maximal zulässige

Größe, z. B. 128 oder 256) haben können, werden die zusammengesetzten Pakete an eine Flusssteuerung **414** gesendet, die bestimmt, ob basierend auf der Anzahl der für das nächste/die nächsten für die Übertragung in der Warteschlange stehende/stehenden TLP(e) erforderlichen Kredite ausreichend Flusssteuerungskredite vorhanden sind, und steuert die Einreihung von Paketen in die Datenverbindungsebene **420**. Genauer gesagt, wie zu sehen ist, werden diesen eingereichten Paketen ein Fehlerdetektor und Sequenzer **422** zugewiesen, die in einer Ausführungsform eine TLP-Sequenznummer und eine CRC erzeugen. Wie weiter zu sehen ist, umfasst die Datenverbindungsebene **420** ferner einen Sendenachrichtenmechanismus **426**, der wiederum DLLPs für die Verbindungsmanagement-Funktionen erzeugt und mit einer Datenverbindungssendesteuerung **425** verkoppelt ist, welche eine Steuerungsfunktion für die Flusssteuerungs- und die Datenverbindungsintegritäts-Mechanismen (ACK/NAK) ist; man beachte, dass dies unterteilt sein kann, so dass diese Funktionen über verschiedene Logikblöcke implementiert werden.

[0047] Wie weiter zu sehen ist, werden die verarbeiteten Datenpakete an einen Wiederholungspuffer **424** weitergegeben, der eine Kopie von jedem TLP speichert, bis diese von der Komponente auf der anderen Seite der Verbindung quittiert wird. Man beachte, dass dies in der Praxis mit Pufferung höher im Stapel implementiert werden kann (in oder über Assembler **412**) und dass diese in entsprechenden Einträgen gespeichert werden können, bis sie zur Übertragung zu der physischen Ebene **430** über einen Daten/Nachrichtenwähler **428** ausgewählt werden. Im allgemeinen können die oben beschriebenen Transaktions- und Datenverbindungsebenen in Übereinstimmung mit bestimmten Modifikationen, wie sie weiter unten beschrieben werden, mit herkömmlichen PCIe™-Protokollstapelschaltungen betrieben werden.

[0048] Stattdessen können bezüglich der physischen Ebene **430** viele weitere Modifikationen bestimmter logischer Komponenten dieser Ebene, z. B. modifiziert von einem PCIe™-Protokollstapel, auftreten sowie eine Schnittstelle zu den tatsächlichen physischen Teilen der physischen Einheit eines anderen Kommunikationsprotokolls bereitgestellt werden. Wie ersichtlich ist, können die ankommenden Pakete auf einen Framing-Generator **432** angewandt werden, der Framing-Symbole auf physischer Ebene hinzufügt und Framing für die Pakete erstellt und diese einem Breiten-/Positionen-Mapper **434** zur Verfügung stellt, der die Bytes in dem Datenpfad verschiebt, um den gewünschten Abgleich für die externe Übertragung zu erzeugen, wobei, wenn notwendig, die Datenpfadbreite angepasst wird und wobei dieser Mapper wiederum mit einem Trainer und Sprungsequenzer **436** gekoppelt ist, die verwendet werden, um Verbindungstraining und Sprungsequenzierung durchzuführen. Wie zu sehen ist, können der Framing-Generator **432**, Trainer/Sequenzer **436** und ein Daten-/Sequenzwähler **438** alle mit einer Übertragungssteuerung auf physischer Ebene **435** gekoppelt sein, die aus einem Sendeempfänger-Teil LTSSM und einer verwandten Logik besteht. Block **436** ist eine Logik, die die Übertragungen auf physischer Ebene wie z. B. Trainingssätze (TS) und geordnete Skip-Listen generiert. Auf diese Weise können die framed Pakete ausgewählt und einer physischen Schaltung zur Verfügung gestellt werden, um Codierung, Serialisierung und Treiben der serialisierten Signale zu einer physischen Verbindung entsprechend der verarbeiteten Pakete auszuführen. In einer Ausführungsform kann das Mapping von Symbolunterschiede zwischen den verschiedenen Kommunikationsprotokollen in dem Frame-Generator **432** ausgeführt werden.

[0049] Wie zu sehen ist, können mehrere einzelne Kanäle oder Spuren für diese physische Verbindung bereitgestellt werden. In der gezeigten Ausführungsform kann jeder physische Kanal oder jede Spur eine eigene unabhängige PHY-Einheits-Sendesaltung **445₀–445_j** umfassen, von denen in einer Ausführungsform jede Teil einer M-PHY-Einheit in Übereinstimmung mit der MIPI-Spezifikation sein kann. Wie hier beschrieben ist, können im Gegensatz zu PCIe™, wo die Anzahl von Sendern und Empfängern übereinstimmt, eine unterschiedliche Anzahl von Sender und Empfänger vorhanden sein. Wie gesehen, kann damit jede Sendesaltung **445** einen Codierer umfassen, der Symbole gemäß einer 8b/10b-Codierung codiert, sowie einen Serializer, um die kodierten Symbole zu serialisieren, und einen Treiber, um die Signale zu der physischen Verbindung zu treiben. Wie weiter zu sehen ist, kann jede Spur oder Kanal mit einer logischen Einheit **440₀–440_j** verbunden sein, die eine logische Schaltung entsprechend der MIPI-Spezifikation für eine M-PHY sein kann und somit die physische Kommunikation über die entsprechende Spur verwaltet.

[0050] Man beachte, dass diese mehreren Spuren konfiguriert werden können, mit verschiedenen Raten betrieben zu werden, und Ausführungsformen können unterschiedliche Anzahlen derartiger Spuren umfassen. Darüber hinaus ist es möglich, eine unterschiedliche Anzahl von Spuren und Spurgeschwindigkeiten in Sende- und Empfangsrichtung zu haben. Obwohl also eine gegebene Logikeinheit **440** den Betrieb einer entsprechenden Spur der PHY **445** steuert, verstehe man, dass die Übertragungssteuerung auf physischer Ebene **435** die gesamte Übertragung von Informationen über die physische Verbindung steuern kann. Man beachte, dass in einigen Fällen eine sehr grundlegende Funktion durch distinkte Logik durchgeführt wird, die mit jeder Spur assoziiert ist; für Fälle, in denen Spuren mehr als einer Verbindung zugeordnet werden können, können mehrere

LTSSM-Instanzen vorgesehen werden; für eine trainierte Verbindung gibt es eine einzige LTSSM in jeder Komponente, die sowohl die Sendeempfänger- als auch die Empfängerseite steuert. Diese Gesamtsteuerung kann Leistungssteuerung, Verbindungsgeschwindigkeits-Steuerung, Verbindungsbreiten-Steuerung, Initialisierung und so weiter umfassen.

[0051] Immer noch mit Bezug auf **Fig. 4**, können eingehende Informationen, die über die physische Verbindungen empfangen werden, in ähnlicher Weise durch physische Ebene **430**, Datenverbindungsebene **420** und Transaktionsebene **410** über den Empfangsmechanismus dieser Ebenen übermittelt werden. In der in **Fig. 4** gezeigten Ausführungsform kann jede PHY-Einheit ferner eine Empfangsschaltung umfassen, nämlich die Empfangsschaltung **455₀–455_k**, die in der gezeigten Ausführungsform für jede Spur der physischen Verbindung präsent sein kann. Man beachte, dass in dieser Ausführungsform die Anzahl von Empfängerschaltungen **455** von der Anzahl der Senderschaltungen **445** abweicht. Wie zu sehen ist, kann diese physische Schaltung einen Eingangspuffer umfassen, um eingehende Informationen zu empfangen, sowie einen Deserialisierer, um die Informationen zu deserialisieren, und einen Decoder, der die übermittelten Symbole in einer 8b/10b-Codierung dekodieren kann. Wie weiter zu sehen ist, kann jede Spur oder Kanal mit einer logischen Einheit **450₀–450_k** verbunden sein, die eine logische Schaltung entsprechend einer gegebenen Spezifikation (wie z. B. die MIPI-Spezifikation für eine M-PHY) sein kann und somit die physische Kommunikation über die entsprechende Spur verwaltet.

[0052] Die decodierten Symbole können wiederum einem logischen Teil der physischen Ebene **430** zur Verfügung gestellt werden, der, wie gesehen, elastische Puffer **460** sein kann, wobei die elastischen Puffer Taktunterschiede zwischen dieser Komponente und der anderen Komponente der Verbindung unterstützen können. Man beachte, dass sich die Position in verschiedenen Implementierungen ändern kann, z. B. unter dem 8b/10b-Decodierer liegen kann, oder mit dem Spur-Deskew-Puffer kombiniert werden kann, und die ankommenden decodierten Symbole zu speichern. Im Gegenzug können die Informationen einem Breiten-/Positionen-Mapper **462** und von da einem Spur-Deskew-Puffer **464** zur Verfügung gestellt werden, der Deskew über mehrere Spuren und für Multispurfälle ausführt, wobei Puffer **464** Variationen im Signalversatz zwischen Spuren verarbeiten kann, um Bytes wieder zu justieren. Im Gegenzug können die versatzkorrigierten Informationen einem Frame-Prozessor **466** zur Verfügung gestellt werden, der das in den eingehenden Informationen vorhandene Framing entfernen kann.

[0053] Wie zu sehen ist, kann eine Empfangssteuerung auf physischer Ebene **465** mit einem elastischen Puffer **460**, Mapper **462**, Deskew-Puffer **464** und Framing-Prozessor **466** gekoppelt sein.

[0054] Immer noch mit Bezug auf **Fig. 4** können die wiederhergestellten Pakete sowohl einem Nachrichtenenempfangs-Mechanismus **478** als auch einem Fehlerdetektor, einer Sequenzprüfung und Link-Level Retry (LLR) Requestor **475** bereitgestellt werden. Diese Schaltung kann eine Fehlerkorrektur-Prüfung der ankommenden Pakete durchführen, z. B. mit der Durchführung von CRC-Prüfsummen-Operationen, Sequenzierungsprüfungen und der Anforderung von Link-Level-Retry der inkorrekt empfangenen Pakete. Sowohl der Nachrichtenenempfangs-Mechanismus **478** als auch der Fehlerdetektor/Requestor **475** können unter der Kontrolle einer Datenverbindungs-Empfangssteuerung **480** sein.

[0055] Immer noch mit Bezug auf **Fig. 4** können die auf dieser Weise in der Einheit **475** verarbeiteten Pakete an die Transaktionsebene **410** und, genauer gesagt, an eine Flusssteuerung **485** weitergegeben werden, die die Flusssteuerung für diese Pakete durchführt, um sie einem Paket-Interpreter **495** bereitzustellen. Der Paket-Interpreter **495** führt eine Interpretation der Pakete aus und leitet sie zu einem ausgewählten Ziel weiter, z. B. zu einem bestimmten Kern oder einer anderen logischen Schaltung des Empfängers. Obwohl dies auf dieser hohen Ebene in der Ausführungsform von **Fig. 4** gezeigt ist, versteht sich, dass dadurch der Umfang der vorliegenden Erfindung in dieser Hinsicht in keiner Weise eingeschränkt wird.

[0056] Man beachte, dass PHYs **440** dieselbe 8b/10b-Codierung für die Übertragung benutzen kann, die durch PCIe™ unterstützt wird. Das 8b/10b-Codierungsschema bietet spezielle Symbole, die unterschiedlich von den Datensymbolen sind, die verwendet werden, um Zeichen darzustellen. Diese Sonderzeichen können für verschiedene Verbindungsmanagement-Mechanismen verwendet werden, wie es in dem Kapitel „Physische Ebene der PCIe™-Spezifikation“ beschrieben ist. Weitere Sonderzeichennutzungen durch die M-PHY sind in der MIPI M-PHY-Spezifikation beschrieben. Ausführungsbeispiele können eine Zuordnung zwischen PCIe™- und MIPI M-PHY-Symbolen vorgeben.

[0057] Unter Bezugnahme auf Tabelle 1 wird jetzt eine beispielhafte Zuordnung von PCIe™-Symbolen zu M-PHY-Symbolen in Übereinstimmung mit einer Ausführungsform der vorliegenden Erfindung gezeigt. Somit

zeigt diese Tabelle die Zuordnung von Sonderzeichen für einen konvergenten Protokollstapel gemäß einer Ausführungsform der vorliegenden Erfindung.

Tabelle 1

Steuersymbole	PCIe-Codierung	Zuordnung von konvergentem Stapel	MIPI M-PHY-Zuordnung	Anmerkung
K28.5	COM	COM	Marker0	
K28.3	IDL	IDL	Marker1	
K28.6	Reserviert	SDP	Marker2	Hinweis: SDP hier zuordnen, da die originale PCIe-Codierung von M-PHY reservierter Codierung zugeordnet ist.
K23.7	PAD	SKP	Marker3	Hinweis: SKP zu K23.7-Symbol, neutrale Disparität, zuordnen, da es das Stör-Entstörgerät nicht vorantreibt.
K27.7	STP	STP	Marker4	
K29.7	ENDE	ENDE	Marker5	
K30.7	EDB	EDB	Marker6	
K28.1	FTS	PAD	Füller	Hinweis: PAD hier zuordnen, da SKP seine Kodierungen übernommen hat. FTS ist nicht erforderlich, da der M-PHY SYNC-Mechanismus verwendet werden kann. M-PHY-SYNC-Mechanismus ist eine bessere Option, da die definierten SYNC-Symbole eine höhere Kantendichte haben. RMMI-Spez. erfordert die Einfügung von M-PHY-Füllern, wenn TX_DORDY niedrig ist.
K28.0	SKP	Reserviert	Reserviert	
K28.2	SDP	Reserviert	Reserviert	

K28.7	EIE	Reserviert	Reserviert	Hinweis: EIE ist nicht erforderlich, da die M-PHY Rauschsperre Übergang von DIF-Z zu N DIF erkennt.
Weitere	Reserviert	Reserviert	Reserviert	

[0058] Die 8b/10b-Decodierregeln sind dieselben, die für die PCIe™-Spezifikation definiert sind. Die einzige Ausnahme für die 8b/10b-Regeln ist bei Erkennung eines TAIL OF BURST, welches eine spezifische Sequenz ist, die die 8b/10b-Regeln verletzt. Gemäß verschiedenen Ausführungsformen kann die physische Ebene **430** eine Benachrichtigung an die Datenverbindungsebene **420** senden, um Fehler während des TAIL OF BURST zu melden.

[0059] In einer Ausführungsform kann das Framing und die Anwendung von Symbolen auf Spuren ausgeführt werde, wie es in der PCIe™-Spezifikation definiert ist, während die Datenverschlüsselung dieselbe sein kann, wie sie in der PCIe™-Spezifikation definiert ist. Man beachte jedoch, dass die Datensymbole, die in der Vorbereitungsphase (PREPARE) einer Kommunikation gemäß der MIPI-Spezifikation übertragen werden, nicht verschlüsselt sind.

[0060] Im Hinblick auf Verbindungsinitialisierung und -training kann der Verbindungsmanager die Konfiguration und Initialisierung der Verbindung übernehmen, was, wie oben erläutert, einen oder mehrere Kanäle von Spuren, die Unterstützung des normalen Datentransfers, die Unterstützung von Zustandsübergängen bei der Erholung von Verbindungsfehlern und Port-Neustart von einem niedrigen Energiezustand umfassen kann.

[0061] Um eine solche Operation durchzuführen, können die folgenden physischen und verbindungsverwandten Funktionen a priori bekannt sein, z. B. vor der Initialisierung: PHY-Parameter (z. B. einschließlich der anfänglichen Verbindungsgeschwindigkeit und unterstützten Geschwindigkeit und die anfängliche Verbindungsbreite und unterstützte Verbindungsbreiten).

[0062] In einer Ausführungsform kann das Training verschiedener Operationen umfassen. Solche Operationen können die Initialisierung der Verbindung mit der konfigurierten Verbindungsgeschwindigkeit und -breite, Bitlock pro Spur, Symbollock pro Spur, Spur Polarität und Signalversatz von Spur zu Spur für mehrspurige Links umfassen. Auf diese Weise kann das Training Spurpolarität erkennen und entsprechende Anpassungen durchführen. Man beachte jedoch, dass das Verbindungstraining in Übereinstimmung mit einer Ausführungsform der vorliegenden Erfindung u. U. keine Verbindungs-Datenraten und Verbindungs-Breiten-Verhandlung, Verbindungsgeschwindigkeit und Breitendegradierung enthalten kann. Stattdessen, wie oben beschrieben, kennen bei der Initialisierung einer Verbindung beide Entitäten a priori die anfängliche Verbindungsbreite und -geschwindigkeit. Damit kann der Zeit- und Rechenaufwand in Verbindung mit den Verhandlungen vermieden werden.

[0063] Geordnete PCIe™-Listen können mit den folgenden Modifikationen verwendet werden: TS1 und TS2 geordnete Listen werden verwendet, um IP-Wiederverwendung zu erleichtern, aber viele Teile des Trainings geordneter Listen werden ignoriert. Schnelle Trainingssequenzen werden ebenfalls nicht verwendet. Eine elektrisch idle geordnete Liste (electrical idle ordered set, EIOS) kann beibehalten werden, um IP-Wiederverwendung zu erleichtern, oder kann eine Skip-OS sein, wobei aber die Frequenz der Skip-OS eine anderen Datenrate als nach der PCIe™ Spezifikation haben kann. Man beachte auch, dass Datenstrom-geordnete Listen und Symbole dieselben wie nach den PCIe™ Spezifikation sein können.

[0064] Die folgenden Ereignisse werden mitgeteilt, um Verbindungstraining und -Management zu erleichtern: (1) Präsenz, was verwendet werden kann, um anzuzeigen, dass eine aktive PHY auf dem remote Ende der Verbindung vorhanden ist; und (2) Konfiguration bereit, was ausgelöst wird, um anzuzeigen, dass die PHY-Parameterkonfiguration abgeschlossen ist und die PHY zur Operation mit dem konfigurierten Profil bereit ist. In einer Ausführungsform können solche Informationen über ein konsolidiertes Seitenbandsignal in Übereinstimmung mit einer Ausführungsform der vorliegenden Erfindung übermittelt werden.

[0065] Zum Zweck der Kontrolle der elektrisch idle Situationen weist die PHY eine TAIL-OF-BURST-Sequenz auf, die verwendet wird, um anzuzeigen, dass der Sender in einen elektrischen Ruhezustand eintritt. In einer

Ausführungsform kann der Seitenbandkanal verwendet werden, um einen Austritt aus dem elektrischen Ruhezustand anzuzeigen. Man beachte, dass diese Indikation zusätzlich zu den PHY-Squelch-Break-Mechanismen angewandt werden kann. Ein ERÖFFNUNG-Sequenz von Symbolen kann als EIOS übertragen werden, um den Eintritt in den elektrischen Ruhezustand anzuzeigen.

[0066] In einigen Ausführungsformen ist eine schnelle Trainingssequenz (fast training sequence, FTS) nicht definiert. Stattdessen kann eine PHY eine bestimmte Sequenz auf physischer Ebene für den Austritt aus dem Pause-/Ruhezustand in einen Burst-Status benutzen, die verwendet werden kann, um Bit-Lock, Symbol-Lock und für Signalversatz von Spur zu Spur zu berücksichtigen. Eine kleine Anzahl von FTS kann als eine Sequenz von Symbolen für die Robustheit definiert werden. Ein geordnete Liste für den Start des Datenstroms sowie die Verbindungs-Fehlerbehebung kann der PCIe™-Spezifikation folgen.

[0067] In Bezug auf die Datenraten können die anfänglichen Datenraten, mit der die Verbindung initialisiert wird, in verschiedenen Ausführungsformen eine vorbestimmte Datenrate sein. Eine Datenraten-Änderung von dieser ersten Verbindungsgeschwindigkeit kann durch einen Durchgang durch einen Wiederherstellungsstatus auftreten. Ausführungsformen können asymmetrische Verbindungsdatenraten unterstützen, wobei die Datenraten in entgegengesetzten Richtungen unterschiedlich sein dürfen.

[0068] In einer Ausführungsform können die unterstützten Verbindungsbreiten denen der PCIe™-Spezifikation folgen. Ferner können, wie oben beschrieben, Ausführungsformen u. U. kein Protokoll zur Verhandlung der Verbindungsbreite unterstützen, da die Verbindungsbreite vorgegeben ist, und als Resultat kann das Verbindungstraining vereinfacht werden. Selbstverständlich können die Ausführungsformen Unterstützung für asymmetrische Verbindungsbreiten in entgegengesetzte Richtungen bieten. Die anfängliche Verbindungsbreite und die anfängliche Datenrate für jede Richtung der Verbindung kann jedoch ebenfalls so konfiguriert werden, dass sie a priori vor dem Training bekannt ist.

[0069] In Bezug auf physische Ports der PHY-Einheit ist die Fähigkeit für einen xN Port, eine xN Verbindung sowie eine x1 Verbindung zu erstellen (wobei N 32, 16, 12, 8, 4, 2 und 1 sein kann), nicht erforderlich und die Fähigkeit für einen xN-Port, eine Verbindung jeder Breite zwischen N und 1 zu erstellen, ist optional. Ein Beispiel für dieses Verhalten beinhaltet einen x16-Port, der nur in eine Verbindung konfiguriert sein kann, wobei aber die Breite der Verbindung auf x12, x8, x4, x2 sowie erforderliche Breiten x16 und x1 konfiguriert sein können. Damit können Entwickler, die Vorrichtungen unter Verwendung eines Protokollstapels in Übereinstimmung mit einer Ausführungsform der vorliegenden Erfindung implementieren wollen, die Ports zwischen zwei verschiedenen Komponenten in einer Weise verbinden, dass diese Komponenten die obigen Anforderungen erfüllen können. Wenn die Ports zwischen Komponenten in einer Weise verbunden werden, die inkonsistent mit dem Verwendungszweck ist, wie er von der Port-Beschreibungen/den Datenblätter der Komponente definiert ist, dann ist das Verhalten undefiniert.

[0070] Darüber hinaus ist die Fähigkeit, einen Port in zwei oder mehr Verbindungen zu spalten, nicht verboten. Wenn eine solche Unterstützung für eine bestimmte Konstruktion angemessen ist, kann der Port im Training konfiguriert werden, eine bestimmte Breite zu unterstützen. Ein Beispiel für dieses Verhalten wäre ein x16-Port, der in der Lage ist, zwei x8-Verbindungen, vier x4-Verbindungen oder 16 x1-Verbindungen zu konfigurieren.

[0071] Bei Verwendung von 8b/10b-Codierung ist ein eindeutiger Signalversatz-Mechanismus von Spur zu Spur, wie in der PCIe™-Spezifikation, das COM-Symbol von geordneten Listen, das in der Trainingssequenz oder SKP geordneten Listen empfangen wird, da geordnete Listen gleichzeitig auf allen Spuren einer konfigurierten Verbindung übertragen werden. Das während der Sync-Sequenz von HS-BURST übertragene MK0-Symbol kann zum Signalversatz von Spur zu Spur verwendet werden.

[0072] Wie oben kurz mit Bezug auf **Fig. 4** beschrieben wurde, kann ein Verbindungsmanager konfiguriert werden, um verschiedene Operationen durchzuführen, einschließlich der Anpassung der oberen Ebenen eines PCIe™-Protokollstapels auf eine PHY-Einheit unterer Ebene mit einem anderen Protokoll. Darüber hinaus kann dieser Verbindungsmanager einzelne oder mehrere Spuren konfigurieren und verwalten und kann Unterstützung für eine symmetrische Verbindungsbandbreite, Kompatibilität der Statusmaschine mit PCIe™-Transaktions- und Datenverbindungsebenen, Verbindungstraining, optionale symmetrischen Verbindungspausenstatus und die Steuerung der Seitenbandsignale für stabile Kommunikation umfassen. Dementsprechend berücksichtigen Ausführungsformen die Implementierung der PCIe™-Transaktions und Datenverbindungsebenen mit beschränkten Modifikationen für verschiedene Verbindungsgeschwindigkeiten und asymmetrische Verbindungen. Darüber hinaus kann durch Verwendung eines Verbindungsmanagers in Übereinstimmung mit einer Ausführungsform der vorliegenden Erfindung Unterstützung für asymmetrische Multispur-

Verbindungsconfiguration, Seitenbandkonsolidierung und dynamische Bandbreiten-Skalierbarkeit erzielt werden, während ebenso die Überbrückung zwischen Ebenen unterschiedlicher Kommunikationsprotokolle ermöglicht wird.

[0073] Unter Bezugnahme auf **Fig. 5** ist ein Zustandsdiagramm **500** für eine Verbindungstraining-Statusmaschine gezeigt, die Teil eines Verbindungsmanagers in Übereinstimmung mit einer Ausführungsform der vorliegenden Erfindung sein kann. Wie in **Fig. 5** gezeigt, kann das Verbindungstraining in einem Erkennungsstatus **510** beginnen. Dieser Zustand tritt beim Einschalt-Reset auf und bezieht sich sowohl auf Upstream- als auch auf Downstream-Ports. Nach vollständiger Ausführung des Reset können alle konfigurierten Spuren in einen bestimmten Status eintreten, nämlich einen HIBERN8-Status, woraufhin jedes Ende der Verbindung ein Signal, z. B. ein Präsenzsignal, mit einem Seitenbandkanal senden kann. Man beachte, dass in diesem Erkennungsstatus ein Hochimpedanzsignal, nämlich ein DIF-Z-Signal, auf allen Spuren getrieben werden kann.

[0074] So geht die Steuerung von einem Erkennungsstatus **510** in den Konfigurationsstatus **520** über, wenn das PRÄSENZ-Ereignis signalisiert und empfangen wird, und diese hohe Impedanz wird auf alle konfigurierten Spuren getrieben. Im Konfigurationszustand **520** können die PHY-Parameter konfiguriert werden und nach Beendigung auf allen konfigurierten Spuren von jedem Ende der Verbindung kann ein Konfiguration-Bereit-Signal (CFG-RDY) angegeben werden, z. B. unter Verwendung der Seitenbandverbindung, während die hohe Impedanz auf allen Spuren beibehalten wird.

[0075] Damit geht die Steuerung nach dem Senden und Empfangen von dieser Konfiguration-Bereit-Anzeige über die Seitenbandverbindung in einen Pause-Status (stall state) **530** über. In diesem L0.STALL-Status geht die PHY in einen Pause-Status über und treibt weiterhin die hohe Impedanz auf alle konfigurierten Spuren. Wie zu sehen ist, kann, je nachdem, ob Daten für Übertragung oder Empfang zur Verfügung stehen, die Steuerung in einen aktiven Status L1 (Status **530**), einen stromsparenden Status (L1-Status **540**), einen noch stromsparenderen Status (L1.OFF-Status **545**) oder zurück in einen Konfigurationsstatus **520** übergehen.

[0076] Damit kann in diesem PAUSE-Status (STALL) ein negatives Treibersignal DIF-N auf allen konfigurierten Spuren kommuniziert werden. Wenn es dann durch den Initiator angeordnet wird, kann eine BURST-Sequenz beginnen. Dementsprechend geht die Steuerung nach der Übertragung eines MARKER-0-(MK0)Symbols in den aktiven Status **530** über.

[0077] In einer Ausführungsform kann ein Empfänger den Austritt aus dem STALL-Status auf allen konfigurierten Spuren erkennen und einen Bit-Lock entsprechend z. B. der MIPI-Spezifikation durchführen. Bei Ausführungsformen mit einer Multispur-Verbindung kann dies MK0-Symbol verwendet werden, um Signalversatz von Spur zu Spur zu etablieren.

[0078] Stattdessen können alle konfigurierte Spuren, wenn sie in einen stromsparenden Status (nämlich den L1-Status **540**) gesendet werden, in einen SLEEP-Status übergehen. Im Gegenzug können alle konfigurierte Spuren, wenn sie in einen stromsparenderen Status (nämlich den L1.OFF-Status **545**) gesendet werden, in den HIBERN8-Status übergehen. Wenn sie schließlich zurück in den Konfigurationsstatus gesendet werden, können in ähnlicher Weise alle konfigurierte Spuren in den HIBERN8-Status übergehen.

[0079] Immer noch mit Bezug auf **Fig. 5** wird für aktive Datenübertragung die Steuerung damit in den aktiven Status **550** übergehen. Insbesondere ist dies der Status, in dem Verbindungs- und Transaktionsebenen beginnen, Informationen unter Verwendung der Datenverbindungsebene-Pakete (DLLPs) und TLP auszutauschen. Als solches kann eine Nutzlastübertragung stattfinden, und am Ende dieser Übertragung kann ein TAIL-OF-BURST-Symbol kommuniziert werden.

[0080] Wie zu sehen ist, kann die Steuerung von diesem aktiven Status in einen STALL-Status **530**, einen Wiederherstellungsstatus **560**, z. B. in Reaktion auf einen Empfangsfehler oder wenn sie ansonsten in diesen Status gesendet wird, oder in einen stromsparenderen (z. B. L2) Status **570** zurückkehren.

[0081] Um zum STALL-Status zurückkehren, kann der Sender eine EIOS-Sequenz gefolgt von einer TAIL-OF-BURST-Indikation zu allen konfigurierten Spuren senden.

[0082] Wenn ein Fehler auftritt oder dies anderweitig angewiesen wird, kann die Steuerung auch in einen Wiederherstellungsstatus **560** übergeben. Hier führt ein Übergang zur Wiederherstellung dazu, dass alle konfigurierten Spuren in beide Richtungen in den STALL-Status übergehen. Um dies zu bewirken, kann ein GO-TO-STALL-Signal auf der Seitenbandverbindung gesendet werden und der Sender dieses Signals kann auf eine

Antwort warten. Wenn dieses STALL-Signal gesendet und empfangen wurde, wie von einer GO-TO-STALL-Indikation auf der Seitenbandverbindung angefordert wurde, geht die Steuerung zurück zum STALL-Status **530**. Man beachte, dass dieser Wiederherstellungsstatus somit das Protokoll mit Hilfe des Seitenbandes etabliert, um den gleichzeitigen Eintritt in den STALL-Status zu koordinieren.

[0083] Im Hinblick auf die stromsparenden Status L1 und L1.OFF läuft der Betrieb nach Status **540** und **545**. Insbesondere geht die Steuerung von dem STALL-Status in den stromsparenderen Status L1 **540** über, so dass die PHY in einen SLEEP-Status versetzt werden kann. In diesem Status wird ein negatives Antriebssignal, nämlich ein DIF-N-Signal, auf alle konfigurierten Spuren getrieben. Wenn die Steuerung angewiesen wird, den Status zu verlassen, geht diese zurück zum STALL-Status **530**, z. B. über die Signalisierung eines PRÄSENZ-Signals über die Seitenbandverbindung.

[0084] Wie ebenfalls zu sehen ist, kann in den energiesparenderen Status L1.OFF eingetreten werden, wenn alle L1.OFF-Bedingungen erfüllt sind. In einer Ausführungsform können diese Bedingungen das vollständige Power-Gating oder Abschalten des Stroms für die PHY-Einheit umfassen. In diesem energiesparenderen Status kann die PHY in den HIBERN8-Status übergehen und das Hochimpedanzsignal auf allen konfigurierten Spuren getrieben werden. Um diesen Status zu verlassen, geht die Steuerung durch das Treiben von DIF-N auf alle konfigurierten Spuren zurück in den STALL-Status.

[0085] Wie weiter in **Fig. 5** gesehen werden kann, können weitere Status präsent sein, nämlich ein noch energiesparenderer Status (L2) **570**, in den von einem aktiven Status eingetreten werden kann, wenn der Strom bereit ist, ausgeschaltet zu werden. In einer Ausführungsform kann dieser Status derselbe sein, wie der der PCIe™ Spezifikation.

[0086] Unter Bezugnahme auf Tabelle 2 ist eine Zuordnung zwischen den LTSSM-Status gemäß der PCIe™-Spezifikation und den entsprechenden M-PHY-Status in Übereinstimmung mit einer Ausführungsform der vorliegenden Erfindung gezeigt.

Tabelle 2

LTSSM-Status	M-PHY-Status	Hinweise
Erkennung, Polling	SPEICHERN	Statusübergänge durch SAVE-Unterstatus
Konfiguration	BURST	BURST-(PREP-, SYNC-)Unterstatus
Wiederherstellung	BURST/SLEEP/STALL	Kann im BURST-Status sein, wird aber durch SLEEP/STALL in BURST übergehen
L0	BURST (Arbeitslast)	BURST-Modus und Austauschtransaktionen
L0s	STALL	STALL-Status
L1	SLEEP	SLEEP-Status
L1.OFF	HIBERN8	HIBERN8
L2	STROMLOS	STROMLOS-Status
Deaktiviert	DEAKTIVIERT	DEAKTIVIERT-Status
Loopback	Keine Aktivität	Die Verbindungsgeschwindigkeit kann sich beim Eintritt von Konfiguration zu Loopback ändern
Warmstart	INLINE RESET	IN-LINE RESET-Status

[0087] Wie oben mit Bezug auf **Fig. 2** beschrieben, stellen Ausführungsformen einen konsolidierten Seitenbandmechanismus zur Verfügung, der für das Verbindungsmanagement zusammen mit optionaler In-Band-Unterstützung verwendet werden kann. Auf diese Weise können unter Verwendung der Seitenbandschaltungen und -verbindung Verbindungsmanagement und -steuerung unabhängig von der Schaltung höherer Geschwindigkeit (und höherem Leistungsverbrauchs) der physischen Ebene für die primären Verbindung ausge-

führt werden. Weiter kann in dieser Weise dieser Seitenband-Kanal verwendet werden, wenn die Teile der PHY-Einheit, die mit der primären Verbindung assoziiert sind, ausgeschaltet sind, wodurch der Stromverbrauch reduziert wird. Außerdem kann dieser konsolidierte Seitenbandmechanismus vor dem Training der primären Verbindung verwendet werden und kann ebenfalls verwendet werden, wenn ein Fehler in der primären Verbindung präsent ist.

[0088] Weiterhin kann über diesen konsolidierten Seitenbandmechanismus eine einzige Verbindung, z. B. ein Paar differentieller Drähte in jeder Richtung, vorhanden sein, wodurch beide Pin-Zahlen reduziert werden und das Hinzufügen neuer Funktionen ermöglicht wird. Ausführungsformen können auch schnelleres und robusteres Clock/Power-Gating ermöglichen und Unklarheiten in herkömmlichen Protokollen, wie z. B. ein PCIe™-Seitenbandmechanismus, der diese Verbindung benutzt, beseitigen.

[0089] Obwohl der Umfang der vorliegenden Erfindung in dieser Hinsicht nicht beschränkt ist, kann in verschiedenen Ausführungsformen die Seitenbandverbindung (z. B. Seitenbandverbindung **270** von **Fig. 2**) als bidirektionales Einzeldraht-Seitenbandsignal, als Doppeldraht mit zwei Richtungen mit einem Satz von unidirektionalen Signalen, als ein Niedergeschwindigkeits-In-Band-Signalisierungsmechanismus, wie z. B. einem M-PHY Impulsbreitenmodulationsmodus (pulse width modulation, PWM) niedriger Leistung oder als In-Band-Hochgeschwindigkeits-Signalmechanismus wie z. B. geordnete Listen auf physischer Ebene oder DLLPs implementiert werden.

[0090] Als Beispiele und nicht zum Zweck der Einschränkung können verschiedene Ansätze für die physische Ebene unterstützt werden. Ein erster Ansatz kann ein Einzeldraht mit bidirektionalem Seitenbandsignal mit niedrigster Pinzahl sein, wenn eine Seitenbandverbindung verwendet wird. In einigen Ausführungsformen kann dieses Signal auf einem vorhandenen Seitenband gemultiplext werden, z. B. PERST#, WAKE#- oder CLKREQ-Signale. Ein zweiter Ansatz kann ein Doppeldraht mit zwei Richtungen mit einem Satz von unidirektionalen Signalen sein, welcher einfacher und etwas effizienter im Vergleich zu dem Einzeldraht-Ansatz sein kann, was jedoch auf Kosten eines zusätzlichen Pins erfolgt. Solche Implementierung kann auf bestehenden Seitenbändern gemultiplext werden, z. B. PERST# für Hostgerät und CLKREQ# für Gerätehost (in diesem Beispiel wird die bestehende Signal-Direktionalität beibehalten, was bimodale Implementierungen vereinfacht). Ein dritter Ansatz kann ein Niedergeschwindigkeits-In-Band-Signalisierungsmechanismus sein, wie z. B. M-PHY LS PWM-Modi, was die Pin-Anzahl relativ zu Seitenbandmechanismen reduziert und noch ähnlich niedrigen Stromverbrauch unterstützen kann. Da dieser Operationsmodus nicht mit Hochgeschwindigkeitsbetrieb kompatibel ist, könnte er mit einem Hochgeschwindigkeits-In-Band-Mechanismus kombiniert werden, wie z. B. geordnete Listen auf physischer Ebene oder DLLPs. Dieser Ansatz hat zwar keinen geringeren Stromverbrauch, maximiert aber die Gemeinsamkeiten mit den bestehenden Hochgeschwindigkeits-IO. In Kombination mit Geringgeschwindigkeits-In-Band-Signalisierung kann dieser Ansatz eine gute stromsparende Lösung sein.

[0091] Um eine oder mehrere dieser Konfigurationen in einem gegebenen System umzusetzen, kann eine semantische Ebene, welche verwendet werden kann, um die Bedeutung der über die physische Ebene auszutauschenden Informationen zu bestimmen, sowie eine Richtlinienebene bereitgestellt werden, die verwendet werden kann, Aktivitäten/Reaktionen auf Gerät-/Plattformebene zu verstehen. In einer Ausführungsform können diese Ebenen in einer SB-PHY-Einheit präsent sein.

[0092] Durch die Bereitstellung eines mehrebenenigen Ansatz können Ausführungsformen Implementierungen auf verschiedenen physischen Ebene erlauben, die beide Seitenband-Funktionen (die in einigen Implementierungen auf Grund der Einfachheit und/oder stromsparendes Betriebs bevorzugt sein kann) und In-Band-Funktionen umfassen, die für andere Implementierungen bevorzugt sein können, z. B. Vermeidung der Notwendigkeit für zusätzliche Pins.

[0093] In einer Ausführungsform kann eine Anzahl von Seitenbandsignalen konfiguriert werden, z. B. über die semantische Ebene in einem einzigen Paket für die Kommunikation über den konsolidierten Seitenbandmechanismus (oder einen In-Band-Mechanismus). In einer Ausführungsform zeigt die nachstehende Tabelle 3 die verschiedenen Signale, die in einer Ausführungsform präsent sein können. In der dargestellten Tabelle wird die logische Richtung der Signale durch den Pfeil angezeigt, wobei ein Pfeil nach oben als die Richtung zu dem Host definiert ist (z. B. einen Root-Complex) und der Pfeil nach unten als die Richtung zu der Vorrichtung definiert ist (z. B. ein Peripheriegerät wie z. B. eine Funklösung).

Tabelle 3

Gerät präsent ↑
 Power Good ↓
 Ausschalten ↓
 Refclock gut ↓
 Fundamentaler Reset ↓
 Konfig. Bereit ↑↓
 Bereit zum Trainieren ↑↓
 Training starten ↑↓
 L1pg anfordern ↑↓
 L1pg ablehnen ↑↓
 L1pg gewähren ↑↓
 OBFF CPU aktiv ↓
 OBFF DMA ↓
 OBFF Idle ↓
 Wakeup ↑
 Ack-Empfang von Handshake ↑↓

[0094] Bezugnehmend auf **Fig. 6** wird ein Ablaufdiagramm für die verschiedenen Status eines Seitenbandmechanismus gemäß einer Ausführungsform der vorliegenden Erfindung gezeigt. Wie in **Fig. 6**, können diese verschiedenen Status sich auf den Root-Complex beziehen (z. B. Host-gesteuerter Betrieb). Zustandsdiagramm **600** kann die Steuerung der verschiedenen Status über den Host bereitstellen. Wie zu sehen ist, beginnt der Betrieb in einem Pre-Boot-Status **610**, in dem ein Präsenzsignal kommuniziert werden kann. Man beachte, dass dieses Präsenzsignal im Hinblick auf die Verbindungsmanagement-Operationen so wie oben beschrieben sein kann. Dann wird die Steuerung einem Boot-Status **620** übergeben, in dem verschiedene Signale übermittelt werden können, nämlich ein Power-Good-Signal, ein Reset-Signal, ein Referenztakt-Statussignal und Bereit-zum-Trainieren-Signal. Man beachte, dass alle diese Signale über ein einziges Paket kommuniziert werden können, wobei jedes dieser Signale einem Indikator oder einem Feld des Pakets entsprechen kann (z. B. ein Ein-Bit-Indikator des Pakets).

[0095] Immer noch mit Bezug auf **Fig. 6** geht die Steuerung dann zu einem aktiven Status **630** über, in dem ein System in einem aktiven Status sein kann (z. B. S0), eine entsprechende Vorrichtung (z. B. die Downstream-Vorrichtung) in einem aktiven Status sein kann (z. B. D0) und die Verbindung in einem aktiven Status, STALL-Status oder stromsparenden Status sein kann (z. B. L0, L0s oder L1). Wie zu sehen ist, können in diesem Zustand verschiedene Signale übermittelt werden, einschließlich eines OBFF Signals, eines Taktanforderungssignals, eines Referenztaktstatus, einen L0-Anforderungssignals und eines Bereit-zum-Trainieren-Signals.

[0096] Dann kann die Steuerung in einen stromsparenden Status **640** übergeben, z. B. nachdem die obige Signalisierung durchgeführt wurde. Wie zu sehen ist, kann in diesem stromsparenden Status **640** das System in einem aktiven Zustand sein, während die Vorrichtung in einem relativ stromsparenden Status niedriger Latenz sein kann (z. B. D3 hot). Darüber hinaus kann die Verbindung in einem bestimmten stromsparenden Status sein (z. B. L2 oder L3). Wie in diesen Status zu sehen ist, können die Signale, die über das konsolidierte Seitenband-Paket übertragen werden, ein Wecksignal, ein Reset-Signal und ein Power-Good-Signal enthalten.

[0097] Wenn das System in stromsparendere Status übergeht, kann es in einen zweiten stromsparenden Status **650** eintreten (z. B. wenn sich das System in einem S0-Status und das Gerät sich in einem D3 kalten Status befindet, und die Verbindung ist ähnlich der in einem L2- oder L3-Status. Wie zu sehen ist, können dieselben Wakeup-, Reset- und Power-Good-Signale übermittelt werden. Wie auch in **Fig. 6** zu sehen ist, können dieselben Signale in einem stromsparenderen Status **660** (z. B. ein stromsparender Systemstatus, S3) und einem stromsparenden Gerätestatus (z. B. D3 kalt) und in denselben stromsparenden Verbindungsstatus L2 und L3 auftreten. Obwohl es in diesem spezifischen Satz von kommunizierten Seitenbandinformationen gezeigt wird, wird dadurch der Umfang der vorliegenden Erfindung in dieser Hinsicht in keiner Weise eingeschränkt.

[0098] Ausführungsformen bieten somit eine erweiterbare Ebenenstruktur, die Einfachheit und niedrige Latenz gegen Flexibilität abwägen kann. Auf diese Weise können bestehende Seitenbandsignale und zusätzliche Seitenbandsignale durch eine kleinere Anzahl von Signalen ersetzt werden und zukünftige Erweiterung der Seitenbandmechanismen ohne Zufügung von Pins ermöglicht werden.

[0099] Bezugnehmend auf **Fig. 7** ist ein Ablaufdiagramm eines Verfahrens gemäß einer Ausführungsform der vorliegenden Erfindung gezeigt. Wie in **Fig. 7** gezeigt wird, kann das Verfahren **700** verwendet werden, um Daten über einen konvergierten Protokollstapel zu übertragen, der obere Ebenen eines Kommunikationsprotokolls und eine untere, z. B. eine physische Ebene eines anderen Kommunikationsprotokolls, beinhaltet. In den gezeigten Beispielen wird ein konvergenter Protokollstapel, wie oben beschrieben, angenommen und zwar mit oberen Transaktions- und Datenverbindungs-Ebenen eines PCIe™-Protokolls und einer physischen Ebene einer unterschiedlichen Spezifikation, z. B. einer MIPI-Spezifikation. Natürlich kann ebenso zusätzliche Logik präsent sein, die die Konvergenz dieser beiden Kommunikationsprotokolle in einem einzigen Protokollstapel ermöglicht, wie z. B. die Logik und Schaltungen die oben mit Bezug auf **Fig. 4** erläutert wurden.

[0100] Wie in **Fig. 7** gesehen werden kann, kann das Verfahren **700** mit dem Empfangen einer ersten Transaktion in einem Protokollstapel des ersten Kommunikationsprotokolls (Block **710**) beginnen. Zum Beispiel versuchen verschiedene Logiken eines Root-Complex, wie z. B. eines Kerns, einer anderen Ausführungsmaschine oder dergleichen, Daten zu einem anderen Gerät zu senden. Dementsprechend können diese Informationen durch die Transaktionsebene übermittelt werden. Wie zu sehen ist, geht die Steuerung zu Block **720** über, wo die Transaktion verarbeitet und einem logischen Teil einer PHY eines zweiten Kommunikationsprotokolls zur Verfügung gestellt werden kann. Diese Verarbeitung kann die verschiedenen oben beschriebenen Vorgänge in Bezug auf den Fluss durch **Fig. 4** umfassen, wo verschiedene Operationen zum Datenempfang und zur Flusskontrolle, sowie Verbindungsoperationen, Paketierungs-Operationen usw. stattfinden können. Zusätzlich können verschiedene Operationen stattfinden, um ein Datenverbindungsebenen-Paket an eine PHY zu liefern. Als nächstes geht die Steuerung zum Block **730** über, wo diese erste Transaktion in einem logischen Teil der PHY in eine Transaktion eines zweiten Formats konvertiert werden kann. Beispielsweise kann (falls nötig) die Konvertierung der Symbole durchgeführt werden. Darüber hinaus können die verschiedenen Übersetzungsoperationen stattfinden, die durchgeführt werden, um somit die Transaktion in ein Format zur Übertragung über die Verbindung umzuwandeln. Dementsprechend kann Kontrolle zu Block **740** übergehen, wo diese zweite formatierte Transaktion von der PHY über eine Verbindung zu dem Gerät übertragen werden kann. Als Beispiel kann die Transaktion des zweiten Formats die serialisierten Daten nach Zeilen-Codierung, Serialisierung usw. sein. Obwohl dies auf dieser hohen Ebene in der Ausführungsform von **Fig. 7** gezeigt ist, versteht sich, dass dadurch der Umfang der vorliegenden Erfindung in dieser Hinsicht in keiner Weise eingeschränkt wird.

[0101] Bezugnehmend auf **Fig. 8**, ist ein Blockdiagramm der Komponenten gezeigt, die in einem zweiten System gemäß einer Ausführungsform der vorliegenden Erfindung präsent sind. Wie in **Fig. 8** gezeigt, kann System **800** viele verschiedene Komponenten enthalten. Diese Komponenten können als IC-Bausteine, Teile davon, diskrete elektronische Geräte oder andere an einer Leiterplatte angepassten Baugruppen, wie einer Hauptplatine oder Erweiterungskarte des Computersystems, oder als Komponenten, die ansonsten innerhalb eines Chassis des Computersystems integriert sind, implementiert werden. Man beachte auch, dass das Blockschaltbild der **Fig. 8** eine Ansicht hoher Ebene der vielen Komponenten des Computersystems zeigen soll. Es ist jedoch zu verstehen, dass zusätzliche Komponenten in bestimmten Ausführungen vorhanden sein können und darüber hinaus unterschiedliche Anordnung der gezeigten Komponenten bei anderen Implementierungen auftreten kann.

[0102] Wie in **Fig. 8** gesehen, kann ein Prozessor **810**, der eine Low-Power-Multicore-Prozessorsockel, wie z. B. ein Ultra-Low-Voltage-Prozessor, sein kann, als eine Hauptverarbeitungseinheit und zentrale Hub für die Kommunikation mit den verschiedenen Komponenten des Systems agieren. Solcher Prozessor kann als SoC implementiert werden. In einer Ausführungsform kann der Prozessor **810** ein auf Intel® Architecture Core™ basierter Prozessor sein, wie z. B. ein i3, i5, i7 oder andere solche Prozessoren von Intel Corporation, Santa Clara, CA. Man verstehe jedoch, dass andere Low-Power-Prozessoren, wie sie von Advanced Micro Devices, Inc. (AMD), Sunnyvale, CA erhältlich sind, ein ARM-basiertes Design von ARM Holdings, Ltd. oder ein MIPS-basiertes Design von MIPS Technologies, Inc., Sunnyvale, CA, oder ihren Lizenznehmern oder Anwendern in anderen Ausführungsformen präsent sein können, wie z. B. ein Apple A5-Prozessor.

[0103] Prozessor **810** kann mit einem Systemspeicher **815** kommunizieren, der in einer Ausführungsform über mehrere Speichervorrichtungen implementiert werden kann, um eine gegebene Menge an Systemspeicher bereitzustellen. Zum Beispiel kann der Speicher ein Speicher in Übereinstimmung mit einem auf Joint Electron Devices Engineering Council (JEDEC) Low-Power Double Data Rate(LPDDR)-basierten Design wie der aktu-

elle LPDDR2-Standard nach JEDEC JESD 209-2E (veröffentlicht April 2009) sein oder eine nächste Generation des LPDDR-Standards, die als LPDDR3 bezeichnet wird und die Erweiterungen des LPDDR2 bieten wird, um die Bandbreite zu erhöhen. Zum Beispiel können 2/4/8 Gigabyte (GB) Systemspeicher vorhanden sein und mit dem Prozessor **810** über eine oder mehrere Speicherverbindungen verbunden sein. Bei verschiedenen Implementierungen können die einzelnen Speichergeräte verschiedene Pakettypen haben, wie z. B. Single-Die-Paket (SDP), Dual-Die-Paket (DDP) oder Quad-Die-Paket (QDP). Diese Geräte können in einigen Ausführungsformen direkt auf einer Hauptplatine aufgelötet werden, um eine Lösung niedrigeren Profils bereitzustellen, während in anderen Ausführungsformen die Vorrichtungen als ein oder mehrere Speichermodule konfiguriert sein können, die wiederum über einen bestimmten Anschluss mit der Hauptplatine gekoppelt sein können.

[0104] Um dauerhafte Speicherung von Informationen, wie z. B. Daten, Anwendungen, ein oder mehrere Betriebssysteme und so weiter, zu unterstützen, kann ein Massenspeicher **820** ebenso mit dem Prozessor **810** gekoppelt sein. In verschiedenen Ausführungsformen kann dieser Massenspeicher über eine SSD implementiert werden, um ein dünneres und leichteres Systemdesign zu ermöglichen sowie die Reaktionsfähigkeit des Systems zu verbessern. In anderen Ausführungsformen kann der Massenspeicher jedoch in erster Linie als ein Festplattenlaufwerk (HDD) mit einer geringeren Menge von SSD-Speicher als SSD-Cache implementiert sein, um die nichtflüchtige Speicherung von Kontextstatus und anderen solchen Informationen beim Abschaltvorgang zu ermöglichen, so dass ein schneller Einschaltvorgang zur Wiederaufnahme der Systemaktivitäten realisiert werden kann. Wie auch in **Fig. 8** gezeigt, kann eine Flash-Vorrichtung **822** an den Prozessor **810** gekoppelt sein, z. B. über eine serielle Peripherieschnittstelle (serial peripheral interface, SPI). Diese Flash-Vorrichtung kann nichtflüchtige Speicherung von Systemsoftware bieten, einschließlich einer Basis-Eingabe/Ausgabe-Software (basic input/output software, BIOS) sowie andere Firmware des Systems.

[0105] Verschiedene Eingabe-/Ausgabe-(IO-)Vorrichtungen können in dem System **800** präsent sein. In der Ausführungsform der **Fig. 8** insbesondere gezeigt ist eine Anzeige **824**, die eine High-Definition-LCD oder ein LED-Panel sein kann, die in einem Deckelteil des Gehäuses konfiguriert sein kann. Dieses Display-Panel kann auch einen Touchscreen **825** bieten, der z. B. extern über das Display-Panel ausgebildet ist, so dass durch die Interaktion eines Benutzers mit diesem Touchscreen Benutzereingaben dem System gegeben werden können, um die gewünschten Operationen zu ermöglichen, z. B. im Zusammenhang mit der Anzeige von Informationen, dem Zugriff auf Informationen und so weiter. In einer Ausführungsform kann der Display **824** mit dem Prozessor **810** über eine Display-Verbindung gekoppelt sein, die als Hochleistungsgrafikverbindung umgesetzt sein kann. Der Touchscreen **825** kann mit dem Prozessor **810** über eine andere Verbindung gekoppelt sein, die in einer Ausführungsform eine I²C-Verbindung sein kann. Wie weiter in **Fig. 8** gezeigt, kann zusätzlich zum Touchscreen **825** die Benutzereingabe auch mittels der Berührung eines Touch-Pads **830** erfolgen, der innerhalb des Gehäuses konfiguriert sein kann und auch mit derselben I²C-Verbindung wie der Touchscreen **825** gekoppelt sein kann.

[0106] Für Perceptual Computing und andere Zwecke können verschiedene Sensoren in dem System vorhanden sein und mit dem Prozessor **810** auf verschiedene Weisen gekoppelt sein. Bestimmte Beschleunigungs- und Umweltsensoren können mit dem Prozessor **810** über einen Sensor-Hub **840**, z. B. über eine I²C-Verbindung gekoppelt sein. In der in **Fig. 8** gezeigten Ausführungsform können diese Sensoren einen Beschleunigungsmesser **841**, einen Umgebungslichtsensor (ambient light sensor, ALS) **842**, einen Kompass **843** und ein Gyroskop **844** umfassen. Andere Umweltsensoren können ein oder mehrere Wärmesensoren **846** umfassen, die in einer Ausführungsform über einen System-Management-Bus (SMBus) mit dem Prozessor **810** gekoppelt sein können. Man verstehe auch, dass ein oder mehrere der Sensoren mit dem Prozessor **810** über eine LPS-Verbindung gemäß einer Ausführungsform der vorliegenden Erfindung gekoppelt sein können.

[0107] Wie auch in **Fig. 8** gesehen, können verschiedenen Peripheriegeräten mit Prozessor **810** über eine Low-Pin-Count-(LPC-)Verbindung gekoppelt sein. In der gezeigten Ausführungsform können verschiedene Komponenten über einen eingebetteten Controller **835** gekoppelt sein. Solche Komponenten können eine Tastatur **836** (z. B. über eine PS2-Schnittstelle gekoppelt), ein Lüfter **837** und ein Thermosensor **839** umfassen. In einigen Ausführungsformen kann das Touchpad **830** auch mit EG-835 über eine PS2-Schnittstelle gekoppelt sein. Zusätzlich kann ein Sicherheitsprozessor wie ein Trusted-Platform-Modul (TPM) **838**, wie z. B. in Übereinstimmung mit der Trusted Computing Group (TCG) TPM-Spezifikation Version 1.2 (2. Oktober 2003), auch mit dem Prozessor **810** über diese LPC-Verbindung gekoppelt sein.

[0108] System **800** kann mit externen Geräten in einer Vielzahl von Arten, einschließlich drahtlos, kommunizieren. In der in **Fig. 8** gezeigten Ausführungsform sind verschiedene Funkmodule präsent, von denen jedes einem Funkgerät entsprechen kann, das für ein bestimmtes drahtloses Kommunikationsprotokoll konfiguriert ist. Eine Weise zur drahtlosen Kommunikation in einem Nahbereich, wie z. B. ein Nahfeld, kann in einer Aus-

führungsform über eine Nahfeld-Kommunikations-(NFC-)Einheit **845** erfolgen, die mit dem Prozessor **810** über einen SMBus kommunizieren kann. Man beachte, dass über diese NFC-Einheit **845** Vorrichtungen in unmittelbarer Nähe miteinander kommunizieren können. Beispielsweise kann ein Benutzer über gemeinsame Anpassung der beiden Geräte in enger Beziehung System **800** erlauben, mit einem anderen (z. B.) tragbaren Gerät, z. B. einem Smartphone des Benutzers, zu kommunizieren, um Informationen wie Identifikationsinformationen, Zahlungsinformationen, Daten, wie z. B. Bilder, oder dergleichen auszutauschen. Drahtlose Energieübertragung kann auch unter Verwendung eines NFC-System erzielt werden.

[0109] Wie weiter in **Fig. 8** gezeigt, können zusätzliche drahtlose Geräte andere drahtlose Nahbereichsgeräte, einschließlich einer WLAN-Einheit **850** und einer Bluetooth-Einheit **852**, umfassen. Unter Verwendung einer WLAN-Einheit **850** kann Wi-Fi™-Kommunikation in Übereinstimmung mit dem gegebenen Institute of Electrical and Electronics Engineers (IEEE) 802.11-Standard erzielt werden, während über Bluetooth-Einheit **852**, Nahbereichskommunikation über ein Bluetooth-Protokoll durchgeführt werden kann. Diese Einheiten können mit dem Prozessor **810** z. B. über eine USB-Verbindung oder eine Universal-Asynchronous-Receiver-Transmitter-(UART)Verbindung kommunizieren. Alternativ können diese Einheiten mit dem Prozessor **810** über eine Verbindung über eine Niederstromverbindung, wie z. B. einer konvergenten PCIe/MIPI-Verbindung, wie hier beschrieben, oder über ein anderes solches Protokoll, wie z. B. einen seriellen Dateneingabe/-ausgabe-(SDIO-)Standard gekoppelt sein. Natürlich kann die tatsächliche physische Verbindung zwischen diesen Peripheriegeräten, die auf einer oder mehreren Zusatzkarten konfiguriert sein können, über die NGFF-Anschlüsse mit einer Hauptplatine erzielt werden.

[0110] Darüber hinaus kann Wireless-Wide-Area-Kommunikation, z. B. gemäß eines mobilen oder anderen drahtlosen WAN-Protokoll, über eine WWAN-Einheit **856** stattfinden, die wiederum mit einem Teilnehmeridentitätsmodul (subscriber identity module, SIM) **857** gekoppelt sein kann. Darüber hinaus kann ebenfalls ein GPS-Modul **855** vorhanden sein, um den Empfang und die Verwendung von Standortinformationen zu ermöglichen. Man beachte, dass in der in **Fig. 8** gezeigten Ausführungsform die WWAN-Einheit **856** und ein integriertes Erfassungsgerät, wie z. B. ein Kameramodul **854**, über ein gegebenes USB-Protokoll, wie z. B. eine USB 2.0 oder 3.0-Verbindung, oder über ein UART- oder I²C-Protokoll kommunizieren können. Auch die tatsächliche physische Verbindung dieser Einheiten kann über Anpassung eines NGFF-Zusatzkarte an einen NGFF-Anschluss erfolgen, der auf der Hauptplatine konfiguriert ist.

[0111] Um Audio-Ein- und Ausgänge zur Verfügung zu stellen, kann ein Audioprozessor über einen digitalen Signalprozessor (DSP) **860** implementiert sein, der mit dem Prozessor **810** über eine High-Definition-Audio-(HDA-)Verbindung gekoppelt sein kann. Ähnlich kann der DSP **860** mit einem integrierten Codierer/Decodierer (CODEC) und einem Verstärker **862** kommunizieren, die wiederum mit Ausgangslautsprechern **863** gekoppelt sind, die innerhalb des Gehäuses implementiert sein können. Ähnlich können Verstärker und CODEC **862** gekoppelt sein, um Audioeingaben von einem Mikrofon **865** zu erhalten, welches in einer Ausführungsform über Dual-Array-Mikrofone implementiert sein kann, um qualitativ hochwertige Audioeingänge bereitzustellen, um sprachaktivierte Steuerung verschiedener Operationen innerhalb des Systems zu ermöglichen. Man beachte auch, dass die Audioausgänge vom Verstärker/CODEC **862** einer Kopfhörerbuchse **864** bereitgestellt werden können.

[0112] Ausführungsformen können somit in vielen unterschiedlichen Umgebungen verwendet werden. Unter Bezugnahme auf **Fig. 9** ist ein Blockdiagramm eines Beispielsystems **900** gezeigt, mit dem Ausführungsformen verwendet werden können. Wie zu sehen ist, kann das System **900** ein Smartphone oder ein anderes drahtloses Kommunikationsgerät sein. Wie in dem Blockdiagramm von **Fig. 9** gezeigt, kann System **900** einen Basisband-Prozessor **910** umfassen, welcher ein Mehrkern-Prozessor sein kann, der sowohl die Basisband-Verarbeitungsaufgaben sowie die Anwendungsverarbeitung bewältigen kann. Somit kann der Basisbandprozessor **910** verschiedene Signalverarbeitungen hinsichtlich der Kommunikation sowie die Rechenoperationen für die Vorrichtung durchführen. Im Gegenzug kann Basisbandprozessor **910** kann mit einer Benutzeroberfläche/Anzeige **920** gekoppelt sein, die in einigen Ausführungsformen durch eine Touchscreen-Anzeige implementiert sein kann. Zusätzlich kann der Basisbandprozessor **910** mit einem Speichersystem, einschließlich in der Ausführungsform der **Fig. 9** eines nichtflüchtigen Speichers, insbesondere eines Flash-Speichers **930** und eines Systemspeichers, und zwar eines dynamischen Direktzugriffsspeicher (dynamic random access memory, DRAM) **935**, gekoppelt sein. Wie weiter zu sehen ist, kann der Basisbandprozessor **910** weiterhin mit einem Erfassungsgerät **940**, wie beispielsweise einem Bilderfassungsgerät, das Video und/oder Standbilder aufnehmen kann, gekoppelt sein.

[0113] Um die Sendung und den Empfang der Kommunikation zu ermöglichen, können verschiedene Schaltungen zwischen dem Basisbandprozessor **910** und einer Antenne **980** gekoppelt sein. Insbesondere können

ein Radiofrequenz-(RF-)Sendeempfänger **970** und ein Wireless-Local-Area-Network(WLAN-)Sendeempfänger **975** präsent sein. Im allgemeinen kann der RF-Sendeempfänger **970** verwendet werden, um drahtlose Daten und Anrufe nach einem vorgegebenen Protokoll der drahtlosen Kommunikation zu übertragen, wie z. B. dem 3G- oder 4G-Kommunikationsprotokoll, wie z. B. in Übereinstimmung mit einem Codemultiplex-Vielfachzugriff (CDMA), Global System for Mobile Communication (GSM), Long Term Evolution (LTE) oder einem anderen Protokoll. Andere drahtlose Kommunikationen wie der Empfang oder die Übertragung von Funksignalen, z. B. Mittelwellen-/UKW- oder Global-Positioning-Satellite-(GPS-)Signale können ebenfalls bereitgestellt werden. Darüber hinaus können über WLAN-Sendeempfänger **975** ebenso lokale, drahtlose Signale, wie z. B. gemäß eines Bluetooth™-Standards oder eines IEEE 802.11-Standards wie IEEE 802.11a/b/g/n realisiert werden. Man beachte, dass die Verbindung zwischen dem Basisbandprozessor **910** und einem oder beiden Sendeeempfängern **970** und **975** über eine konvergierte Niederstromverbindung erfolgen kann, die die Funktionalität einer PCIe™-Verbindung und einer Niederstromverbindung, wie z. B. eine MIPI-Verbindung, kombiniert und zuordnet. Obwohl dies auf dieser hohen Ebene in der Ausführungsform von **Fig. 9** gezeigt ist, versteht sich, dass dadurch der Umfang der vorliegenden Erfindung in dieser Hinsicht in keiner Weise eingeschränkt wird.

[0114] Im Gegensatz zu üblicherweise verwendeten Verbindungen wie PCIe™, wo die Betriebsart der Verbindung wie Breite und Geschwindigkeit beim Verbindungstraining ausgehandelt werden, führen die Ausführungsformen keine solche Verhandlung durch. Da das PCIe™-Protokoll verhandlungsbasiert ist, bestimmt die Verbindung den höchsten unterstützten gemeinsamen Betriebsmodus und betreibt die Verbindung dann in diesem Modus. Dies wird komplizierter, wenn eine Verbindung asymmetrische Spuren und Geschwindigkeiten unterstützt.

[0115] Ausführungsformen bieten stattdessen ein Architekturprotokoll zum dynamischeren Ändern des Verbindungsbetriebsmodus, was während der Laufzeit erfolgen kann und ohne dass die Verhandlung heruntergefahren werden muss. Gemäß dieses Architekturprotokolls werden während der Initialisierung der Verbindung die von Agenten unterstützten Betriebsmodi an gegenüberliegenden Enden der Verbindung bestimmt; Software oder Hardware kann dann die höchste gemeinsame Betriebsart bestimmen; wenn ein Verbindungsbetriebsmodus verändert werden soll, tauschen die Agenten auf beiden Seiten der Verbindung Meldungen aus, um die Betriebsmodusänderung anzufordern und zu bestätigen. Man beachte, dass das Neustraining der Verbindung vereinfacht ist, da die exakten Parameter übermittelt werden, so dass beide Enden der Verbindung genau wissen, welche neuen Parameter verwendet werden sollen, um die Verbindung zu betreiben.

[0116] Wie oben erläutert können in einigen Ausführungsformen die physischen Ebenen auf beiden Seiten einer Verbindung M-PHYs sein. In solchen Ausführungsformen kann die Verbindung serielle Leitungen umfassen, die differentielle Pulsweitenmodulations(PWM-)Signale übertragen. Diese differentiellen PWM-Signale, die über die Verbindung kommuniziert werden, können sich in einem oder mehreren Datenratenbereichen der M-PHY-Spezifikation („Gears“ genannt) befinden. Zum Beispiel kann die Verbindung unter GEAR 1 (3 Megabit pro Sekunde (Mbps) bis 9 Mbps, GEAR 2 (6 Mbps bis 18 Mbps, GEAR 3 (12 Mbps bis 36 Mbps), Gear 4 (24 Mbps bis 72 Mbps), GEAR 5 (48 Mbps bis 144 Mbps), GEAR 6 (96 Mbps bis 288 Mbps), GEAR 7 (192 Mbps bis 576 Mbps) usw. arbeiten. Unter Verwendung von Ausführungsformen der vorliegenden Erfindung kann dynamische Rekonfiguration der Verbindung zum Betrieb in einem ausgewählten Bereich dieser Bereiche (oder in anderen Bereichen) stattfinden.

[0117] Ausführungsformen unterstützen eine dynamische Verbindungs-Bandbreiten-Änderung, so dass beide Enden der Verbindung übereinstimmen, die Verbindungsparameter zu ändern, um eine neue Bandbreite zu unterstützen. Es gibt zwei Konfigurations-Szenarien: Anfängliche Konfiguration, gefolgt von einem Einschalt-Reset oder von einem Link-Down-Status; und eine Laufzeit-Rekonfiguration der Verbindung ohne Durchgang durch Link-Down. Die anfängliche Konfiguration nutzt den LS-Betriebsmodus, wobei nur eine einzige Spur (z. B. lane0) verwendet wird. In einigen Ausführungsformen ist die Verbindungs-Rekonfiguration optional und kann während des normalen Betriebs der Verbindung initiiert werden.

[0118] Für die Rekonfiguration können beide Enden der Verbindung für einen Rekonfigurationsauslöser (re-configuration trigger, RCT) eingerichtet sein, und dies gilt nur für solche Register, die RCT fähig sind. Handshake-Mechanismen für Protokolle höherer Ebene können verwendet werden, um offline Register zu lesen und zu schreiben und RCT einzurichten. RCT wird in einer Ausführungsform ausgelöst, wenn alle aktiven Spuren in den SAVE-Status eintreten. Man beachte, dass die PHY offline Register direkt von Software oder implizit von Hardware geschrieben werden können (Handshake auf Protokollebene ist abgeschlossen). Wenn sich die Konfiguration ändert, wenn zum Beispiel Breiten- oder Geschwindigkeitsänderungen auftreten, fordert der Agent, der die Änderung initiiert, die Konfigurationsänderung an.

[0119] Das Beispiel von **Fig. 10** zeigt Port A, der einen Antrag auf Rekonfiguration initiiert, und Port B, der die Anforderung bestätigt, worauf auf allen Spuren in die Wiederherstellung und in den STALL-Status eingetreten wird. RCT wird ausgelöst und die neuen Konfigurationseinstellungen werden wirksam. In bestimmten Ausführungsformen kann es zwei unterschiedliche Methoden zur Durchführung der Rekonfiguration geben: unter Verwendung von Datenverbindungsebene-Paketen (DLLPs) und mit Trainingssequenzen.

[0120] Wie in **Fig. 10** gesehen wird, ist Port A, der aus einer PHY-Einheit von einer ersten Vorrichtung bestehen kann, z. B. einer Upstream-Vorrichtung, in Kommunikation über eine physische Verbindung mit einem Port B einer zweiten Vorrichtung, der eine Downstream-Vorrichtung sein kann. Wie zu sehen ist, kann in einem aktiven Verbindungsstatus (z. B. L0) Port A eine Vielzahl von DLLPs, und zwar Bandbreiten-Änderungsanforderungs-DLLPs, zu Port B übertragen.

[0121] Als Reaktion auf solche Pakete kann Port B über seine PHY-Einheit eine Vielzahl von Bestätigungs-DLLPs vermitteln, und zwar eine Vielzahl von Bandbreiten-Änderungsbestätigungs-DLLPs zurück zu Port A. Dann kann Port A wiederum einen STALL-Status der Verbindung verursachen, indem her eine STALL-Status-Übertragung über eine EEA, gefolgt von einem DIF-N-Signal, initiiert. Im Gegenzug kann der Port B ähnliche Signale (nämlich ein EEA), gefolgt von einem DIF-N-Signal, kommunizieren. Als Reaktion auf solche Kommunikationen können beide Geräte ein hohes Impedanz-Signal senden, wodurch die Verbindung in einen HIBERN8-Status eintritt. Als solches kann ein RCT auftreten, um die Verbindungs-Rekonfiguration durchzuführen, worauf die Verbindung über Kommunikation von DIF-N-Signale mit den neuen Einstellungen austreten kann. Obwohl dies auf dieser hohen Ebene in der Ausführungsform von **Fig. 10** gezeigt ist, versteht sich, dass dadurch der Umfang der vorliegenden Erfindung in dieser Hinsicht in keiner Weise eingeschränkt wird.

[0122] Für DLLP-basierte Rekonfiguration liest die Software während der Initialisierung die Funktionsregister an beiden Enden der Verbindung und aktualisiert das Register für jeden Port mit den Kapazitäten der Verbindungspartner. Für einen Anforderer, der eine Rekonfiguration der Verbindungsparameter initialisiert, wird der Anforderer kontinuierlich eine BwChange_Request-DLLP-Meldung senden. Man beachte, dass der Anforderer keine neuen Konfigurationsanforderungen machen darf, bis die erste Verbindungskonfiguration-Anforderung abgeschlossen ist. Der Anforderer wartet dann auf die BwChange_Acknowledge-DLLP-Meldung. Bei Empfang der Bestätigungsmeldung sendet der Anforderer EIOS und richtet ein RCT-Ereignis ein. Dieses Ereignis bewirkt, dass die Register mit den entsprechenden M-PHY-Parametern geschrieben werden, und das RCT führt dazu, dass die Parameter auf die Verbindung angewendet werden. Wenn das RCT-Bit gesetzt ist, leitet der Anforderer alle Spuren auf einen HIBERN8-Status über und wartet, bis alle Empfänger-Spuren HIBERN8 eingehen. Sobald sowohl Sender- als auch Empfänger-Spuren in HIBERN8 sind, führt der Anforderer die entsprechenden Konfigurationsänderungen durch, und RCT ist erreicht.

[0123] Wenn ein remote Ende die Rekonfiguration der Verbindungsparameter anfordert, sendet der Empfänger kontinuierlich eine BwChange_Acknowledge-DLLP-Meldung bis EIOS empfangen wird oder empfangene Spuren den Übergang zu HIBERN8 beginnen. Dann wird das RCT-Bit gesetzt. Wenn das RCT-Bit gesetzt ist, überträgt der Empfänger EIOS und leitet dann alle Spuren auf einen HIBERN8-Status über. Es ist zulässig, dass der Empfänger sofort in HIBERN8 eintritt; die Empfangsverbindung kann die Rekonfiguration ihrer offline Register beginnen. Nachdem sowohl der Sender als auch der Empfänger im HIBERN8-Status sind, führt die empfangende Verbindung RCT aus. Danach findet der Austritt aus dem HIBERN8-Status mit den aktualisierten Parameter statt.

[0124] Ausführungsformen unterstützen die Fähigkeit, Verbindungsbandbreite ohne einen Durchgang durch Link-Down zu ändern. Die dynamische Verbindungs-Rekonfiguration unterstützt die Änderung der Verbindungsbandbreite durch Änderung der Ratenserie oder High-Speed-Gears identisch auf Modulen auf beiden Seiten der Verbindung und die Änderung der Anzahl der konfigurierten Spuren pro Unterverbindung. Der Verbindungsbandbreitenänderung kann asymmetrisch sein. Das heißt, ein Modul kann mit einer unterschiedlichen Anzahl von TX-Spuren und RX-Spuren konfiguriert sein. Der Software wird jede Änderung in Ratenserie oder Verbindungsgeschwindigkeit und Verbindungsbandbreite durch den Verbindungsbandbreiten-Benachrichtigungsmechanismus nach PCIe™ gemeldet.

[0125] In einer Ausführungsform können die Verbindungsbandbreiten-Parameter, die während der Laufzeit rekonfiguriert werden können, Folgendes umfassen: Ratenserie (A oder B); High-Speed-Gears; und Unterverbindungsbandbreite. Die Verbindungsratenserie und das Verbindungsgeschwindigkeits-Management kann zwischen dem LTSSM und den Software-Nachverfolgungs- und Kontrollmechanismen nach PCIe™ koordiniert werden.

[0126] In einer Ausführungsform stellt das Ziel-Verbindungsgeschwindigkeits-Feld in dem Link-Control 2-Register die obere Grenze der Verbindungsgeschwindigkeit ein. Außer wie unten beschrieben, versucht die Upstream-Komponente, die Verbindung auf der niedrigeren Geschwindigkeit der Zielverbindungsgeschwindigkeit oder der höchsten Geschwindigkeit, die von beiden Komponenten der Verbindung unterstützt wird (wie durch die Werte in den Trainingsätzen gemeldet wurde), beizubehalten.

[0127] Wenn eine dynamische Ratenserien-Rekonfiguration aktiviert ist, kann die Komponente die Verbindungsratenserie autonom einstellen und die dynamische Verbindungs-Rekonfigurations-Funktionalität nutzen, um eine Ratenserien-Änderung anzufordern.

[0128] Wenn das Disable-Bit für Hardware-autonome Geschwindigkeit im Verbindungssteuerung-2-Register nicht gesetzt ist, kann die Komponente die Verbindungsratenserie autonom einstellen und High-Speed-Gears mit implementierungsspezifischen Kriterien verbinden. Jede der Komponenten kann die Verbindungsgeschwindigkeit senken oder erhöhen, indem die entsprechenden Verbindungs-High-Speed-Gears aus der Liste der unterstützten High-Speed-Gears, die in den Trainingsätzen von der Komponente gemeldet wurden, entfernt oder dieser hinzugefügt werden. Die Kriterien für das Entfernen oder Hinzufügen von High-Speed-Gears sind implementierungsspezifisch und sind unabhängig von der Einstellung des Disable-Bit für Hardware-autonome Geschwindigkeit.

[0129] Die Software kann die maximale Geschwindigkeit des Verbindungsbetriebs einschränken und die bevorzugte Verbindungsgeschwindigkeit einstellen, indem sie den Wert in das Ziel-Verbindungsgeschwindigkeit-Feld der Upstream-Komponente einstellt. Nach dem Ändern des Werts im Ziel-Verbindungsgeschwindigkeit-Feld löst die Software Verbindungsneutraining durch das Schreiben eines vorbestimmten Wertes im Verbindungsneutraining-Bit aus.

[0130] Das Verbindungsbreiten-Management kann zwischen dem LTSSM und den Software-Nachverfolgungs- und Kontrollmechanismen nach PCIe™ koordiniert werden. Die Upstream-Komponente versucht, jede Unterverbindung mit der höchsten Anzahl von der Spuren zu konfigurieren, die von den M-Ports auf gegenüberliegenden Seiten der Verbindung unterstützt wird.

[0131] Wenn das Disable-Bit für Hardware-autonome Breite im Steuerregister beider Komponenten nicht gesetzt ist, kann die Komponente Breitenänderung der Unterverbindung unter Verwendung von implementierungsspezifischen Kriterien autonom anfordern. Jede Komponente kann die Unterverbindungsbreite senken oder erhöhen, indem die entsprechenden TX-Spurbreiten- und RX-Spurbreiten-Einstellung aus der Liste der unterstützten Unterverbindungsbreiten-Einstellungen, die die Komponente in den Trainingsätzen meldet, entfernt oder dieser hinzufügt. Die Kriterien für das Entfernen oder Hinzufügen von Verbindungsbreiten-Einstellungen sind implementierungsspezifisch und sind unabhängig von der Einstellung des Disable-Bit für Hardware-autonome Breite.

[0132] Unter Bezugnahme auf **Fig. 11** wird ein Zeitdiagramm in Übereinstimmung mit einer Ausführungsform der vorliegenden Erfindung gezeigt, das die Verbindungskonfiguration illustriert, die von einem Downstream-Port initiiert wird. Wie in **Fig. 11** gezeigt, kann die Verbindungs-Rekonfiguration dadurch beginnen, dass der Downstream-Port eine Änderung anfordert, indem er eine geordnete Trainingsliste (TS1) mit einer Bandbreitenänderung kommuniziert. Diese Meldung kann eine vorbestimmte Anzahl von Malen gesendet werden. Als Reaktion kann die Upstream-Komponenten diese empfangen und dieselbe Meldung eine vorgegebene Anzahl von Malen kommunizieren. Nach Beschluss durch den Downstream-Port kann sie eine geordnete Trainingsliste TS2 mit den beschlossenen Werten senden, während der Upstream-Port Meldungen mit geordneten Trainingslisten TS2 mit der Bandbreitenänderung und der Identifizierung der unterstützten Werte kommunizieren kann. Nach Eintritt in einen STALL-Status, z. B. über Senden von EEA- und DIF-N-Signalen, kann die Verbindung den STALL-Status mit den neuen Konfigurationseinstellungen verlassen.

[0133] Für Trainingssequenz-basierte Rekonfiguration initiiert eine Komponente dynamische Verbindungs-Rekonfiguration durch Senden von geordneten Trainingslisten TS1, wobei das Bandbreitenänderungs-Bit gesetzt ist. Die Komponente, die die Bandbreitenänderungs-Anforderung initiiert, setzt die Felder in der geordneten Trainingsliste für die angeforderte Ratenserie, High-Speed-Gears, TX-Spurbreite und RX-Spurbreite. Komponenten können die angeforderte Bandbreiten-Einstellungen in den geordneten Trainingslisten nicht ändern, nachdem der dynamische Verbindungs-Rekonfigurations-Prozess gestartet ist.

[0134] Für Verbindungs-Bandbreiten-Änderungen, die von der Upstream-Komponente initiiert wurden, antwortet die Downstream-Komponente nach erfolgreichem Empfang von TS1 mit der geordneten Trainingsliste

TS1, wobei das Bandbreiten-Änderungs-Anforderungs-Bit gesetzt ist. Die Downstream-Komponente setzt die TS1-Felder auf Basis der Unterstützung für Ratenserie, High-Speed-Gears und Unterverbindungsbreite.

[0135] Die Upstream-Komponente ist verantwortlich für die Ermittlung der neuen Ratenserie, High-Speed-Gears und Unterverbindungsbreite entsprechend der Verbindungs-Rekonfigurations-Richtlinien wie unten beschrieben. Diese neuen Einstellungen werden von der Upstream-Komponente durch Übertragung von TS2 mit gesetztem Bandbreiten-Änderungs-Bit und den neuen Bandbreiteneinstellungen gesendet. Die Downstream-Komponente quittiert den Empfang von neuen Bandbreiteneinstellungen durch Übertragung der Trainingsliste TS2, wobei das Verbindungs-Bandbreiten-Bestätigungs-Bit gesetzt ist.

[0136] Wenn eine Komponente versucht, eine bestimmte Verbindungsbandbreite auszuhandeln und dieses abgewiesen wird, darf die Komponente für ein vorbestimmtes Zeitintervall nach der Rückkehr zu L0 nach dem vorherigen Versuch, oder bis die andere Komponente der Verbindung eine Verbindungsbandbreiten-Anfrage initiiert, keine dynamische Verbindungs-Rekonfiguration anfordern.

[0137] Im Rahmen der dynamischen Verbindungs-Rekonfiguration trifft die Upstream-Komponente auf der Grundlage der folgenden Regeln eine Entscheidung über den neuen Ziel-High-Speed-Gear, die neue Ratenserie und die Breite. Die Upstream-Komponente verwendet TS2-geordnete Liste, um die für den Betrieb gewählten Werte mitzuteilen. Der Upstream-Komponente trifft die Entscheidung, die neue Verbindungs-Bandbreiten-Werte zu wählen, indem sie die Werte in TS1 mit dem gesetzten Verbindungsbandbreiten-Änderungs-Bit, die von der Downstream-Komponente empfangen wurde, mit den von der Upstream-Komponenten unterstützten Werte vergleicht, wobei dies unabhängig von den Werten geschieht, die die Upstream-Komponente in TS1 verwendet haben kann, um eine Verbindungs-Bandbreiten-Änderung anzufordern.

[0138] Wenn der neue angeforderte High-Speed-Gear sich von dem aktuellen High-Speed-Gear unterscheidet und die Komponenten auf beiden Seiten der Verbindung zumindest einen gemeinsamen unterstützten High-Speed-Gear haben, wird der neue High-Speed-Gear der höchste beidseitig unterstützte High-Speed-Gear sein.

[0139] Wenn der neue angeforderte High-Speed-Gear sich von dem aktuellen High-Speed-Gear unterscheidet und die Komponenten auf beiden Seiten der Verbindung nicht zumindest einen gemeinsamen unterstützten High-Speed-Gear haben, wird der neue High-Speed-Gear der aktuelle High-Speed-Gear bleiben.

[0140] Wenn die neue angeforderte Unterverbindungsbreite sich von der aktuellen Unterverbindungsbreite unterscheidet und die Komponenten auf beiden Seiten der Verbindung zumindest eine gemeinsame unterstützte Unterverbindungsbreite haben, wird die neue Unterverbindungsbreite die höchste beidseitig unterstützte Unterverbindungsbreite sein.

[0141] Wenn die neue angeforderte Unterverbindungsbreite sich von der aktuellen Unterverbindungsbreite unterscheidet und die Komponenten auf beiden Seiten der Verbindung nicht zumindest eine gemeinsame unterstützte Unterverbindungsbreite haben, wird die neue Unterverbindungsbreite die aktuelle Unterverbindungsbreite bleiben.

[0142] Wenn die neue angeforderte Ratenserie sich von der aktuellen Ratenserie unterscheidet und die Komponenten auf beiden Seiten der Verbindung zumindest eine gemeinsame unterstützte Ratenserie haben, wird die neue Ratenserie die höchste beidseitig unterstützte Ratenserie sein.

[0143] Man beachte, dass bei der Anforderung, die Ratenserie zu ändern, die anfragende Komponente das entsprechende Ratenserie-Unterstützungs-Bit für die neue Ratenserie in den Trainingssequenzen setzt und das Ratenserie-Unterstützungs-Bit für die aktuelle Ratenserie löscht. Im Gegenzug setzt die antwortende Komponente das entsprechende Ratenserie-Unterstützungs-Bit für die Ratenserie, die sie unterstützt, was mehr als eine Ratenserie sein kann.

[0144] Wenn die neue angeforderte Ratenserie sich von der aktuellen Ratenserie unterscheidet und die Komponenten auf beiden Seiten der Verbindung nicht zumindest eine gemeinsame unterstützte Ratenserie haben, wird die neue Ratenserie die aktuelle Ratenserie bleiben.

[0145] In einigen Ausführungsformen können High-Speed-Gear, Ratenserien und Unterverbindungsbreiten-Änderungen gleichzeitig angefordert werden.

[0146] Mit Bezug auf Tabelle 4 wird eine TS1-geordnete Liste gemäß einer Ausführungsform der vorliegenden Erfindung gezeigt.

Tabelle 4

Symbolnummer	Beschreibung
0	COM (K28.5) für Symbol-Ausrichtung.
1	<p>RATEN-Serie und GEAR-Unterstützung. Bit 0 – VERBINDUNGS-Bandbreiten-Änderung. 0b: Nicht-Bestätigung. Wenn dieses Bit gelöscht ist, zeigen die Bits 2 bis 7 unterstützte Werte an. Wenn das entsprechende Bit gesetzt ist, zeigt dies die Unterstützung für das Feld von diesem Bit an. 1b: Bestätigung Wenn dieses Bit gesetzt ist, melden die Bits 1 bis 7 angeforderte Werte. Wenn das entsprechende Bit gesetzt ist, zeigt dies an, dass das Feld, das von diesem Bit repräsentiert wird, angefordert wird. Es können mehrere Werte eingestellt werden</p> <p>Bit 1 – Reserviert Bit 2 – High-Speed-GEAR 1. 0b: HIGH-SPEED-GEAR 1 wird nicht unterstützt. 1b: HIGH-SPEED-GEAR 1 wird unterstützt. Bit 3 – High-Speed-GEAR 2. 0b: HIGH-SPEED-GEAR 2 wird nicht unterstützt. 1b: HIGH-SPEED-GEAR 2 wird unterstützt. Bit 4 – High-Speed-GEAR 3. 0b: HIGH-SPEED-GEAR 3 wird nicht unterstützt. 1b: HIGH-SPEED-GEAR 3 wird unterstützt. Bit 5 – Reserviert. Bit 6 – RATEN-Serie A. 0b: RATEN-Serie A wird nicht unterstützt. 1b: RATEN-Serie A wird unterstützt. Bit 7 – RATEN-Serie B. 0b: RATEN-Serie B wird nicht unterstützt. 1b: RATEN-Serie B wird unterstützt.</p>
2	TX-Spurnummer in VERBINDUNG für Sender und RX-Spurnummer in VERBINDUNG für Empfänger.

3	<p>TX-UNTERVERBINDUNGS-Breite.</p> <p>Für Downstream-Ports Wenn das VERBINDUNGS-Bandbreiten-Änderungsfeld (Bit 0 von Symbol 1 von TS2) der TS2-geordneten Liste nicht gesetzt ist, zeigen Bits 0 bis 7 die unterstützten TX UNTERVERBINDUNGS-Breiten an. Wenn das entsprechende Bit gesetzt ist, bedeutet dies, dass die UNTERVERBINDUNGS-Breite, die von diesem Bit repräsentiert wird, unterstützt ist.</p> <p>Wenn das UNTERVERBINDUNGS-Breiten-Änderungsfeld (Bit 0 von Symbol 1 von TS1) der TS1-geordneten List gesetzt ist, zeigen Bits 0 bis 7 die TX-UNTERVERBINDUNGS-Breiten an, mit denen der Port betrieben werden kann. Wenn das entsprechende Bit gesetzt ist, bedeutet dies, dass die UNTERVERBINDUNG mit der TX-UNTERVERBINDUNGS-Breite betrieben werden kann, die von diesem Bit repräsentiert ist.</p> <p>Bit 0 – x1 TX-UNTERVERBINDUNG Immer auf 1 gesetzt. Unterstützung für x1 TX-UNTERVERBINDUNG ist obligatorisch. Bit 1 – x2 TX-UNTERVERBINDUNG-Unterstützung. Bit 2 – x4 TX-UNTERVERBINDUNG Bit 3 – x8 TX-UNTERVERBINDUNG 0b: x8 TX-UNTERVERBINDUNG wird nicht unterstützt. 1b: x8 TX-UNTERVERBINDUNG wird unterstützt. Bit 4 – x12 TX-UNTERVERBINDUNG Bit 5 – x16 TX-UNTERVERBINDUNG Bit 6 – x32 TX-UNTERVERBINDUNG</p> <p>Alle anderen Bits sind reserviert.</p>
4	<p>RX-UNTERVERBINDUNGS-Breite.</p> <p>Für Downstream-Ports Wenn das VERBINDUNGS-Bandbreiten-Änderungsfeld (Bit 0 von Symbol 1 von TS2) der TS2-geordneten Liste nicht gesetzt ist, zeigen Bits 0 bis 7 die unterstützten RX UNTERVERBINDUNGS-Breiten an. Wenn das entsprechende Bit gesetzt ist, bedeutet dies, dass die UNTERVERBINDUNGS-Breite, die von diesem Bit repräsentiert wird, unterstützt ist.</p> <p>Wenn das UNTERVERBINDUNGS-Breiten-Änderungsfeld (Bit 0 von Symbol 1 von TS1) der TS1-geordneten List gesetzt ist, zeigen Bits 0 bis 7 die RX-UNTERVERBINDUNGS-Breiten an, mit denen der Port betrieben werden kann. Wenn das entsprechende Bit gesetzt ist, bedeutet dies, dass die UNTERVERBINDUNG mit der RX-UNTERVERBINDUNGS-Breite betrieben werden kann, die von diesem Bit repräsentiert ist.</p> <p>Bit 0 – x1 RX-UNTERVERBINDUNG Immer auf 1 gesetzt. Unterstützung für x1 RX-UNTERVERBINDUNG ist obligatorisch. Bit 1 – x2 RX-UNTERVERBINDUNG Bit 2 – x4 RX-UNTERVERBINDUNG Bit 3 – x8 RX-UNTERVERBINDUNG Bit 4 – x12 RX-UNTERVERBINDUNG Bit 5 – x16 RX-UNTERVERBINDUNG Bit 6 – x32 RX-UNTERVERBINDUNG</p> <p>Alle anderen Bits sind reserviert.</p>

5	<p>Trainingssteuerung.</p> <p>Bit 0 – Warmstart. 0b: Nicht-Bestätigung. 1b: Bestätigung</p> <p>Bit 1 – Verbindung abgeschaltet. 0b: Nicht-Bestätigung. 1b: Bestätigung</p> <p>Bit 2 – Reserviert. Reserviert.</p> <p>Bit 3 – Verschlüsselung deaktivieren. 0b: Nicht-Bestätigung. 1b: Bestätigung</p> <p>Bit 4:7 – reserviert.</p>
6–15	Standard TS1-geordnete Listen codieren dieses Symbol als einen TS1-Identifikator, D10.2 (4Ah).

[0147] Mit Bezug auf Tabelle 5 wird eine TS2-geordnete Liste gemäß einer Ausführungsform der vorliegenden Erfindung gezeigt.

Tabelle 5

Symbol Nummer	Beschreibung
0	COM (K28.5) für Symbol-Ausrichtung.
1	<p>RATEN-Serie und GEAR-Unterstützung.</p> <p>Bit 0 – VERBINDUNGS-Bandbreiten-Änderung 0b: Nicht-Bestätigung 1b: Bestätigung</p> <p>Bit 1 – VERBINDUNGS-Bandbreiten-Änderung bestätigen. 0b: Nicht-Bestätigung. Wenn dieses Bit gelöscht ist, zeigen die Bits 1 bis 7 unterstützte Werte an. Wenn das entsprechende Bit gesetzt ist, zeigt dies die Unterstützung für das Feld von diesem Bit an. 1b: Bestätigung Für Downstream-Port: Wenn dieses Bit gesetzt ist, melden die Bits 1 bis 7 für von dem Downstream-Port angefordertes TS2 die gewählten Werte. Wenn das entsprechende Bit gesetzt ist, zeigt dies an, dass das Feld, das von diesem Bit repräsentiert wird, gewählt wurde. Nur High-Speed-Gear und eine RATEN-Serie soll gewählt werden. Für Upstream-Port: Wenn dieses Bit gesetzt ist, melden die Bits 1 bis 7 für von dem Upstream-Port angefordertes TS2 die gewählten Werte. Wenn das entsprechende Bit gesetzt ist, zeigt dies Unterstützung für das Feld an, das von diesem Bit repräsentiert wird.</p> <p>Bit 2 – High-Speed-GEAR 1. Bit 3 – High-Speed-GEAR 2. Bit 4 – High-Speed-GEAR 3. Bit 5 – Reserviert. Bit 6 – RATEN-Serie A. Bit 7 – RATEN-Serie B.</p>
2	TX-Spurnummer in VERBINDUNG für Sender und RX-Spurnummer in VERBINDUNG für Empfänger.

3	<p>TX-UNTERVERBINDUNGS-Breiten-Unterstützung.</p> <p>Für Downstream-Ports Wenn das VERBINDUNGS-Bandbreiten-Änderung-Bestätigen-Feld (Bit 1 von Symbol 1 von TS2) der TS2-geordneten Liste nicht gesetzt ist, zeigen Bits 0 bis 7 die unterstützten TX UNTERVERBINDUNGS-Breiten an. Wenn das entsprechende Bit gesetzt ist, bedeutet dies, dass die UNTERVERBINDUNGS-Breite, die von diesem Bit repräsentiert wird, unterstützt ist.</p> <p>Wenn das VERBINDUNGS-Bandbreiten-Änderung-Bestätigen-Feld (Bit 1 von Symbol 1 von TS2) der TS2-geordneten Liste nicht für TS2-geordneten Listen, die von dem Downstream-Port gesendet werden, gesetzt ist, zeigen Bits 0 bis 7 die gewählten TX UNTERVERBINDUNGS-Breiten an. Wenn das entsprechende Bit gesetzt ist, bedeutet dies, dass die UNTERVERBINDUNGS-Breite, die von diesem Bit repräsentiert wird, für den Betrieb gewählt wurde. Nur eine TX-UNTERVERBINDUNGS-Breite darf ausgewählt werden.</p> <p>Für Upstream-Ports: Die Bits 0 bis 7 zeigen immer die unterstützten TX-UNTERVERBINDUNGS-Breiten an. Bit 0 – x1 TX-UNTERVERBINDUNGS-Unterstützung. Bit 1 – x2 TX-UNTERVERBINDUNGS-Unterstützung. Bit 2 – x4 TX-UNTERVERBINDUNGS-Unterstützung. Bit 3 – x8 TX-UNTERVERBINDUNGS-Unterstützung. Bit 4 – x12 TX-UNTERVERBINDUNGS-Unterstützung. Bit 5 – x16 TX-UNTERVERBINDUNGS-Unterstützung. Bit 6 – x32 TX-UNTERVERBINDUNGS-Unterstützung.</p> <p>Alle anderen Bits sind reserviert.</p>
4	<p>RX-UNTERVERBINDUNGS-Breiten-Unterstützung.</p> <p>Für Downstream-Ports Wenn das VERBINDUNGS-Bandbreiten-Änderungsfeld (Bit 0 von Symbol 1 von TS2) der TS2-geordneten Liste nicht gesetzt ist, zeigen Bits 0 bis 7 die unterstützten RX UNTERVERBINDUNGS-Breiten an. Wenn das entsprechende Bit gesetzt ist, bedeutet dies, dass die UNTERVERBINDUNGS-Breite, die von diesem Bit repräsentiert wird, unterstützt ist.</p> <p>Wenn das VERBINDUNGS-Bandbreiten-Änderung-Feld (Bit 0 von Symbol 1 von TS2) der TS2-geordneten Liste nicht für TS2-geordneten Listen, die von dem Downstream-Port gesendet werden, gesetzt ist, zeigen Bits 0 bis 7 die gewählten RX-UNTERVERBINDUNGS-Breiten an. Wenn das entsprechende Bit gesetzt ist, bedeutet dies, dass die UNTERVERBINDUNGS-Breite, die von diesem Bit repräsentiert wird, für den Betrieb gewählt wurde. Nur eine RX-UNTERVERBINDUNGS-Breite darf ausgewählt werden.</p> <p>Für Upstream-Ports: Die Bits 0 bis 7 zeigen immer die unterstützten TX-UNTERVERBINDUNGS-Breiten an. Bit 0 – x1 RX-UNTERVERBINDUNGS-Unterstützung. Bit 1 – x2 RX-UNTERVERBINDUNGS-Unterstützung. Bit 2 – x4 RX-UNTERVERBINDUNGS-Unterstützung. Bit 3 – x8 RX-UNTERVERBINDUNGS-Unterstützung. Bit 4 – x12 RX-UNTERVERBINDUNGS-Unterstützung. Bit 5 – x16 RX-UNTERVERBINDUNGS-Unterstützung. Bit 6 – x32 RX-UNTERVERBINDUNGS-Unterstützung.</p> <p>Alle anderen Bits sind reserviert.</p>

5	Trainingssteuerung. Bit 0 – Warmstart. 0b: Nicht-Bestätigung. 1b: Bestätigung Bit 1 – Verbindung abgeschaltet. 0b: Nicht-Bestätigung. 1b: Bestätigung Bit 2 – Reserviert. Reserviert. Bit 3 – Verschlüsselung deaktivieren. 0b: Nicht-Bestätigung. 1b: Bestätigung Bit 4:7 – reserviert.
6–15	Standard TS2-geordnete Listen codieren dieses Symbol als einen TS2-Identifikator, D5.2 (45h)

[0148] Es wird nun auf **Fig. 12** Bezug genommen, die ein Blockdiagramm einer Recovery-Unterstatus-Maschine gemäß einer Ausführungsform der vorliegenden Erfindung illustriert. Wie in **Fig. 12** gezeigt, kann diese Wiederaufnahme-Unterstatus-Maschine von einem L0- oder L1-Status in einen Wiederaufnahme-Unterstatus-Maschine-Status **1210** eintreten. Von dort aus kann die Steuerung je nach den empfangenen Werten zu einem Recovery.Reconfig-Unterstatus **1220** und dann in einen Configuration.Update-Unterstatus übergeben.

[0149] Alternativ kann die Steuerung von Unterstatus **1210** zu einem Recovery.Complete-Unterstatus **1230** und danach zu einem Recovery.Idle-Status **1240** (in den der Recovery.Reconfig-Unterstatus eintreten kann) übergehen. Danach kann die Wiederaufnahme-Unterstatusmaschine den Recovery.Idle Unterstatus **1240** verlassen und die Steuerung an den L0-aktiven Zustand übergehen.

[0150] In der Recovery.Entry-Statusmaschine sendet der Sender TS1 geordnete Listen auf allen konfigurierten TX-Spuren. Die Komponente, die die Bandbreiten-Änderung TS1 einleitet, sendet geordneten Listen, wobei das Verbindungsbandbreiten-Änderungs-Bit gesetzt ist, und die aktualisierten Verbindungs-Bandbreiten-Parameter auf allen konfigurierten TX-Spuren. Nach dem Empfang von zwei oder mehr TS1- oder TS2-geordneten Listen mit auf diesen Zustand gesetztem Verbindungs-Bandbreiten-Änderungs-Bit, sendet die Komponente, die die Bandbreiten-Änderung nicht initiiert, sendet TS1-geordneten Listen mit gesetztem Verbindungs-Bandbreiten-Änderungs-Bit. Die Verbindungs-Bandbreiten-Parameter repräsentieren die unterstützten Konfigurationen.

[0151] Der nächste Status ist Recovery.ReConfig, der auftreten kann, wenn eine vorbestimmte Anzahl (z. B. 8) von aufeinander folgenden TS1- oder TS2-geordneten Listen auf allen konfigurierten RX-Spuren empfangen werden, wobei das Verbindungsbandbreiten-Änderungs-Bit gesetzt ist, das mit dem Verbindungsbandbreiten-Änderungs-Bit auf der Seite des Senders korrespondiert.

[0152] Der nächste Status ist Recovery.Complete, der auftreten kann, wenn eine vorbestimmte Anzahl (z. B. 8) von aufeinander folgenden TS1- oder TS2-geordneten Listen auf allen konfigurierten RX-Spuren empfangen werden, wobei das Verbindungsbandbreiten-Änderungs-Bit gelöscht ist, das mit dem Verbindungsbandbreiten-Änderungs-Bit auf der Seite des Senders korrespondiert.

[0153] Andernfalls ist nach einem vorbestimmten Timeout-Intervall der nächste Zustand ERKENNEN, wobei dieses Intervall in einem Ausführungsbeispiel 24 Millisekunden (ms) betragen kann. Hier senden alle konfigurierten TX-Spuren mindestens einen EIOS, der von HS-BURST abgeschlossen ist, gehen in STALL ein. Sobald alle konfigurierten TX-Spuren und RX-Spuren in STALL eingegangen sind, wird RCT ausgelöst, um die konfigurierten TX-Spuren und RX-Spuren in HIBERN8 eintreten zu lassen.

[0154] In der Recovery.ReConfig-Statusmaschine für den Downstream-Portsendet der Sender TS2-geordnete Listen auf allen konfigurierten TX-Spuren, wobei das Verbindungs-Bandbreiten-Änderungs-Bit und Verbindungsbandbreite Verbindungs-Bandbreiten-Änderungsbestätigungs-Bit gesetzt ist, wenn eine Verbindungs-Bandbreiten-Änderung auf Basis der TS1-geordneten Listen-Verhandlung im Recovery.Entry-Status und der Richtlinien, die in dynamische Verbindungs-Rekonfiguration beschrieben sind, erforderlich ist. Ansonsten ist das Verbindungs-Bandbreiten-Änderungs-Bit gelöscht.

[0155] Wenn eine Verbindungs-Bandbreiten-Änderung erfolgen soll, repräsentieren die Verbindungsbandbreiten-Parameter in der TS2-geordneten Liste die neue Ziel-Verbindungsbandbreite, die auf Basis der Richtlinien, die in dynamische Verbindungs-Rekonfiguration beschrieben sind, bestimmt wird. Der Upstream-Port akzeptiert die neue Ziel-Verbindungsbandbreite, wenn 2 oder mehr TS2-geordnete Listen empfangen wurde, für die das Verbindungsbandbreiten-Änderungs-Bit und das Verbindungsbandbreiten-Änderungbestätigungs-Bit gesetzt sind. Wenn die Verbindungsbandbreiten-Änderung nicht stattfinden soll, dann ist der nächste Status Recovery.Idle, unmittelbar nachdem alle konfigurierten RX-Spuren acht aufeinanderfolgende TS2-geordnete Listen empfangen, wobei das Verbindungsbandbreiten-Änderungs-Bit gesetzt und das Verbindungsbandbreiten-Änderungbestätigungs-Bit gelöscht ist. Es werden dann 16 TS2-geordnete Listen gesendet, wobei das Verbindungsbandbreiten-Änderungs-Bit gesetzt und das Verbindungsbandbreiten-Änderungs-Bit gelöscht ist, nachdem eine TS2-geordnete Liste empfangen wurde.

[0156] Für den Upstream-Port sendet der Sender TS2 geordnete Listen auf allen konfigurierten TX-Spuren, wobei das Verbindungsbandbreiten-Änderungs-Bit gesetzt und das Verbindungsbandbreiten-Änderungbestätigungs-Bit gelöscht ist, bis er auf allen konfigurierten RX-Spuren 2 oder mehr TS2-geordneten Listen empfängt, wobei das Verbindungsbandbreiten-Änderungs-Bit und das Verbindungsbandbreiten-Änderungbestätigungs-Bit gesetzt sind. Daraufhin beginnt der Sender, TS2-geordnete Listen auf allen konfigurierten TX-Spuren zu übertragen, wobei das Verbindungsbandbreiten-Änderungs-Bit und das Verbindungsbandbreiten-Änderungbestätigungs-Bit gesetzt sind. Die Link-Bandbreiten-Parameter in den TS2-geordneten Listen zeigen weiterhin die unterstützten Konfigurationen an, unabhängig davon, ob das Verbindungsbandbreiten-Änderungbestätigungs-Bit gesetzt oder gelöscht ist. Der nächste Status ist Recovery.Idle, unmittelbar nachdem alle konfigurierten RX-Spuren acht aufeinanderfolgende TS2-geordnete Listen empfangen, wobei das Verbindungsbandbreiten-Änderungs-Bit gesetzt und das Verbindungsbandbreiten-Änderungbestätigungs-Bit gelöscht ist. Es werden dann 16 TS2-geordnete Listen gesendet, wobei das Verbindungsbandbreiten-Änderungs-Bit gesetzt und das Verbindungsbandbreiten-Änderungs-Bestätigen-Bit gelöscht ist, nachdem eine TS2-geordnete Liste empfangen wurde.

[0157] Für sowohl die Downstream als auch die Upstream-Ports ist, wenn eine Verbindungsbandbreiten-Änderung gewählt ist, der nächste Status Configuration.Update, unmittelbar nachdem alle konfigurierten RX-Spuren acht aufeinanderfolgende TS2-geordneten Listen erhalten haben, wobei das Verbindungsbandbreiten-Änderungs-Bit und das Verbindungsbandbreiten-Änderungbestätigungs-Bit gesetzt sind, und 16 TS2-geordneten Listen gesendet wurden, wobei das Verbindungsbandbreiten-Änderungs-Bit und das Verbindungsbandbreiten-Änderungbestätigungs-Bit gesetzt sind, nachdem Empfang eine TS2-geordnete Liste empfangen wurde und alle konfigurierten TX-Spuren und RX-Spuren in den HIBERN8 Zustand eingetreten sind. Hier treten alle konfigurierten TX-Spuren in den STALL-Status ein, und RCT wird ausgelöst, um den Betriebsmodus auf den neuen High-Speed-Gear und die neue Ratenserie zu wechseln und die TX-Spuren und RX-Spuren zu HIBERN8 übergehen zu lassen, nachdem alle konfigurierten TX-Spuren und RX-Spuren in den STALL-Status übergegangen sind.

[0158] Andernfalls ist nach einem vorbestimmten Timeout-Intervall, welches 2 ms sein kann, der nächste Zustand ERKENNEN (wie oben beschrieben).

[0159] In der Recovery.Complete-Statusmaschine sendet der Sender TS2 geordnete Listen auf allen konfigurierten TX-Spuren. Man beachte, dass der Signalversatz von Spur zu Spur vor dem Verlassen dieses Status beendet ist. Der nächste Status ist Recovery.Idle, unmittelbar nachdem alle konfigurierten RX-Spuren acht aufeinanderfolgende TS2-geordneten Listen empfangen und 16 TS2-geordnete Listen gesendet werden, nachdem eine TS2-geordnete Liste erhalten wurde. Andernfalls ist, wie oben beschrieben, der nächste Status ERKENNEN.

[0160] In der Recovery.Idle-Statusmaschine ist der nächste Status deaktiviert, wenn dies angewiesen wird. Man beachte, dass "wenn angewiesen" sich auf einen Downstream-Port bezieht, der von einer höheren Ebene angewiesen wird, das Verbindung abgeschaltet-Bit (TS1 und TS2) der Verbindung zu bestätigen. Der nächste Status ist Warmstart, wenn dies angewiesen wird. Man beachte, dass "wenn angewiesen" sich auf einen Downstream-Port bezieht, der von einer höheren Ebene angewiesen wird, das Warmstart-Bit (TS1 und TS2) der Verbindung zu bestätigen. Der nächste Status ist sofort deaktiviert, nachdem alle konfigurierten RX-Spuren TS1-geordneten Listen erhalten haben, wobei das Verbindung abgeschaltet-Bit in zwei aufeinanderfolgend empfangenen TS1-geordneten Listen bestätigt war. Hinweis: Dieses Verhalten trifft nur auf Upstream-Ports zu. Der nächste Status ist sofort Warmstart, nachdem alle konfigurierten RX-Spuren TS1-geordneten Listen erhalten hat, wobei das Warmstart-Bit in zwei aufeinanderfolgenden TS1-geordneten Listen bestätigt war. Ruhe-Daten werden auf allen konfigurierten TX-Spuren übertragen. Wenn zu einem der oben aufgeführten Status

angewiesen wird, brauchen vor dem Übergang in die anderen Status keine Ruhe-Daten geschickt werden. Der nächste Status ist L0, wenn acht aufeinanderfolgenden Symbolzeiten von Ruhe-Daten auf allen konfigurierten Spuren empfangen und 16 Ruhe-Daten-Symbole nach Erhalt eines Ruhe-Daten-Symbols gesendet werden. Andernfalls ist, wie oben beschrieben, der nächste Status ERKENNEN.

[0161] Wie hierin beschrieben, erlauben Ausführungsformen einem herkömmlichen PCIe™-Protokollstapel, Kommunikation, die über eine nicht-PCIe™ physische Ebene und Verbindung stattfinden, zu verknüpfen. Die hierin beschriebenen Ausführungsformen beziehen sich auf eine PHY-Einheit (M-PHY) in Übereinstimmung mit der MIPI-Spezifikation. Selbstverständlich sind Ausführungsformen in diesem Bezug nicht eingeschränkt.

[0162] Um Software-Kompatibilität mit einem vorhandenen PCIe™-Software-Stapel zu ermöglichen, können Ausführungsformen eine Zuordnung von verschiedenen Steuer- und Statusinformationen entsprechend dem PCIe™-Protokoll einer physischen Ebene eines anderen Protokolls bereitstellen. Genauer gesagt, können Ausführungsformen hier einen Mechanismus zur Verfügung stellen, um einen Bus-Treiber zu aktivieren, auf einen Konfigurationsraum zugreift, um eine bestimmte Verbindung unter Verwendung von Informationen in dem Konfigurationsraum zu verwalten und zu steuern, verschiedene Operationen entlang der Verbindung auszuführen, einschließlich Initialisierung, Konfiguration/Rekonfiguration und Kommunikation und anderes.

[0163] Es wird nun auf **Fig. 13** Bezug genommen, die ein Blockdiagramm eine Funktionsstruktur gemäß einer Ausführungsform der vorliegenden Erfindung zeigt. Wie in **Fig. 13** gezeigt, kann die Funktionsstruktur **1300** eine Datenstruktur sein, die z. B. im Konfigurationsraum der Vorrichtung gespeichert sein kann, welcher eine Vielzahl von Registern umfassen kann, um Funktions-, Konfigurations-, Status- und Steuerinformation für die Verwendung durch verschiedene Agenten, einschließlich eines oder mehrerer Bus-Treiber, zu speichern. Natürlich kann in anderen Ausführungsformen diese Kapazitätsstruktur an anderen Orten innerhalb einer Vorrichtung gespeichert werden.

[0164] Wie in **Fig. 13** gesehen, kann Struktur **1300** ein PCIe gehören™-Funktions-Register **1305** und eine Zeigerregister für nächste Funktionen **1307** umfassen, und ein PCIe™-Funktions-Identifikator-Register **1309**, das gezeigt ist, kann sich in einem ersten Satz von Bytes befinden. Weiterhin können für Geräteinformationen, Gerät-Funktions-Register **1310** sowie mit ein Gerät-Status-Register **1314** und ein Gerät-Steuer-Register **1312** vorhanden sein. Für Verbindungsinformationen können ein Verbindungs-Funktions-Register **1315** sowie ein Verbindungs-Status-Register **1325** und ein Verbindungs-Steuer-Register **1320** bereitgestellt werden. Für Slot-Informationen kann ein Slot-Funktions-Register **1313** sowie ein Slot-Statusregister **1304** und ein Slot-Steuer-Register **1332** bereitgestellt werden. Für Root-Informationen, z. B. eines Root-Complex-Geräts, kann ein Root-Funktions-Register **1338** und ein Root Steuer-Register **1336** sowie ein Root-Status-Register **1340** bereitgestellt werden.

[0165] Für zusätzliche Informationen kann Funktions-Struktur **1300** zusätzliche Informationen für Geräte, Verbindungen und Slots umfassen. Insbesondere können, wie gesehen, zusätzliche Geräteinformationen über ein Geräte-Funktionen-2-Register **1350**, ein Geräte-Status-2-Register **1354** und ein Geräte-Steuerung-2-Register **1352** zur Verfügung gestellt werden. Für weitere Verbindungsinformationen können ein Verbindungs-Funktions-2-Register **1360** sowie ein Verbindungs-Status-2-Register **1364** und ein Verbindungs-Steuer-2-Register **1362** bereitgestellt werden. Für die Zwecke zusätzlicher Slot-Informationen, können ebenfalls ein Slot-Funktionen-2 Register **1370**, ein Slot-Status-2-Register **1374** und ein Slot-Steuer-2-Register **1372** vorgesehen werden. Schließlich kann die Funktions-Struktur **1300** die Steuerung der Verbindungsbreite und -geschwindigkeit sein, wie hier beschrieben, und ferner eine Mobile Express™-Funktion ein und Steuerregister **1380** umfassen. Obwohl dies in dieser spezifischen Illustration in der Ausführungsform von **Fig. 13** gezeigt ist, versteht sich, dass dadurch der Umfang der vorliegenden Erfindung in dieser Hinsicht in keiner Weise eingeschränkt wird. Zum Beispiel kann Register **1380** in einigen Ausführungsformen Teil einer separaten Funktions-Struktur sein.

[0166] Um zu erlauben, dass Informationen in der erweiterten Funktions-Struktur der **Fig. 13** in Verbindung mit einer nicht-PCIe™ physischen Ebene verwendet werden und dennoch mit zuvor existierenden PCIe™-Software-Stapel kompatibel sind, können bestimmte Register-Schnittstellen-Werte vorgesehen sein, um zumindest einige dieser erweiterten Funktions-Register, die unterschiedliche Verbindungsgeschwindigkeiten, Gears und Verbindungsbreiten bieten, mit einer nicht-PCIe™ physischen Ebene zu verknüpfen.

[0167] Unter Bezugnahme auf Tabelle 6 wird eine Auflistung bestimmter Funktions-Register und empfohlenen Einstellungen für bestimmte Felder der Register gezeigt, um älteren Software-Stapeln zu ermöglichen, mit einer nicht-PCIe™ physischen Ebene zu kommunizieren. Obwohl diese in der Tabelle 6 gezeigten Regis-

tereinstellungen für die Verbindung mit einer M-PHY gemäß einer MIPI-Spezifikation sind, versteht sich, dass dadurch der Umfang der vorliegenden Erfindung in dieser Hinsicht in keiner Weise eingeschränkt wird.

Tabelle 6

Funktions-Register	Registerfeld	Empfohlene Einstellungen/Kommentar	
Gerätfunktions-Register	Endpunkt L0s akzeptable Latenz	Dieses Feld gilt für L0_Stall	
	Endpunkt L1 akzeptable Latenz	Dieses Feld gilt für L0_Stall	
Verbindungsfunktions-Register	Max. Verbindungsgeschwindigkeit	Siehe den Unterstützte Verbindungsgeschwindigkeiten-Vektor für die Zuordnung zu M-PHY-Gears	
	Maximale Verbindungsbreite	Zeigt x1 oder Niedrigere der Max. Verbindungsbreite in jede Richtung an	
Verbindungs-Steuerregister	Active State Power Management (ASPM) Steuerung	L0s wird L0_stall zugeordnet	
	Verbindung abgeschaltet	Wie PCI Express	
	Neutraining der Verbindung	Neutraining von Mobile Express	
	Erweiterte Synch	Nicht anwendbar und Register kann RW sein	
Verbindungsstatus-Register	Aktuelle Verbindungsgeschwindigkeit	Siehe den Unterstützte Verbindungsgeschwindigkeiten-Vektor für die Zuordnung zu M-PHY-Gears	
	Verhandelte Verbindungsbreite	Zeigt x1 oder Niedrigere der konfiguriert. Verbindungsbreite in jede Richtung an	
Verbindungsfunktion-2-Register	Unterstützte-Verbindungsgeschwindigkeit-Vektor	Bit-Definitionen für Mobile Express sind	
		Bit 0:	HS-G1, HS-G2
		Bit 1:	HS-G3
		Bit 6: 2	RsvdP
Verbindungssteuerung-2-Register	Ziel-Verbindungsgeschwindigkeit	Siehe den Unterstützte Verbindungsgeschwindigkeiten-Vektor für die Zuordnung zu M-PHY-Gears	
Mobile Express Funktions- und Steuerregister	Mobile Express Max. Verbindungsgeschwindigkeit-Funktion	Wird verwendet, um die Unterstützung für HS-G1 oder HS-G2 zu deklarieren, kann verwendet werden, wenn die Max. Verbindungsgeschwindigkeit-Codierung gleich 0001b ist.	
		Bit-Definitionen für Mobile Express sind wie folgt. Wenn ein Bit gesetzt ist, zeigt es an, dass der entsprechende HS-Gear unterstützt wird.	
		Bit 0:	HS-G1
		Bit 1:	HS-G2
	Maximale TX-Spurbreiten-Funktion	Wie TX-LANE_WIDTH_SUPPORTED-Feld in Mobile Express-Funktionsattribute, Attribut-ID 0x01	
	Maximale RX-Spurbreiten-B22-Funktion	Wie RX-LANE_WIDTH_SUPPORTED-Feld in Mobile Express-Funktionsattribute, Attribut-ID 0x02	

Mobile Express Ziel-Verbindungsgeschwindigkeit-Erweiterungs-Steuerung	Wird verwendet, um die Ziel-Verbindungsgeschwindigkeit auf HS-G1 oder HS-G2 einzustellen	
	Bit-Definitionen für Mobile Express sind wie folgt. Wenn ein Bit gesetzt ist, zeigt es an, dass der entsprechende HS-Gear ausgewählt ist.	
	Bit 0:	HS-G1
	Bit 1:	HS-G2
TX-Spurbreiten-Status	Dieses Feld entspricht dem TX-LANE_WIDTH-Register der Mobile Express Konfigurationsattribute Attribut-ID 0x01. Dies ist ein RO-Register	
RX-Spurbreiten-Status	Dieses Feld entspricht dem RX-LANE_WIDTH-Register der Mobile Express Konfigurationsattribute Attribut-ID 0x02. Dies ist ein RO-Register	

[0168] Man beachte, dass für die Zwecke der von Gerätetreibern ausgeführten Operationen sowie der hier beschriebenen Statusmaschinen die Informationen in diesen Feldern der Register der Funktionsstruktur verwendet werden kann. Weiterhin ist zu beachten, dass die empfohlenen Einstellungen für bestimmte dieser Felder das Unterstützte-Verbindungsgeschwindigkeit-Vektor-Funktionen-2-Register identifizieren oder sich darauf beziehen. Dies kann auf verschiedene Weisen implementiert werden. Zum Beispiel können die Werte aus diesem Vektor in die Felder der angegebenen Register kopiert werden, oder einen Wert der Felder dieser anderen Register (z. B. ein Maximale-Verbindungsgeschwindigkeit Feld des Verbindungs-Funktions-Registers) kann einen Wert haben, um so als ein Zeiger auf diesen Unterstützte-Verbindungsgeschwindigkeit-Vektor des Link-Funktionen-2-Registers zu agieren.

[0169] Bezugnehmend auf **Fig. 14** ist ein Ablaufdiagramm eines Verfahrens gemäß einer anderen Ausführungsform der vorliegenden Erfindung gezeigt. Insbesondere kann Verfahren **1400** aus **Fig. 14** verwendet werden, z. B. durch einen Gerätetreiber oder eine andere Logik, um Informationen in verschiedenen dieser Konfigurationsregister der Funktionsstruktur zu verwenden, um bestimmte Einstellungen zu bestimmen, wie beispielsweise für eine maximale Verbindungsgeschwindigkeit, Verbindungsbreite oder dergleichen. Zum Zweck der Veranschaulichung sei angenommen, dass die gewünschte Konfigurationsinformationen, die in **Fig. 14** erhalten und verwendet, Informationen für eine maximale Verbindungsgeschwindigkeit seien. Es versteht sich jedoch, dass dadurch der Umfang der vorliegenden Erfindung in dieser Hinsicht in keiner Weise eingeschränkt wird.

[0170] Das Verfahren **1400** kann bei Block **1410** durch den Zugriff auf ein erstes Feld eines ersten Verbindungsfunktions-Register mit einem Gerätetreiber beginnen. Zum Beispiel mit Rückbezug auf Tabelle 6 kann dieses erste Verbindungsfunktions-Register das Verbindungsfunktions-Register sein, und das spezifische Feld, auf das zugegriffen wird, kann das Max. Verbindungsgeschwindigkeit-Feld sein. Dann kann im Block **1420** dieses Feld als Zeigerwert zu einer Position in einem zweiten Verbindungsfunktions-Register verwendet werden. Genauer gesagt kann der Wert dieses ersten Feld einen bestimmten Ort innerhalb eines anderen Konfigurationsregister angeben, z. B. das Verbindungsfunktions-2-Register der Tabelle 6, um somit ein bestimmtes Bit innerhalb des Unterstützte-Verbindungsgeschwindigkeit-Vektors anzuzeigen.

[0171] So geht die Steuerung zu Block **1440** über, wo auf dieses zweite Verbindungsfunktions-Register mit dem Gerätetreiber zugegriffen werden kann, um Informationen zu dem identifizierten Ort zu erhalten. Man nehme in diesem Fall an, dass das erste Feld des ersten Verbindungsfunktions-Register einen Wert von 01b hat, was anzeigt, dass es auf das Bit 1 Unterstützte-Verbindungsgeschwindigkeit-Vektors zeigt. Damit kann bei Block **1440** ein Gerätetreiber auf dieses Register zugreifen und den Wert dieses Bits erhalten, welches anzeigt, wenn es gesetzt ist, dass die Verbindung eine hohe High-Speed-Gear-3-Geschwindigkeit (HS-G3) unterstützen kann.

[0172] Immer noch mit Bezug auf **Fig. 14** kann dann unter Verwendung der Informationen, die an dem angegebenen Ort erhalten wurden, eine Verbindungsoperation durchgeführt werden. Zum Beispiel kann ein Gerätetreiber auf der Grundlage dieser ermittelten maximalen Verbindungsgeschwindigkeits-Einstellung einen Verbindungs-Rekonfigurations- oder andere Mechanismen initiieren, um bei dieser angegebenen Maximal-Verbindungsgeschwindigkeit zu arbeiten. Obwohl dies auf dieser hohen Ebene in der Ausführungsform von **Fig. 14** gezeigt ist, versteht sich, dass dadurch der Umfang der vorliegenden Erfindung in dieser Hinsicht in keiner Weise eingeschränkt wird.

[0173] Ausführungsformen können in vielen verschiedenen Arten von Systemen benutzt werden. Zum Beispiel wird in einer Ausführungsform eine Kommunikationseinrichtung eingestellt, um die verschiedenen Verfahren und Techniken, die hierin beschrieben sind, auszuführen. Natürlich ist der Schutzzumfang der vorliegenden Erfindung nicht auf eine Kommunikationsvorrichtung beschränkt, und stattdessen können andere Ausführungsformen sich auf andere Typen von Vorrichtungen zur Verarbeitung von Anweisungen beziehen oder auf ein oder mehrere maschinenlesbare Medien, darunter Anweisungen, die als Reaktion auf ihre Durchführung auf einer Rechenvorrichtung dazu führen, dass die Vorrichtung eine oder mehrerer der hier beschriebenen Methoden und Techniken ausführt.

[0174] Ausführungsformen können in Code implementiert werden und auf einem nicht flüchtigen Speichermedium gespeichert werden, auf dem Befehle gespeichert sind, die verwendet werden können, um ein System zu programmieren, die Befehle auszuführen. Das Speichermedium kann einschließen, ist aber nicht beschränkt auf, jede Art Disks, u. a. Floppy Disks, optische Disks, Solid State-Laufwerke (SSDs), Compact Disk Read-Only Memories (CD-ROMs), Compact Disk Rewritables (CD-RWs) und magnetooptische Disks (MO), Halbleiter-Geräte, wie Read-Only Memories (ROMs), Random Access Memories (RAMs), wie Dynamic Random Access Memories (DRAMs), Static Random Access Memories (SRAMs), Erasable Programmable Read-Only Memories (EPROMs), Flash Memories, Electrically Erasable Programmable Read-Only Memories (EEPROMs), magnetische oder optische Karten oder jede andere Art Speichermedium, das sich für das Speichern von elektronischen Befehlen eignet.

[0175] Obwohl die vorliegende Erfindung im Hinblick auf eine begrenzte Anzahl von Ausführungsformen beschrieben wurde, sind sich Fachleute bewusst, dass viele weitere Modifikationen und Varianten hiervon möglich sind. Die beigefügten Ansprüche sollen alle solche Modifikationen und Varianten abdecken, die dem Sinn und Wirkungsbereich der vorliegenden Erfindung entsprechen.

Patentansprüche

1. Vorrichtung, umfassend:

ein Protokollstapel für einen Peripheral Component Interconnect Express™ (PCIe™) Kommunikationsprotokoll, wobei der Protokollstapel eine Transaktionsebene und eine Verbindungsebene umfasst;
eine physische (PHY-)Einheit, die mit dem Protokollstapel gekoppelt ist, um die Kommunikation zwischen der Vorrichtung und einem mit der Vorrichtung über eine physische Verbindung gekoppelten Gerät zu erlauben, welches eine PHY-Einheit eines stromsparenden Übertragungsprotokoll ist und eine Schaltung einer physischen Einheit gemäß des stromsparenden Übertragungsprotokolls und eine logische Ebene umfasst, um den Protokollstapel mit der physischen Schaltungseinheit-Schnittstelle zu verknüpfen; und
a Funktionsstruktur, die eine Vielzahl von Registern umfasst, um jeweils Funktionsinformationen, Statusinformationen oder Steuerinformationen zu speichern.

2. Vorrichtung nach Anspruch 1, die ferner ein erstes Verbindungs-Steuerregister der Funktionsstruktur umfasst, um eine Energieverwaltungs-Einstellung zu speichern, um einen ersten stromsparenden Verbindungszustand des PCIe™-Kommunikationsprotokoll zu verursachen, um zu einem zweiten stromsparenden Status des stromsparenden Kommunikationsprotokolls zu verknüpfen.

3. Vorrichtung nach Anspruch 2, wobei das erste Verbindungs-Steuerregister weiterhin eine Verbindungs-Neutraining-Einstellung speichert, die dazu führt, dass eine Vielzahl von Trainingssequenzen in Reaktion auf eine Verbindungs-Neutraining-Anfrage übermittelt werden.

4. Vorrichtung nach Anspruch 1, 2 oder 3, die ferner ein erstes Verbindungsfunktions-Register umfasst, um eine maximale Verbindungsgeschwindigkeit-Einstellung zu speichern, um eine Gear-Geschwindigkeit des stromsparenden Übertragungsprotokoll mit einer entsprechenden Geschwindigkeit des PCIe™-Protokolls zu verknüpfen.

5. Vorrichtung nach Anspruch 4, die ferner ein zweites Verbindungsfunktions-Register umfasst, um einen Vektor mit einer Vielzahl von Bits zu speichern, die jeweils identifizieren, ob ein entsprechender einer Vielzahl von Gears des stromsparenden Übertragungsprotokolls unterstützt wird.

6. Vorrichtung nach Anspruch 5, wobei Treiber auf die maximale Verbindungsgeschwindigkeit-Einstellung des ersten Verbindungsfunktions-Registers zugreift und diese nutzt, um die maximale Verbindungsgeschwindigkeit-Einstellung als Zeiger zu benutzen, um dann auf einen weiteren Vektor des zweiten Verbindungsfunktions-Register zuzugreifen, um eine maximale Verbindungsgeschwindigkeit der Verbindung zu bestimmen.

7. Vorrichtung nach Anspruch 5, die ferner eine Verbindungstraining-Statusmaschine umfasst, um die Verbindung zu konfigurieren, wobei die Verbindungstraining-Statusmaschine auf die maximale Verbindungsgeschwindigkeit-Einstellung des ersten Verbindungsfunktions-Registers zugreift und die maximale Verbindungsgeschwindigkeit-Einstellung als Zeiger nutzt.

8. Vorrichtung nach Anspruch 1, 2 oder 3, wobei die physische Verbindung eine asymmetrische Breite in Richtung von der Vorrichtung zu dem Gerät und in der Richtung von dem Gerät zu der Vorrichtung hat und die physische Verbindung konfigurierbar ist, mit einer asymmetrischen Frequenz in Richtung von der Vorrichtung zu dem Gerät und in der Richtung von dem Gerät zu der Vorrichtung zu arbeiten.

9. Verfahren, umfassend:

das Zugreifen auf eine erstes Feld eines ersten Verbindungsfunktions-Registers einer ersten Vorrichtung, die einen Protokollstapel, der eine Transaktionsebene und einer Verbindungsebene gemäß einem Peripheral Component Interconnect Express™-(PCIe™-)Kommunikationsprotokoll, und eine physische Ebene des Protokollstapels hat, welche eine physische (PHY-)Einheit eines zweiten Kommunikationsprotokolls hat, wobei die erste Vorrichtung mit einer zweiten Vorrichtung über eine physische Verbindung gekoppelt ist; die Verwendung des ersten Felds als ein Zeigerwert zu einem Ort in einem zweiten Verbindungsfunktions-Register der ersten Vorrichtung; und die Verwendung von Informationen aus dem Ort in dem zweiten Verbindungsfunktions-Register, um eine Konfigurationsoperation für die physische Verbindung durchzuführen.

10. Verfahren nach Anspruch 9, das ferner den Zugriff auf das zweite Verbindungsfunktions-Register mit einem Treiber umfasst, um Konfigurationsinformationen aus dem Ort in dem zweiten Verbindungsfunktions-Register zu erhalten.

11. Verfahren nach Anspruch 9, das ferner die Verwaltung der physischen Verbindung umfasst, um gemäß asymmetrischer Verbindungsbreiten in einer ersten Richtung und einer zweiten Richtung zu kommunizieren.

12. Verfahren nach Anspruch 11, das ferner die Verwaltung der physischen Verbindung umfasst, um gemäß asymmetrischer Verbindungsgeschwindigkeiten in der ersten Richtung und der zweiten Richtung zu kommunizieren.

13. Das Verfahren nach Anspruch 9, das ferner den Zugriff auf die maximale Verbindungsgeschwindigkeit-Einstellung des ersten Verbindungsfunktions-Registers umfasst und diese nutzt, um die maximale Verbindungsgeschwindigkeit-Einstellung als Zeiger zu benutzen, um dann auf einen weiteren Vektor des zweiten Verbindungsfunktions-Register zuzugreifen, um eine maximale Verbindungsgeschwindigkeit der physischen Verbindung zu bestimmen.

14. Verfahren nach Anspruch 9, das ferner das Speichern einer maximalen Verbindungsgeschwindigkeit im ersten Verbindungsfunktions-Register umfasst, wobei die maximale Verbindungsgeschwindigkeit-Einstellung eine Gear-Geschwindigkeit des zweiten Kommunikationsprotokolls mit einer entsprechenden Geschwindigkeit des PCIe™-Protokolls verknüpft.

15. Ein Verfahren nach Anspruch 14, das ferner das Speichern eines Vektors mit einer Vielzahl von Bits in einem zweiten Verbindungsfunktions-Register umfasst, wobei die Bits jeweils identifizieren, ob ein entsprechender einer Vielzahl von Gear-Geschwindigkeiten des zweiten Übertragungsprotokolls unterstützt wird.

16. System umfassend:

eine erste Vorrichtung, umfassend:

eine Transaktionsebene in Übereinstimmung mit einem Peripheral Component Interconnect Express(PCIe™-) Kommunikationsprotokoll;

eine Datenverbindungsebene gemäß des PCIe™-Kommunikationsprotokolls;

eine physische Ebene, die eine Übertragungssteuerung für die physische Ebene und eine Sendeschaltung einer physischen (PHY) Einheit in Übereinstimmung mit einem zweiten Kommunikationsprotokoll umfasst, wobei die Übertragungssteuerung für die physische Ebene die Sendeschaltung der PHY-Einheit an die Transaktionsebene und die Datenverbindungsebene des PCIe™-Kommunikationsprotokolls anpasst; und

eine erste Funktionsstruktur, die eine Vielzahl von Registern enthält, die jeweils zur Speicherung von Funktionsinformationen, Statusinformationen oder Steuerinformationen, zumindest einigen der Funktionsinformationen, die auf vorbestimmte Werte eingestellt werden, um an die PHY-Einheit-Sendeschaltung an die Transak-

tionsebene und Datenverbindungsebene des PCIe™-Kommunikationsprotokoll angepasst zu werden, benutzt werden; und
eine zweite Vorrichtung, die mit der ersten Vorrichtung über eine Verbindung des zweiten Kommunikationsprotokolls gekoppelt ist.

17. Das System nach Anspruch 16, das ferner ein erstes Verbindungsfunktions-Register umfasst, um eine maximale Verbindungsgeschwindigkeit-Einstellung zu speichern, um eine Gear-Geschwindigkeit des zweiten Übertragungsprotokoll mit einer entsprechenden Geschwindigkeit des PCIe™-Protokolls zu verknüpfen.

18. Das System nach Anspruch 17, das ferner ein zweites Verbindungsfunktions-Register umfasst, um einen Vektor mit einer Vielzahl von Bits zu speichern, die jeweils identifizieren, ob ein entsprechender einer Vielzahl von Gears des zweiten Übertragungsprotokolls unterstützt wird.

19. Das System nach Anspruch 18, das ferner eine Treiber umfasst, um auf die maximale Verbindungsgeschwindigkeit-Einstellung des ersten Verbindungsfunktions-Registers zuzugreifen und diese zu nutzen, um die maximale Verbindungsgeschwindigkeit-Einstellung als Zeiger zu benutzen, um dann auf einen weiteren Vektor des zweiten Verbindungsfunktions-Register zuzugreifen, um eine maximale Verbindungsgeschwindigkeit der Verbindung zu bestimmen.

20. Das System nach Anspruch 16, 17, 18 oder 19, das ferner eine Funktionsstruktur umfasst, um Funktionsinformationen der Verbindung zu speichern.

21. Ein Kommunikationsgerät, das angeordnet ist, um das Verfahren nach einem der Ansprüche 1 bis 11 auszuführen.

22. Mindestens ein maschinenlesbares Medium, das aus einer Vielzahl von Befehlen besteht, die bei ihrer Ausführung auf einem Computergerät das Computergerät veranlassen, ein Verfahren gemäß eines Verfahrens nach einem der Ansprüche 9 bis 15 auszuführen.

23. Eine Vorrichtung für die Ausführung von Anweisungen, die so konfiguriert ist, dass sie das Verfahren nach einem der Ansprüche 9–15 ausführt.

24. Eine Vorrichtung, die Mittel umfasst, um das Verfahren nach einem der Ansprüche 9 bis 15 auszuführen.

Es folgen 14 Seiten Zeichnungen

Anhängende Zeichnungen

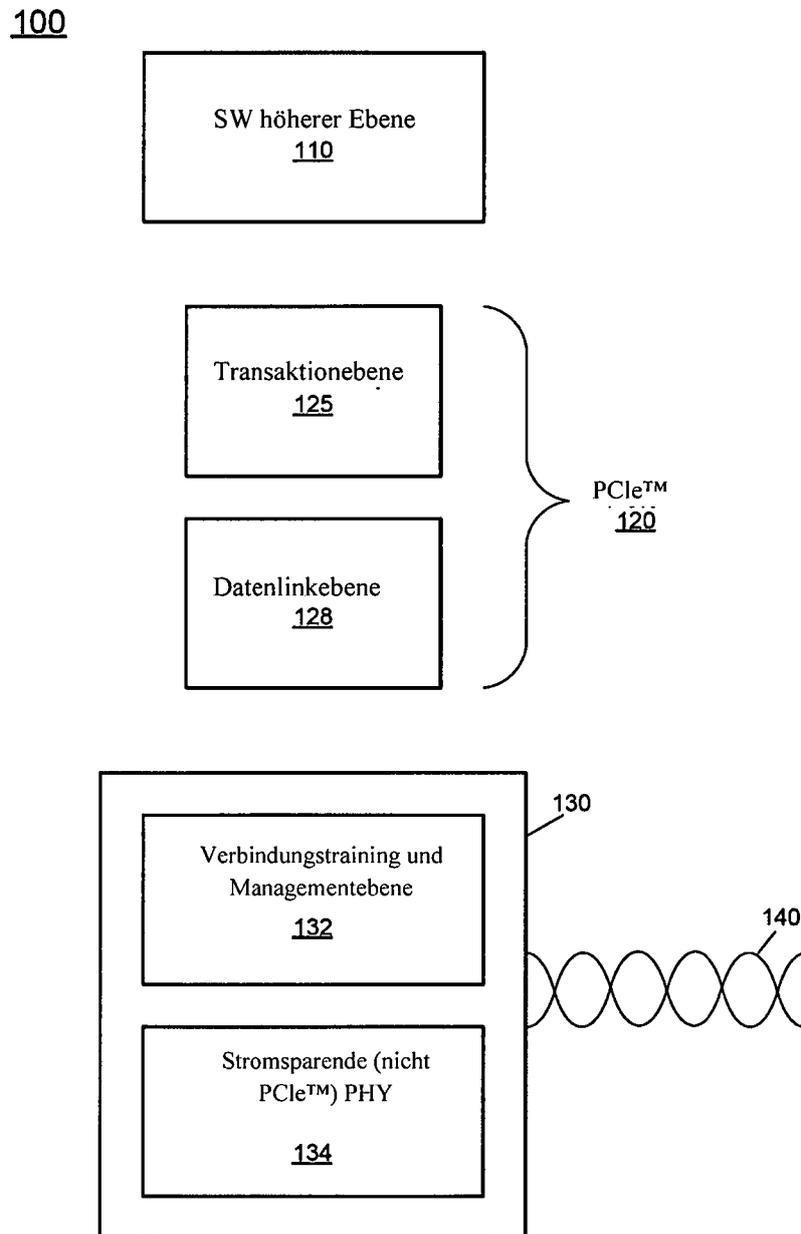


FIG. 1

200

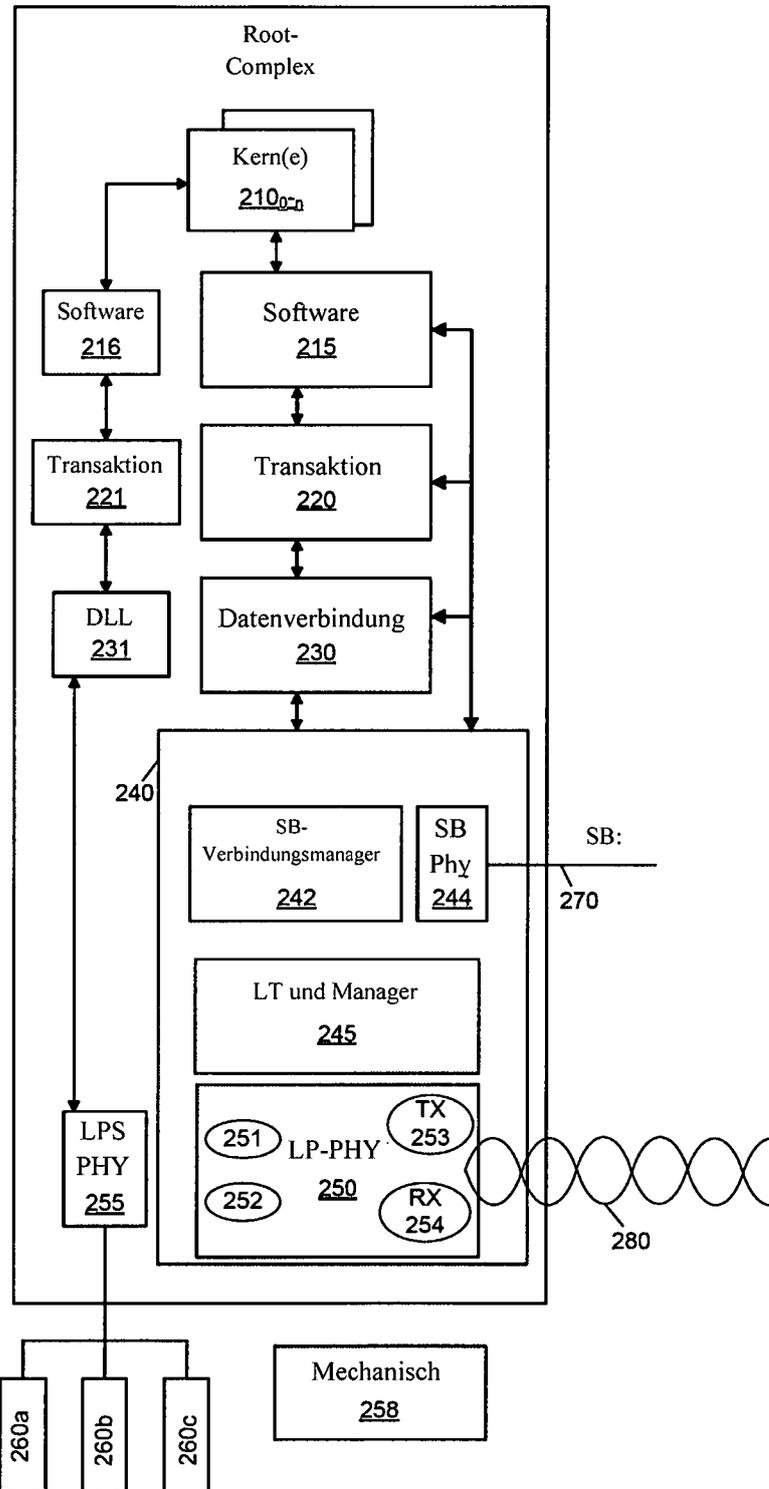


FIG. 2

300

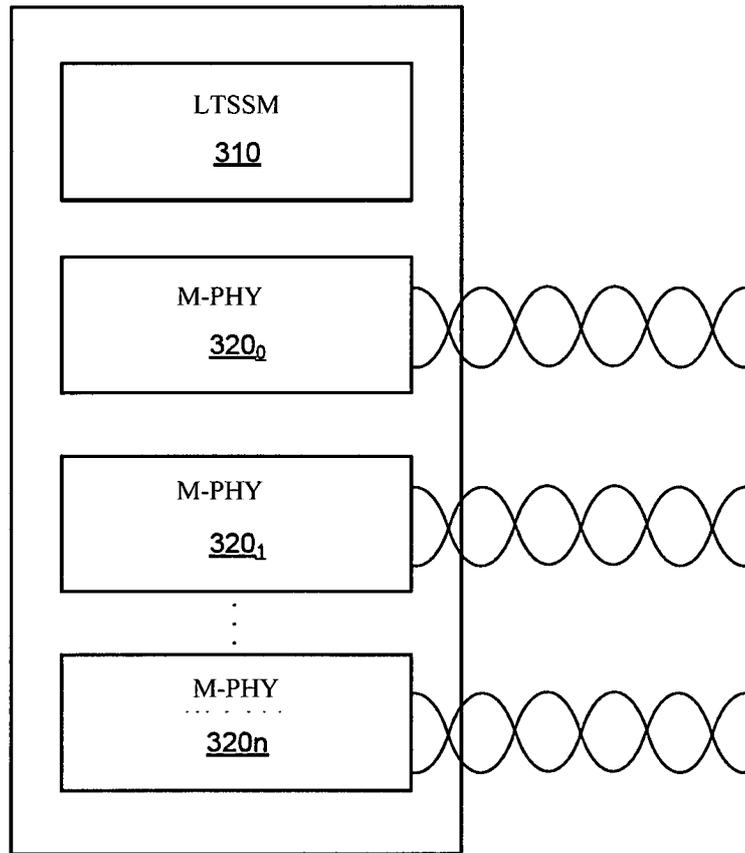


FIG. 3

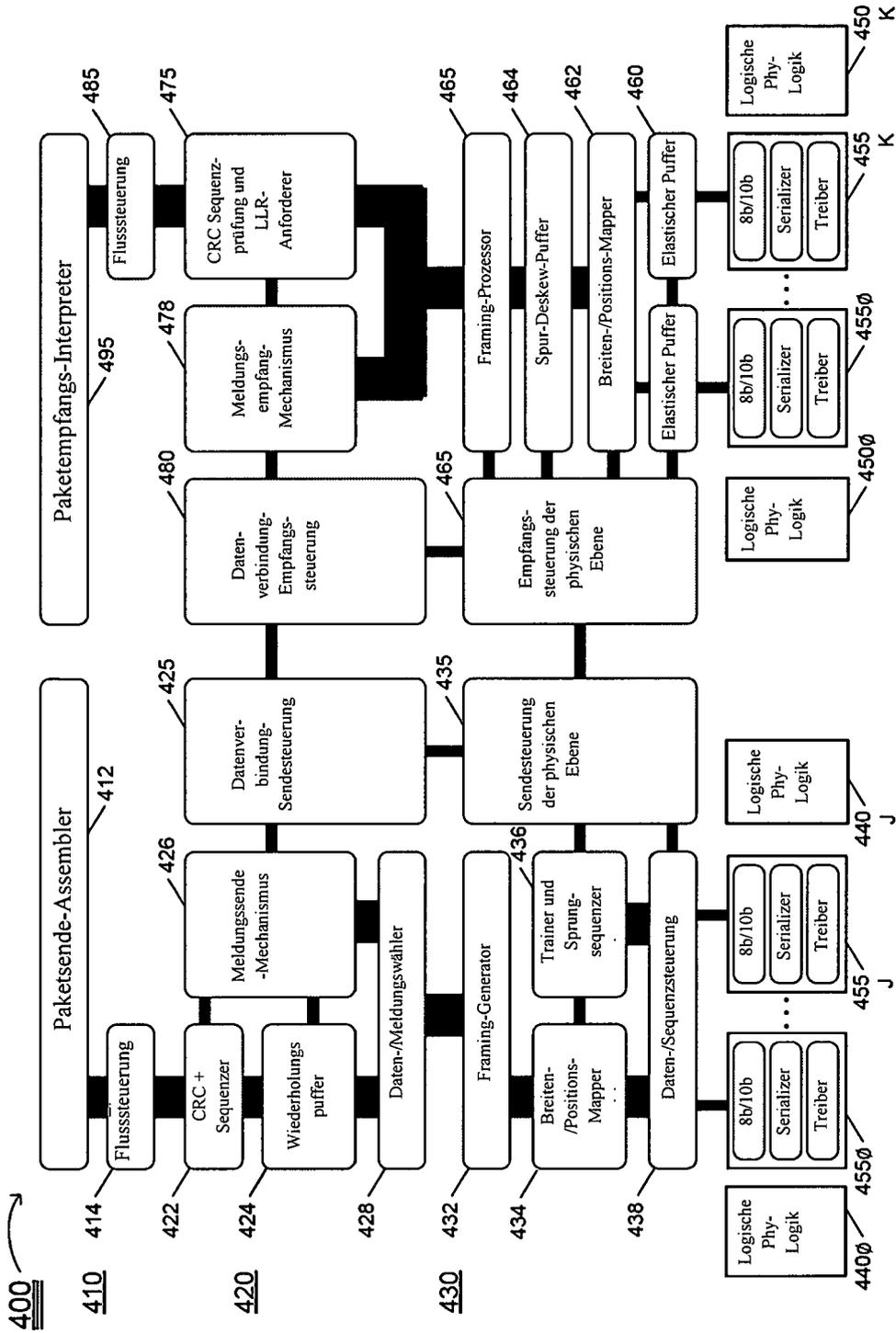


FIG. 4

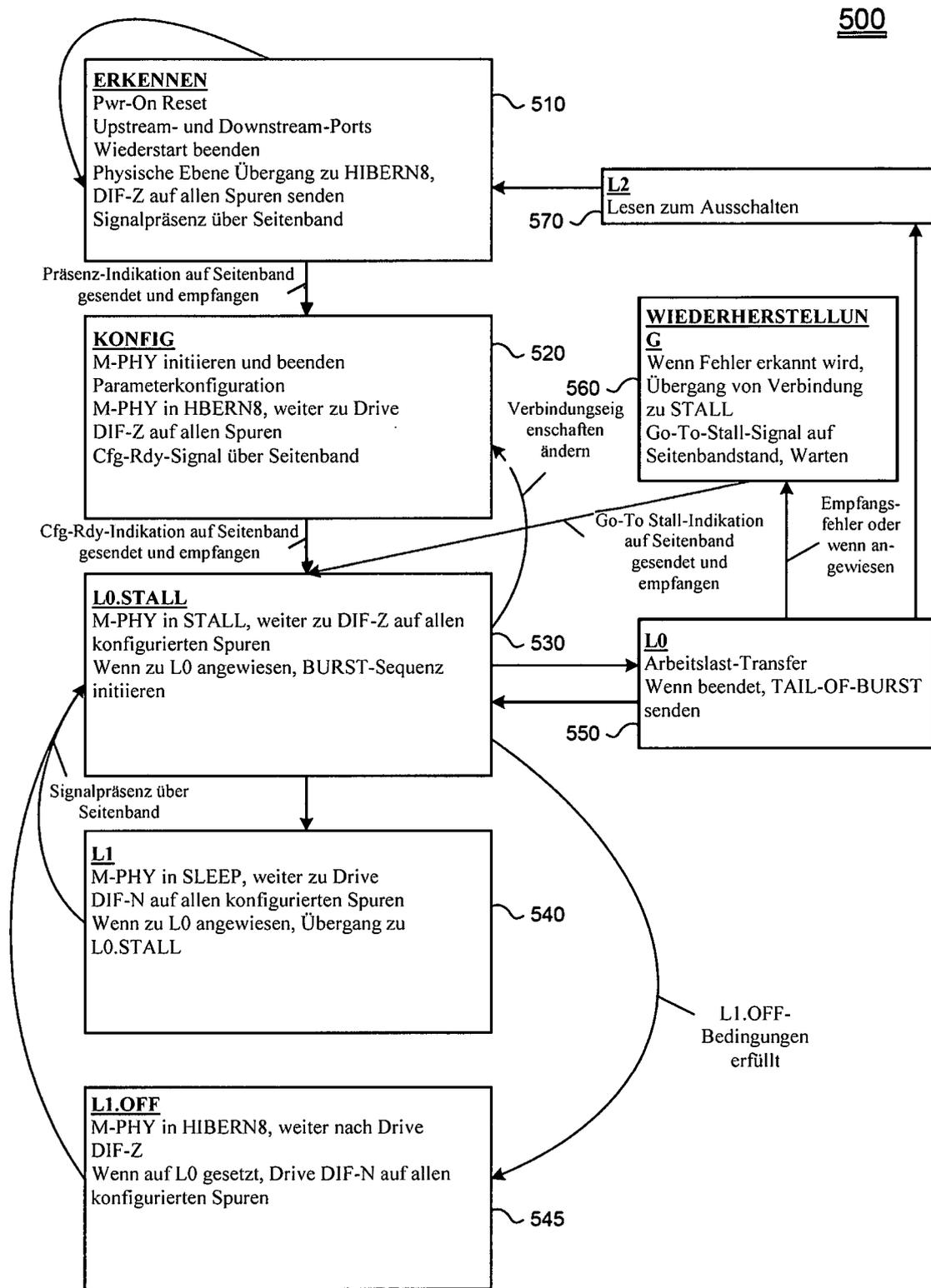


FIG. 5

600

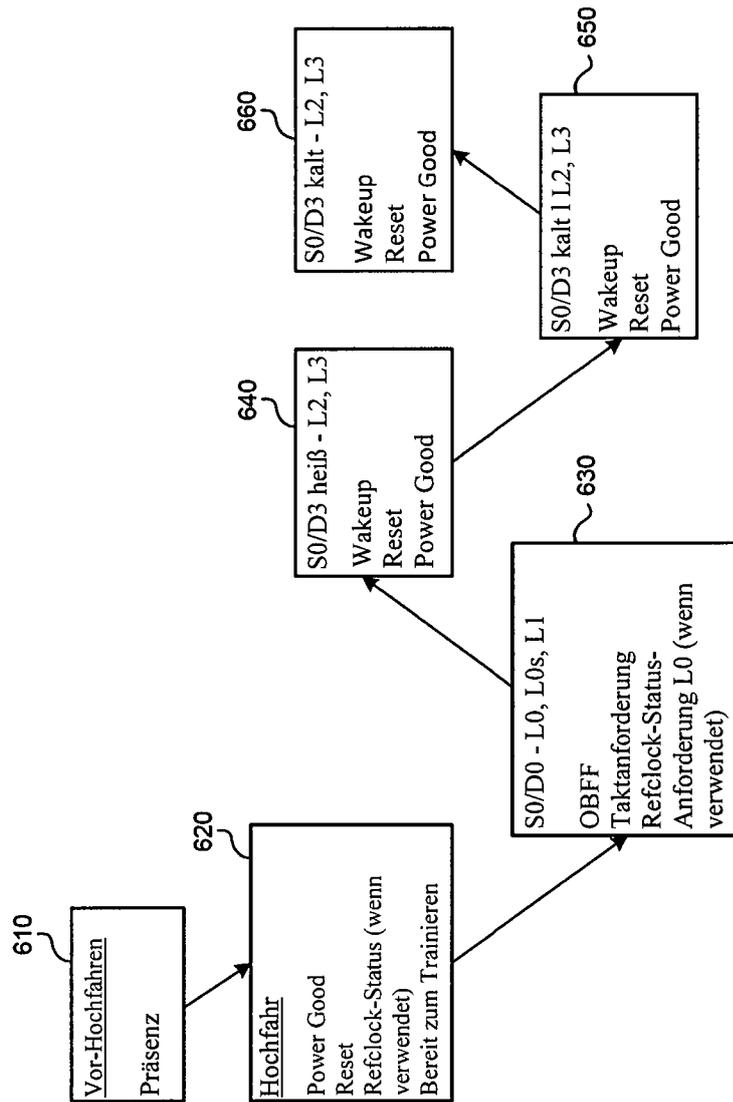


FIG. 6

700

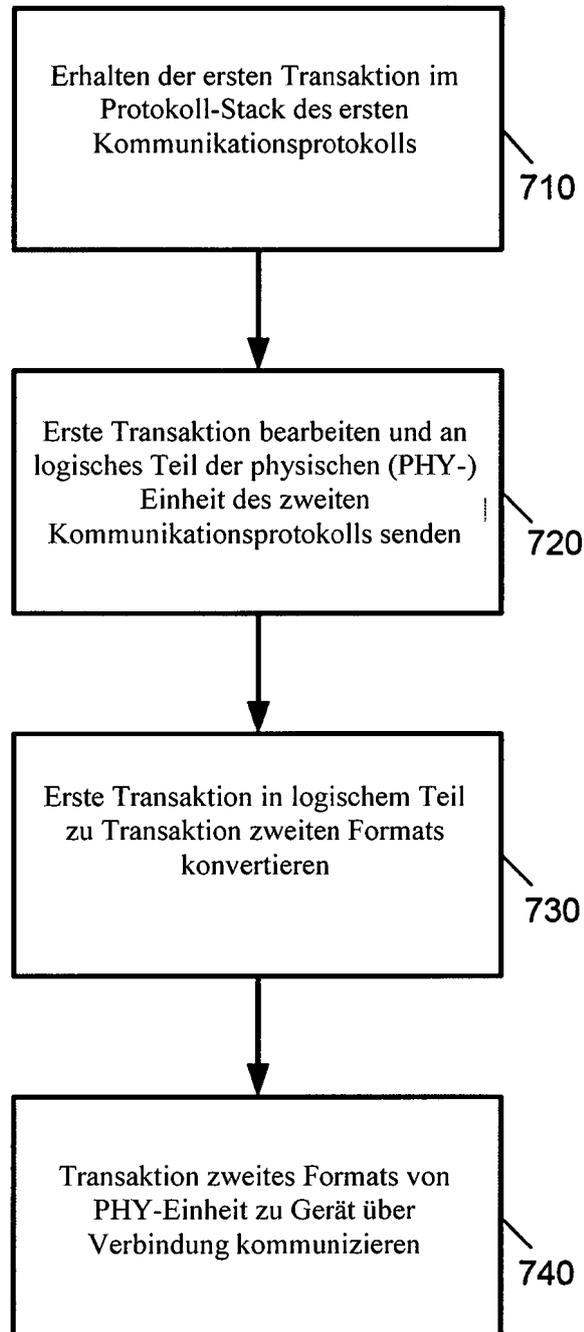


FIG. 7

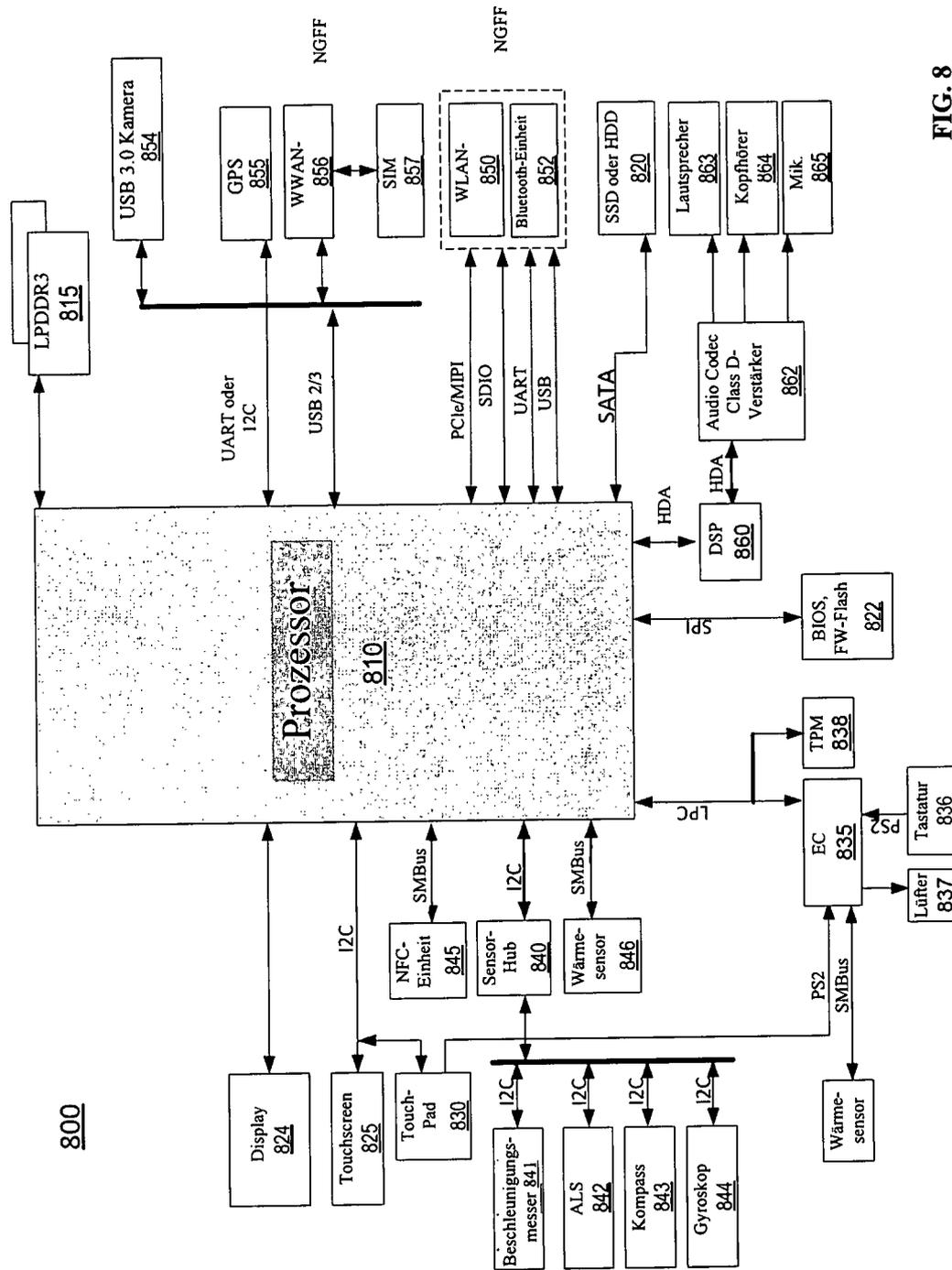


FIG. 8

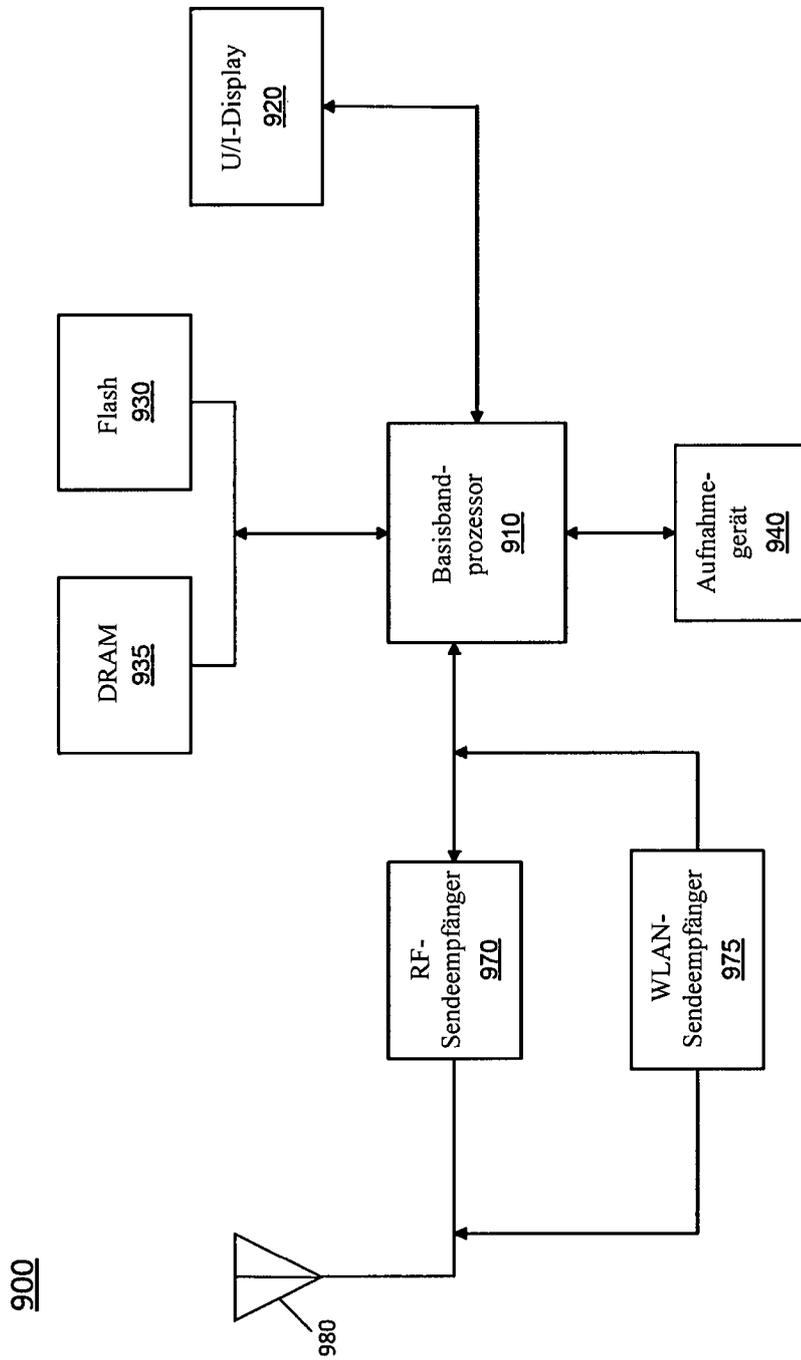


FIG. 9

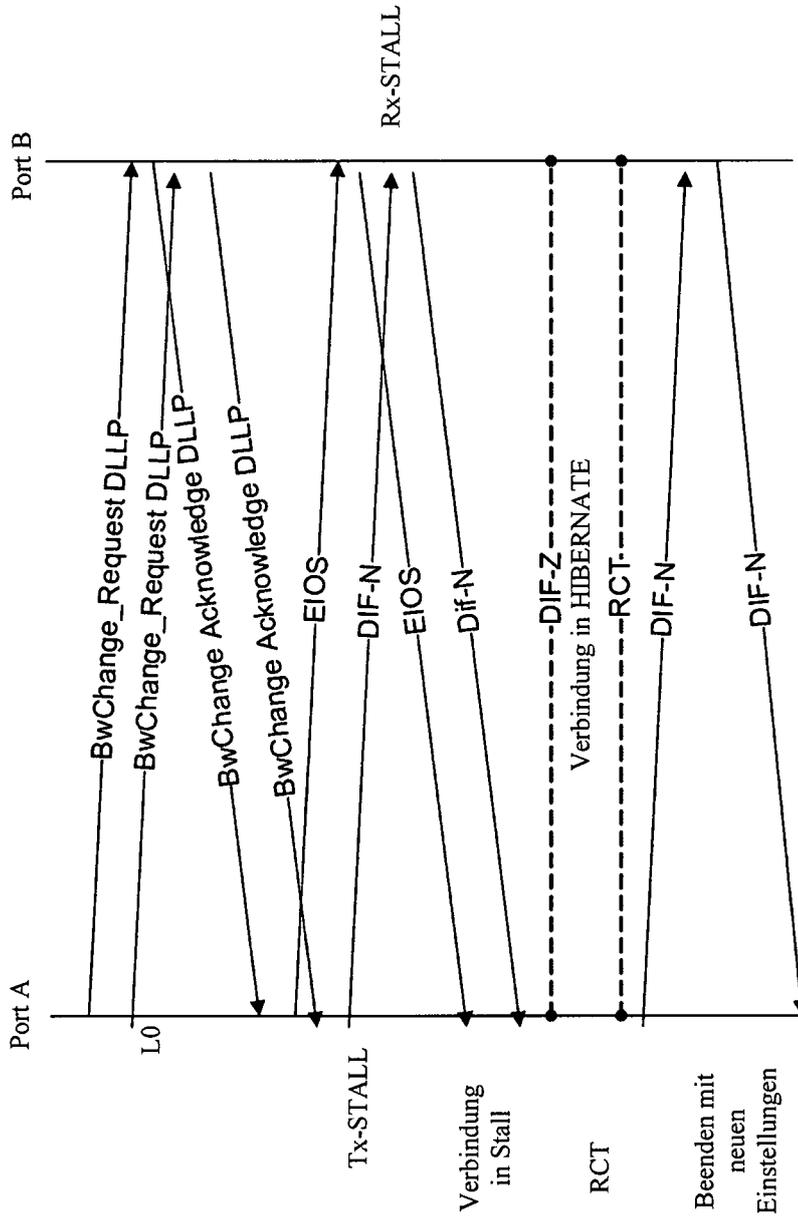


FIG. 10

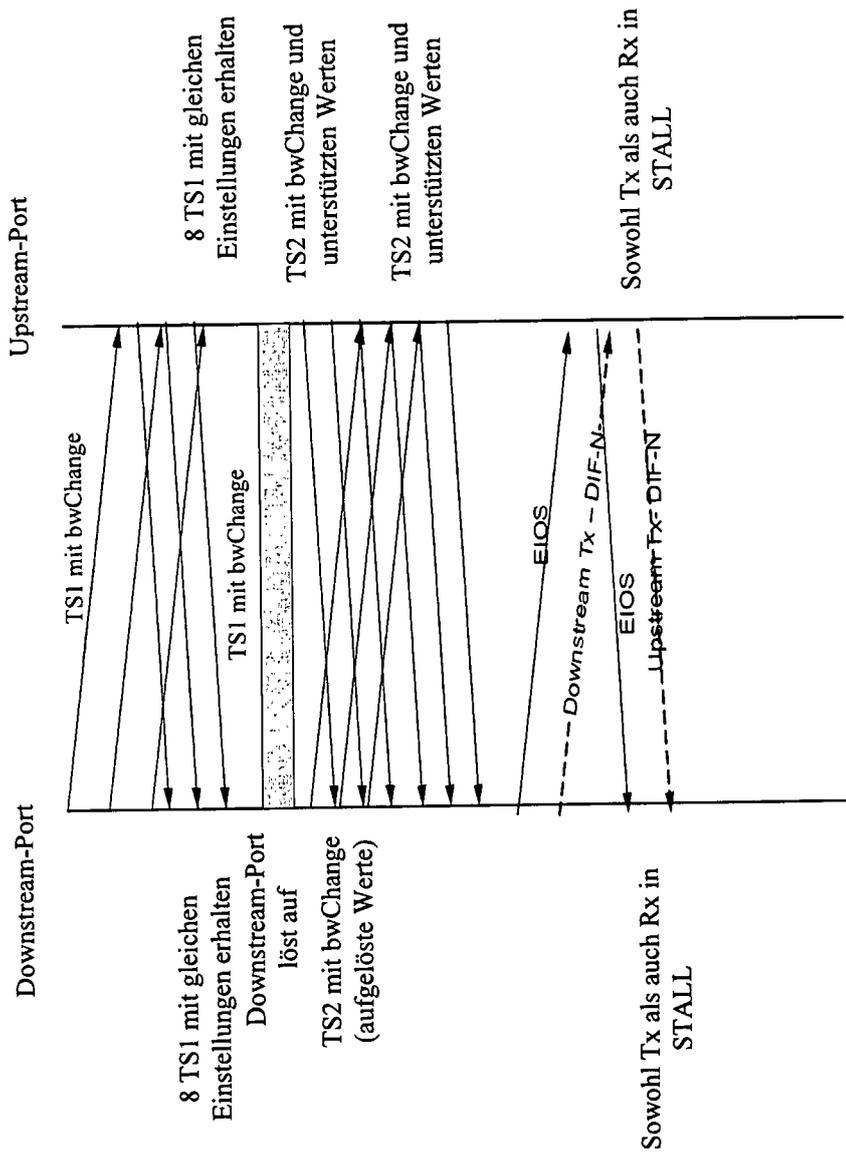


FIG. 11

1200

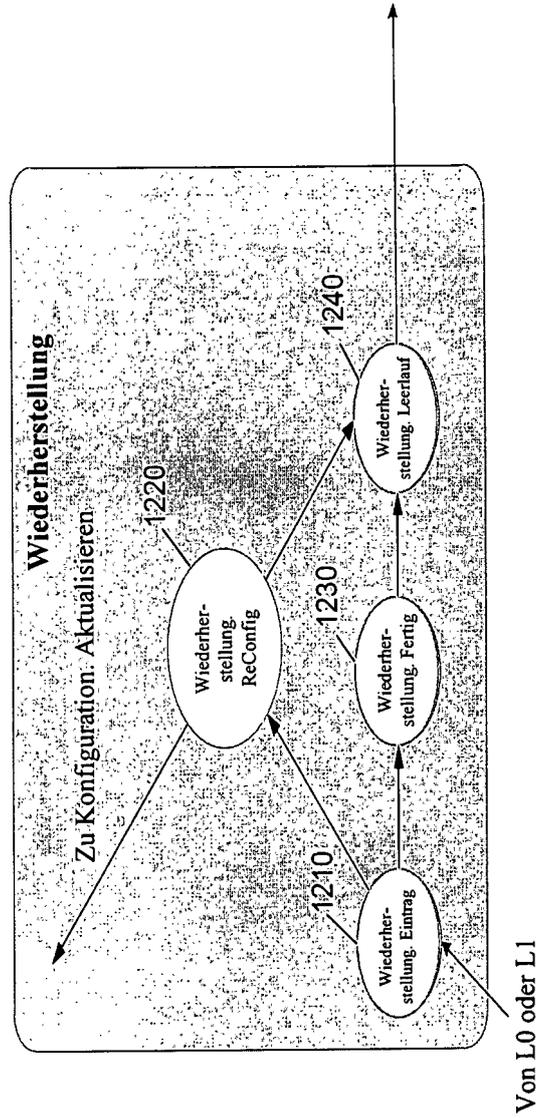


FIG. 12

1300

PCI Express Funktionsregister <u>1305</u>	Nächster Cap-Zeiger <u>1307</u>	PCI Express Cap ID <u>1309</u>
Geräte-Funktionen <u>1310</u>		
Gerätestatus <u>1314</u>	Geräte-Steuerung <u>1312</u>	
Verbindungsfunktionen <u>1315</u>		
Verbindungsstatus <u>1325</u>	Verbindungssteuerung <u>1320</u>	
Slot-Funktionen <u>1330</u>		
Slot-Status <u>1334</u>	Slot-Steuerung <u>1332</u>	
Root-Funktionen <u>1338</u>	Root-Steuerung <u>1336</u>	
Root-Status <u>1340</u>		
Gerätefunktionen 2 <u>1350</u>		
Gerätestatus 2 <u>1354</u>	Geräte-Steuerung 2 <u>1352</u>	
Verbindungsfunktionen 2 <u>1360</u>		
Verbindungsstatus 2 <u>1364</u>	Verbindungssteuerung 2 <u>1362</u>	
Slot-Funktionen 2 <u>1370</u>		
Slot-Status 2 <u>1374</u>	Slot-Steuerung 2 <u>1372</u>	
Mobile Express Kompatibilität und Steuerung <u>1380</u>		

FIG. 13

1400

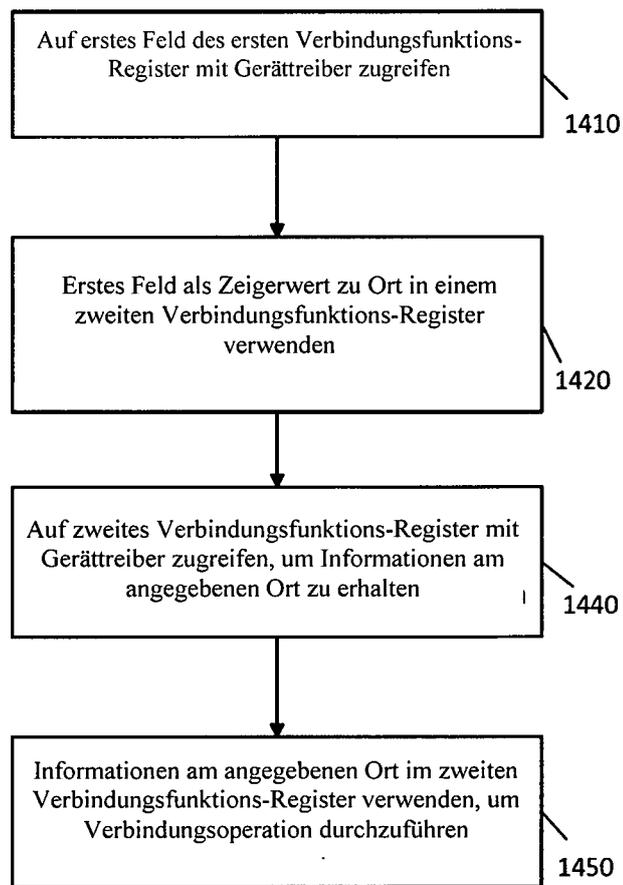


FIG. 14