



(10) **DE 11 2020 004 541 T5** 2022.06.15

(12) **Veröffentlichung**

der internationalen Anmeldung mit der
(87) Veröffentlichungs-Nr.: **WO 2021/061664**
in der deutschen Übersetzung (Art. III § 8 Abs. 2
IntPatÜbkG)
(21) Deutsches Aktenzeichen: **11 2020 004 541.2**
(86) PCT-Aktenzeichen: **PCT/US2020/052031**
(86) PCT-Anmeldetag: **22.09.2020**
(87) PCT-Veröffentlichungstag: **01.04.2021**
(43) Veröffentlichungstag der PCT Anmeldung
in deutscher Übersetzung: **15.06.2022**

(51) Int Cl.: **A63F 13/67 (2014.01)**

A63F 13/422 (2014.01)
G06N 3/02 (2006.01)
G06N 20/00 (2019.01)
G06V 30/18 (2022.01)
G06V 30/19 (2022.01)
G06T 7/20 (2017.01)

(30) Unionspriorität:
16/581,969 **25.09.2019** **US**

(71) Anmelder:
NVIDIA Corporation, Santa Clara, CA, US

(74) Vertreter:
**Kraus & Weisert Patentanwälte PartGmbB, 80539
München, DE**

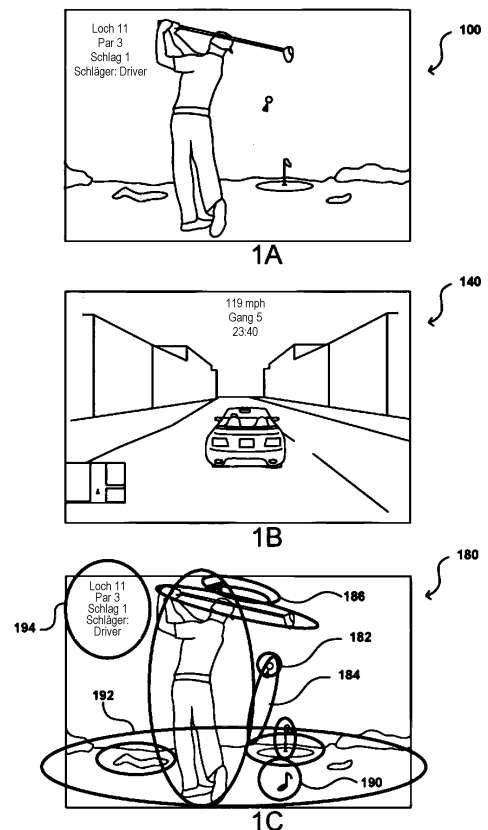
(72) Erfinder:
**Pardeshi, Siddhant, Santa Clara, CA, US; Kothari,
Pranit P., Santa Clara, CA, US; Gaikwad, Vinayak
Vilas, Santa Clara, CA, US**

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **SPIELERANALYSE UNTER VERWENDUNG EINES ODER MEHRERER NEURONALER NETZE**

(57) Zusammenfassung: Es werden Geräte, Systeme und Techniken vorgestellt, um Feedback für einen Benutzer zu bestimmen. In mindestens einer Ausführungsform wird einem oder mehreren Spielern eines Spiels Feedback bereitgestellt, um diesen Spielern zu helfen, ihre Leistung beim Spielen des Spiels zu verbessern.



Beschreibung

QUERVERWEISE AUF VERWANDTE ANWENDUNGEN

[0001] Dies ist eine PCT-Anmeldung und beansprucht Priorität gegenüber US-Anmeldung Nr. 16/581.969 mit dem Titel „SPIELERANALYSE UNTER VERWENDUNG EINES ODER MEHRERER NEURONALER NETZE“, eingereicht am 25. September 2019, von der die vollständige Offenbarung dieser Anmeldung durch Bezugnahme zu allen Zwecken in diese Schrift aufgenommen wird.

TECHNISCHES GEBIET

[0002] Mindestens eine Ausführungsform bezieht sich auf Verarbeitungsressourcen, die zur Durchführung und Ermöglichung künstlicher Intelligenz verwendet werden. Zum Beispiel bezieht sich mindestens eine Ausführungsform auf Prozessoren oder Rechensysteme, die zum Trainieren von neuronalen Netzen gemäß verschiedenen in dieser Schrift beschriebenen neuartigen Techniken verwendet werden.

ALLGEMEINER STAND DER TECHNIK

[0003] Mit einer zunehmenden Anzahl von Videospielen, die für Online-Multiplayer-Sitzungen verfügbar sind, und mit dem Aufkommen von E-Sports, wird Gaming immer wettbewerbsfähiger. Um die Fähigkeiten eines Spielers zu verbessern, hat dieser Spieler begrenzte Möglichkeiten, wie zum Beispiel statische Tutorials zu lesen, Videos von anderen Spielern anzusehen oder professionelle Trainer einzustellen.

KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0004] Verschiedene Ausführungsformen gemäß der vorliegenden Offenbarung werden unter Bezugnahme auf die Zeichnungen beschrieben, in denen Folgendes gilt:

Fig. 1A, Fig. 1B und Fig. 1C veranschaulichen Darstellungen von Spielinhalten, die gemäß mindestens einer Ausführungsform analysiert werden können;

Fig. 2 veranschaulicht Komponenten eines Videoanalysesystems gemäß mindestens einer Ausführungsform;

Fig. 3 veranschaulicht Schlüsselwort und Empfehlungen, die gemäß mindestens einer Ausführungsform in Bezug auf eine Instanz von Spielinhalten erzeugt werden können;

Fig. 4 veranschaulicht einen Prozess zur Empfehlung von Spielinhalten gemäß mindestens einer Ausführungsform;

Fig. 5 veranschaulicht einen Prozess zur Empfehlung von Inhalt gemäß mindestens einer Ausführungsform;

Fig. 6A, Fig. 6B und Fig. 6C veranschaulichen Spielerempfehlungen gemäß mindestens einer Ausführungsform;

Fig. 7 veranschaulicht ein Spielerempfehlungssystem gemäß mindestens einer Ausführungsform;

Fig. 8 veranschaulicht einen Prozess zum Bestimmen von Spielerempfehlungen gemäß mindestens einer Ausführungsform;

Fig. 9A, Fig. 9B, Fig. 9C und Fig. 9D veranschaulichen beispielhafte Gameplayzustände gemäß mindestens einer Ausführungsform;

Fig. 10 veranschaulicht einen Prozess zum Bestimmen von Spielerreaktionszeiten gemäß mindestens einer Ausführungsform;

Fig. 11A und Fig. 11B veranschaulichen beispielhafte Spielerermüdungsempfehlungen gemäß mindestens einer Ausführungsform;

Fig. 12 veranschaulicht einen Prozess zum Bestimmen von Spielerermüdung gemäß mindestens einer Ausführungsform;

Fig. 13 veranschaulicht eine Umgebung gemäß mindestens einer Ausführungsform;

Fig. 14 veranschaulicht ein System zum Trainieren eines Bildsynthese-Netzes, das gemäß mindestens einer Ausführungsform verwendet werden kann;

- Fig. 15** veranschaulicht Schichten eines statistischen Modells, das gemäß mindestens einer Ausführungsform verwendet werden kann;
- Fig. 16A** veranschaulicht Inferenz- und/oder Trainingslogik gemäß mindestens einer Ausführungsform;
- Fig. 16B** veranschaulicht Inferenz- und/oder Trainingslogik gemäß mindestens einer Ausführungsform;
- Fig. 17** veranschaulicht ein beispielhaftes Rechenzentrumssystem gemäß mindestens einer Ausführungsform;
- Fig. 18** veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;
- Fig. 19** veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;
- Fig. 20** veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;
- Fig. 21** veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;
- Fig. 22A** veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;
- Fig. 22B** veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;
- Fig. 22C** veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;
- Fig. 22D** veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;
- Fig. 22E** und **Fig. 22F** veranschaulichen ein gemeinsam genutztes Programmiermodell gemäß mindestens einer Ausführungsform;
- Fig. 23** veranschaulicht beispielhafte integrierte Schaltungen und zugehörige Grafikprozessoren gemäß mindestens einer Ausführungsform;
- Fig. 24A-24B** veranschaulichen beispielhafte integrierte Schaltungen und zugehörige Grafikprozessoren gemäß mindestens einer Ausführungsform;
- Fig. 25A-25B** veranschaulichen zusätzliche beispielhafte Grafikprozessorlogik gemäß mindestens einer Ausführungsform;
- Fig. 26** veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;
- Fig. 27A** veranschaulicht einen Parallelprozessor gemäß mindestens einer Ausführungsform;
- Fig. 27B** veranschaulicht eine Partitionseinheit gemäß mindestens einer Ausführungsform;
- Fig. 27C** veranschaulicht ein Verarbeitungsknoten gemäß mindestens einer Ausführungsform;
- Fig. 27D** veranschaulicht einen Grafik-Multiprozessor gemäß mindestens einer Ausführungsform;
- Fig. 28** veranschaulicht ein System mit mehreren Grafikverarbeitungseinheiten (graphics processing unit- GPU) gemäß mindestens einer Ausführungsform;
- Fig. 29** veranschaulicht einen Grafikprozessor gemäß mindestens einer Ausführungsform;
- Fig. 30** veranschaulicht die Mikroarchitektur eines Prozessors gemäß mindestens einer Ausführungsform;
- Fig. 31** veranschaulicht einen Deep-Learning-Anwendungsprozessorgemäß mindestens einer Ausführungsform;
- Fig. 32** veranschaulicht einen beispielhaften neuromorphen Prozessor gemäß mindestens einer Ausführungsform;
- Fig. 33** und **Fig. 34** veranschaulichen zumindest Abschnitte eines Grafikprozessors gemäß mindestens einer Ausführungsform;
- Fig. 35** veranschaulicht zumindest Abschnitte eines Grafikprozessorkerns gemäß mindestens einer Ausführungsform;
- Fig. 36A-36B** veranschaulicht zumindest Abschnitte eines Grafikprozessorkerns gemäß mindestens einer Ausführungsform;
- Fig. 37** veranschaulicht eine Parallelverarbeitungseinheit (parallel processing unit - „PPU“) gemäß mindestens einer Ausführungsform;

Fig. 38 veranschaulicht ein allgemeines Verarbeitungscluster (general processing cluster - „GPC“) gemäß mindestens einer Ausführungsform;

Fig. 39 veranschaulicht eine Speicherpartitionseinheit einer Parallelverarbeitungseinheit („PPU“) gemäß mindestens einer Ausführungsform; und

Fig. 40 veranschaulicht einen Streaming-Multiprozessor gemäß mindestens einer Ausführungsform.

DETAILLIERTE BESCHREIBUNG

[0005] In mindestens einer Ausführungsform kann ein Benutzer über ein Computergerät auf digitale Inhalte zugreifen. In mindestens einer Ausführungsform können diese digitalen Inhalte auch Spiel-Inhalte beinhalten. In mindestens einer Ausführungsform kann ein Spiel, das ein Benutzer spielt, mehrere verschiedene Gameplayaspekte oder Arten möglicher Interaktionen aufweisen. In mindestens einer Ausführungsform können Interaktionen mit verschiedenen Arten von Spielen, Minispielen, Handlungen oder Gameplaystilen zusammenhängen. In mindestens einer Ausführungsform kann ein Spiel eine primäre Storyline haben, die einer ersten Art von Interaktion entspricht, aber auch andere Minispiele oder Erfahrungen enthalten, die ein Spieler als Teil dieses Spiels spielen kann. In mindestens einer Ausführungsform, wie in einer Gameplayansicht 100 von **Fig. 1A** veranschaulicht, kann ein Spiel einen Abschnitt, eine Ebene oder ein Minispiel aufweisen, das es einem Spieler ermöglicht, Golf zu spielen. In mindestens einer Ausführungsform kann dies auftreten, wenn ein Spieler ein Fahrzeug durch eine virtuelle Stadt fährt und auf einen Golfplatz trifft, wobei ein Spieler einen Avatar aus einem Fahrzeug aussteigen und auf einen Golfplatz gehen lassen kann, was es diesem Spieler dann ermöglicht, Golf zu spielen. In mindestens einer Ausführungsform kann dieser Spieler dann als Teil des Spiels auch mindestens einen Teil des Gameplays damit verbringen, ein Fahrzeug in einer oder mehreren virtuellen Städten herumzufahren, wie in einer zweiten Gameplayansicht 140 von **Fig. 1B** veranschaulicht. In mindestens einer Ausführungsform sind Handlungen wie das Fahren oder Golfen möglicherweise nicht Teil eines primären Gameplaystils für ein Spiel, das sich stattdessen auf ein Abenteuerspiel, ein Strategiespiel, einen Third-Person-Shooter oder einen Ego-Shooter beziehen könnte.

[0006] In mindestens einer Ausführungsform werden Informationen über das Spiel einer dritten Person in einem Profil oder einem anderen mit dem Benutzer verknüpften Speicher gespeichert.

[0007] In mindestens einer Ausführungsform kann eine Einheit, die diesem Benutzer Inhalte empfehlen möchte, auf Informationen in diesem Profil zugreifen und diese Informationen dazu verwenden, verwandte Inhalte zu empfehlen. In mindestens einer Ausführungsform könnte eine Empfehlung auf der Grundlage dieser Art von Spiel, das mit einer primären Art von Gameplay verbunden ist, zu Empfehlungen für andere Third-Person-Spiele im Abenteuerstil führen. In mindestens einer Ausführungsform verbringt ein Benutzer jedoch die meiste Zeit in diesem Spiel mit einem Minispiel oder einer bestimmten Art von Aktivität oder Gameplay. In mindestens einer Ausführungsform, und wie in Bezug auf **Fig. 1A** und **Fig. 1B** diskutiert, könnte dies einem Benutzer entsprechen, der/die meiste Zeit in diesem Spiel damit verbringt, Golf zu spielen oder durch eine virtuelle Stadt zu fahren, anstatt sich an einem Third-Person-Shooter-, Abenteuer- oder Rollenspiel-Gameplay zu beteiligen, das eine primäre Art von Gameplay für ein bestimmtes Spiel sein könnte. In mindestens einer Ausführungsform könnte bestimmt werden, dass Golf- oder Rennspiele dann für die Interessen dieses bestimmten Benutzers relevanter sein können als Third-Person-Abenteuerspiele.

[0008] In mindestens einer Ausführungsform kann ein Gameplay-Analysesystem Gameplay-Daten analysieren, um zu versuchen, Informationen über das Gameplay für einen bestimmten Benutzer zu ermitteln. In mindestens einer Ausführungsform umfasst dies die Analyse von Videos von mindestens Teilen, Segmenten oder Untergruppen einer Spielsitzung, um zu versuchen, Aspekte wie Szenen, Objekte und Handlungen zu identifizieren, die innerhalb eines Spiels auftreten. In mindestens einer Ausführungsform können Schlüsselwörter, die mit diesen Aspekten in Verbindung stehen, erzeugt und dann im Laufe der Zeit aggregiert werden, um einen genaueren Eindruck von den Interessen eines Benutzers zu geben, als dies allein anhand von Daten auf Spielebene möglich wäre. In mindestens einer Ausführungsform können diese Schlüsselwörter auch für andere Zwecke verwendet werden, z. B. um ein Profil für einen Benutzer zu erstellen, das für Matchmaking, Spielerauswahl oder Level-Einstellung verwendet werden kann.

[0009] In mindestens einer Ausführungsform kann ein Gameplay-Video mit einem Gameplay-Analysesystem analysiert werden, um verschiedene Arten von Objekten oder Ereignissen im Gameplay zu identifizieren, die auf eine Szene, ein Objekt oder eine Handlung hindeuten können. In mindestens einer Ausführungsform können Videosegmente, die für das Gameplay repräsentativ sind, analysiert werden. In mindestens einem Ausführungsbeispiel können tatsächliche Gameplay- oder Interaktionsdaten während des Streamings analysiert

oder zur Anzeige bereitgestellt werden. In mindestens einer Ausführungsform kann, wie in einer Spielsicht 180 von **Fig. 1C** veranschaulicht, ein Gameplay-Analyse-System (oder -service) Gameplayaudio und -video analysieren, um zu versuchen, verschiedene Arten von Objekten, die für dieses Spiel definiert wurden, zu bestimmen. In mindestens einer Ausführungsform, könnte die Ansicht 180 von **Fig. 1C** analysiert werden, um Objekte wie einen Golfball 182, Golfschläger, Sandbunker und Flagge zu identifizieren, die mit einem Golfspiel in Verbindung stehen. In mindestens einer Ausführungsform könnte Ansicht 180 analysiert werden, um Handlungen zu identifizieren, wie das Schwingen eines Golfschlägers in einer Golfbewegung 186, das Schlagen eines Golfballs und das Bewirken, dass sich ein Golfball in Richtung eines Lochs mit einer Flagge bewegt 184, was ein Hinweis darauf sein kann, Golf zu spielen oder zumindest einen Ball zu schlagen. In mindestens einer Ausführung können Objekte wie Wasserhindernisse, Sandbunker, Fairways, Grüns und Flaggen auf eine Szene hinweisen, die einem Golfplatz entspricht. In mindestens einer Ausführungsform kann ein erzeugter Klang 190, der einem Golfschläger entspricht, der einen Golfball trifft, auch verwendet werden, um eine spezifische Szene oder Handlung zu identifizieren, die Golf entspricht. In mindestens einer Ausführungsform können diese Elemente identifiziert und verwendet werden, um eine Reihe von Schlüsselwörtern zu erzeugen, die für eine Art von Gameplay über einen entsprechenden Teil oder ein Segment des Gameplays stehen. In mindestens einer Ausführungsform können Schlüsselwörter erzeugt werden, die einer Szene „Golfplatz“, Objekten „Golfschläger“ oder „Golfball“ und Handlungen „Golf spielen“ und „Schlagen von Bällen“ entsprechen können. In mindestens einer Ausführungsform können diese Schlüsselwörter verwendet werden, um ein Profil eines Spielers zu erzeugen oder zu aktualisieren, welches verwendet werden kann, um Inhalte zu empfehlen, die für diesen bestimmten Spieler von Interesse sein könnten.

[0010] In mindestens einer Ausführungsform können auf Computervision und maschinellem Lernen basierende Techniken verwendet werden, um Spielinhalte, wie z. B. Gameplay-Video-Highlights, zu verarbeiten und Spielempfehlungen zu erzeugen. In mindestens einer Ausführungsform können Spielinhalte analysiert werden, um bestimmte Arten von Funktionen in einer Szene zu erkennen, wie z. B. Szenen, in denen Gameplay auftritt, in einer Spielsitzung erkannte Objekte, die sich auf Gameplay beziehen, und Handlungen, die von einem Spieler (oder einem Avatar oder einem vom Spieler gesteuerten Gameplay-Element) während einer oder mehrerer Spielsitzungen ausgeführt werden. In mindestens einer Ausführungsform können ein oder mehrere Gameplay-Segmente für eine Spielszene analysiert werden, und ein trainiertes neuronales Netzwerkmodell kann eine Reihe von Schlüsselwörtern erzeugen, die die für diese Spielszene bestimmten Funktionen repräsentieren. In mindestens einer Ausführungsform können diese Schlüsselwörter aggregiert und an eine Empfehlungsmaschine weitergeleitet werden. In mindestens einer Ausführungsform kann eine Empfehlungsmaschine diesen Schlüsselwörtern Gewichtungen zuweisen und diese in eine Liste wahrscheinlicher Spiele übersetzen, die zumindest teilweise auf der Schlüsselwort-Aggregation aus einem Empfehlungsprofil für diesen Benutzer (oder Spieler) basiert.

[0011] In mindestens einer Ausführungsform kann ein Inhalt-Empfehlungssystem 200 verwendet werden, wie in **Fig. 2** veranschaulicht. In mindestens einer Ausführungsform werden Spielinhalte auf einem Streaming-Server 208 empfangen. In mindestens einer Ausführungsform kann der Streaming-Server 208 einen Gameplay-Videostream als Eingabe akzeptieren, das von einer Spielekonsole, einem Spielecomputer oder einem Spielehost bereitgestellt wird. In mindestens einer Ausführungsform kann ein empfangener Gameplay-Videostream aus zufällig ausgewählten oder spezifisch ausgewählten Highlights eines Gameplay-Videostreams eines bestimmten Spielers bestehen. In mindestens einer Ausführungsform kann der Streaming-Server 208 ein oder mehrere trainierte künstliche neuronale Netzwerke (artificial neural networks - ANNs) nutzen, um Funktionen wie Szene, Objekte und Handlungen aus einzelnen Szenen des Eingabe-Gameplay-Videos zu erkennen, um eine Sammlung von Schlüsselwörtern zu erzeugen, die bestimmten Kategorien entsprechen, die für ein bestimmtes Spiel bestimmt wurden. In mindestens einer Ausführungsform ist eine trainierte ANN ein Netzwerk, das aus einem ResNet (Restneuronales Netzwerk) oder einem anderen solchen Netzwerk oder dessen Ableitung abgeleitet wurde. In mindestens einer Ausführungsform kann eine Szene einer Umgebung entsprechen, in der Gameplay auftritt, z. B. ein/e Stadtbild, Wüste, Meer, Dschungel, Büro, Straße, Gebäude, Berg, Gletscher, Arena, Stadion, Rennstrecke oder fremde Welt. In mindestens einer Ausführungsform kann ein Objekt einem grafischen Element entsprechen, das während einer Gameplay-Sitzung angezeigt oder auf andere Weise präsentiert oder angetroffen wird, wie z. B. ein/e Waffe, Buch, Baum, Auto, Container, Treppenhaus, Zug, Schreibtisch oder Avatar-Sammlerobjekt. In mindestens einer Ausführungsform kann eine erfassbare Handlung eine Handlung eines Avatars oder eines vom Spieler steuerbaren Spielelements beinhalten, wie z. B. Laufen, Springen, Kämpfen, Schwingen, Fahren, Sport treiben, Schießen, Fahren, Fliegen, Schwimmen, Klettern, Reiten oder Tanzen. In mindestens einer Ausführungsform kann die Erfassung solcher Merkmale in einem empfangenen Videostream verwendet werden, um eine Reihe von Schlüsselwörtern, Phrasen oder anderen Indikatoren zu erzeugen, die für diese Merkmale repräsentativ sind. In mindestens einer Ausführungsform entsprechen die erzeugten Schlüsselwörter dann eher den Hand-

lungen und Entscheidungen, die ein Spieler während einer Gameplay-Szene getroffen hat, als einer allgemeinen Reihe von Schlüsselwörtern, die mit diesem Spiel als Ganzes verbunden sind.

[0012] In mindestens einer Ausführungsform werden diese Schlüsselwörter als Eingabe an einen Empfehlungsserver 202 weitergeleitet und zur weiteren Verarbeitung an eine Empfehlungsmaschine 206 weitergeleitet. In mindestens einer Ausführungsform ist die Empfehlungsmaschine 206 so programmiert, dass sie den empfangenen Schlüsselwörtern Gewichte zuweist. In mindestens einer Ausführungsform werden diese Gewichtungen zumindest teilweise auf der Grundlage von Faktoren wie Kategorie, Ereignishäufigkeit und Dauer bestimmt. In mindestens einer Ausführungsform ist die Empfehlungsmaschine 206 auch so programmiert, dass sie nach Spiel erzeugte Schlüsselwörter erzeugt und eine Nachschlagetabelle 204 für die Zuordnung von Spielen zu Schlüsselwörtern sowie von Schlüsselwörtern zu Spielen verwaltet. In mindestens einer Ausführungsform hätte jedes Spiel nach Ebene eine Reihe von Schlüsselwörtern, die mögliche Ausgaben eines trainierten Modells sind. In mindestens einer Ausführungsform können basierend auf der Gewichtung dieser Schlüsselwörter aus dieser Nachschlagetabelle eine oder mehrere Empfehlungen für Spiele ermittelt werden, die mit diesen Schlüsselwörtern verknüpft sind. In mindestens einer Ausführungsform würde jedes Schlüsselwort mit einem Konfidenzwert von über 75 % mit einem entsprechenden Spiel in der Nachschlagetabelle 204 assoziiert werden. In mindestens einer Ausführungsform können andere Konfidenzschwellenwerte festgelegt werden, wie beispielsweise mindestens 50 % oder mindestens 90 %, was von verschiedenen Faktoren abhängen kann.

[0013] In mindestens einer Ausführungsform kann Empfehlungsserver 202 ein oder mehrere trainierte neuronale Netze 216 nutzen. In mindestens einer Ausführungsform kann ein neuronales Netz trainiert werden, um eine Eingabe von Schlüsselwörtern einem Empfehlungsprofil eines Benutzers zuzuordnen, wo diese Schlüsselwörter durch ein Spiel abgebildet werden. In mindestens einer Ausführungsform kann eine Testphase verwendet werden, um eine Liste empfohlener Spiele für einen Benutzer zu erstellen, die eine Eingabe von Schlüsselwörtern enthält, basierend auf Informationen wie einer Gameplay-Sitzung und einem zusammengefassten Profil.

[0014] In mindestens einer Ausführungsform kommuniziert der Streaming-Server 208 auch mit mindestens einem Streaming-Client 212. In mindestens einer Ausführungsform kann der Streaming-Client 212 eine Streaming-Media-Anwendung ausführen, die einen Videostream 210 und einen Audio-Stream (oder einen Medien-Stream, der beides enthält) empfangen und diesen über eine Schnittstelle des Streaming-Clients 212 streamen kann. In mindestens einer Ausführungsform werden die Video-Feeds eines Spieles an den Empfehlungsserver 202 weitergeleitet, um ein Empfehlungsprofil für dieses bestimmte Spiel zu trainieren. In mindestens einer Ausführungsform werden diese Video-Feeds verarbeitet und die Ergebnisse dieser Verarbeitung werden dem Empfehlungsserver 202 zur Verfügung gestellt. In mindestens einer Ausführungsform handelt es sich bei diesem Training um einen serverseitigen Prozess, um zumindest die Rechenleistung zu optimieren. In mindestens einer Ausführungsform kann der Streaming-Client 212 mit dem Empfehlungsserver 202 kommunizieren, z. B. über eine Media-Anwendung oder ein dediziertes Software-Entwicklungskit. In mindestens einer Ausführungsform kann der Streaming-Client 212 mit Gewichtungen eines trainierten neuronalen Netzes ausgeliefert werden, das in der Lage ist, Szenen-, Video- und Objektfunktionen in einem vom Client gerenderten Videostream des Spiels zu erkennen, der zur weiteren Verarbeitung an einen Empfehlungsserver gesendet werden kann. In mindestens einer Ausführungsform empfängt oder lädt der Streaming-Client 212 Empfehlungen vom Empfehlungsserver 202 herunter und stellt mindestens einige dieser Empfehlungen über eine Benutzeroberfläche (user interface - UI) auf einem Streaming-Client dar, wie zum Beispiel über eine Anwendung, mit der Empfehlungsinhalte über einen Bildschirm und einen oder mehrere Lautsprecher des Streaming-Clients 212 dargestellt werden können.

[0015] In mindestens einer Ausführungsform können mehrere trainierte neuronale Netze eingesetzt werden. In mindestens einer Ausführungsform kann ein erstes neuronales Netzwerkmodell 214 trainiert werden, um präzise Schlüsselwörter für die Kategorien von Funktionen im Gameplay zu erzeugen, z. B. für ausgewählte Gameplay-Szenen. In mindestens einer Ausführungsform können Trainingsdaten für dieses neuronale Netzwerkmodell eine manuelle Kennzeichnung oder Modellierung in mindestens einer ersten Trainingsphase erfordern. In mindestens einer Ausführungsform kann ein zweites neuronales Netzwerkmodell trainiert werden, um ein Empfehlungsprofil für einen Spieler zu aktualisieren. In mindestens einer Ausführungsform kann dieses zweite neuronale Netzwerkmodell 216, während ein Spieler verschiedene Spiele spielt, Schlussfolgerungen erzeugen, die verwendet werden können, um das Empfehlungsprofil des Spielers im Laufe der Zeit für die Daten zu aktualisieren, die aus verschiedenen Gameplay-Sitzungen gesammelt wurden. In mindestens einer Ausführungsform kann das neuronale Netz 216 Schlüsselwörter für ein Gameplay-Segment analysieren, um Themenmodelle durchzuführen, oder um ein Thema oder eine Art von Gameplay, das in erster

Linie während dieses Segments durchgeführt wird, zu bestimmen, wie z. B. ein Golfthema basierend auf Merkmalen, die in Bezug auf eine Ansicht von **Fig. 1C** bestimmt werden. In mindestens einer Ausführungsform können die für ein oder mehrere dieser Modelle aktualisierten Gewichtungen auch auf den Streaming-Client 212 übertragen werden, so dass auf dem Streaming-Client 212 mindestens eine gewisse Anzahl von Schlussfolgerungen durchgeführt werden kann. In mindestens einer Ausführungsform können dem Streaming-Client 212 Modelle und Gewichtungen für Spiele bereitgestellt werden, die ein korrespondierender Spieler installiert oder spielt, so dass auf Streaming-Client 212 mindestens ein gewisses Maß an Funktionsidentifikation und/oder Profilaktualisierung durchgeführt werden kann. In mindestens einer Ausführungsform können diese Modelle nach dem Training für die clientseitige Echtzeitanalyse eingesetzt werden.

[0016] In mindestens einer Ausführungsform sind Streaming-Server 208, Streaming-Client 212 und Empfehlungsserver 202 separate Geräte oder Systeme, die von verschiedenen Einheiten an verschiedenen Standorten betrieben werden und über mindestens ein Netzwerk, wie z. B. ein Mobilfunknetz, ein lokales Netzwerk oder das Internet, zugänglich sind. In mindestens einer Ausführungsform können der Streaming-Server 214 und der Empfehlungsserver 202 von einer einzigen Einheit in einer Umgebung eines Ressourcenanbieters betrieben werden oder von verschiedenen Einheiten betrieben werden, die von einem einzigen Ressourcenanbieter bereitgestellt werden. In mindestens einer Ausführungsform kann Streaming-Client 212 jedes geeignete Gerät sein, das Empfehlungen und/oder Spielinhalte präsentieren kann, wie z. B. ein Smartphone, Tablet-Computer, Desktop-Computer, Notebook, Spielekonsole, Set-Top-Box, Smart-TV oder ein anderes Computergerät. In mindestens einer Ausführungsform können Medieninhalte über verschiedene Mechanismen oder Kanäle bereitgestellt werden, beispielsweise durch Download oder andere Übertragung als Streaming.

[0017] In mindestens einer Ausführungsform können auf der Grundlage unterschiedlicher Merkmale, die in einem Spiel erfasst wurden, unterschiedliche Schlüsselwörter erzeugt werden, wie in der Ansicht 300 von **Fig. 3** veranschaulicht ist. In mindestens einer Ausführungsform beziehen sich die für eine erste Gameplay-Sitzung 302 erfassten Schlüsselwörter auf das Golfspiel und umfassen Schlüsselwörter wie Golfplatz, Golfball, Golfschwung und Golfschläger, was dazu führen kann, dass Empfehlungen für Golfspiele erzeugt werden. In mindestens einer Ausführungsform beziehen sich die für eine zweite Gameplay-Sitzung 304 erfassten Schlüsselwörter auf das Fahr-Gameplay und umfassen Schlüsselwörter wie Stadtbild, Fahrzeug, Fahren und Lenkrad, was dazu führen kann, dass Empfehlungen erzeugt werden, die das Fahren in der Stadt betreffen. In mindestens einer Ausführungsform werden keine Schlüsselwörter für das Gameplay erzeugt, sondern es werden Empfehlungen basierend auf einem jeweiligen Spiel als Ganzes abgegeben, was dazu führen kann, dass Empfehlungen erzeugt werden, die Abenteuerspiele sein können, die mehrere Arten von Handlungen oder Gameplay beinhalten, von denen viele für einen Benutzer möglicherweise nicht von Interesse sind. In mindestens einer Ausführungsform werden Empfehlungen für einzelne Funktionen veranschaulicht, die für einen Spieler auf der Grundlage der von ihm in einem Spiel durchgeführten Handlungen wesentlich relevanter sind als Empfehlungen, die auf einem Spiel als Ganzes basieren.

[0018] In mindestens einer Ausführungsform kann ein trainiertes neuronales Netz, das zur Aktualisierung eines Profils verwendet wird, erzeugte Schlüsselwörter als Eingabe verwenden und Schlüsselwörter, Themen oder andere Informationen bestimmen, die zur Aktualisierung eines Spielerprofils verwendet werden sollen. In mindestens einer Ausführungsform kann ein Schlüsselwort, das einen Golfplatz als Szene für eine Spielsitzung angibt, darauf hindeuten, dass ein Spieler Golf spielt, insbesondere in Kombination mit Handlungen wie dem Schwingen eines Schlägers, um einen Ball zu treffen, und Objekten wie einem Golfball, Golfschläger und einem Putting Green. In mindestens einer Ausführungsform könnte ein neuronales Netz diese Informationen stattdessen verwenden, um zu bestimmen, dass dieser Benutzer gerne Golf spielt, anstatt diese einzelnen Schlüsselwörter in einem Profil zu speichern, und kann ein entsprechendes Profil mit Informationen aktualisieren, die auf das Golf-Gameplay hinweisen, und nicht mit einzelnen Schlüsselwörtern, die für diese Entscheidung verwendet werden. In mindestens einer Ausführungsform können auch zugehörige Informationen wie Anzahl der Ereignisse, Häufigkeit, Zeitdauer oder Prozentsatz des Gameplays gespeichert werden, die dem Golf-Gameplay in einem Spiel gewidmet sind. In mindestens einer Ausführungsform können mehrere aufeinander folgende Gameplay-Bilder von einem trainierten Netzwerkmodell analysiert werden, um Handlungen oder Kombinationen von Handlungen für ein entsprechendes Segment zu bestimmen. In mindestens einer Ausführungsform kann ein Modell den resultierenden Schlüsselwörtern Gewichtungen zuweisen, die einem Konfidenzgrad oder einer Bedeutung in einer Szene entsprechen können.

[0019] In mindestens einer Ausführungsform kann eine Empfehlungsmaschine auch andere Informationen enthalten, wie Spielerhistorie, Käufe, Interessen, Wunschliste und Ausdrücke. In mindestens einer Ausführungsform können auch Empfehlungen auf Spielebene verwendet werden. In mindestens einer Ausführungs-

form können diese und andere Faktoren, die sich auf das Benutzerinteresse beziehen, mit erzeugten Schlüsselwörtern bei dem Bestimmen von Empfehlungen für einen Spieler verwendet werden. In mindestens einer Ausführungsform kann eine Empfehlungsmaschine diese Faktoren berücksichtigen, um mit einem erheblichen Maß an Konfidenz darauf zu schließen, was ein Benutzer versucht zu tun und wo er sich in einem Spiel befindet. In mindestens einer Ausführungsform kann ein Empfehlungssystem bestimmen, dass Golf-bezogene Spiele empfohlen werden sollten, wenn die höchste Wahrscheinlichkeit besteht, dass ein Spieler Golf spielt. In mindestens einer Ausführungsform können diese Empfehlungen zunehmen, wenn ein Spieler im Laufe der Zeit mehr Golf spielt, da die Gewichtung oder der Einfluss des Golfs auf die Empfehlungen zunimmt. In mindestens einer Ausführungsform können Empfehlungen im Laufe der Zeit abgestimmt oder aktualisiert werden. In mindestens einer Ausführungsform können die Empfehlungen spezifisch für eine aktuelle Gameplay-Sitzung, eine kürzliche Gameplay-Sitzung, eine Reihe von kürzlichen Gameplay-Sitzungen oder alle relevanten Gameplay-Sitzungen sein, die dazu führen können, dass ein Spieler Empfehlungen basierend auf aktuellen oder kürzlichen Handlungen, oder aggregierten Handlungen über einen oder mehrere Zeiträumen, erhält.

[0020] In mindestens einer Ausführungsform wird pro Spiel mindestens ein neuronales Netz trainiert. In mindestens einer Ausführungsform wird pro Spiel eine Reihe neuronaler Netze trainiert, wobei verschiedene Netze trainiert werden, um verschiedene Arten von Merkmalen wie Szenen, Handlungen oder Objekte zu erkennen. In mindestens einer Ausführungsform kann ein Netz trainiert werden, das für Inferenzen über eine Vielzahl von Spielen oder zumindest über Spiele einer bestimmten Art oder einer bestimmten Kategorie mit zumindest einigermaßen ähnlichem Gameplay verwendet werden kann. In mindestens einer Ausführungsform könnte ein erstes Modell so ausgebildet sein, dass es Merkmale einer Art von Spiel wie einen Ego-Shooter erkennt, während ein anderes Modell so ausgebildet sein könnte, dass es Merkmale einer Art von Spiel wie ein Jump'n'Run- oder Third-Person-Abenteuerspiel erfasst, da es verschiedene Arten von Funktionen zu erfassen gäbe. In mindestens einer Ausführungsform können die zu erfassenden Funktionen je nach Spiel oder Art von Spiel variieren. In mindestens einer Ausführungsform können Trainingsdaten für diese Modelle Videostreams enthalten, einschließlich Anmerkungen zu Merkmalen von Typen, die für dieses Spiel oder diese Art von Spiel erkannt werden. In mindestens einer Ausführungsform werden diese Anmerkungen manuell oder mit Hilfe der Modellierung durchgeführt. In mindestens einer Ausführungsform kann ein Modell so konfiguriert werden, dass ein oder mehrere erfasste Merkmals-Schlüsselwörter mit entsprechenden Konfidenzwerten ausgegeben werden, und Schlüsselwörter mit höheren Konfidenzwerten oder Werten, die mindestens ein minimales Konfidenzkriterium erfüllen, können zur Aktualisierung eines Spielerprofils oder zur Erzeugung von Empfehlungen verwendet werden.

[0021] In mindestens einer Ausführungsform kann ein trainiertes Modell zur Inferenzzeit zehn bis dreißig Sekunden Medieneingabe als Eingabe akzeptieren, um die in Medieninhalten dargestellten Merkmale zu bestimmen. In mindestens einer Ausführungsform wird ein Stream von Videoinhalten für eine Spielsitzung zur Analyse in mehrere 30-Sekunden-Segmente unterteilt. In mindestens einer Ausführungsform können ein oder mehrere Hooks für bestimmte Arten von Handlungen in ein Spiel programmiert werden, sodass Videosegmente einschließlich dieser Hooks analysiert werden, anstatt einen gesamten Videostream zu analysieren. In mindestens einem Ausführungsbeispiel können automatische Algorithmen zur Erzeugung von Hervorhebung verwendet werden, um diese Hooks zu identifizieren und entsprechende Videosegmente für die Analyse zu erzeugen oder zu benennen. In mindestens einer Ausführungsform kann die Analyse nur von Hervorhebungen in einem Videostream dazu beitragen, unbedeutende oder redundante Teile herauszufiltern, z. B. Teile eines Spiels, in denen ein Spieler verfügbare Minispiele ansieht und dann ein Spiel auswählen kann, das für einen Benutzer von Interesse ist. In mindestens einer Ausführungsform kann eine Hervorhebung Merkmale einer resultierenden Auswahl anstelle eines Prozesses zur Bestimmung der Auswahl enthalten. In mindestens einem Ausführungsbeispiel können Informationen für Minispiele gespeichert werden, die nicht ausgewählt wurden, um ein Spielerprofil mit Informationen über Inhalte oder Gameplay zu aktualisieren, die für einen Spieler möglicherweise nicht von Interesse sind, oder die ist vielleicht weniger interessant als andere Arten von Gameplay sind. In mindestens einer Ausführungsform können diese Informationen mit Informationen über die Menge oder Häufigkeit, mit der ein Benutzer bestimmte Arten von Spielen ausübt, kombiniert werden, um die relative Gewichtung dieser Arten von Spielen für einen Spieler zu bestimmen. In mindestens einer Ausführungsform kann auch die Zeit, die für eine aktuelle Art von Gameplay aufgewendet wurde, mit der durchschnittlichen Zeit verglichen werden, die für diese Art von Gameplay für dieses Spiel aufgewendet wurde. Bei mindestens einer Ausführungsform kann bestimmt werden, dass Spieler in der Regel etwa 15 % der Zeit in einem bestimmten Spiel verbringen, um ein Minispiel wie Golf zu spielen. In mindestens einer Ausführungsform kann, wenn bestimmt wird, dass ein Spieler in diesem Spiel deutlich mehr Zeit, z. B. 30 % oder mehr, mit Golfspielen verbracht hat, bestimmt werden, dass diesem Spieler Golfspiele wahrscheinlich gefallen. In mindestens einer Ausführungsform kann, wenn ein Spieler deutlich

weniger Zeit, z. B. weniger als 5 %, mit dem Golfspielen verbringt, auch bestimmt werden, dass diesem Spieler Golfspiele nicht gefallen, was die Empfehlungen verbessern kann, indem weniger Golfempfehlungen präsentiert werden.

[0022] In mindestens einer Ausführungsform können Empfehlungen unter Verwendung eines in **Fig. 4** veranschaulichten Verfahren 400 erzeugt werden. In mindestens einer Ausführungsform wird ein Mediendaten-Stream empfangen 402, der einer Spielsitzung für einen Spieler entspricht. In mindestens einer Ausführungsform kann es sich um einen direkten Feed von einer Spielkonsole oder einem Game-Server handeln oder es kann ein Stream von einem Content-Streaming-Dienst mit Zugriff auf Gameplay-Inhalte sein. In mindestens einer Ausführungsform kann ein Stream alle Inhalte einer Spielsitzung oder ausgewählte Inhalte einer Spielsitzung enthalten, z. B. Highlights oder bestimmte Arten von Handlungen. In mindestens einer Ausführungsform können Segmente dieser Mediendaten als Eingabe für ein oder mehrere trainierte neuronale Netze bereitgestellt werden 404. In mindestens einer Ausführungsform wird ein Stream in eine Vielzahl von Segmenten segmentiert, und mindestens eine Teilmenge dieser Segmente kann als Eingabe in ein neuronales Netz bereitgestellt werden. In mindestens einer Ausführungsform sind diese Netze trainiert, Arten von Spielmerkmalen für ein Spiel oder eine Art von Spiel zu erkennen. In mindestens einer Ausführungsform wird eine Ausgabe empfangen 406, die eine Reihe von Schlüsselwörtern beinhaltet, wobei diese Schlüsselwörter auf Gameplay-Merkmale wie Szenen, Objekte oder Handlungen hinweisen, die aus analysierten Segmenten abgeleitet wurden. In mindestens einer Ausführungsform werden mindestens einige dieser Schlüsselwörter, zusammen mit Informationen über einen entsprechenden Spieler, einem Empfehlungssystem bereitgestellt 408. Dieses Empfehlungssystem kann Schlüsselwörter zusammenfassen 410 und mögliche Themen, die für diesen Spieler von Interesse sind, zumindest teilweise anhand des tatsächlichen Gameplays dieses Spielers bestimmen. In mindestens einer Ausführungsform kann ein Empfehlungssystem Spielinhalte zu diesen interessierenden Themen bestimmen 412. In mindestens einer Ausführungsform können diese Empfehlungen Empfehlungen für Spiele beinhalten, die von einem entsprechenden Spieler bezogen werden können oder auf die er zugreifen kann. In mindestens einer Ausführungsform können mindestens einige dieser Empfehlungen für einen entsprechenden Spieler bereitgestellt werden 414, wobei dieser Spieler bestimmen kann, wobei dieser Spieler entscheiden kann, ob er den entsprechenden Spielinhalt spielen, darauf zugreifen oder beziehen möchte.

[0023] In mindestens einer Ausführungsform kann ein Prozess 500, der in **Fig. 5** veranschaulicht ist, verwendet werden, um Inhaltsempfehlungen zu bestimmen. In mindestens einer Ausführungsform können während des Gameplays zu bestimmen 502. In mindestens einer Ausführungsform können diese Interaktionen verwendet werden, um ein oder mehrere zweite Spiele zu bestimmen 504, wenn diese Interaktionen auf ein potenzielles Interesse an dieser Nutzung hinweisen. In mindestens einer Ausführungsform kann mindestens ein zweites Spiel zur Präsentation für einen entsprechenden Spieler als eine Empfehlung bereitgestellt werden 506, wobei ein entsprechender Spieler entscheiden kann, ob er auf ein empfohlenes zweites Spiel zugreifen, es beziehen oder spielen möchte.

[0024] In mindestens einer Ausführungsform kann ein Empfehlungssystem das Empfehlungsprofil eines Benutzers für Videospiele basierend auf der Videoverarbeitung des Gameplays mithilfe von Deep Learning aggregieren. In mindestens einer Ausführungsform kann dies zumindest teilweise durch den Abgleich von visueller Gameplay-Erfahrung und Spielmetadaten erfolgen. In mindestens einer Ausführungsform kann ein System durch die Verarbeitung visueller Merkmale von Spielen bestimmte visuelle Merkmale, die einem Benutzer gefallen, bei der Erstellung von Empfehlungen berücksichtigen. In mindestens einer Ausführungsform ist dies ein wichtiger Faktor, der ein Abgleichen durch die tatsächliche visuelle Gameplay-Erfahrung ermöglicht. In mindestens einer Ausführungsform berücksichtigt ein System neben visuellen Merkmalen auch Spielmetadaten wie Genre, Herausgeber, Plattform und Erscheinungsjahr, um Empfehlungen nach Bedarf zu filtern. In mindestens einer Ausführungsform kann es zu einer Aggregation von Empfehlungsgewichtungen und Erfahrungswerten kommen. In mindestens einer Ausführungsform kann ein System Empfehlungen und visuelle Gameplay-Erfahrungen über einen bestimmten Zeitraum aggregieren. In mindestens einer Ausführungsform kann dies zu einem Erfahrungswert führen, der die Verarbeitung von Gameplay-Eigenschaften mit erhöhter Genauigkeit der Empfehlungen ermöglicht. In mindestens einer Ausführungsform eignet sich ein solcher Ansatz für Game-Streaming-Plattformen wie NVIDIA GeForce NOW® oder GeForce Experience®, die den Videostream eines Spiels mit Genauigkeit und konsistenten visuellen Einstellungen verarbeiten können. In mindestens einer Ausführungsform kann ein Player wie ein NVIDIA SHIELD TV® Streaming Media Player zur Darstellung von Medieninhalten verwendet werden. In mindestens einer Ausführungsform kann ein solches System zusätzlich zur Verarbeitung von visuellen Merkmalen und Spielmetadaten die erzeugten Empfehlungen nach Benutzerprofil-Metadaten wie Alter, Bewertung, geografischem Standort und Vorlieben filtern.

[0025] In mindestens einer Ausführungsform können Ereignisdaten, die mithilfe eines solchen Prozesses ermittelt werden, verwendet werden, um das Fähigkeitsniveau eines Spielers zu bestimmen. In mindestens einer Ausführungsform können diese Informationen auch verwendet werden, um Empfehlungen, Tipps und Anleitungen zur Verbesserung eines Fähigkeitsniveaus oder der Leistung eines Spielers bereitzustellen. In mindestens einer Ausführungsform können Gameplay-Medien analysiert werden, wie z. B. ein Bild 600 von Gameplay- oder Videodaten, wie in **Fig. 6A** veranschaulicht. In mindestens einer Ausführungsform kann ein Analysesystem oder -Dienst eine Vielzahl verschiedener Handlungen im Spiel bestimmen, z. B. einen Spieler, der einen Golfschläger schwingt, einen Ball schlägt und einen Golfball an einen bestimmten Ort treibt. In mindestens einer Ausführungsform kann ein solches System auch zugehörige Meilensteine oder Erfolge bestimmen, wie z. B. einen Golfball mit einer bestimmten Anzahl von Schlägen in einen Becher zu bekommen, einen Ball auf ein Grün zu bekommen, einen Birdie zu erzielen oder einen Kurs zu beenden. In mindestens einer Ausführungsform kann ein solches System auch ein Beziehen oder Erhalten von zugehörigen Belohnungen oder Auszeichnungen bestimmen, wie z. B. den Gewinn eines Turniers, das Erreichen eines neuen Fähigkeitsniveaus oder das Aufstellen eines neuen persönlichen Rekords. In mindestens einer Ausführungsform können diese Informationen verwendet werden, um das Fähigkeitsniveau eines bestimmten Spielers zu bestimmen.

[0026] In mindestens einer Ausführungsform können Handlungen und Meilensteine im Spiel anhand von Informationen zu einer aktuellen Szene, erfassten Objekten und bestimmten Handlungen sowie beliebigen Texten oder Metadaten, die für eine bestimmte Gameplay-Sitzung bestimmt wurden, bestimmt werden. In mindestens einer Ausführungsform können solche Daten einem trainierten neuronalen Netz zur Bestimmung verschiedener Handlungen und Meilensteine bereitgestellt werden. In mindestens einer Ausführungsform beinhalten Beispiele für Handlungen oder Meilensteine das Töten eines Gegners, das Öffnen einer Beutekiste, das Abschließen einer Mission, das Gewinnen eines Spiels oder das Driften eines Autos. In mindestens einer Ausführungsform können auch Belohnungen für jede vom Spiel zugewiesene Handlung ermittelt werden, z. B. die Anzahl der Tötungen, die für Handlungen im Spiel gesammelten XP-Punkte, gewonnene Matches und Turniere (einschließlich Trophäen und Abzeichen im Spiel), die Fortschrittsstufe für Spiele, basierend auf dem Fortschreiten der Handlung, oder Ressourcen oder Objekte, die für bestimmte Arten von Aufgaben gesammelt wurden. In mindestens einer Ausführungsform kann ein solches System solchen Handlungen und Meilensteinen weitere Tags zuweisen und diese nach Spieler, Spiel und Spielplan oder Stadium in einem bestimmten Spiel oder Art eines Spiels verfolgen.

[0027] In mindestens einer Ausführungsform können diese Informationen verwendet werden, um einen Wert zu berechnen, z. B. ein Fähigkeitsniveau, einen Index oder eine Bewertung des Spielers. In mindestens einer Ausführungsform kann ein System spezielle Modelle für jedes Spiel oder jede Art von Spiel verwenden, da die Handlungen und Meilensteine im Spiel sehr spielspezifisch sein können. In mindestens einer Ausführungsform kann ein Bewertungssystem die Spielerleistung anhand verschiedener Kategorien bewerten, was zumindest teilweise von einer Art des Spiels und einem Spielmodus abhängen kann. Im Allgemeinen können diese Kategorien spezifische Beurteilungsbereiche umfassen, die sich überschneiden können oder nicht. In mindestens einer Ausführungsform kann eine Strategiekategorie Beurteilungen bezüglich der Verwendung von Aussichtspunkten (z. B. Camping), Aggression (z. B. Zeit, die für Angriffe aufgewendet wird), Vorbereitung (z. B. Zeit, die für den Erwerb von Sammlerstücken aufgewendet wird), Auswahl von Waffen, Charakter, Fahrzeug oder Ausrüstung; Angriffsart (z. B. Einzelperson vs. Gruppe, heimlich vs. offen); Verwendung von Lachgas (z. B. in Rennspielen); Genauigkeit; Schussgenauigkeit (z. B. % des Alters der abgefeuerten Kugeln, Kopfschüsse vs. Körperschüsse); Schusspräzision (z. B. Konsistenz); Fahrgenauigkeit (z. B. Fahren auf der Strecke oder Fahren abseits der Strecke) oder Fahrpräzision (z. B. Bremsen vs. Driften auf Kurven) beinhalten.

[0028] In mindestens einer Ausführungsform kann eine Fortschrittskategorie Aspekte wie den prozentualen Anteil am Spielabschluss (z. B. für Einzelspieler-, Storyline-Spiele), die Beendigung der Nebenmission, die Ansammlung von Sammlerstücken; das Freischalten und Einsammeln von Fahrzeugen, Eigentum oder Waffen; die Geschwindigkeit; die Bewegungsrate (z. B. einschließlich Handlungen wie Ausweichen und Ducken zur Verteidigung oder das Weglaufen vor Bedrohungen); den Waffenwechsel und Nachladungen; die Zeit für den Abschluss von Karten, Missionen oder Meilensteinen unter Berücksichtigung von Sammelobjekten und Nebenmissionen; oder die Reaktionszeit beinhalten. In mindestens einer Ausführungsform kann eine Wirkungskategorie Aspekte wie die Anzahl der Tötungen, von sich selbst und dem Team gewonnene Matches, angesammelte XP oder freigeschaltete Trophäen und Sammlerstücke enthalten. In mindestens einer Ausführungsform können Artikel in einer bestimmten Kategorie unterschiedlich gewichtet werden, was zumindest teilweise auf den damit verbundenen Belohnungen basiert.

[0029] In mindestens einer Ausführungsform können diese Informationen verwendet werden, um ein Benchmarking von Spielern anhand eines Fähigkeitsindex durchzuführen. In mindestens einer Ausführungsform können diese Informationen verwendet werden, um Verbesserungen vorzuschlagen, ohne dass ein unfairer Vorteil während des Spiels entsteht. In mindestens einer Ausführungsform können Informationen für einen relativen Vergleich der Spielerleistung in jedem Spiel, jeder Karte oder jedem Level verwendet werden, um Verbesserungsmaßnahmen in Bezug auf die Analytik vorzuschlagen. In mindestens einer Ausführungsform kann dies mit einem separaten Training durchgeführt werden, bei dem das Training im Einzelspielermodus stattfinden kann, um unfairen Vorteil in Multiplayer-Szenarien zu vermeiden. In mindestens einer Ausführungsform kann es vorkommen, dass viele Spieler dazu neigen, an einem bestimmten Ort auf einer Karte zu „campen“, wo sie mit einem Scharfschützengewehr eine hohe Anzahl von Tötungen erzielen. In mindestens einer Ausführungsform kann dies dann ein sehr guter Aussichtspunkt für ein Scharfschützengewehr sein, wenn ein Spieler diese Waffe zur Verfügung hat oder damit ausgerüstet ist. In einer mindestens Ausführungsform könnte, wenn das Driften durch eine bestimmte Kurve auf einer Karte in einem bestimmten Winkel eine höhere Geschwindigkeit im Renn-Gameplay ermöglicht, diese Handlung die Chancen auf den Gewinn eines Rennens verbessern. In mindestens einer Ausführungsform kann ein bestimmtes Sammlerstück den Spielern ermöglichen, einen hohen Schaden pro Angriff zu verursachen, und kann von Spielern, die solche Kämpfe gewinnen, sehr häufig verwendet werden, wodurch es im Interesse eines Spielers liegen kann, dieses Sammlerstück zu erwerben. In mindestens einer Ausführungsform kann ein Spielerempfehlungssystem dem Spieler vorschlagen, dieses Sammlerstück zu erwerben, ohne notwendigerweise die verschiedenen Schritte des Erwerbs so detailliert zu beschreiben, wie es dieses Spiel erlaubt.

[0030] In mindestens einer Ausführungsform kann eine Empfehlung oder ein Tipp 602 an einen Spieler bereitgestellt werden, der gerade einen Golfball in einem Golfspiel in eine Sandbunker geschlagen hat, wie in Bild 600 von **Fig. 6A** veranschaulicht. In mindestens einer Ausführungsform könnte ein Empfehlungssystem bestimmen, dass Spieler in einer ähnlichen Situation erfolgreich einen bestimmten Schläger und eine bestimmte Technik eingesetzt haben, und dieser Tipp kann zur Anzeige bereitgestellt werden. In mindestens einer Ausführungsform handelt es sich um einen Tipp 642 oder eine Empfehlung, die sich nicht auf ein bestimmtes Ereignis oder eine bestimmte Handlung bezieht, sondern für einen bestimmten Kontext relevant sein kann. In mindestens einer Ausführungsform könnte ein Empfehlungssystem bestimmen, dass ein Spieler auf einer langen geraden Straße oder Strecke fährt, und bestimmen, dass viele Spieler einen Lachgas-Booster verwenden, um schneller voranzukommen. In mindestens einer Ausführungsform kann dieser Tipp wie in Bild 640 von **Fig. 6B** veranschaulicht, bereitgestellt werden, wenn sich ein Spieler auf einem solchen geraden Abschnitt oder in der Nähe davon befindet und innerhalb einer bestimmten Zeitspanne kein anderer Tipp angezeigt wurde, zumindest wenn eine maximale Tipphäufigkeit festgelegt ist. In mindestens einer Ausführungsform kann ein Empfehlungssystem einen Tipp für bestimmte, aber nicht alle Ereignisse einer Art von Handlung geben. In mindestens einer Ausführungsform könnte ein Empfehlungssystem bestimmen, dass die meisten Spieler auf ein Ziel schießen, das gerade erschien, wie in Bild 680 von **Fig. 6C**, aber dass die meisten Spieler nachladen, um die Möglichkeit zu haben, dieses Ziel zu treffen. In mindestens einer Ausführungsform kann ein Tipp bereitgestellt werden, um einen Benutzer anzuweisen, eine aktuelle Waffe nachzuladen, um dieses Ziel zu treffen. In mindestens einer Ausführungsform, in der ein Spieler eine Waffe während einer Sitzung viele Male nachladen muss, kann dieser Tipp nicht bei jeder dieser Gelegenheiten bereitgestellt werden, sondern kann nach bestimmten Kriterien, z. B. wenn das Nachladen kritisch ist oder wenn ein Spieler mehrere Male Ziele verfehlt hat, weil er nicht nachgeladen hat, bereitgestellt werden.

[0031] In mindestens einer Ausführungsform können unbeaufsichtigte Machine-Learning-Techniken verwendet werden, um Gameplay-Video-Highlights zu verarbeiten und Empfehlungen für Spieler zu generieren. In mindestens einer Ausführungsform kann ein solches System Dimensionsreduktion und Clustering durchführen, um diese Empfehlungen zu erzeugen. In mindestens einer Ausführungsform ermöglicht die Anwendung dieser Algorithmen auf einen Gameplay-Clip einem Machine-Learning-Modell, eine Darstellung der wichtigsten Merkmale in einer Szene zu extrahieren und sie auf der Grundlage der Ähnlichkeit zu clustern. In mindestens einer Ausführungsform können die relativen Abstände zwischen den Spielen in einem Cluster als Ranking-Kriterien für Empfehlungen verwendet werden. In mindestens einer Ausführungsform kann eine Empfehlungsmaschine die Clusterzuordnung und die Abstände über einen bestimmten Zeitraum hinweg aggregieren, um Überschneidungen zu berücksichtigen und ein intelligentes Empfehlungsprofil zu erstellen.

[0032] In mindestens einer Ausführungsform kann ein Spieler-Empfehlungssystem als Eingabe einen Gameplay-Videostream akzeptieren, der aus zufällig ausgewählten oder speziell ausgewählten Highlights aus dem Gameplay-Sitzungs-Videostream eines Spielers besteht. In mindestens einer Ausführungsform kann ein Videostream mit Hilfe eines Merkmalsauswahl- oder Filterprozesses analysiert werden, um Faktoren für Empfehlungen zu definieren, wobei Prozesse wie die oben in Bezug auf Spielempfehlungen besprochenen

verwendet werden. In mindestens einer Ausführungsform kann eine Merkmalsgruppierung oder ein Clustering durchgeführt werden, und diese Cluster können analysiert werden, um ein Ranking der Empfehlungen zu erstellen. In mindestens einer Ausführungsform kann ein Spielerempfehlungssystem 700 Komponenten enthalten, wie in **Fig. 7** veranschaulicht. In mindestens einer Ausführungsform beinhaltet dieses System sowohl client- als auch serverseitige Komponenten sowie Komponenten, die sich über beide überbrücken.

[0033] In mindestens einer Ausführungsform kann ein Modul 706 zur Auswahl und Extraktion von Merkmalen verwendet werden, das das Trainingsvideo 702 auf einer Server-Seite und das Testvideo 704 auf einer Client-Seite akzeptiert. In mindestens einer Ausführungsform kann dieses Modul 706 die Dimensionalitätsreduktion nutzen, um wichtige oder wesentliche Merkmale für die weitere Verarbeitung zu extrahieren und um Rauschen für nachfolgende Schritte zu eliminieren. In mindestens einer Ausführungsform kann die Dimensionalitätsreduktion auch als Komprimierungsmaßnahme dienen, um Darstellungen für Anwendungsfälle zu speichern, die eine große Anzahl von Merkmalen aufweisen. In mindestens einer Ausführungsform kann die Dimensionsreduktion als Maßnahme zur Merkmalsauswahl dienen, um relevante Merkmale aus einer Stichprobe zu identifizieren und auszuwählen, sowie zur Merkmalsextraktion, um neue Merkmale durch Kombination vorhandener Merkmale zu extrahieren. In mindestens einer Ausführungsform kann die Dimensionsreduktion durch Verwendung von Autocodierern durchgeführt werden.

[0034] In mindestens einer Ausführungsform können die von der Dimensionsreduktion ausgegebenen Rohmerkmale dann als Eingabe für ein Clustering-Modul 708 bereitgestellt werden. In mindestens einer Ausführungsform umfasst das Clustermodul 708 ein Erstellungs- oder Segregationsmodul 710 sowie ein Cluster-Zuweisungsmodul 712. In mindestens einer Ausführungsform können relevante Merkmale, da sie verfügbar sind, auf der Grundlage ihrer Ähnlichkeit in Gruppen geclustert werden. In mindestens einer Ausführungsform kann dieses Clustering mit einem einer Reihe von Clustering-Algorithmen wie K-Means oder OPTICS in Kombination mit Autocodierern wie Generative Adversarial Networks (GANs) erfolgen. In mindestens einer Ausführungsform können Hyperparameter zum Teil von der Wahl des Clustering-Algorithmus abhängen. In mindestens einer Ausführungsform können nach der Identifizierung von Clustern die Informationen für diese Cluster als Eingabe für eine Spielerempfehlungsmaschine 714 bereitgestellt werden. In mindestens einer Ausführungsform kann eine Spielerempfehlungsmaschine 714 ein Mappingmodul 716 und ein Zuweisungsmodul 718 enthalten. In mindestens einer Ausführungsform kann diese Maschine 714 eine Rangliste von Empfehlungen für eine eingehende Probe identifizieren, die zumindest teilweise auf relativen Abständen innerhalb eines zugewiesenen Clusters basiert. In mindestens einer Ausführungsform kann die Verarbeitung verschiedener Punkte eines Videoclips zu einem beliebigen Zeitpunkt zu seiner Klassifizierung als Teil verschiedener Cluster führen. In mindestens einer Ausführungsform kann es zu Überschneidungen zwischen Clustern kommen, die zumindest zum Teil auf das Vorhandensein von Duplikaten zurückzuführen sind. In mindestens einer Ausführungsform behandelt die Empfehlungsmaschine 714 solche Konflikte und Überschneidungen auf der Grundlage von Techniken wie der einfachen Aggregation zwischen sich überschneidenden Clustern, relativen Abständen zu anderen Spielen in einem Cluster und zusätzlichem Kontext unter Verwendung von Metadaten. In mindestens einer Ausführungsform ist es möglich, überwachtes maschinelles Lernen für die Empfehlungsmaschine 714 zu verwenden, um in diesem Prozess zu lernen und ihn zu verbessern. In mindestens einer Ausführungsform hilft die Aggregation von Clustern und Empfehlungen für jede Eingabeprobe über einen bestimmten Zeitraum hinweg der Empfehlungsmaschine 714 dabei, ein Profil für die Empfehlungen eines Spielers zu erstellen und zu pflegen, das als Verhaltensparameter für das Ranking nachfolgender Empfehlungen dienen kann.

[0035] In mindestens einer Ausführungsform kann ein Empfehlungssystem mit einem oder mehreren Streaming-Servern und mehreren Client-Geräten integriert werden. In mindestens einer Ausführungsform können auf einer Server-Seite ein unüberwachtes Machine-Learning-Modell und eine überwachte Machine-Learning-Empfehlungsmaschine 714 auf Gameplay-Videos 702 trainiert werden, wobei ein Clustering-Modell und eine Empfehlung serverseitig installiert werden. In mindestens einer Ausführungsform kann auf einer Client-Seite ein von diesen Clients gesehener Gameplay-Videostream 704 vorverarbeitet und in eine Client-Seite dieses Systems eingespeist werden, wodurch Merkmale extrahiert und in komprimierter Form an einen Server zur weiteren Verarbeitung gesendet werden können.

[0036] In mindestens einer Ausführungsform gibt es in einem solchen System mindestens zwei Arten von Training und Testen. In mindestens einer Ausführungsform umfasst ein Dimensionsreduktions- und Clustermodell für Spiele das Trainieren eines Modells für die Merkmalsauswahl, die Merkmalsextraktion und das Clustering. In mindestens einer Ausführungsform wird dieses Training für jedes zu analysierende Spiel oder jede Art von Spiel durchgeführt. In mindestens einer Ausführungsform kann das Empfehlungsprofil eines Benutzers, der eine Reihe von Spielen spielt, über verschiedene Gameplay-Sessions hinweg aggregiert wer-

den. In mindestens einer Ausführungsform kann durch die Aggregation über einen bestimmten Zeitraum eine ausreichende Genauigkeit der Empfehlungen erreicht werden, die zum Teil auf Erfahrungen mit Überschneidungen und Konflikten zwischen Clustern beruht.

[0037] In mindestens einer Ausführungsform können Videostreamdaten für ein Abenteuerspiel mit einem solchen Empfehlungssystem verarbeitet werden. In mindestens einer Ausführungsform ermöglicht dieses Spiel dem Benutzer, zusätzlich zu einer Handlung, die Schieß-, Fahr- und Flugmissionen beinhaltet, ein Golfspiel zu spielen. In mindestens einer Ausführungsform kann bestimmt werden, dass ein bestimmter Spieler gerne dieses Golf-Teilspiel spielt und viel Zeit mit diesem Teilspiel und möglicherweise verwandten Spielen verbringt. In mindestens einer Ausführungsform kann dieses System Merkmale erkennen, die ein Golfspiel repräsentieren, das gerade gespielt wird, und dieses System kann mit nicht gekennzeichneten Daten arbeiten. Im Gegensatz zu überwachten Machine-Learning- und Deep-Learning-Algorithmen können unüberwachte Machine-Learning-Algorithmen, die in einem Spielerempfehlungssystem verwendet werden, mit nicht gekennzeichneten Daten arbeiten, was die Kosten für die Datenerfassung und -kennzeichnung verringern kann.

[0038] In mindestens einer Ausführungsform können Spielerempfehlungen erzeugt werden, die sich auf die Kalibrierung und Verbesserung der Spielerfähigkeiten beziehen. In mindestens einer Ausführungsform umfasst dies die Erfassung von Handlungen oder interessierenden Ereignissen, wie oben beschrieben. In mindestens einer Ausführungsform können die Arten von Handlungen oder Ereignissen, die von Interesse sind, je nach Spiel oder Art des Spiels variieren. In mindestens einer Ausführungsform wird eine einleitende Reihe von Ereignissen oder Handlungen, die von Interesse sind, präsentiert und definiert. In mindestens einer Ausführungsform kann die Definition von Handlungen erforderlich sein, da es verschiedene Eingaben geben kann, die einer Art von Handlung entsprechen, da eine „Tötung“ je nach Art des Spiels beinhalten kann, dass eine Figur erschossen, von einer Vorsprung gestoßen oder angesprungen wird. In mindestens einer Ausführungsform können mit bestimmten Handlungen auch Belohnungen verbunden sein, wie z. B. eine Erhöhung der Gesundheit oder des Schutzschildes für das Aufnehmen eines Power-Ups oder eine Erhöhung der Tötungs-Anzahl für eine weitere Tötung. In mindestens einer Ausführungsform kann eine Belohnung implizit sein, wie z. B. das Erreichen von 100 Erfahrungspunkten, die nur vorübergehend auf dem Bildschirm angezeigt werden, kurz nachdem eine Handlung auftritt. In mindestens einer Ausführungsform können die Belohnungen durch eine von Bildschirmanalyse, Audioanalyse, Textanalyse oder Metadatenanalyse ermittelt werden.

[0039] In mindestens einer Ausführungsform können diese Handlungen in verschiedene Muster kategorisiert werden. In mindestens einer Ausführungsform können „Fähigkeiten“ komplex und vielfältig sein. In mindestens einer Ausführungsform können die Arten von Fähigkeiten anhand einer Reihe von Faktoren definiert werden. Bei mindestens einer Ausführungsform werden fünf Fähigkeitsfaktoren verwendet. In mindestens einer Ausführungsform werden die Handlungen zur Trainingszeit in eine dieser Kategorien kategorisiert. In mindestens einer Ausführungsform löst die Erfassung einer interessierenden Handlung nicht nur eine Analyse der Art der Handlung und der damit verbundenen Belohnung aus, sondern auch aller verfügbaren Metadaten oder anderen Informationen, die dazu beitragen können, eine Handlung in eine dieser Fähigkeitskategorien einzuordnen. In mindestens einer Ausführungsform kann ein Spieler einen Gegner mit einer ganz bestimmten Technik getötet haben, z. B. wenn er sich an einem bestimmten Ort in Bezug auf den Gegner befand und eine Tötung mit einer bestimmten Waffe mit einem bestimmten Grad an Genauigkeit durchgeführt wurde. In mindestens einer Ausführungsform können diese zusätzlichen Informationen dazu beitragen, eine Handlung von der Art und Weise, wie eine Handlung ausgeführt wurde, zu unterscheiden, was eine genauere Bestimmung der Fähigkeiten des Spielers bereitstellen kann, da einige Techniken schwieriger sind als andere. In mindestens einer Ausführungsform können sowohl momentane als auch über eine Sitzung oder eine Reihe von Sitzungen aggregierte Spielerdaten verwendet werden, wie sie beispielsweise für die Schussgenauigkeit nützlich sein können. In mindestens einer Ausführungsform wird für einen Spieler ein Index erstellt, der Gameplay- oder Fähigkeitsinformationen für jede Kategorie enthält, die verwendet werden können, um individuelle Fähigkeitswertungen in bestimmten Kategorien sowie eine Gesamtfähigkeitswertung zu bestimmen.

[0040] In mindestens einer Ausführungsform können die Spieler anhand dieses spieterspezifischen Indexes gebenchmarkt werden. In mindestens einer Ausführungsform können Gameplay-Empfehlungen für einen Spieler zumindest teilweise anhand dieses Benchmarkings bestimmt werden. In mindestens einer Ausführungsform kann ein trainiertes Modell Handlungs- und Belohnungsdaten akzeptieren und diese Daten in verschiedene Indizes zum Vergleich mit anderen Spielern kategorisieren. In mindestens einer Ausführungsform können Spieler direkt durch den Vergleich von Daten für eine bestimmte Anzahl oder eine Reihe von Hand-

lungen gebenchmarkt werden. In mindestens einer Ausführungsform kann ein Benchmarking durchgeführt werden, indem gewichtete Fähigkeitsgrade aus verschiedenen Kategorien verglichen werden. In mindestens einer Ausführungsform kann eine einfache Handlung, wie z. B. das Erschießen eines Gegners in der Nähe und in Sichtweite, eine geringere Gewichtung haben als eine geschicktere Tötung, wie z. B. durch Camping und strategisches Töten eines Gegners, wie z. B. durch einen schallgedämpften Kopfschuss zu einem Zeitpunkt, an dem es unwahrscheinlich ist, dass er bemerkt wird. In mindestens einer Ausführungsform können diese Informationen verwendet werden, um Spieler anhand eines Fähigkeitsindex zu benchmarken. In mindestens einer Ausführungsform können einem Benutzer, wenn ein Spieler lokal und ohne Multiplayer-Funktionalität spielt, ein oder mehrere Tipps oder Vorschläge bereitgestellt werden, die teilweise auf diesem Benchmarking basieren. In mindestens einer Ausführungsform können die Tipps auf aktuellen Aspekten eines Spiels basieren, wie z. B. einer Position auf einer Karte, einer aktiven Waffe und dem Gesundheitszustand, und es können geeignete Vorschläge für einen Spieler mit diesem Fähigkeitsniveau bestimmt und ein aktuelles Fähigkeitsniveau oder Benchmarking präsentiert werden. In mindestens einer Ausführungsform werden Tipps auf der Grundlage von Handlungen gegeben, die für Spieler mit höherem Fähigkeitsniveau bestimmt wurden, um einem Spieler zu helfen, ein höheres Fähigkeitsniveau zu erreichen. In mindestens einer Ausführungsform können mehrere trainierte Modelle parallel verwendet werden, um einen eingehenden Videostream zu verarbeiten und die Wertungen für verschiedene Kategorien von Fähigkeiten zu bestimmen.

[0041] In mindestens einer Ausführungsform kann ein Prozess des Benchmarkings und des Vorschlags von Verbesserungen mindestens zwei neuronale Netze verwenden. In mindestens einer Ausführungsform kann ein erstes neuronales Netz als Eingabe Daten für verschiedene Handlungen in verschiedenen Kategorien akzeptieren, um einen Index des Fähigkeitsniveaus zu erzeugen, der für das Benchmarking nützlich ist. In mindestens einer Ausführungsform kann ein zweites neuronales Netz dieselben oder ähnliche Daten zu bestimmten Zeitpunkten während des Spiels als Eingabe akzeptieren. In mindestens einer Ausführungsform kann dieses zweite neuronale Netz Informationen für einen aktuellen Spielerzustand nehmen und diese mit Handlungen vergleichen, die von anderen Spielern in einem ähnlichen Zustand durchgeführt wurden, z. B. an einer ähnlichen Kartenposition unter ähnlichen Gameplay-Bedingungen, und kann Tipps oder Vorschläge ausgeben oder ableiten, die einem Spieler für diesen aktuellen Spielerzustand bereitgestellt werden. In mindestens einer Ausführungsform könnte dies die Weitergabe eines Tipps beinhalten, nach Gegenständen zu suchen, wenn festgestellt wird, dass eine Beutekiste in der Nähe ist, oder zu einer anderen Waffe zu wechseln und sich zu ducken, wenn es wahrscheinlich ist, dass eine bestimmte Art von Gegner in der Nähe ist. In mindestens einer Ausführungsform können Tipps erzeugt werden, indem Metadaten für Handlungen ermittelt werden, die von anderen Spielern in ähnlichen Gameplay-Situationen durchgeführt wurden, und eine Tippvorlage verwendet wird, um einen Vorschlag zu erzeugen, der einem Spieler zu einem entsprechenden Zeitpunkt bereitgestellt wird.

[0042] In mindestens einer Ausführungsform können solche Modelle, die auf einem Client-Gerät während des Gameplays ausgeführt werden, dazu beitragen, Metadaten oder Schlüsselwörter für bestimmte Handlungen zu beziehen, wie hier beschrieben. In mindestens einer Ausführungsform werden diese Schlüsselwörter oder Metadaten zur Kategorisierung an einen Server übertragen. In mindestens einer Ausführungsform kann ein trainiertes Modell, das auf diesem Client-Gerät läuft, Metadaten akzeptieren, ein Benchmarking durchführen und relevante Tipps für die Bereitstellung bestimmen. In mindestens einer Ausführungsform kann ein Tipperzeugungsmodell Indexwerte und Kategorie-Metadaten akzeptieren, um relevante Tipps abzuleiten. In mindestens einer Ausführungsform kann ein Tipp-Manager neben anderen solchen Aspekten auch bestimmen, wann ein bestimmter Tipp angezeigt werden soll, wo ein Tipp angezeigt werden soll und wie oft ein Tipp angezeigt werden soll. In mindestens einer Ausführungsform werden Tipps über Textvorlagen bereitgestellt. In mindestens einer Ausführungsform können Tipps Elemente wie Bilder, Animationen, Video, Audio, Sprache oder Hyperlinks zu verwandten Inhalten beinhalten. In mindestens einer Ausführungsform kann ein Spieler anpassen, wo Tipps angezeigt werden, sowie Faktoren wie die Häufigkeit der Anzeige von Tipps und welche Arten von Tipps angezeigt werden.

[0043] In mindestens einer Ausführungsform kann ein bestimmtes Format oder eine Vorlage für eine Art von Tipp verwendet werden. In mindestens einer Ausführungsform kann eine Tippvorlage bestimmte Kombinationen von Daten und Werten angeben, die bei der Erstellung dieses Tipps verwendet werden sollen. In mindestens einer Ausführungsform können sich die Kategorien der Tipps in Abhängigkeit von den Fähigkeitsbewertungen und den Fähigkeitskategorien unterscheiden. In mindestens einer Ausführungsform sind die Tipps nicht fest kodiert, sondern es kann ein bestimmter Satz von Tipparten bereitgestellt werden. In mindestens einer Ausführungsform werden die Tipps nur einzeln mit einer maximalen Häufigkeit angezeigt. In mindestens einer Ausführungsform kann ein bestimmter Tipp- oder Vorschlagsbereich festgelegt werden, und in die-

sem Bereich können mehrere Tipps gleichzeitig angezeigt werden. In mindestens einer Ausführungsform kann ein Spieler die Möglichkeit haben, bereitgestellte Vorschläge zu speichern, zu löschen oder zu suchen.

[0044] In mindestens einer Ausführungsform können Teile eines Spielerempfehlungsprozesses auf einem Gerät, auf einem Spielservers oder in einer Cloud-Umgebung durchgeführt werden. In mindestens einer Ausführungsform können Spielmodelle und Standardgewichtungen in Spielbibliotheken enthalten sein, die auf ein Client-Gerät übertragen werden. In mindestens einer Ausführungsform können kleinere Modelle auf Client-Geräten eingesetzt werden, während größere Modelle serverseitig ausgeführt werden. In mindestens einer Ausführungsform werden die Spielerdaten lokal für Vorschläge verarbeitet, dann aber serverseitig für ein globales Benchmarking oder eine Verhaltensanalyse aggregiert.

[0045] In mindestens einer Ausführungsform kann zumindest ein Teil dieser Funktionalität von einem Entwickler in ein Spiel integriert werden. In mindestens einer Ausführungsform kann die Fähigkeit, Ereignisse innerhalb eines Spiels zu identifizieren, für eine höhere Genauigkeit und geringere Latenz sorgen. In mindestens einer Ausführungsform kann die Fähigkeit, auf ein Benutzerprofil zuzugreifen, dazu beitragen, Daten zu aggregieren und zusätzliche Metadaten zu erhalten, die für die Indizierung und das Benchmarking nützlich sein können. In mindestens einer Ausführungsform kann auch auf Informationen über zuvor bereitgestellte Tipps zugegriffen werden, was bei der Bestimmung der anzuzeigenden Tipps helfen kann. In mindestens einer Ausführungsform kann ein bestimmter Tipp nur mit einer maximalen Häufigkeit oder Anzahl von Ereignissen bereitgestellt werden, und eine Fähigkeit, die sich nach einer Anzahl dieser Tipps nicht verbessert hat, kann dazu führen, dass ein anderer Tipp oder eine andere Art von Tipp erzeugt und bereitgestellt wird, um dazu beizutragen, diese bestimmte Fähigkeit für diesen Spieler zu verbessern. In mindestens einer Ausführungsform kann die Fähigkeit, Handlungs- und Belohnungsdaten aus dem Spielcode zu beziehen, die Notwendigkeit beseitigen, einen einleitenden Satz von Modell- und Merkmalsextraktion auf Spielmedien durchzuführen. In mindestens einer Ausführungsform kann der Zugriff auf den Spielcode auch dazu führen, dass zumindest einige Tipps kontextbezogener sind, oder dass Tipps stattdessen in ein Spiel integriert oder als Overlays zu diesem Spiel oder als Tipps außerhalb eines Spiels bereitgestellt werden können. In mindestens einer Ausführungsform kann ein Kamerawinkel oder eine Ansicht geändert werden, um einen spezifischen Kontext für einen bestimmten Tipp zu schaffen.

[0046] In mindestens einer Ausführungsform kann ein Prozess 800 zur Bereitstellung von Spielerempfehlungen verwendet werden, wie in **Fig. 8** veranschaulicht. In mindestens einer Ausführungsform wird ein Video einer Gameplay-Sitzung für einen Spieler empfangen 802. In mindestens einer Ausführungsform kann dieses Video nahezu in Echtzeit während dieser Sitzung oder danach in einer Videodatei oder einem Stream empfangen werden. In mindestens einer Ausführungsform ist das Video zusammen mit Medieninhalten einschließlich Audio und Metadaten enthalten. In mindestens einer Ausführungsform werden aus diesem Video eine oder mehrere interessierende Handlungen mit den dazugehörigen Belohnungen bestimmt 804. In mindestens einer Ausführungsform werden diese Handlungen oder Ereignisse anhand verschiedener detektierter Merkmale bestimmt, wie an anderer Stelle hierin beschrieben. In mindestens einer Ausführungsform werden diese ermittelten Handlungen in eine von mehreren Fähigkeitskategorien kategorisiert 806. In mindestens einer Ausführungsform wird ein Spielerfähigkeitsindex auf der Grundlage von Spielerdaten für diese Fähigkeitskategorien, wie Daten und Metadaten für erfasste Handlungen, die in jede dieser Kategorien fallen, und zugehörige Fähigkeitswerte, die für diese Kategorien zumindest teilweise auf der Grundlage dieser Daten bestimmt werden, erzeugt 808 oder aktualisiert. In mindestens einer Ausführungsform kann die Fähigkeit des Spielers anhand dieses Fähigkeitsindexes gebenchmarkt werden 810, was nützlich sein kann, um diesen Spieler mit anderen Spielern zu vergleichen. In mindestens einer Ausführungsform können ein oder mehrere Vorschläge zur Verbesserung der Fähigkeiten bestimmt werden 812, um sie dem Spieler zu einem geeigneten Zeitpunkt bereitzustellen, wobei die Vorschläge zumindest teilweise auf der Grundlage eines entsprechenden Fähigkeitsindex und relevanter Metadaten bestimmt werden. In mindestens einer Ausführungsform kann ein Vorschlag auf einer Kategorie einer in einer aktuellen Spielsitzung auszuführenden Handlung, einem aktuellen Fähigkeitsniveau dieses Spielers für diese Handlungskategorie und Metadaten für einen aktuellen Zustand dieser Spielsitzung, in der diese Handlung ausgeführt werden soll, basieren.

[0047] In mindestens einer Ausführungsform können die Fähigkeitsinformationen zumindest teilweise auf der ermittelten Reaktionszeit für eine Handlungsart basieren. In mindestens einer Ausführungsform kann die zur Bestimmung der Fähigkeit verwendete Reaktionszeit mindestens zwei Arten von Reaktionen umfassen, einschließlich der Reaktionszeit, um ein Ereignis zu bemerken, und der Reaktionszeit, um eine angemessene Reaktion zu erzeugen. In mindestens einer Ausführungsform kann die Reaktionszeit getrennt von der Fähigkeitsstufe verfolgt und gemeldet werden. In mindestens einer Ausführungsform kann ein Spielerempfehlungssystem die Reaktionszeiten eines Spielers bestimmen und überwachen und Empfehlungen oder Vorschläge

zur Verbesserung der Reaktionszeiten bereitstellen. In mindestens einer Ausführungsform kann die Reaktionszeit in einen Fähigkeitsindex integriert und zum Benchmarking eines Spielers mit anderen Spielern verwendet werden.

[0048] In mindestens einer Ausführungsform können die Erfassungs- und Reaktionszeiten eines Spielers anhand von Videobildern bestimmt werden, wie den in **Fig. 9A-9D** veranschaulicht. In mindestens einer Ausführungsform veranschaulicht ein erstes Bild 900 von Videodaten einen Spielzustand zu einem ersten Zeitpunkt $t=0$, wobei in diesem Bild kein Ziel zu sehen ist. In mindestens einer Ausführungsform könnte ein Ziel 922 zum Zeitpunkt $t=1$ erscheinen, wie in Bild 920 von **Fig. 9B** veranschaulicht. In mindestens einer Ausführungsform kann bestimmt werden, dass ein Spieler das Ziel 922 bemerkt hat, weil ein Avatar 902 für einen Spieler in einer Weise bewegt wurde, die anzeigt, dass dieser Spieler dieses Ziel bemerkt hat, wie in Bild 940 von **Fig. 9C** veranschaulicht, was zu einem Zeitpunkt $t=1+m$ auftritt. In mindestens einer Ausführungsform kann m ein Maß für die Erfassungszeit für diesen Spielers sein. In mindestens einer Ausführungsform kann die Erfassung bestimmt werden, wenn irgendeine Handlung erfasst wird, die einem Spieler entspricht, der ein Objekt oder ein Ereignis bemerkt, was durch die Analyse des Gameplays für mehrere Spieler und die Bestimmung ihrer Reaktionen bestimmt werden kann, wenn ein solches Objekt oder Ereignis im Gameplay erfasst werden kann. In mindestens einer Ausführungsform, veranschaulicht das Bild 960 von **Fig. 9D** einen Spieler, der den Avatar 902 veranlasst, dieses Ziel zu erschießen und zu zerstören. In mindestens einer Ausführungsform tritt dies zur Zeitpunkt $t=1+m+n$ auf, wobei n ein Maß für die Reaktionszeit ab der Erfassungszeit und $m+n$ ein Maß für die Gesamtreaktionszeit ist. In mindestens einer Ausführungsform können Webcam-Daten zur Bestimmung des Erfassungszeitpunkts herangezogen werden, wenn sie verfügbar sind, da die Augen eines Spielers möglicherweise eine schnelle Bewegung in diese Richtung machen oder ein Spieler eine andere Geste oder Bewegung macht. In mindestens einer Ausführungsform können Audiodaten zur Bestimmung der Erfassungszeit herangezogen werden, wenn sie verfügbar sind, z. B. wenn ein Spieler kurz nach dem Erscheinen eines Objekts eine Äußerung macht.

[0049] In mindestens einer Ausführungsform ist eine Erfassungsphase durch Metadaten gekennzeichnet, die eine Startzeitreaktion und eine entsprechende Kinetik beinhalten können. In mindestens einer Ausführungsform können diese Metadaten Informationen über einen Punkt enthalten, an dem ein Spieler ein Objekt oder ein Ereignis erkannt hat, sowie Informationen darüber, was dieser Spieler getan hat (z. B. ortsfest stehen, sich bewegen oder klettern), sowie über seine Richtung und seinen Standort. In mindestens einer Ausführungsform können diese Variablen als Eingabe für ein trainiertes neuronales Netz zur Bestimmung der Erfassungs- und/oder Reaktionszeit bereitgestellt werden. In mindestens einer Ausführungsform kann dieser Spieler dazu bestimmt werden, nach der Erkennung eines Objekts oder Ereignisses eine Handlung vorzunehmen. In mindestens einer Ausführungsform können ähnliche Metadaten bereitgestellt werden, um eine von der Erfassungszeit getrennte Reaktionszeit zu bestimmen. In mindestens einer Ausführungsform kann die Unterteilung der Gesamtreaktion in Erfassung und Reaktion dazu beitragen, die Reaktionszeit für verschiedene Benutzer zu verbessern, wobei einige Benutzer schnell reagieren, aber langsam bemerken oder erfassen können, während andere Benutzer eine Veränderung schnell erfassen, aber relativ langsam reagieren könnten. In mindestens einer Ausführungsform kann die Reaktionszeit des Spielers in vielen Fällen durch Verbesserung der Fähigkeiten und Vertrautheit mit den Kontrollschemata verbessert werden, während die Erfassungszeit durch andere Vorschläge als die Änderung der relativen Position eines Spielers zu einer Anzeige, die Änderung der Art oder Größe der Anzeige oder andere derartige Änderungen schwieriger zu beheben sein kann. In mindestens einer Ausführungsform können die zur Erfassung verwendeten Aktionen jede geeignete Handlung umfassen, wie z. B. Schießen, Ausweichen, Weglaufen oder das Ziehen einer Waffe. In mindestens einer Ausführungsform kann eine Reaktionshandlung zum Teil von der Art des erkannten Objekts oder Ereignisses abhängen.

[0050] In mindestens einer Ausführungsform wird ein Video einer Gameplay-Sitzung für einen Spieler empfangen 1002. In mindestens einer Ausführungsform kann dieses Video nahezu in Echtzeit während dieser Sitzung oder danach in einer Videodatei oder einem Stream empfangen werden. In mindestens einer Ausführungsform ist das Video zusammen mit Medieninhalten einschließlich Audio und Metadaten enthalten. In mindestens einer Ausführungsform wird ein interessierendes Ereignis aus einem Spielvideo bestimmt 1004. In mindestens einer Ausführungsform werden die Gameplay-Daten nach diesem Ereignis analysiert 1006, um eine oder mehrere nachfolgende Spielerhandlungen zu bestimmen. In mindestens einer Ausführungsform werden diese Handlungen oder Ereignisse anhand verschiedener detektierter Merkmale bestimmt, wie an anderer Stelle herein beschrieben. In mindestens einer Ausführungsform werden diese bestimmten Handlungen zusammen mit den entsprechenden Metadaten als Eingabe für mindestens ein trainiertes neuronales Netz bereitgestellt 806. In mindestens einer Ausführungsform werden Schlussfolgerungen in Bezug auf die Erfassungs- und Reaktionszeiten der Spieler in auf dieses Ereignis empfangen 1010. In mindestens einer

Ausführungsform wird ein Spielerfähigkeitsindex zumindest teilweise auf der Grundlage der Erfassungs- und Reaktionszeiten oder der Gesamtreaktionszeit aktualisiert 1012. In mindestens einer Ausführungsform können ein oder mehrere Vorschläge zur Verbesserung der Reaktion bestimmt werden 1014, die dem Spieler zu einem geeigneten Zeitpunkt bereitgestellt werden.

[0051] In mindestens einer Ausführungsform kann eine Verschlechterung der Erfassungs- oder Reaktionszeit über eine oder mehrere Gameplay-Sitzungen ein Zeichen für die Ermüdung eines Spielers sein. In mindestens einer Ausführungsform können andere Faktoren analysiert werden, um zu versuchen, die Ermüdung des Spielers zu bestimmen. In mindestens einer Ausführungsform kann die Abweichung der Reaktionszeit ein primärer Faktor bei der Bestimmung der Ermüdung sein, und es kann ein Schwellenwert festgelegt werden, bei dessen Überschreitung eine Verschlechterung auf Ermüdung hinweist. In mindestens einer Ausführungsform kann dieser Schwellenwert je nach Spieler variieren und mit Hilfe eines trainierten neuronalen Netzes bestimmt werden. In mindestens einer Ausführungsform können die Reaktionszeiten eines bestimmten Spielers als Eingabe in ein Netzwerk eingespeist werden, und es kann eine Schlussfolgerung hinsichtlich des Ermüdungsgrades gezogen werden. In mindestens einer Ausführungsform können auch andere Arten von Beeinträchtigungen anhand von Änderungen der Reaktionszeiten ermittelt werden. In mindestens einer Ausführungsform kann eine Verschlechterung der Reaktionszeit dazu führen, dass ein Spieler für einen bestimmten Zeitraum einen niedrigeren Fähigkeitsindex erhält, während eine Verlängerung der Reaktionszeit dazu führen kann, dass ein Spieler einen höheren Fähigkeitsgrad erhält, was für das Matchmaking oder andere derartige Zwecke nützlich sein kann.

[0052] In mindestens einer Ausführungsform können andere Daten als Eingabe für ein neuronales Netz zur Bestimmung der Ermüdung bereitgestellt werden. In mindestens einer Ausführungsform können analysierte Mikrofondaten bereitgestellt werden, die Daten enthalten können, die repräsentativ für Atemmuster, Änderungen in Sprachmustern, Änderungen im Tonfall, Gähnen oder Ausrufe in Bezug auf Müdigkeit sind. In mindestens einer Ausführungsform können Videodaten analysiert werden, um zu versuchen, Metadaten zu ermitteln, die für die Bestimmung von Müdigkeit nützlich sind, wie z. B. Augenbelastung, Änderungen der Augenbewegungen, Änderungen der Blinzelmuster, Änderungen der Kopfposition oder Änderungen der Körperhaltung. In mindestens einer Ausführungsform können auch Daten über die Genauigkeit oder nachlassende Fähigkeiten auf Ermüdung hinweisen. In mindestens einer Ausführungsform könnte ein neuronales Netz schlussfolgern, dass ein Benutzer eine Ermüdungsschwelle oder -bestimmung erreicht hat. In mindestens einer Ausführungsform kann diesem Spieler ein Vorschlag zur Verbesserung seiner Fähigkeiten bereitgestellt werden, z. B. eine Pause einzulegen oder für eine gewisse Zeit eine andere Aufgabe auszuführen. In mindestens einer Ausführungsform kann auch ein Vorschlag angezeigt werden, der einen Tipp zur Überwindung der Müdigkeit enthält. In mindestens einer Ausführungsform kann eine vom Spiel unabhängige Benachrichtigung angezeigt werden, z. B. eine Nachricht an ein Telefon, die den Spieler dazu veranlasst, das Spiel zu verlassen. In mindestens einer Ausführungsform kann ein Spieler Präferenzen für solche Benachrichtigungen festlegen. In mindestens einer Ausführungsform können diese Benachrichtigungen zu geeigneten Zeitpunkten erfolgen, z. B. nach dem Ende eines Levels oder während eines Ladebildschirms, um die Auswirkungen auf das Gameplay zu minimieren und den optimalen Zeitpunkt für eine Pause oder Verhaltensänderung zu nutzen. In mindestens einer Ausführungsform kann ein Spiel, das nach einem Tod oder dem Ende eines Levels einen Zuschauermodus zulässt, dem Spieler vorschlagen, länger zuzuschauen, oder das Zuschauen oder Laden etwas länger als gewöhnlich dauern lassen, um dem Spieler eine kleine Pause zu gönnen, insbesondere wenn sich eine ernsthafte Ermüdung einstellt.

[0053] In mindestens einer Ausführungsform kann ein Verfahren 1200 zur Bestimmung der Spielerermüdung verwendet werden, wie in **Fig. 12** veranschaulicht. In mindestens einer Ausführungsform wird ein Video von einer Gameplay-Sitzung empfangen 1202. In mindestens einer Ausführungsform werden auch Verhaltensdaten für eine Gameplay-Sitzung empfangen 1204. In mindestens einer Ausführungsform können die Verhaltensdaten Video- oder Audiodaten eines Spielers während des Spiels, biometrische Daten dieses Spielers oder Eingaben dieses Spielers sowie andere Arten von Verhaltensdaten umfassen. In mindestens einer Ausführungsform werden diese Daten analysiert 1206, um Handlungs- und Reaktionsdaten für diesen Spieler zu extrahieren, wie hier beschrieben. In mindestens einer Ausführungsform werden diese Daten als Eingabe für ein oder mehrere trainierte neuronale Netze bereitgestellt 1208. In mindestens einer Ausführungsform kann ein neuronales Netz so trainiert werden, dass es den Zustand des Spielers bestimmt, der sich auf den Bewusstseinsgrad, die Beeinträchtigung oder die Müdigkeit beziehen kann. In mindestens einer Ausführungsform kann eine Schlussfolgerung bezüglich des Zustands dieses Spielers empfangen werden 1210. In mindestens einer Ausführungsform können, wenn bestimmt wird 1212, dass ein Spieler ermüdet ist oder mehr als ein normales oder akzeptables Maß an Müdigkeit aufweist, Maßnahmen ergriffen werden, um dem Benutzer zu helfen, mit dieser Müdigkeit umzugehen. In mindestens einer Ausführungsform kann dem Spie-

ler ein Vorschlag bereitgestellt werden 1214, um ihm zu helfen, mit der Müdigkeit umzugehen, z. B. eine Pause einzulegen oder eine andere Aktivität auszuführen. In mindestens einer Ausführungsform kann dieser Vorschlag unabhängig von einer aktuellen Gameplay-Sitzung bereitgestellt werden. In mindestens einer Ausführungsform können Maßnahmen ergriffen werden, um ein Spiel zu modifizieren, um dieser Ermüdung Rechnung zu tragen, wie z. B. eine Änderung des Schwierigkeitsgrads oder der Zusammenstellung der Spieler für eine nachfolgende Sitzung.

TRAINING UND ENTWICKLUNG EINES NEURONALEN NETZES

[0054] Eine zunehmende Vielfalt von Branchen und Anwendungen nutzt die Vorteile von maschinellem Lernen. In mindestens einer Ausführungsform wurden auf Prozessoren entwickelte tiefe neuronale Netze (deep neural networks - DNNs) für verschiedene Anwendungsfälle verwendet, von selbstfahrenden Autos bis zur schnelleren Entwicklung von Medikamenten, von der automatischen Bildanalyse für Sicherheitssysteme bis zur intelligenten Echtzeit-Sprachübersetzung in Video-Chat-Anwendungen. In mindestens einer Ausführungsform ist Deep Learning eine Technik, die den neuronalen Lernprozess des menschlichen Gehirns modelliert und dabei kontinuierlich lernt, immer intelligenter wird und mit der Zeit schneller genauere Ergebnisse liefert. Ein Kind lernt zunächst von einem Erwachsenen, verschiedene Formen richtig zu erkennen und zu klassifizieren und ist schließlich in der Lage, Formen ohne Nachhilfe zu erkennen. In ähnlicher Weise müsste in mindestens einer Ausführungsform ein Deep-Learning- oder neuronales Lernsystem, das dazu ausgelegt ist, eine ähnliche Aufgabe zu erfüllen, trainiert werden, damit es intelligenter und effizienter bei der Identifizierung von grundlegenden Objekten, verdeckten Objekten usw. wird, während es diesen Objekten außerdem einen Kontext zuweist.

[0055] In mindestens einer Ausführungsform betrachten die Neuronen in einem menschlichen Gehirn verschiedene Eingaben, die empfangen werden, werden jeder dieser Eingaben Bedeutungsstufen zugewiesen und wird die Ausgabe an andere Neuronen weitergegeben, um darauf zu reagieren. Ein künstliches Neuron oder Perzeptron ist das grundlegendste Modell eines neuronalen Netzes. In mindestens einer Ausführungsform kann ein Perzeptron eine oder mehrere Eingaben empfangen, die verschiedene Merkmale eines Objekts darstellen, für dessen Erkennung und Klassifizierung ein Perzeptron trainiert ist, und wird jedem dieser Merkmale auf Grundlage der Bedeutung dieses Merkmals beim Definieren einer Form eines Objekts eine bestimmte Gewichtung zugewiesen.

[0056] Ein tiefes neuronales Netz (DNN) beinhaltet mehrere Schichten mit vielen verbundenen Perzeptronen (z. B. Knoten), die mit enormen Mengen von Eingabedaten trainiert werden können, um komplexe Probleme schnell und mit hoher Genauigkeit zu lösen. In einem Beispiel zerlegt eine erste Schicht eines DNN-Modells ein Eingangsbild eines Autos in verschiedene Abschnitte und sucht nach grundlegenden Mustern, wie etwa Linien und Winkeln. Die zweite Schicht setzt die Linien zusammen, um nach übergeordneten Mustern, wie etwa Rädern, Windschutzscheiben und Spiegeln, zu suchen. Eine nächste Schicht identifiziert einen Fahrzeugtyp und einige letzte Schichten erzeugen ein Label für ein Eingangsbild, das ein Modell einer bestimmten Automarke identifiziert. Sobald ein DNN trainiert ist, kann dieses DNN eingesetzt und zur Identifizierung und Klassifizierung von Objekten oder Mustern in einem als Inferenz bekannten Prozess verwendet werden. Beispiele für Inferenz (einen Prozess, bei dem ein DNN nützliche Informationen aus einer gegebenen Eingabe extrahiert) beinhalten die Identifizierung handgeschriebener Zahlen auf Schecks, die in Geldautomaten eingezahlt werden, die Identifizierung von Bildern von Freunden auf Fotos, die Bereitstellung von Filmpfehlungen, die Identifizierung und Klassifizierung verschiedener Arten von Automobilen, Fußgängern und Gefahren auf der Straße in fahrerlosen Autos oder die Übersetzung menschlicher Sprache in nahezu Echtzeit.

[0057] Während des Trainings fließen Daten in einer Vorwärtspropagationsphase durch ein DNN, bis eine Vorhersage erzeugt wird, die ein der Eingabe entsprechendes Label anzeigt. Wenn ein neuronales Netz die Eingaben nicht korrekt kennzeichnet, werden Fehler zwischen einem korrekten Label und einem vorhergesagten Label analysiert und die Gewichtungen werden für jedes Merkmal während einer Rückwärtspropagationsphase angepasst, bis ein DNN die Eingabe und andere Eingaben in einem Trainingsdatensatz korrekt kennzeichnet. Das Trainieren komplexer neuronaler Netze erfordert enorme Mengen von paralleler Rechenleistung, einschließlich Gleitkommamultiplikationen und -additionen, die unterstützt werden. Inferenzieren ist weniger rechenintensiv als Training, da es sich um einen latenzabhängigen Prozess handelt, bei dem ein trainiertes neuronales Netz auf neue Eingaben angewendet wird, die es nicht zuvor gesehen hat, um Bilder zu klassifizieren, Sprache zu übersetzen und neue Informationen abzuleiten.

[0058] Neuronale Netze stützen sich in hohem Maße auf mathematische Matrixoperationen und komplexe mehrschichtige Netze erfordern enorme Mengen an Gleitkommarechenleistung und Bandbreite sowohl für Effizienz als auch für Geschwindigkeit. Mit Tausenden von Verarbeitungskernen, die für mathematische Matrixoperationen optimiert sind und eine Rechenleistung von mehreren Dutzend bis Hunderten von TFLOPS liefern, kann eine Computerplattform die Rechenleistung liefern, die für Anwendungen für künstliche Intelligenz und maschinelles Lernen auf der Grundlage tiefer neuronaler Netze erforderlich ist.

[0059] Fig. 13 veranschaulicht Komponenten eines Systems 1300, das in mindestens einer Ausführungsform verwendet werden kann, um maschinelles Lernen zu trainieren und zu verwenden. Wie erörtert wird, können verschiedene Komponenten durch verschiedene Kombinationen von Rechenvorrichtungen und Ressourcen oder durch ein einziges Rechnersystem bereitgestellt werden, die unter der Kontrolle einer einzigen Entität oder mehrerer Entitäten stehen kann. Darüber hinaus können Aspekte durch verschiedene Entitäten ausgelöst, initiiert oder angefordert werden. In mindestens einer Ausführungsform kann das Trainieren eines neuronalen Netzes durch einen Anbieter angewiesen werden, der einer Anbieterumgebung 1306 zugeordnet ist, während in mindestens einer Ausführungsform das Trainieren von einem Kunden oder einem anderen Benutzer angefordert werden kann, der über eine Client-Vorrichtung 1302 oder eine andere derartige Ressource Zugang zu einer Anbieterumgebung hat. In mindestens einer Ausführungsform können Trainingsdaten (oder durch ein trainiertes neuronales Netz zu analysierende Daten) von einem Anbieter, einem Benutzer oder einem Drittanbieter von Inhalten 1324 bereitgestellt werden. In mindestens einer Ausführungsform kann die Client-Vorrichtung 1302 zum Beispiel ein Fahrzeug oder ein Objekt sein, das im Auftrag eines Benutzers navigiert werden soll und das Anforderungen übermitteln und/oder Anweisungen empfangen kann, die die Navigation einer Vorrichtung unterstützen.

[0060] In mindestens einer Ausführungsform können Anforderungen über mindestens ein Netzwerk 1304 übermittelt werden, um in einer Anbieterumgebung 1306 empfangen zu werden. In mindestens einer Ausführungsform kann es sich bei einer Client-Vorrichtung um beliebige geeignete elektronische Vorrichtungen und/oder Rechenvorrichtungen handeln, die es einem Benutzer ermöglichen, derartige Anforderungen zu erzeugen und zu senden und die Desktop-Computer, Notebook-Computer, Computer-Server, Smartphones, Tablet-Computer, Spielkonsolen (tragbar oder nicht), Computer-Prozessoren, Computer-Logik und Set-Top-Boxen beinhalten können. Das/die Netzwerk(e) 1304 kann/können ein beliebiges geeignetes Netzwerk zum Übertragen einer Anforderung oder anderer derartiger Daten beinhalten, das das Internet, ein Intranet, ein Ethernet, ein Mobilfunknetz, ein lokales Netzwerk (LAN), ein Netzwerk direkter drahtloser Verbindungen zwischen Peers usw. beinhalten kann.

[0061] In mindestens einer Ausführungsform können Anforderungen an einer Schnittstellenschicht 1308 empfangen werden, die in diesem Beispiel Daten an einen Trainings- und Inferenzmanager 1310 weiterleiten kann. Bei diesem Manager kann es sich um ein System oder einen Dienst handeln, das bzw. der Hardware und Software für die Verwaltung von Anforderungen und Diensten, die Daten oder Inhalten entsprechen, beinhaltet. In mindestens einer Ausführungsform kann dieser Manager eine Anforderung zum Trainieren eines neuronalen Netzes empfangen und Daten für eine Anforderung an einen Trainingsmanager 1312 bereitstellen. In mindestens einer Ausführungsform kann der Trainingsmanager 1312 ein geeignetes zu verwendendes Modell oder Netz auswählen, wenn es nicht durch eine Anforderung vorgegeben ist, und ein Modell unter Verwendung relevanter Trainingsdaten trainieren. In mindestens einer Ausführungsform können Trainingsdaten ein Stapel von Daten sein, die in einem Trainingsdatenspeicher 1314 gespeichert sind, von der Client-Vorrichtung 1302 empfangen oder von einem Drittanbieter 1324 erhalten werden. In mindestens einer Ausführungsform kann der Trainingsmanager 1312 für die Trainingsdaten verantwortlich sein, wie etwa durch Verwendung eines LARC-basierten Ansatzes, wie in dieser Schrift erörtert. Ein Netz kann ein beliebiges geeignetes Netz sein, wie etwa ein rekurrentes neuronales Netz (RNN) oder ein faltungsorientiertes neuronales Netz (convolutional neural network - CNN). Sobald ein Netz trainiert und erfolgreich bewertet wurde, kann ein trainierte Netz zum Beispiel in einem Modellspeicher 1316 gespeichert werden, der verschiedene Modelle oder Netze für Benutzer, Anwendungen oder Dienste usw. speichern kann. In mindestens einer Ausführungsform kann es mehrere Modelle für eine einzige Anwendung oder Entität geben, die auf der Grundlage einer Reihe verschiedener Faktoren verwendet werden können.

[0062] In mindestens einer Ausführungsform kann zu einem nachfolgenden Zeitpunkt eine Anforderung von Inhalten (z. B. Pfadbestimmungen) oder Daten von der Client-Vorrichtung 1302 (oder einer anderen derartigen Vorrichtung) empfangen werden, die zumindest teilweise durch ein trainiertes neuronales Netz bestimmt oder beeinflusst werden. Diese Anforderung kann zum Beispiel Eingabedaten beinhalten, die unter Verwendung eines neuronalen Netzes verarbeitet werden sollen, um eine oder mehrere Inferenzen oder andere Ausgabewerte, Klassifizierungen oder Vorhersagen zu erhalten. In mindestens einer Ausführungsform können

Eingabedaten an der Schnittstellenschicht 1308 empfangen und an das Inferenzmodul 1318 geleitet werden, obwohl auch ein anderes System oder ein anderer Dienst verwendet werden kann. In mindestens einer Ausführungsform kann das Inferenzmodul 1318 ein geeignetes trainiertes Netz, wie etwa ein trainiertes tiefes neuronales Netz (DNN), wie in dieser Schrift beschrieben, aus dem Modellspeicher 1316 erhalten, wenn es nicht bereits lokal in dem Inferenzmodul 1318 gespeichert ist. Das Inferenzmodul 1318 kann Daten als Eingabe für ein trainiertes Netz bereitstellen, das dann eine oder mehrere Inferenzen als Ausgabe erzeugen kann. Dies kann zum Beispiel eine Klassifizierung einer Instanz von Eingabedaten beinhalten. In mindestens einer Ausführungsform können die Inferenzen dann an die Client-Vorrichtung 1302 zur Anzeige oder anderen Kommunikation an einen Benutzer übertragen werden. In mindestens einer Ausführungsform können Kontextdaten für einen Benutzer auch in einem Benutzerkontextdatenspeicher 1322 gespeichert werden, der Daten über einen Benutzer beinhalten kann, die als Eingabe für ein Netz beim Erzeugen von Inferenzen oder Bestimmen von Daten, die nach dem Erhalten von Instanzen an einen Benutzer zurückgegeben werden. In mindestens einer Ausführungsform können relevante Daten, die zumindest einige von Eingabe- oder Inferenzdaten beinhalten können, auch in einer lokalen Datenbank 1320 zur Verarbeitung zukünftiger Anforderungen gespeichert werden. In mindestens einer Ausführungsform kann ein Benutzer Konto- oder andere Informationen verwenden, um auf Ressourcen oder Funktionalität einer Anbieterumgebung zuzugreifen. In mindestens einer Ausführungsform, können, sofern zulässig und verfügbar, auch Benutzerdaten gesammelt und zum weiteren Trainieren von Modellen verwendet werden, um genauere Inferenzen für zukünftige Anforderungen bereitzustellen. In mindestens einer Ausführungsform können Anforderungen über eine Benutzerschnittstelle in einer Anwendung für maschinelles Lernen 1326, die auf der Client-Vorrichtung 1302 ausgeführt wird, empfangen und die Ergebnisse über dieselbe Schnittstelle angezeigt werden. Eine Client-Vorrichtung kann Ressourcen, wie etwa einen Prozessor 1328 und einen Speicher 1330, zum Erzeugen einer Anforderung und Verarbeiten von Ergebnissen oder einer Antwort sowie mindestens ein Datenspeicherelement 1332 zum Speichern von Daten für die Anwendung für maschinelles Lernen 1326 beinhalten.

[0063] In mindestens einer Ausführungsform ist ein Prozessor 1328 (oder ein Prozessor des Trainingsmanagers 1312 oder des Inferenzmoduls 1318) eine zentrale Verarbeitungseinheit (central processing unit - CPU). Wie bereits erwähnt, können Ressourcen in derartigen Umgebungen jedoch GPUs verwenden, um Daten zumindest für bestimmte Arten von Anforderungen zu verarbeiten. Mit Tausenden von Kernen sind GPUs dazu ausgelegt, umfangreiche parallele Arbeitslasten zu bewältigen und sind daher beim Deep Learning zum Trainieren neuronaler Netze und Erzeugen von Vorhersagen beliebt geworden. Während die Verwendung von GPUs für Offline-Erstellungen ein schnelleres Training größerer und komplexerer Modelle ermöglichte, impliziert die Offline-Erzeugung von Vorhersagen, dass entweder Eingabemerkmale zur Anforderungszeit nicht verwendet werden können oder dass Vorhersagen für alle Permutationen von Merkmalen erzeugt und in einer Lookup-Tabelle gespeichert werden müssen, um Echtzeitanforderungen zu bedienen. Wenn ein Deep-Learning-Rahmen einen CPU-Modus unterstützt und ein Modell klein und einfach genug ist, um eine Vorwärtskopplung auf einer CPU mit einer angemessenen Latenz durchzuführen, könnte ein Dienst auf einer CPU-Instanz ein Modell hosten. In diesem Fall kann das Trainieren offline auf einer GPU und die Inferenz in Echtzeit auf einer CPU durchgeführt werden. Wenn ein CPU-Ansatz nicht praktikabel ist, kann ein Dienst auf einer GPU-Instanz laufen. Da GPUs jedoch andere Eigenschaften hinsichtlich Rechenleistung und Kosten als CPUs aufweisen, kann das Ausführen eines Dienstes, der einen Laufzeitalgorithmus auf eine GPU auslagert, erfordern, dass er anders als ein CPU-basierter Dienst ausgelegt ist.

[0064] **Fig. 14** veranschaulicht ein System 1400, das in mindestens einer Ausführungsform verwendet werden kann, um Daten zu klassifizieren oder Inferenzen zu erzeugen. In mindestens einer Ausführungsform kann sowohl überwachtes als auch nicht überwachtes Training in mindestens einer in dieser Schrift erörterten Ausführungsform verwendet werden. In mindestens einer Ausführungsform wird ein Satz von Trainingsdaten 1402 (z. B. klassifizierten oder gekennzeichneten Daten) als Eingabe bereitgestellt, um als Trainingsdaten zu fungieren. In mindestens einer Ausführungsform können Trainingsdaten Instanzen von mindestens einem Objekttyp, für den ein neuronales Netz trainiert werden soll, sowie Informationen, die diesen Objekttyp identifizieren, beinhalten. In mindestens einer Ausführungsform können Trainingsdaten einen Satz von Bildern beinhalten, die jeweils eine Darstellung eines Objekttyps beinhalten, wobei jedes Bild zudem ein Label, Metadaten, eine Klassifizierung oder eine andere Information, die einen in einem jeweiligen Bild dargestellten Objekttyp identifizieren, beinhaltet oder diesen zugeordnet ist. Als Trainingsdaten können auch verschiedene andere Datentypen verwendet werden, die Textdaten, Audiodaten, Videodaten usw. beinhalten können. In mindestens einer Ausführungsform werden die Trainingsdaten 1402 als Trainingseingabe einem Trainingsmanager 1404 bereitgestellt. In mindestens einer Ausführungsform kann der Trainingsmanager 1404 ein System oder Dienst sein, das/der Hardware und Software beinhaltet, wie etwa eine oder mehrere Rechenvorrichtungen, die eine Trainingsanwendung zum Trainieren eines neuronalen Netzes (oder eines anderen Modells oder Algorithmus usw.) ausführen. In mindestens einer Ausführungsform empfängt der Trainingsma-

nager 1404 eine Anweisung oder Anforderung, die eine Art des Modells angibt, das für das Training verwendet werden soll. In mindestens einer Ausführungsform kann ein Modell ein beliebiges geeignetes statistisches Modell, ein beliebiges geeignetes Netz oder ein beliebiger geeigneter Algorithmus sein, das/der für derartige Zwecke nützlich ist, und ein künstliches neuronales Netz, einen Deep-Learning-Algorithmus, einen Learning Classifier, ein Bayes'sches Netz usw. beinhalten. In mindestens einer Ausführungsform kann der Trainingsmanager 1404 ein Ausgangsmodell oder ein anderes untrainiertes Modell aus einem geeigneten Speicher 1406 auswählen und Trainingsdaten 1402 verwenden, um ein Modell zu trainieren und ein trainiertes Modell 1408 (z. B. ein trainiertes tiefes neuronales Netz) erzeugen, das verwendet werden kann, um ähnliche Datentypen zu klassifizieren oder andere derartige Inferenzen zu erzeugen. In mindestens einer Ausführungsform, in der keine Trainingsdaten verwendet werden, kann dennoch ein geeignetes Ausgangsmodell für das Trainieren an Eingabedaten per Trainingsmanager 1404 ausgewählt werden.

[0065] In mindestens einer Ausführungsform kann ein Modell auf eine Reihe von verschiedenen Arten trainiert werden, die teilweise von einer Art des ausgewählten Modells abhängig sind. In mindestens einer Ausführungsform kann einem Algorithmus für maschinelles Lernen ein Satz von Trainingsdaten bereitgestellt werden, wobei ein Modell ein durch einen Trainingsprozess erstelltes Modellartefakt ist. In mindestens einer Ausführungsform enthält jede Instanz von Trainingsdaten enthält eine richtige Antwort (z. B. eine Klassifizierung), die als Ziel oder Zielattribut bezeichnet werden kann. In mindestens einer Ausführungsform findet ein Lernalgorithmus in den Trainingsdaten Muster, die die Attribute der Eingabedaten auf ein Ziel, eine vorherzusagende Antwort, abbilden, und wird ein Modell für maschinelles Lernen ausgegeben, das diese Muster erfasst. In mindestens einer Ausführungsform kann dann ein Modell für maschinelles Lernen verwendet werden, um Vorhersagen für neue Daten zu erhalten, für die kein Ziel vorgegeben ist.

[0066] In mindestens einer Ausführungsform kann ein Trainingsmanager 1404 aus einem Satz von Modellen für maschinelles Lernen auswählen, der binäre Klassifizierungs-, Multiklassen-Klassifizierungs- und Regressionsmodelle beinhaltet. In mindestens einer Ausführungsform kann die Art des zu verwendenden Modells zumindest teilweise von der Art des vorherzusagenden Ziels abhängig sein. In mindestens einer Ausführungsform sagen Modelle für maschinelles Lernen für binäre Klassifizierungsprobleme ein binäres Ergebnis vorher, wie etwa eine von zwei möglichen Klassen. In mindestens einer Ausführungsform kann ein Lernalgorithmus, wie etwa logistische Regression, verwendet werden, um binäre Klassifizierungsmodelle zu trainieren. In mindestens einer Ausführungsform ermöglichen Modelle für maschinelles Lernen für Multiklassen-Klassifizierungsprobleme, dass Vorhersagen für mehrere Klassen erzeugt werden, wie etwa um eines von mehr als zwei Ergebnissen vorherzusagen. Multinomiale logistische Regression kann für das Trainieren von Multiklassenmodellen nützlich sein. Modelle für maschinelles Lernen für Regressionsprobleme sagen einen numerischen Wert vorher. Lineare Regression kann für das Trainieren von Regressionsmodellen nützlich sein.

[0067] In mindestens einer Ausführungsform muss ein Trainingsmanager, um ein Modell für maschinelles Lernen gemäß einer Ausführungsform zu trainieren, eine Eingabe-Trainingsdatenquelle sowie andere Informationen, wie etwa den Namen eines Datenattributs, das ein vorherzusagendes Ziel enthält, erforderliche Datenumwandlungsanweisungen und Trainingsparameter zur Steuerung eines Lernalgorithmus, bestimmen. In mindestens einer Ausführungsform kann ein Trainingsmanager 1404 während eines Trainingsprozesses automatisch einen geeigneten Lernalgorithmus auf Grundlage einer in einer Trainingsdatenquelle vorgegebenen Zielart auswählen. In mindestens einer Ausführungsform können Algorithmen für maschinelles Lernen Parameter annehmen, die verwendet werden, um bestimmte Eigenschaften eines Trainingsprozesses und eines daraus resultierenden Modells für maschinelles Lernen zu steuern. Diese werden in dieser Schrift als Trainingsparameter bezeichnet. In mindestens einer Ausführungsform kann ein Trainingsmanager, wenn keine Trainingsparameter vorgegeben sind, Standardwerte verwenden, die bekanntermaßen für einen breiten Bereich von Aufgaben für maschinelles Lernen gut funktionieren. Beispiele für Trainingsparameter, für die Werte vorgegeben werden können, beinhalten eine maximale Modellgröße, eine maximale Anzahl von Durchläufen über Trainingsdaten, einen Mischtyp, einen Regularisierungstyp, eine Lernrate und eine Regularisierungsmenge. Es können Standardeinstellungen vorgegeben werden, mit Optionen zur Anpassung der Werte zur Feinabstimmung der Rechenleistung.

[0068] In mindestens einer Ausführungsform ist eine maximale Modellgröße eine Gesamtgröße, in Byte-Einheiten, der Muster, die während des Trainierens eines Modells erstellt werden. In mindestens einer Ausführungsform kann ein Modell standardmäßig mit einer bestimmten Größe erstellt werden, wie etwa ein Modell mit 100 MB. Wenn ein Trainingsmanager nicht in der Lage ist, genügend Muster zu bestimmen, um eine Modellgröße zu füllen, kann ein kleineres Modell erstellt werden. Wenn ein Trainingsmanager mehr Muster findet als in eine vorgegebene Größe passen, kann eine maximale Obergrenze erzwungen werden, indem

die Muster abgeschnitten werden, die die Qualität eines gelernten Modells am wenigsten beeinträchtigen. Die Wahl einer Modellgröße stellt Kontrolle über einen Kompromiss zwischen der Vorhersagequalität eines Modells und den Nutzungskosten bereit. In mindestens einer Ausführungsform können kleinere Modelle dazu führen, dass ein Trainingsmanager viele Muster entfernt, um in eine maximale Größenbegrenzung zu passen, was die Qualität von Vorhersagen beeinträchtigt. In mindestens einer Ausführungsform können bei größeren Modellen die Kosten für die Abfrage von Echtzeit-Vorhersagen höher sein. In mindestens einer Ausführungsform führen größere Eingabedatensätze nicht unbedingt zu größeren Modellen, da Modelle Muster und keine Eingabedaten speichern. In mindestens einer Ausführungsform ist bei wenigen und einfachen Mustern ein resultierendes Modell klein. Bei Eingabedaten, die eine große Anzahl von Rohattributen (Eingabespalten) oder abgeleiteten Merkmalen (Ausgaben von Datentransformationen) aufweisen, ist es wahrscheinlich, dass mehr Muster gefunden und während eines Trainingsprozesses gespeichert werden.

[0069] In mindestens einer Ausführungsform kann der Trainingsmanager 1404 mehrere Durchläufe oder Iterationen über Trainingsdaten durchführen, um zu versuchen, Muster zu entdecken. In mindestens einer Ausführungsform kann es eine Standardanzahl von Durchläufen, wie etwa zehn Durchläufe, geben, während in mindestens einer Ausführungsform eine maximale Anzahl von Durchläufen festgelegt werden kann, wie etwa bis zu hundert Durchläufe. In mindestens einer Ausführungsform kann kein Maximum festgelegt sein oder es kann ein Konvergenzkriterium oder ein anderer Faktor festgelegt sein, der ein Ende eines Trainingsprozesses auslöst. In mindestens einer Ausführungsform kann der Trainingsmanager 1404 eine Qualität von Mustern (wie etwa für die Modellkonvergenz) während des Trainings überwachen und das Training automatisch beenden, wenn es keine weiteren Datenpunkte oder Muster mehr zu entdecken gibt. In mindestens einer Ausführungsform können Datensätze mit nur wenigen Beobachtungen mehrere Durchläufe über die Daten erfordern, um eine ausreichend hohe Modellqualität zu erreichen. Größere Datensätze können viele ähnliche Datenpunkte enthalten, was eine Notwendigkeit einer großen Anzahl von Durchläufen reduzieren kann. Eine mögliche Auswirkung der Wahl von mehreren Datendurchläufen über Daten ist, dass das Modelltraining länger dauern und im Hinblick auf Ressourcen und Systemauslastung mehr kosten kann.

[0070] In mindestens einer Ausführungsform werden die Trainingsdaten vor dem Training oder zwischen Durchläufen von Training gemischt. In mindestens einer Ausführungsform ist das Mischen ein zufälliges oder pseudozufälliges Mischen, um eine wirklich zufällige Reihenfolge zu erzeugen, obwohl es einige Beschränkungen geben kann, um sicherzustellen, dass es keine Gruppierung bestimmter Datentypen gibt, oder gemischte Daten können neu gemischt werden, wenn eine derartige Gruppierung besteht, usw. In mindestens einer Ausführungsform wird durch das Mischen eine Reihenfolge oder Anordnung, in der Daten zum Trainieren verwendet werden, geändert, sodass ein Trainingsalgorithmus nicht auf Gruppierungen ähnlicher Datentypen oder auf einen einzigen Datentyp für zu viele Beobachtungen in Folge trifft. In mindestens einer Ausführungsform könnte ein Modell trainiert werden, um ein Objekt vorherzusagen. In mindestens einer Ausführungsform könnten Daten vor dem Hochladen nach Objekttyp sortiert werden. In mindestens einer Ausführungsform kann dann ein Algorithmus die Daten alphabetisch nach Objekttyp verarbeiten, wobei er zuerst nur auf Daten für einen bestimmten Objekttyp trifft. In mindestens einer Ausführungsform beginnt ein Modell, Muster für diesen Objekttyp zu lernen. In mindestens einer Ausführungsform trifft ein Modell dann nur auf Daten für einen zweiten Objekttyp und versucht ein Modell so anzupassen, dass es zu diesem Objekttyp passt, was zu einer Verschlechterung von Mustern führen kann, die zu einem ersten Objekttyp passen. Dieser plötzliche Wechsel zwischen Objekttypen kann ein Modell erzeugen, das nicht lernt, Objekttypen genau vorherzusagen. In mindestens einer Ausführungsform kann das Mischen in mindestens einer Ausführungsform durchgeführt werden, bevor ein Trainingsdatensatz in Trainings- und Bewertungsteilsätze aufgeteilt wird, sodass eine relativ gleichmäßige Verteilung von Datentypen für beide Phasen verwendet wird. In mindestens einer Ausführungsform kann der Trainingsmanager 1404 die Daten automatisch mischen, zum Beispiel unter Verwendung einer pseudozufälligen Mischtechnik.

[0071] In mindestens einer Ausführungsform kann der Trainingsmanager 1404 beim Erstellen eines Modells für maschinelles Lernen in mindestens einer Ausführungsform einem Benutzer ermöglichen, Einstellungen vorzugeben oder benutzerdefinierte Optionen anzuwenden. In mindestens einer Ausführungsform kann ein Benutzer eine oder mehrere Bewertungseinstellungen vorgeben, die einen Abschnitt der Eingabedaten angeben, der für die Bewertung einer Vorhersagequalität eines Modells für maschinelles Lernen reserviert werden soll. In mindestens einer Ausführungsform kann ein Benutzer eine Richtlinie vorgeben, die angibt, welche Attribute und Attributtransformationen für das Modelltraining verfügbar sind. In mindestens einer Ausführungsform kann der Benutzer außerdem verschiedene Trainingsparameter vorgeben, die bestimmte Eigenschaften eines Trainingsprozesses und eines daraus resultierenden Modells steuern.

[0072] In mindestens einer Ausführungsform kann, sobald ein Trainingsmanager bestimmt hat, dass das Training eines Modells abgeschlossen ist, wie etwa unter Verwendung mindestens eines der in dieser Schrift erörterten Endkriterien, das trainierte Modell 1408 zur Verwendung durch einen Klassifikator 1414 beim Klassifizieren von (oder anderweitigen Erzeugung von Inferenzen für) Validierungsdaten 1412 bereitgestellt werden. In mindestens einer Ausführungsform beinhaltet dies einen logischen Übergang zwischen einem Trainingsmodus für ein Modell und einem Inferenzmodus für ein Modell. In mindestens einer Ausführungsform wird das trainierte Modell 1408 jedoch zuerst an einen Bewerter 1410 weitergegeben, der eine Anwendung, einen Prozess oder einen Dienst beinhalten kann, die/der auf mindestens einer Rechenressource (z. B. einer CPU oder GPU mindestens eines Servers) zum Bewerten einer Qualität (oder eines anderen derartigen Aspekts) eines trainierten Modells ausgeführt wird. In mindestens einer Ausführungsform wird ein Modell bewertet, um zu bestimmen, ob dieses Modell bei der Vorhersage eines Ziels an neuen und zukünftigen Daten zumindest ein akzeptables Mindest- oder Schwellenniveau an Rechenleistung bereitstellt. Ist dies nicht der Fall, kann der Trainingsmanager 1404 damit fortfahren, dieses Modell zu trainieren. In mindestens einer Ausführungsform kann, da zukünftige Dateninstanzen oft unbekannte Zielwerte aufweisen, es wünschenswert sein, eine Genauigkeitsmetrik des maschinellen Lernens an Daten zu überprüfen, für die eine Zielantwort bekannt ist, und diese Bewertung als Ersatz für die Vorhersagegenauigkeit an zukünftigen Daten zu verwenden.

[0073] In mindestens einer Ausführungsform wird ein Modell unter Verwendung eines Teilsatzes von Trainingsdaten 1402 bewertet, die für das Training bereitgestellt wurden. Dieser Teilsatz kann unter Verwendung eines Misch- und Aufteilansatzes, wie vorstehend erörtert, bestimmt werden. In mindestens einer Ausführungsform wird dieser Teilsatz von Bewertungsdaten mit einem Ziel gekennzeichnet und kann somit als Ground-Truth-Quelle für die Bewertung dienen. Das Bewerten einer Vorhersagegenauigkeit eines Modells für maschinelles Lernen mit denselben Daten, die zum Trainieren verwendet wurden, ist nicht sinnvoll, da positive Bewertungen für Modelle erzeugt werden könnten, die sich an die Trainingsdaten erinnern anstatt aus ihnen zu verallgemeinern. In mindestens einer Ausführungsform wird, sobald das Training abgeschlossen ist, der Teilsatz der Bewertungsdaten unter Verwendung des trainierten Modells 1408 verarbeitet und der Bewerter 1410 kann die Genauigkeit dieses Modells bestimmen, indem er die Ground-Truth-Daten mit entsprechenden Ausgaben (oder Vorhersagen/Beobachtungen) dieses Modells vergleicht. In mindestens einer Ausführungsform kann der Bewerter 1410 in mindestens einer Ausführungsform eine Zusammenfassung oder Rechenleistungsmetrik bereitstellen, die angibt, wie gut vorhergesagte und wahre Werte übereinstimmen. In mindestens einer Ausführungsform kann, wenn ein trainiertes Modell nicht zumindest ein Kriterium für die minimale Rechenleistung oder einen anderen derartigen Genauigkeitsschwellenwert erfüllt, dann der Trainingsmanager 1404 angewiesen werden, weiteres Training durchzuführen oder in einigen Fällen versuchen, ein neues oder anderes Modell zu trainieren. In mindestens einer Ausführungsform kann, wenn das trainierte Modell 1408 die relevanten Kriterien erfüllt, dann ein trainiertes Modell zur Verwendung durch den Klassifikator 1414 bereitgestellt werden.

[0074] In mindestens einer Ausführungsform kann es beim Erstellen und Trainieren eines Modells für maschinelles Lernen in mindestens einer Ausführungsform wünschenswert sein, Modelleinstellungen oder Trainingsparameter vorzugeben, die zu einem Modell führen, dass in der Lage ist, genaue Vorhersagen zu treffen. In mindestens einer Ausführungsform beinhalten die Parameter die Anzahl von durchzuführenden Durchläufen (vorwärts und/oder rückwärts), eine Regularisierung oder Verfeinerung, eine Modellgröße und einen Mischtyp. In mindestens einer Ausführungsform könnte das Auswählen von Modellparametereinstellungen, die eine beste Vorhersageleistung an Bewertungsdaten erzeugen, zu einer Überanpassung eines Modells führen. In mindestens einer Ausführungsform kommt es zu einer Überanpassung, wenn sich ein Modell Muster gemerkt hat, die in Trainings- und Bewertungsdatenquellen vorkommen, aber Muster in Daten nicht verallgemeinert hat. Eine Überanpassung tritt häufig auf, wenn die Trainingsdaten alle in einer Bewertung verwendeten Daten beinhalten. In mindestens einer Ausführungsform kann ein Modell, das überangepasst wurde, bei der Bewertung eine gute Leistung erbringen, aber an neuen oder anderweitigen Validierungsdaten keine genauen Vorhersagen treffen. In mindestens einer Ausführungsform kann ein Trainingsmanager, um zu vermeiden, dass ein überangepasstes Modell als bestes Modell ausgewählt wird, zusätzliche Daten reservieren, um die Rechenleistung eines Modells zu validieren. Zum Beispiel könnte der Trainingsdatensatz in 60 Prozent für das Training und 40 Prozent für die Bewertung oder Validierung aufgeteilt werden, die wiederum in zwei oder mehr Phasen unterteilt werden können. In mindestens einer Ausführungsform kann nach dem Auswählen von Modellparametern, die für die Evaluierungsdaten gut funktionieren und zu einer Konvergenz bei einer Teilmenge von Validierungsdaten, wie etwa der Hälfte dieser Validierungsdaten, führen, eine zweite Validierung mit einem Rest dieser Validierungsdaten durchgeführt werden, um die Rechenleistung dieses Modells sicherzustellen. Wenn dieses Modell die Erwartungen an die Validierungsdaten erfüllt, dann erfolgt keine Überanpassung der Daten durch dieses Modell. In mindestens einer

Ausführungsform kann ein Testsatz oder ein zurückgehaltener Satz zum Testen der Parameter verwendet werden. In mindestens einer Ausführungsform hilft das Verwenden eines zweiten Validierungs- oder Testschritts bei der Auswahl geeigneter Modellparameter, um eine Überanpassung zu verhindern. Durch das Zurückhalten mehrere Daten aus einem Trainingsprozess zur Validierung sind jedoch weniger Daten für das Training verfügbar. Dies kann bei kleineren Datensätzen problematisch sein, da nicht genügend Daten für das Training verfügbar sein können. In mindestens einer Ausführungsform besteht ein Ansatz in einer derartigen Situation darin, eine Kreuzvalidierung durchzuführen, wie an anderer Stelle in dieser Schrift erörtert.

[0075] In mindestens einer Ausführungsform gibt es viele Metriken oder Einblicke, die verwendet werden können, um die Vorhersagegenauigkeit eines bestimmten Modells zu überprüfen und zu bewerten. In mindestens einer Ausführungsform enthält ein Bewertungsergebnis eine Metrik für die Vorhersagegenauigkeit, um einen Gesamterfolg eines Modells anzugeben, sowie Visualisierungen, um dabei zu helfen, die Genauigkeit eines Modells über eine Metrik für die Vorhersagegenauigkeit hinaus zu untersuchen. Ein Ergebnis kann auch eine Möglichkeit bereitstellen, die Auswirkung des Festlegens einer Bewertungsschwelle, wie etwa für eine binäre Klassifizierung, zu überprüfen, und kann Warnmeldungen zu Kriterien zur Überprüfung der Gültigkeit einer Bewertung erzeugen. Eine Wahl einer Metrik und einer Visualisierung kann zumindest teilweise von einer Art des zu bewertenden Modells abhängig sein.

[0076] In mindestens einer Ausführungsform kann ein trainiertes Modell für maschinelles Lernen, sobald es trainiert und zufriedenstellend bewertet wurde, verwendet werden, um eine Anwendung für maschinelles Lernen zu erstellen oder zu unterstützen. In einer Ausführungsform ist das Erstellen einer Anwendung für maschinelles Lernen ein iterativer Prozess, der eine Abfolge von Schritten beinhaltet. In mindestens einer Ausführungsform kann/können (ein) Kernproblem(e) des maschinellen Lernens in Bezug darauf, was beobachtet wird und welche Antwort ein Modell vorhersagen soll, formuliert werden. In mindestens einer Ausführungsform können dann Daten gesammelt, bereinigt und aufbereitet werden, um die Daten für die Verwendung durch Trainingsalgorithmen für Modelle für maschinelles Lernen geeignet zu machen. Diese Daten können visualisiert und analysiert werden, um Integritätsprüfungen durchzuführen, um eine Qualität von Daten zu validieren und die Daten zu verstehen. Es kann sein, dass Rohdaten (z. B. Eingabevariablen) und Antwortdaten (z. B. ein Ziel) nicht in einer Weise dargestellt sind, die zum Trainieren eines Modells mit hoher Vorhersageleistung verwendet werden kann. Daher kann es wünschenswert sein, Eingabedarstellungen oder -merkmale mit besserer Vorhersageleistung aus Rohvariablen zu konstruieren. Die sich daraus ergebenden Merkmale können einem Lernalgorithmus zugeführt werden, um Modelle zu erstellen und eine Qualität von Modellen an Daten zu bewerten, die aus der Modellbildung zurückgehalten wurden. Ein Modell kann dann verwendet werden, um Vorhersagen einer Zielantwort für neue Dateninstanzen zu erzeugen.

[0077] In mindestens einer Ausführungsform wird in dem System 1400 aus **Fig. 14** ein trainiertes Modell 1410 nach der Auswertung einem Klassifikator 1414 bereitgestellt oder zur Verfügung gestellt, der in der Lage ist, ein trainiertes Modell zu verwenden, um Validierungsdaten zu verarbeiten. In mindestens einer Ausführungsform kann dies zum Beispiel Daten beinhalten, die von Nutzern oder Dritten empfangen werden und nicht klassifiziert sind, wie etwa Abfragebilder, die nach Informationen darüber suchen, was in diesen Bildern dargestellt ist. In mindestens einer Ausführungsform können die Validierungsdaten durch einen Klassifikator unter Verwendung eines trainierten Modells verarbeitet werden, und die Ergebnisse 1416 (wie etwa Klassifizierungen oder Vorhersagen), die erzeugt werden, können an jeweilige Quellen zurückgesendet oder anderweitig verarbeitet oder gespeichert werden. In mindestens einer Ausführungsform und sofern eine derartige Verwendung zulässig ist, können diese nun klassifizierten Dateninstanzen in einem Trainingsdatenspeicher gespeichert werden, der für zukünftiges Trainieren des trainierten Modells 1408 durch einen Trainingsmanager verwendet werden kann. In mindestens einer Ausführungsform wird ein Modell kontinuierlich trainiert, wenn neue Daten verfügbar sind, aber in mindestens einer Ausführungsform werden diese Modelle periodisch neu trainiert, wie etwa einmal pro Tag oder Woche, in Abhängigkeit von Faktoren, wie etwa einer Größe eines Datensatzes oder einer Komplexität eines Modells.

[0078] In mindestens einer Ausführungsform kann der Klassifikator 1414 geeignete Hardware und Software zur Verarbeitung von Validierungsdaten 1412 unter Verwendung eines trainierten Modells beinhalten. In mindestens einer Ausführungsform beinhaltet ein Klassifikator einen oder mehrere Computerserver, die jeweils eine oder mehrere Grafikverarbeitungseinheiten (GPUs) aufweisen, die in der Lage sind, Daten zu verarbeiten. In mindestens einer Ausführungsform können die Konfiguration und das Design von GPUs sie für die Verwendung bei der Verarbeitung von Daten für maschinelles Lernen wünschenswerter machen als CPUs oder andere derartige Komponenten. In mindestens einer Ausführungsform kann ein trainiertes Modell in mindestens einer Ausführungsform in den GPU-Speicher geladen und eine empfangene Dateninstanz einer GPU zur Verarbeitung bereitgestellt werden. GPUs können eine viel größere Anzahl von Kernen aufweisen

als CPUs, und GPU-Kerne können auch viel weniger komplex sein. In mindestens einer Ausführungsform kann eine bestimmte GPU in der Lage sein, Tausende von Dateninstanzen gleichzeitig über verschiedene Hardware-Threads zu verarbeiten. In mindestens einer Ausführungsform kann eine GPU zudem dazu konfiguriert sein, den Gleitkommadurchsatz zu maximieren, was bedeutende zusätzliche Verarbeitungsvorteile für einen großen Datensatz bereitstellen kann.

[0079] In mindestens einer Ausführungsform können derartige Aufgaben selbst bei Verwendung von GPUs, Beschleunigern und anderer derartiger Hardware zur Beschleunigung von Aufgaben wie etwa dem Trainieren eines Modells oder der Klassifizierung von Daten unter Verwendung eines derartigen Modells immer noch bedeutende Zeit, Ressourcenzuweisung und Kosten erfordern. In mindestens einer Ausführungsform müssen, wenn ein Modell für maschinelles Lernen in 1400 Durchläufen trainiert werden soll und ein Datensatz 1.000.000 Dateninstanzen beinhaltet, die für das Trainieren verwendet werden sollen, alle Millionen Instanzen für jeden Durchlauf verarbeitet werden. Verschiedene Abschnitte einer Architektur können zudem von verschiedenen Arten von Vorrichtungen unterstützt werden. In mindestens einer Ausführungsform kann das Training unter Verwendung einer Reihe von Servern an einem logisch zentralisierten Ort durchgeführt werden, die als Dienst angeboten werden können, während die Klassifizierung der Rohdaten durch einen derartigen Dienst oder auf einer Client-Vorrichtung durchgeführt werden kann. Diese Vorrichtungen können auch im Besitz einer gleichen Entität oder mehrerer Entitäten sein und von diesen betrieben oder gesteuert werden.

[0080] In mindestens einer Ausführungsform kann ein in **Fig. 15** veranschaulichtes beispielhaftes neuronales Netz 1500 in mindestens einer Ausführungsform trainiert oder anderweitig verwendet werden. In mindestens einer Ausführungsform ist ein statistisches Modell ein künstliches neuronales Netz (artificial neural network - ANN), das mehrere Schichten von Knoten enthält, einschließlich einer Eingabeschicht 1502, einer Ausgabeschicht 1506 und mehrerer Schichten 1504 von Zwischenknoten, die oft als „verborgene“ Schichten bezeichnet werden, da die internen Schichten und Knoten in neuronalen Netzen normalerweise nicht sichtbar oder zugänglich sind. Obwohl in mindestens einer Ausführungsform zu Erklärungszwecken nur einige wenige Zwischenschichten dargestellt sind, versteht es sich, dass es keine Begrenzung für die Anzahl der Zwischenschichten gibt, die verwendet werden können, und dass eine beliebige Begrenzung der Schichten oft ein Faktor der Ressourcen oder der Zeit ist, die für die Verarbeitung unter Verwendung eines Modells erforderlich sind. In mindestens einer Ausführungsform können zusätzliche Arten von Modellen, Netzen, Algorithmen oder Prozessen ebenfalls verwendet werden, die andere Anzahlen oder Auswahlen von Knoten und Schichten beinhalten können. In mindestens einer Ausführungsform können die Validierungsdaten durch Schichten eines Netzes verarbeitet werden, um einen Satz von Inferenzen oder Inferenzwerten zu erzeugen, die dann einer Verlustfunktion 1508 zugeführt werden können.

[0081] In mindestens einer Ausführungsform sind alle Knoten einer bestimmten Schicht mit allen Knoten einer benachbarten Schicht verbunden. In mindestens einer Ausführungsform sind die Knoten einer Zwischenschicht dann jeweils mit den Knoten zweier benachbarter Schichten verbunden. In mindestens einer Ausführungsform werden die Knoten in einigen Modellen auch als Neuronen oder verbundene Einheiten bezeichnet und die Verbindungen zwischen den Knoten werden als Kanten bezeichnet. Jeder Knoten kann eine Funktion für empfangene Eingaben ausführen, wie etwa durch Verwenden einer vorgegebenen Funktion. In mindestens einer Ausführungsform können Knoten und Kanten während des Trainings unterschiedliche Gewichtungen erhalten und einzelne Schichten von Knoten können bestimmte Arten von Transformationen an empfangenen Eingaben durchführen, wobei diese Transformationen auch während des Trainings gelernt oder angepasst werden können. In mindestens einer Ausführungsform kann das Lernen überwachtes oder nicht überwachtes Lernen sein, was zumindest teilweise von einer Art der in einem Trainingsdatensatz enthaltenen Informationen abhängig sein kann. In mindestens einer Ausführungsform können verschiedene Arten von neuronalen Netzen verwendet werden, die ein neuronales Faltungsnetz (convolutional neural network - CNN), das eine Reihe von Faltungsschichten und einen Satz von Pooling-Schichten beinhaltet, beinhalten können und sich für Anwendungen wie etwa Bilderkennung als vorteilhaft erwiesen haben. CNNs können aufgrund einer relativ kleinen Anzahl von zu bestimmenden Parametern auch einfacher zu trainieren sein als andere Netze.

[0082] In mindestens einer Ausführungsform kann ein derartiges komplexes Modell für maschinelles Lernen unter Verwendung verschiedener Abstimmungsparametern trainiert werden. Das Auswählen der Parameter, das Anpassen eines Modells und das Bewerten eines Modells sind Teile eines Modellabstimmungsprozesses, der oft als Hyperparameteroptimierung bezeichnet wird. Eine derartige Abstimmung kann in mindestens einer Ausführungsform Introspektion eines zugrundeliegenden Modells oder zugrundeliegender Daten beinhalten. In einer Trainings- oder Produktionsumgebung kann ein robuster Arbeitsablauf wichtig sein, um eine Überanpassung von Hyperparametern zu vermeiden, wie an anderer Stelle in dieser Schrift erörtert. Die

Kreuzvalidierung und das Hinzufügen von Gaußschem Rauschen zu einem Trainingsdatensatz sind Techniken, die nützlich sein können, um eine Überanpassung an einen beliebigen Datensatz zu vermeiden. Für die Optimierung der Hyperparameter kann es wünschenswert sein, die Trainings- und Validierungssätze fest zu halten. In mindestens einer Ausführungsform können die Hyperparameter in bestimmten Kategorien abgestimmt werden, die Datenvorverarbeitung (wie etwa Übersetzung von Wörtern in Vektoren), Definition der CNN-Architektur (wie etwa Filtergrößen, Anzahl der Filter), Parameter des stochastischen Gradientenabstiegs (stochastic gradient descent - SGD) (zum Beispiel Lernrate) und Regularisierung oder Verfeinerung (zum Beispiel Dropout-Wahrscheinlichkeit) beinhalten können.

[0083] In mindestens einer Ausführungsform können die Instanzen eines Datensatzes während der Vorverarbeitung in einen niedrigeren dimensional Raum mit einer bestimmten Größe eingebettet werden. In mindestens einer Ausführungsform ist die Größe dieses Raums ein abzustimmender Parameter. In mindestens einer Ausführungsform enthält eine Architektur eines CNN viele abstimmbare Parameter. Ein Parameter für Filtergrößen kann eine Interpretation von Informationen darstellen, die einer Größe einer zu analysierenden Instanz entspricht. In der Computerlinguistik wird dies als n-Gramm-Größe bezeichnet. Ein beispielhaftes CNN verwendet drei verschiedene Filtergrößen, die potenziell unterschiedliche n-Gramm-Größen darstellen. Eine Anzahl von Filtern pro Filtergröße kann einer Tiefe eines Filters entsprechen. Jeder Filter versucht, etwas anderes aus einer Struktur einer Instanz zu lernen, wie etwa eine Satzstruktur für Textdaten. In einer Faltungsschicht kann eine Aktivierungsfunktion eine gleichgerichtete lineare Einheit und ein Pooling-Typ sein, der als Max-Pooling festgelegt ist. Die Ergebnisse können dann zu einem eindimensionalen Vektor verkettet werden und eine letzte Schicht ist vollständig mit einer zweidimensionalen Ausgabe verbunden. Dies entspricht einer binären Klassifizierung, auf die eine Optimierungsfunktion angewendet werden kann. Eine derartige Funktion ist eine Implementierung eines quadratischen Mittelwert-(Root Mean Square - RMS-)Propagationsverfahrens des Gradientenabstiegs, wobei beispielhafte Hyperparameter die Lernrate, die Batchgröße, die maximale Gradientennormale und Epochen beinhalten können. Bei neuronalen Netzen kann die Regularisierung ein äußerst wichtiger Faktor sein. In mindestens einer Ausführungsform können die Eingabedaten relativ spärlich sein. Ein wichtiger Hyperparameter in einer derartigen Situation kann ein Dropout auf einer vorletzten Schicht sein, der einen Anteil der Knoten darstellt, die nicht in jedem Trainingszyklus „feuern“. Ein beispielhafter Trainingsprozess kann verschiedene Hyperparameterkonfigurationen auf Grundlage einer Rückmeldung über die Rechenleistung vorheriger Konfigurationen vorschlagen. Dieses Modell kann mit einer vorgeschlagenen Konfiguration trainiert werden, an einem designierten Validierungssatz bewertet werden und die Rechenleistung kann berichtet werden. Dieser Prozess kann wiederholt werden, um zum Beispiel einen Kompromiss zwischen Exploration (mehr über verschiedene Konfigurationen lernen) und Exploitation (vorheriges Wissen nutzen, um bessere Ergebnisse zu erreichen) zu finden.

[0084] Da das Training von CNNs parallelisiert werden kann und GPU-fähige Rechenressourcen genutzt werden können, können mehrere Optimierungsstrategien für verschiedene Szenarien versucht werden. Ein komplexes Szenario ermöglicht das Abstimmen der Modellarchitektur und der Parameter für die Vorverarbeitung und den stochastischen Gradientenabstieg. Dadurch wird ein Modellkonfigurationsraum erweitert. In einem Basisszenario werden nur die Parameter für die Vorverarbeitung und den stochastischen Gradientenabstieg abgestimmt. In einem komplexen Szenario kann es eine größere Anzahl von Konfigurationsparametern geben als in einem Basisszenario. Das Abstimmen in einem gemeinsamen Raum kann unter Verwendung einer linearen oder exponentiellen Anzahl von Schritten durchgeführt werden, wobei eine Optimierungsschleife für Modelle durchlaufen wird. Die Kosten für einen derartigen Abstimmungsprozess können bedeutend geringer sein als für Abstimmungsprozesse, wie etwa Zufallssuche und Gittersuche, ohne einen bedeutenden Verlust der Rechenleistung.

[0085] In mindestens einer Ausführungsform kann Rückpropagierung verwendet werden, um einen Gradienten zu berechnen, der zur Bestimmung von Gewichtungen für ein neuronales Netz verwendet wird. Rückpropagierung ist eine Form der Differenzierung und kann von einem Gradientenabstiegs-Optimierungsalgorithmus verwendet werden, um die auf verschiedene Knoten oder Neuronen angewendeten Gewichtungen anzupassen, wie vorstehend erörtert. Die Gewichtungen können unter Verwendung eines Gradienten einer relevanten Verlustfunktion bestimmt werden. Die Rückpropagierung kann eine Ableitung einer Verlustfunktion in Bezug auf die durch ein statistisches Modell erzeugte Ausgabe verwenden. Wie bereits erwähnt, können verschiedene Knoten zugehörige Aktivierungsfunktionen aufweisen, die die Ausgabe jeweiliger Knoten definieren. Verschiedene Aktivierungsfunktionen können nach Bedarf verwendet werden und radiale Basisfunktionen (RBFs) und Sigmoiden beinhalten, die von verschiedenen Stützvektormaschinen (support vector machines - SVM) zur Transformation von Daten verwendet werden können. Eine Aktivierungsfunktion einer Zwischenschicht von Knoten wird in dieser Schrift als Kernel des inneren Produkts bezeichnet. Diese Funk-

tionen können zum Beispiel Identitätsfunktionen, Stufenfunktionen, Sigmoidfunktionen, Rampenfunktionen usw. beinhalten. Aktivierungsfunktionen können auch linear oder nicht linear sein.

[0086] In mindestens einer Ausführungsform wird ein untrainiertes neuronales Netz unter Verwendung eines Trainingsdatensatzes trainiert. In mindestens einer Ausführungsform ist der Trainingsrahmen ein PyTorch-Rahmen, ein TensorFlow-, Boost-, Caffe-, Microsoft-Cognitive-Toolkit/CNTK-, MXNet-, Chainer-, Keras-, Deeplearning4j- oder ein anderer Trainingsrahmen. In mindestens einer Ausführungsform trainiert der Trainingsrahmen ein untrainiertes neuronales Netz und ermöglicht, dass es unter Verwendung von in dieser Schrift beschriebenen Verarbeitungsressourcen trainiert wird, um ein trainiertes neuronales Netz zu erzeugen. In mindestens einer Ausführungsform können die Gewichtungen zufällig oder durch Vortraining unter Verwendung eines Deep-Belief-Netzes gewählt werden. In mindestens einer Ausführungsform kann das Training entweder auf überwachter, teilweise überwachter oder nicht überwachter Weise durchgeführt werden.

[0087] In mindestens einer Ausführungsform wird das untrainierte neuronale Netz unter Verwendung von überwachtem Lernen trainiert, wobei der Trainingsdatensatz eine Eingabe beinhaltet, die mit einer gewünschten Ausgabe für eine Eingabe gepaart ist, oder wobei der Trainingsdatensatz eine Eingabe beinhaltet, die eine bekannte Ausgabe aufweist, und eine Ausgabe des neuronalen Netzes manuell bewertet wird. In mindestens einer Ausführungsform wird das untrainierte neuronale Netz auf überwachter Weise trainiert, verarbeitet es Eingaben aus dem Trainingsdatensatz und vergleicht es die resultierenden Ausgaben mit einem Satz von erwarteten oder gewünschten Ausgaben. In mindestens einer Ausführungsform werden Fehler dann durch das untrainierte neuronale Netz rückpropagiert. In mindestens einer Ausführungsform passt der Trainingsrahmen Gewichtungen an, die das untrainierte neuronale Netz steuern. In mindestens einer Ausführungsform beinhaltet der Trainingsrahmen Werkzeuge, um zu überwachen, wie gut das untrainierte neuronale Netz zu einem Modell konvergiert, wie etwa dem trainierten neuronalen Netz, das dazu geeignet ist, korrekte Antworten zu erzeugen, wie etwa in dem Ergebnis, die auf bekannten Eingabedaten, wie etwa neuen Daten, basieren. In mindestens einer Ausführungsform trainiert der Trainingsrahmen das untrainierte neuronale Netz wiederholt, während die Gewichtungen angepasst werden, um eine Ausgabe des untrainierten neuronalen Netzes unter Verwendung einer Verlustfunktion und eines Anpassungsalgorithmus, wie etwa des stochastischen Gradientenabstiegs, zu verfeinern. In mindestens einer Ausführungsform trainiert der Trainingsrahmen das untrainierte neuronale Netz, bis das untrainierte neuronale Netz eine gewünschte Genauigkeit erreicht. In mindestens einer Ausführungsform kann das trainierte neuronale Netz dann eingesetzt werden, um eine beliebige Anzahl von Operationen für maschinelles Lernen zu implementieren.

[0088] In mindestens einer Ausführungsform wird das untrainierte neuronale Netz unter Verwendung von nicht überwachtem Lernen trainiert, wobei das untrainierte neuronale Netz versucht, sich selbst unter Verwendung von nicht gekennzeichneten Daten zu trainieren. In mindestens einer Ausführungsform beinhaltet der Trainingsdatensatz für nicht überwachtes Lernen Eingabedaten ohne beliebige zugeordnete Ausgabedaten oder „Ground-Truth“-Daten. In mindestens einer Ausführungsform kann das untrainierte neuronale Netz Gruppierungen innerhalb des Trainingsdatensatzes lernen und bestimmen, wie einzelne Eingaben mit dem untrainierten Datensatz in Beziehung stehen. In mindestens einer Ausführungsform kann nicht überwachtes Training verwendet werden, um eine selbstorganisierende Karte zu erzeugen, die eine Art von trainiertem neuronalen Netz ist, das in der Lage ist, Operationen durchzuführen, die beim Reduzieren der Dimensionalität neuer Daten nützlich sind. In mindestens einer Ausführungsform kann nicht überwachtes Training auch verwendet werden, um Anomaliedetektion durchzuführen, was die Identifizierung von Datenpunkten in einem neuen Datensatz ermöglicht, die von normalen Mustern des neuen Datensatzes abweichen.

[0089] In mindestens einer Ausführungsform kann halbüberwachtes Lernen verwendet werden, wobei es sich um eine Technik handelt, bei welcher der Trainingsdatensatz eine Mischung aus gekennzeichneten und nicht gekennzeichneten Daten beinhaltet. In mindestens einer Ausführungsform kann der Trainingsrahmen verwendet werden, um inkrementelles Lernen durchzuführen, wie etwa durch Transferlernetzwerke. In mindestens einer Ausführungsform ermöglicht das inkrementelle Lernen es dem trainierten neuronalen Netz, sich an neue Daten anzupassen, ohne das Wissen zu vergessen, das dem trainierten neuronalen Netz während des anfänglichen Trainings beigebracht wurde.

INFERENZ- UND TRAININGSLOGIK

[0090] Fig. 16A veranschaulicht eine Inferenz- und/oder Trainingslogik 1615, die verwendet wird, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet

sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt.

[0091] In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 ohne Einschränkung Code- und/oder Datenspeicher 1601 beinhalten, um Vorwärts- und/oder Ausgabegewichtungs- und/oder Eingabe-/Ausgabedaten und/oder andere Parameter zu speichern, um Neuronen oder Schichten eines neuronalen Netzes zu konfigurieren, das in Aspekten einer oder mehrerer Ausführungsformen trainiert und/oder zum Inferenzieren verwendet wird. In mindestens einer Ausführungsform kann die Trainingslogik 1615 einen Code- und/oder Datenspeicher 1601 beinhalten oder an diesen gekoppelt sein, um Graphencode oder andere Software zum Steuern der Zeitsteuerung und/oder Reihenfolge zu speichern, in der Gewichtungs- und/oder andere Parameterinformationen geladen werden sollen, um Logik zu konfigurieren, einschließlich Integer- und/oder Gleitkommaeinheiten (zusammen als arithmetisch-logische Einheiten (arithmetic logic units - ALUs) bezeichnet). In mindestens einer Ausführungsform lädt Code, wie etwa Graphencode, Gewichtungs- oder andere Parameterinformationen in Prozessor-ALUs auf Grundlage einer Architektur eines neuronalen Netzes, dem dieser Code entspricht. In mindestens einer Ausführungsform speichert der Code- und/oder Datenspeicher 1601 Gewichtungssparameter und/oder Eingabe-/Ausgabedaten jeder Schicht eines neuronalen Netzes, die in Verbindung mit einer oder mehreren Ausführungsformen trainiert oder verwendet wird, während der Vorwärtspropagierung von Eingabe-/Ausgabedaten und/oder Gewichtungsparametern während des Trainierens und/oder Inferenzierens unter Verwendung von Aspekten einer oder mehrerer Ausführungsformen. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Code- und/oder Datenspeichers 1601 in einem anderen chipinternen oder chipexternen Datenspeicher, einschließlich des L1-, L2- oder L3-Caches oder Systemspeichers eines Prozessors, enthalten sein.

[0092] In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Code- und/oder Datenspeichers 1601 zu einem oder mehreren Prozessoren oder anderen Hardware-Logikvorrichtungen oder -Schaltungen intern oder extern sein. In mindestens einer Ausführungsform kann der Code und/oder Code- und/oder Datenspeicher 1601 Cache-Speicher, dynamischer zufällig adressierbarer Speicher (dynamic randomly addressable memory - „DRAM“), statischer zufällig adressierbarer Speicher (static randomly addressable memory - „SRAM“), nicht flüchtiger Speicher (z. B. Flash-Speicher) oder anderer Speicher sein. In mindestens einer Ausführungsform kann die Wahl, ob der Code und/oder Code- und/oder Datenspeicher 1601 zu einem Prozessor, der zum Beispiel aus DRAM, SRAM, Flash oder einer anderen Speicherart besteht, intern oder extern ist, von dem verfügbaren Speicher auf dem Chip oder außerhalb des Chips, den Latenzanforderungen der durchzuführenden Trainings- und/oder Ableitungsfunktionen, der Batchgröße der Daten, die bei der Ableitung und/oder dem Training eines neuronalen Netzes verwendet werden, oder einer Kombination dieser Faktoren abhängen.

[0093] In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 ohne Einschränkung einen Code- und/oder Datenspeicher 1605 beinhalten, um Rückwärts- und/oder Ausgabegewichtungs- und/oder Eingabe-/Ausgabedaten zu speichern, die Neuronen oder Schichten eines neuronalen Netzes entsprechen, das in Aspekten einer oder mehrerer Ausführungsformen trainiert und/oder zum Inferenzieren verwendet wird. In mindestens einer Ausführungsform speichert der Code- und/oder Datenspeicher 1605 Gewichtungssparameter und/oder Eingabe-/Ausgabedaten jeder Schicht eines neuronalen Netzes, die in Verbindung mit einer oder mehreren Ausführungsformen trainiert oder verwendet wird, während der Rückwärtspropagierung von Eingabe-/Ausgabedaten und/oder Gewichtungssparametern während des Trainierens und/oder Inferenzierens unter Verwendung von Aspekten einer oder mehrerer Ausführungsformen. In mindestens einer Ausführungsform kann die Trainingslogik 1615 einen Code- und/oder Datenspeicher 1605 beinhalten oder an diesen gekoppelt sein, um Graphencode oder andere Software zum Steuern der Zeitsteuerung und/oder Reihenfolge zu speichern, in der Gewichtungs- und/oder andere Parameterinformationen geladen werden sollen, um Logik zu konfigurieren, einschließlich Integer- und/oder Gleitkommaeinheiten (zusammen als arithmetisch-logische Einheiten (ALUs) bezeichnet). In mindestens einer Ausführungsform lädt Code, wie etwa Graphencode, Gewichtungs- oder andere Parameterinformationen in Prozessor-ALUs auf Grundlage einer Architektur eines neuronalen Netzes, dem dieser Code entspricht. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Code- und/oder Datenspeichers 1605 in einem anderen chipinternen oder chipexternen Datenspeicher, einschließlich des L1-, L2- oder L3 - Caches oder Systemspeichers eines Prozessors, enthalten sein. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Code- und/oder Datenspeichers 1605 zu einem oder mehreren Prozessoren oder anderen Hardware-Logikvorrichtungen oder -Schaltungen intern oder extern sein. In mindestens einer Ausführungsform kann der Code- und/oder Datenspeicher 1605 Cache-Speicher, DRAM, SRAM, nicht flüchtiger Speicher (z. B. Flash-Speicher) oder anderer Speicher sein. In mindestens einer Ausführungsform kann die Wahl, ob der Code- und/oder Datenspeicher 1605 zu einem Prozessor, der zum Beispiel aus DRAM, SRAM, Flash

oder einer anderen Speicherart besteht, intern oder extern ist, von dem verfügbaren Speicher auf dem Chip oder außerhalb des Chips, den Latenzanforderungen der durchzuführenden Trainings- und/oder Ableitungsfunktionen, der Batchgröße der Daten, die bei der Ableitung und/oder dem Training eines neuronalen Netzes verwendet werden, oder einer Kombination dieser Faktoren abhängen.

[0094] In mindestens einer Ausführungsform können der Code- und/oder Datenspeicher 1601 und der Code- und/oder Datenspeicher 1605 separate Speicherstrukturen sein. In mindestens einer Ausführungsform können der Code- und/oder Datenspeicher 1601 und der Code- und/oder Datenspeicher 1605 dieselbe Speicherstruktur sein. In mindestens einer Ausführungsform können der Code- und/oder Datenspeicher 1601 und der Code- und/oder Datenspeicher 1605 teilweise dieselbe Speicherstruktur und teilweise separate Speicherstrukturen sein. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Code- und/oder Datenspeichers 1601 und/oder des Code- und/oder Datenspeichers 1605 in einem anderen chipinternen oder chipexternen Datenspeicher, einschließlich des L1-, L2- oder L3-Caches oder Systemspeichers eines Prozessors, enthalten sein.

[0095] In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 ohne Einschränkung eine oder mehrere arithmetisch-logische Einheit(en) („ALU(s)“) 1610 beinhalten, einschließlich Integer- und/oder Gleitkommaeinheiten, um logische und/oder mathematische Operationen durchzuführen, die zumindest teilweise auf Trainings- und/oder Inferenzcode (z. B. Graphencode) basieren oder dadurch angegeben werden, wobei ein Ergebnis davon Aktivierungen (z. B. Ausgabewerte von Schichten oder Neuronen innerhalb eines neuronalen Netzes) produzieren kann, die in einem Aktivierungsspeicher 1620 gespeichert sind und die Funktionen von Eingabe/Ausgabe- und/oder Gewichtungssparameterdaten sind, die in dem Code- und/oder Datenspeicher 1601 und/oder dem Code- und/oder Datenspeicher 1605 gespeichert sind. In mindestens einer Ausführungsform werden in dem Aktivierungsspeicher 1620 gespeicherte Aktivierungen gemäß linearer algebraischer und/oder matrixbasierter Mathematik erzeugt, die durch die ALU(s) 1610 als Reaktion auf das Durchführen von Anweisungen oder anderem Code durchgeführt wird, wobei in dem Code- und/oder Datenspeicher 1605 und/oder dem Code- und/oder Datenspeicher 1601 gespeicherte Gewichtungswerte als Operanden zusammen mit anderen Werten, wie etwa Verzerrungswerten, Gradienteninformationen, Momentwerten oder anderen Parametern oder Hyperparametern, verwendet werden, von denen beliebige oder alle in dem Code- und/oder Datenspeicher 1605 oder dem Code- und/oder Datenspeicher 1601 oder einem anderen chipinternen oder -externen Speicher gespeichert sein können.

[0096] In mindestens einer Ausführungsform sind die ALU(s) 1610 innerhalb eines oder mehrerer Prozessoren oder anderer Hardware-Logikvorrichtungen oder -Schaltungen enthalten, während in einer anderen Ausführungsform die ALU(s) 1610 zu einem Prozessor oder einer anderen Hardware-Logikvorrichtung oder -Schaltung extern sein können, der/die sie verwendet (z. B. ein Koprozessor). In mindestens einer Ausführungsform können die ALUs 1610 innerhalb der Ausführungseinheiten eines Prozessors oder anderweitig innerhalb einer Bank von ALUs enthalten sein, worauf die Ausführungseinheiten eines Prozessors zugreifen können, entweder innerhalb des gleichen Prozessors oder verteilt auf unterschiedliche Prozessoren unterschiedlichen Typs (z. B. zentrale Verarbeitungseinheiten, Grafikverarbeitungseinheiten, Festfunktionseinheiten usw.). In mindestens einer Ausführungsform können sich der Code- und/oder Datenspeicher 1601, der Code- und/oder Datenspeicher 1605 und der Aktivierungsspeicher 1620 auf demselben Prozessor oder einer anderen Hardware-Logikvorrichtung oder -schaltung befinden, während sie sich in einer anderen Ausführungsform in verschiedenen Prozessoren oder anderen Hardware-Logikvorrichtungen oder -schaltungen oder in einer Kombination aus gleichen und verschiedenen Prozessoren oder anderen Hardware-Logikvorrichtungen oder -schaltungen befinden können. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Aktivierungsspeichers 1620 in einem anderen chipinternen oder chipexternen Datenspeicher, einschließlich des L1-, L2- oder L3-Caches oder Systemspeichers eines Prozessors, enthalten sein. Des Weiteren kann der Inferenz- und/oder Trainingscode mit anderem Code gespeichert sein, auf den ein Prozessor oder eine andere Hardware-Logik oder -Schaltung zugreifen kann und der unter Verwendung der Abruf-, Decodier-, Planungs-, Ausführungs-, Ausscheidungs- und/oder anderen Logikschaltungen eines Prozessors abgerufen und/oder verarbeitet wird.

[0097] In mindestens einer Ausführungsform kann der Aktivierungsspeicher 1620 Cache-Speicher, DRAM, SRAM, nicht flüchtiger Speicher (z. B. Flash-Speicher) oder anderer Speicher sein. In mindestens einer Ausführungsform kann sich der Aktivierungsspeicher 1620 ganz oder teilweise innerhalb oder außerhalb eines oder mehrerer Prozessoren oder anderer logischer Schaltungen befinden. In mindestens einer Ausführungsform kann die Wahl, ob der Code- und/oder Aktivierungsspeicher 1620 zu einem Prozessor, der zum Beispiel aus DRAM, SRAM, Flash oder einem anderen Speichertyp besteht, intern oder extern ist, von dem verfügbaren Speicher auf dem Chip oder außerhalb des Chips, den Latenzanforderungen der durchzuführenden Trai-

nings- und/oder Ableitungsfunktionen, der Batchgröße der Daten, die bei der Ableitung und/oder dem Training eines neuronalen Netzes verwendet werden, oder einer Kombination dieser Faktoren abhängig sein. In mindestens einer Ausführungsform kann die in **Fig. 16A** veranschaulichte Inferenz- und/oder Trainingslogik 1615 in Verbindung mit einer anwendungsspezifischen integrierten Schaltung (application-specific integrated circuit - ASIC) verwendet werden, wie etwa der TensorFlow® Processing Unit von Google, einer Inferenzverarbeitungseinheit (inference processing unit - IPU) von Graphcore™ oder einem Prozessor vom Typ Nervana® (z. B. „Lake Crest“) der Intel Corp. In mindestens einer Ausführungsform kann die in **Fig. 16A** veranschaulichte Inferenz- und/oder Trainingslogik 1615 in Verbindung mit Hardware einer zentralen Verarbeitungseinheit („CPU“), Hardware einer Grafikverarbeitungseinheit („GPU“) oder anderer Hardware, wie etwa feldprogrammierbaren Gate-Arrays („FPGAs“) verwendet werden.

[0098] **Fig. 16B** veranschaulicht eine Inferenz- und/oder Trainingslogik 1615 gemäß mindestens einer oder mehreren Ausführungsformen. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 ohne Einschränkung Hardware-Logik beinhalten, in der Rechenressourcen dediziert oder anderweitig ausschließlich in Verbindung mit Gewichtungswerten oder anderen Informationen verwendet werden, die einer oder mehreren Schichten von Neuronen innerhalb eines neuronalen Netzes entsprechen. In mindestens einer Ausführungsform kann die in **Fig. 16B** veranschaulichte Inferenz- und/oder Trainingslogik 1615 in Verbindung mit einer anwendungsspezifischen integrierten Schaltung (ASIC) verwendet werden, wie etwa der TensorFlow® Processing Unit von Google, einer Inferenzverarbeitungseinheit (inference processing unit - IPU) von Graphcore™ oder einem Prozessor vom Typ Nervana® (z. B. „Lake Crest“) der Intel Corp. In mindestens einer Ausführungsform kann die in **Fig. 16B** veranschaulichte Inferenz- und/oder Trainingslogik 1615 in Verbindung mit Hardware einer zentralen Verarbeitungseinheit („CPU“), Hardware einer Grafikverarbeitungseinheit („GPU“) oder anderer Hardware, wie etwa feldprogrammierbaren Gate-Arrays („FPGAs“) verwendet werden. In mindestens einer Ausführungsform beinhaltet die Inferenz- und/oder Trainingslogik 1615 ohne Einschränkung den Code- und/oder Datenspeicher 1601 und den Code- und/oder Datenspeicher 1605, die zum Speichern von Code (z.B. Graphencode), Gewichtungswerten und/oder anderen Informationen, einschließlich Verzerrungswerten, Gradienteninformationen, Momentwerten und/oder anderer Parameter- oder Hyperparameterinformationen, verwendet werden können. In mindestens einer in **Fig. 16B** veranschaulichten Ausführungsform ist jeder von dem Code- und/oder Datenspeicher 1601 und dem Code- und/oder Datenspeicher 1605 einer dedizierten Rechenressource, wie etwa der Rechen-Hardware 1602 bzw. der Rechen-Hardware 1606, zugeordnet. In mindestens einer Ausführungsform umfasst jede der Rechen-Hardware 1602 und der Rechen-Hardware 1606 eine oder mehrere ALUs, die mathematische Funktionen, wie etwa lineare algebraische Funktionen, nur an Informationen durchführen, die in dem Code- und/oder Datenspeicher 1601 bzw. dem Code- und/oder Datenspeicher 1605 gespeichert sind, wobei das Ergebnis davon in dem Aktivierungsspeicher 1620 gespeichert wird.

[0099] In mindestens einer Ausführungsform entspricht jeder der Code- und/oder Datenspeicher 1601 und 1605 und die entsprechende Rechen-Hardware 1602 bzw. 1606 verschiedenen Schichten eines neuronalen Netzes, sodass die resultierende Aktivierung von einem „Speicher-/Rechenpaar 1601/1602“ des Code- und/oder Datenspeichers 1601 und der Rechenhardware 1602 als Eingabe für das „Speicher-/Rechenpaar 1605/1606“ des Code- und/oder Datenspeichers 1605 und der Rechen-Hardware 1606 bereitgestellt wird, um die konzeptionelle Organisation eines neuronalen Netzes zu spiegeln. In mindestens einer Ausführungsform kann jedes der Speicher-/Rechenpaare 1601/1602 und 1605/1606 mehr als einer Schicht eines neuronalen Netzes entsprechen. In mindestens einer Ausführungsform können zusätzliche Speicher-/Rechenpaare (nicht gezeigt) im Anschluss an oder parallel zu den Speicher-/Rechenpaaren 1601/1602 und 1605/1606 in die Inferenz- und/oder Trainingslogik 1615 integriert werden.

RECHENZENTRUM

[0100] **Fig. 17** veranschaulicht ein beispielhaftes Rechenzentrum 1700, in dem mindestens eine Ausführungsform verwendet werden kann. In mindestens einer Ausführungsform beinhaltet das Rechenzentrum 1700 eine Rechenzentrumsinfrastrukturschicht 1710, eine Rahmenschicht 1720, eine Softwareschicht 1730 und eine Anwendungsschicht 1740.

[0101] In mindestens einer Ausführungsform, wie in **Fig. 17** gezeigt, kann die Rechenzentrumsinfrastrukturschicht 1710 einen Ressourcen-Orchestrator 1712, gruppierte Rechenressourcen 1714 und Knoten-Rechenressourcen (node computing resources - „Knoten-C.R.s“) 1716(1)-1716(N) beinhalten, wobei „N“ eine beliebige positive ganze Zahl darstellt. In mindestens einer Ausführungsform können die Knoten-C.R.s 1716(1)-1716(N) eine beliebige Anzahl von zentralen Verarbeitungseinheiten („CPUs“) oder anderen Prozessoren (einschließlich Beschleunigern, feldprogrammierbaren Gate-Arrays (FPGAs), Grafikprozessoren usw.),

Arbeitsspeichervorrichtungen (z. B. dynamischer Festwertspeicher), Datenspeichervorrichtungen (z. B. Solid-State- oder Festplattenlaufwerke), Netzwerk-Eingabe-/Ausgabe-(„NW-E/A“-)Vorrichtungen, Netzwerk-Switches, virtuellen Maschinen („VMs“), Leistungsmodulen und Kühlmodulen usw. beinhalten, sind aber nicht darauf beschränkt. In mindestens einer Ausführungsform kann es sich bei einer oder mehreren Knoten-C.R.s unter den Knoten-C.R.s 1716(1)-1716(N) um einen Server handeln, der eine oder mehrere der vorstehend erwähnten Rechenressourcen aufweist.

[0102] In mindestens einer Ausführungsform können die gruppierten Rechenressourcen 1714 separate Gruppierungen von Knoten-C.R.s beinhalten, die innerhalb eines oder mehrerer Racks (nicht gezeigt) oder vieler Racks untergebracht sind, die in Rechenzentren an verschiedenen geografischen Standorten untergebracht sind (ebenfalls nicht gezeigt). Separate Gruppierungen von Knoten-C.R.s innerhalb der gruppierten Rechenressourcen 1714 können gruppierte Rechen-, Netzwerk-, Arbeitsspeicher- oder Datenspeicherressourcen beinhalten, die zur Unterstützung einer oder mehrerer Arbeitslasten konfiguriert oder zugewiesen sein können. In mindestens einer Ausführungsform können mehrere Knoten-C.R. s, die CPUs oder Prozessoren beinhalten, in einem oder mehreren Racks gruppiert sein, um Rechenressourcen bereitzustellen, um eine oder mehrere Arbeitslasten zu unterstützen. In mindestens einer Ausführungsform können ein oder mehrere Racks auch eine beliebige Anzahl von Leistungsmodulen, Kühlmodulen und Netzwerk-Switches in beliebiger Kombination beinhalten.

[0103] In mindestens einer Ausführungsform kann der Ressourcen-Orchestrator 1712 eine oder mehrere Knoten-C.R.s 1716(1)-1716(N) und/oder gruppierte Rechenressourcen 1714 konfigurieren oder anderweitig steuern. In mindestens einer Ausführungsform kann der Ressourcen-Orchestrator 1712 eine Verwaltungsentität für Softwaregestaltungsinfrastruktur (software design infrastructure - „SDI“) für das Rechenzentrum 1700 beinhalten. In mindestens einer Ausführungsform kann der Ressourcen-Orchestrator Hardware, Software oder eine Kombination davon beinhalten.

[0104] In mindestens einer Ausführungsform, wie in **Fig. 17** gezeigt, beinhaltet die Rahmenschicht 1720 einen Aufgabenplaner 1722, einen Konfigurations-Manager 1724, einen Ressourcen-Manager 1726 und ein verteiltes Dateisystem 1728. In mindestens einer Ausführungsform kann die Rahmenschicht 1720 einen Rahmen zum Unterstützen von Software 1732 der Software-Schicht 1730 und/oder einer oder mehreren Anwendung(en) 1742 der Anwendungsschicht 1740 beinhalten. In mindestens einer Ausführungsform kann/können die Software 1732 oder die Anwendung(en) 1742 jeweils webbasierte Dienstsoftware oder -anwendungen beinhalten, wie etwa diejenigen, die von Amazon Web Services, Google Cloud und Microsoft Azure bereitgestellt werden. In mindestens einer Ausführungsform kann es sich bei der Rahmenschicht 1720 um eine Art von freiem und Open-Source-Software-Webanwendungsrahmen, wie etwa Apache Spark™ (im Folgenden „Spark“) handeln, der das verteilte Dateisystem 1728 für die Verarbeitung großer Datenmengen (z. B. „Big Data“) verwenden kann, ohne darauf beschränkt zu sein. In mindestens einer Ausführungsform kann der Aufgabenplaner 1722 einen Spark-Treiber beinhalten, um die Planung von Arbeitslasten zu erleichtern, die durch verschiedene Schichten des Rechenzentrums 1700 unterstützt werden. In mindestens einer Ausführungsform kann der Konfigurations-Manager 1724 in der Lage sein, unterschiedliche Schichten, wie etwa die Software-Schicht 1730 und die Rahmenschicht 1720, einschließlich Spark und des verteilten Dateisystems 1728, zu konfigurieren, um die Verarbeitung großer Datenmengen zu unterstützen. In mindestens einer Ausführungsform kann der Ressourcen-Manager 1726 in der Lage sein, geclusterte oder gruppierte Rechenressourcen zu verwalten, die zur Unterstützung des verteilten Dateisystems 1728 und des Aufgaben-Planers 1722 abgebildet oder zugewiesen sind. In mindestens einer Ausführungsform können geclusterte oder gruppierte Rechenressourcen die gruppierte Rechenressource 1714 auf der Rechenzentrumsinfrastrukturschicht 1710 beinhalten. In mindestens einer Ausführungsform kann sich der Ressourcen-Manager 1726 mit dem Ressourcen-Orchestrator 1712 koordinieren, um diese abgebildeten oder zugewiesenen Rechenressourcen zu verwalten.

[0105] In mindestens einer Ausführungsform kann die Software 1732, die in der Software-Schicht 1730 enthalten ist, Software beinhalten, die zumindest von Abschnitten der Knoten-C.R.s 1716(1)-1716(N), den gruppierten Rechenressourcen 1714 und/oder dem verteilten Dateisystem 1728 der Rahmenschicht 1720 verwendet wird. Eine oder mehrere Arten von Software können Software zum Durchsuchen von Internet-Webseiten, Software zum Scannen von E-Mails auf Viren, Datenbank-Software und Software zum Streaming von Videoinhalten beinhalten, ohne darauf beschränkt zu sein.

[0106] In mindestens einer Ausführungsform können die in der Anwendungsschicht 1740 enthaltenen Anwendung(en) 1742 eine oder mehrere Arten von Anwendungen beinhalten, die zumindest von Abschnitten der Knoten-C.R.s 1716(1)-1716(N), den gruppierten Rechenressourcen 1714 und/oder dem verteilten Datei-

system 1728 der Rahmenschicht 1720 verwendet werden. Zu einer oder mehreren Arten von Anwendungen können eine beliebige Anzahl von genomischen Anwendungen, eine kognitive Berechnung und eine Anwendung für maschinelles Lernen gehören, einschließlich Trainings- oder Inferenz-Software, Rahmen-Software für maschinelles Lernen (z. B. PyTorch, TensorFlow, Caffe usw.) oder andere Anwendungen für maschinelles Lernen, die in Verbindung mit einer oder mehreren Ausführungsformen verwendet werden, ohne darauf beschränkt zu sein.

[0107] In mindestens einer Ausführungsform können beliebige des Konfigurations-Managers 1724, des Ressourcen-Managers 1726 und des Ressourcen-Orchestrators 1712 eine beliebige Anzahl und Art von selbstmodifizierenden Handlungen implementieren, die auf einer beliebigen Menge und Art von Daten basieren, die auf eine beliebige technisch machbare Weise erfasst wurden. In mindestens einer Ausführungsform können selbstmodifizierende Handlungen einen Rechenzentrumsbetreiber des Rechenzentrums 1700 dahingehend entlasten, möglicherweise schlechte Konfigurationsentscheidungen zu treffen und möglicherweise nicht ausgelastete und/oder schlecht funktionierende Abschnitte eines Rechenzentrums zu vermeiden.

[0108] In mindestens einer Ausführungsform kann das Rechenzentrum 1700 Werkzeuge, Dienste, Software oder andere Ressourcen beinhalten, um ein oder mehrere Modelle für maschinelles Lernen zu trainieren oder Informationen unter Verwendung eines oder mehrerer Modelle für maschinelles Lernen gemäß einer oder mehrerer in dieser Schrift beschriebener Ausführungsformen vorherzusagen oder abzuleiten. Zum Beispiel kann in mindestens einer Ausführungsform ein Modell für maschinelles Lernen trainiert werden, indem Gewichtungsparmeter gemäß einer Architektur eines neuronalen Netzes unter Verwendung von Software- und Rechenressourcen berechnet werden, die vorstehend in Bezug auf das Rechenzentrum 1700 beschrieben sind. In mindestens einer Ausführungsform können trainierte Modelle für maschinelles Lernen, die einem oder mehreren neuronalen Netzen entsprechen, verwendet werden, um Informationen unter Verwendung der vorstehend in Bezug auf das Rechenzentrum 1700 beschriebenen Ressourcen abzuleiten oder vorherzusagen, indem Gewichtungsparmeter verwendet werden, die durch eine oder mehrere in dieser Schrift beschriebene Trainingstechniken berechnet werden.

[0109] In mindestens einer Ausführungsform kann das Rechenzentrum CPUs, anwendungsspezifische integrierte Schaltungen (ASICs), GPUs, FPGAs oder andere Hardware verwenden, um Training und/oder Inferenzieren unter Verwendung der vorstehend beschriebenen Ressourcen durchzuführen. Darüber hinaus können eine oder mehrere der vorstehend beschriebenen Software- und/oder Hardware-Ressourcen als Dienst konfiguriert sein, um Benutzern das Trainieren oder Durchführen des Inferenzierens von Informationen zu ermöglichen, wie etwa Bilderkennung, Spracherkennung oder andere Dienste der künstlichen Intelligenz.

[0110] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 in dem System aus **Fig. 17** für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparmetern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0111] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

COMPUTER SYSTEMS

[0112] **Fig. 18A** ist ein Blockdiagramm, das ein beispielhaftes Computersystem veranschaulicht, das ein System mit miteinander verbundenen Vorrichtungen und Komponenten, ein System auf einem Chip (system-on-a-chip - SOC) oder eine Kombination davon 1800 sein kann, gebildet mit einem Prozessor, der Ausführungseinheiten zur Ausführung einer Anweisung gemäß mindestens einer Ausführungsform beinhalten kann. In mindestens einer Ausführungsform kann das Computersystem 1800 ohne Einschränkung eine Komponente, wie etwa einen Prozessor 1802, beinhalten, um Ausführungseinheiten einschließlich Logik zur Durchführung von Algorithmen für Prozessdaten gemäß der vorliegenden Offenbarung zu verwenden, wie etwa in der in dieser Schrift beschriebenen Ausführungsform. In mindestens einer Ausführungsform kann das Computersystem 1800 Prozessoren beinhalten, wie etwa PENTIUM®-Prozessorfamilie, Mikroprozessoren von

Xeon™, Itanium®, XScale™ und/oder StrongARM™, Intel® Core™ oder Intel® Nervana™, die von Intel Corporation aus Santa Clara, Kalifornien, erhältlich sind, obwohl auch andere Systeme (die PCs mit anderen Mikroprozessoren, Engineering-Workstations, Set-Top-Boxen und dergleichen beinhalten) verwendet werden können. In mindestens einer Ausführungsform kann das Computersystem 1800 eine Version des Betriebssystems WINDOWS ausführen, das von der Microsoft Corporation in Redmond, Washington, erhältlich ist, obwohl auch andere Betriebssysteme (z. B. UNIX und Linux), eingebettete Software und/oder grafische Benutzeroberflächen verwendet werden können.

[0113] Ausführungsformen können in anderen Vorrichtungen wie etwa tragbaren Vorrichtungen und eingebetteten Anwendungen verwendet werden. Einige Beispiele für tragbare Vorrichtungen beinhalten Mobiltelefone, Internetprotokoll-Vorrichtungen, Digitalkameras, persönliche digitale Assistenten („PDAs“) und tragbare PCs. In mindestens einer Ausführungsform können eingebettete Anwendungen einen Mikrocontroller, einen digitalen Signalprozessor (digital signal processor- „DSP“), ein System auf einem Chip, Netzcomputer („NetPCs“), Set-Top-Boxen, Netz-Hubs, Switches für ein Weitverkehrsnetz (wide area network - „WAN“) oder ein beliebiges anderes System beinhalten, das eine oder mehrere Anweisungen gemäß mindestens einer Ausführungsform durchführen kann.

[0114] In mindestens einer Ausführungsform kann das Computersystem 1800 ohne Einschränkung den Prozessor 1802 beinhalten, der ohne Einschränkung eine oder mehrere Ausführungseinheiten 1808 beinhalten kann, um Trainieren und/oder Inferenzieren eines Modells für maschinelles Lernen gemäß den in dieser Schrift beschriebenen Techniken durchzuführen. In mindestens einer Ausführungsform ist das Computersystem 1800 ein Einzelprozessor-Desktop- oder -Serversystem, aber in einer anderen Ausführungsform kann das Computersystem 1800 ein Multiprozessorsystem sein. In mindestens einer Ausführungsform kann der Prozessor 1802 ohne Einschränkung einen Mikroprozessor für einen Complex Instruction Set Computer („CISC“), einen Mikroprozessor für Reduced Instruction Set Computing („RISC“), einen Very-Long-Instruction-Word-(„VLIW“-)Mikroprozessor, einen Prozessor, der eine Kombination von Anweisungssätzen implementiert, oder eine beliebige andere Prozessorvorrichtung, wie etwa einen digitalen Signalprozessor, beinhalten. In mindestens einer Ausführungsform kann der Prozessor 1802 an einen Prozessorbus 1810 gekoppelt sein, der Datensignale zwischen dem Prozessor 1802 und anderen Komponenten in dem Computersystem 1800 übertragen kann.

[0115] In mindestens einer Ausführungsform kann der Prozessor 1802 ohne Einschränkung einen internen Cache-Speicher („Cache“) 1804 der Ebene 1 (Level 1 - „L1“) beinhalten. In mindestens einer Ausführungsform kann der Prozessor 1802 einen einzelnen internen Cache oder mehrere Ebenen von internem Cache aufweisen. In mindestens einer Ausführungsform kann sich der Cache-Speicher extern zu dem Prozessor 1802 befinden. Andere Ausführungsformen können in Abhängigkeit von der konkreten Implementierung und den Anforderungen auch eine Kombination aus sowohl internen als auch externen Caches beinhalten. In mindestens einer Ausführungsform kann die Registerdatei 1806 unterschiedliche Arten von Daten in verschiedenen Registern speichern, einschließlich ohne Einschränkung Integerregistern, Gleitkommaregistern, Statusregistern und Anweisungszeigerregistern.

[0116] In mindestens einer Ausführungsform befindet sich die Ausführungseinheit 1808, einschließlich ohne Einschränkung der Logik zum Durchführen von Ganzzahl- und Gleitkommaoperationen, ebenfalls in dem Prozessor 1802. In mindestens einer Ausführungsform kann der Prozessor 1802 auch Festwertspeicher (read only memory - „ROM“) für Mikrocode („µCode“) beinhalten, der Mikrocode für bestimmte Makrobefehle speichert. In mindestens einer Ausführungsform kann die Ausführungseinheit 1808 Logik zum Handhaben eines gepackten Anweisungssatzes 1809 beinhalten. In mindestens einer Ausführungsform können durch die Aufnahme des gepackten Anweisungssatzes 1809 in einen Anweisungssatz eines Universalprozessors 1802 zusammen mit zugeordneten Schaltungen zum Ausführen von Anweisungen Operationen, die durch viele Multimedia-Anwendungen verwendet werden, unter Verwendung von gepackten Daten in einem Universalprozessor 1802 durchgeführt werden. In einer oder mehreren Ausführungsformen können viele Multimedia-Anwendungen beschleunigt und effizienter ausgeführt werden, indem die volle Breite des Datenbusses eines Prozessors für die Durchführung von Operationen an Paketdaten genutzt wird, wodurch die Notwendigkeit entfällt, kleinere Dateneinheiten über den Datenbus des Prozessors zu übertragen, um einen oder mehrere Operationen an einem Datenelement nach dem anderen durchzuführen.

[0117] In mindestens einer Ausführungsform kann die Ausführungseinheit 1808 auch in Mikrocontrollern, eingebetteten Prozessoren, Grafikvorrichtungen, DSPs und anderen Arten von Logikschaltungen verwendet werden. In mindestens einer Ausführungsform kann das Computersystem 1800 ohne Einschränkung einen Speicher 1820 beinhalten. In mindestens einer Ausführungsform kann der Speicher 1820 als dynamische

Direktzugriffsspeicher- („DRAM“-)Vorrichtung, statische Direktzugriffsspeicher- („SRAM“-)Vorrichtung, Flash-Speichervorrichtung oder eine andere Speichervorrichtung implementiert sein. In mindestens einer Ausführungsform kann der Speicher 1820 Anweisung(en) 1819 und/oder Daten 1821 speichern, die durch Datensignale dargestellt werden, die durch den Prozessor 1802 ausgeführt werden können.

[0118] In mindestens einer Ausführungsform kann ein Systemlogikchip mit dem Prozessorbus 1810 und dem Speicher 1820 gekoppelt sein. In mindestens einer Ausführungsform kann der Systemlogikchip ohne Einschränkung einen Speichersteuerungs-Hub (memory controller hub - „MCH“) 1816 beinhalten und kann der Prozessor 1802 mit dem MCH 1816 über den Prozessorbus 1810 kommunizieren. In mindestens einer Ausführungsform kann der MCH 1816 dem Speicher 1820 einen Speicherpfad 1818 mit hoher Bandbreite für die Anweisungs- und Datenspeicherung sowie für die Speicherung von Grafikbefehlen, Daten und Texturen bereitstellen. In mindestens einer Ausführungsform kann der MCH 1816 Datensignale zwischen dem Prozessor 1802, dem Speicher 1820 und anderen Komponenten in dem Computersystem 1800 leiten und Datensignale zwischen dem Prozessorbus 1810, dem Speicher 1820 und einer System-E/A 1822 überbrücken. In mindestens einer Ausführungsform kann der Systemlogikchip einen Grafikport zur Kopplung mit einer Grafiksteuerung bereitstellen. In mindestens einer Ausführungsform kann der MCH 1816 durch einen Speicherpfad 1818 mit hoher Bandbreite mit dem Speicher 1820 gekoppelt sein und kann die Grafik-/Videokarte 1812 durch eine Accelerated-Graphics-Port- („AGP“-)Zusammenschaltung 1814 mit dem MCH 1816 gekoppelt sein.

[0119] In mindestens einer Ausführungsform kann das Computersystem 1800 die System-E/A 1822 verwenden, die ein proprietärer Hub-Schnittstellenbus ist, um den MCH 1816 mit dem E/A-Steuerungs-Hub (I/O controller hub - „ICH“) 1830 zu koppeln. In mindestens einer Ausführungsform kann der ICH 1830 direkte Verbindungen zu einigen E/A-Vorrichtungen über einen lokalen E/A-Bus bereitstellen. In mindestens einer Ausführungsform kann der lokale E/A-Bus ohne Einschränkung einen Hochgeschwindigkeits-E/A-Bus zur Verbindung von Peripheriegeräten mit dem Speicher 1820, dem Chipsatz und dem Prozessor 1802 beinhalten. Beispiele können ohne Einschränkung eine Audiosteuerung 1829, einen Firmware-Hub („Flash-BIOS“) 1828, einen drahtlosen Transceiver 1826, einen Datenspeicher 1824, eine Legacy-E/A-Steuerung 1823 mit Benutzereingabe- und Tastaturschnittstellen 1825, einen seriellen Erweiterungsport 1827, wie etwa Universal Serial Bus („USB“), und eine Netzwerksteuerung 1834 beinhalten. Der Datenspeicher 1824 kann ein Festplattenlaufwerk, ein Diskettenlaufwerk, eine CD-ROM-Vorrichtung, eine Flash-Speichervorrichtung oder eine andere Massenspeichervorrichtung umfassen.

[0120] In mindestens einer Ausführungsform veranschaulicht **Fig. 18A** ein System, das zusammengeschaltete Hardware-Vorrichtungen oder „Chips“ beinhaltet, wohingegen in anderen Ausführungsformen **Fig. 18A** ein beispielhaftes System auf einem Chip („SoC“) veranschaulichen kann. In mindestens einer Ausführungsform können die Vorrichtungen mit proprietären Zusammenschaltungen, standardisierten Zusammenschaltungen (z. B. PCIe) oder einer Kombination davon zusammengeschaltet sein. In mindestens einer Ausführungsform sind eine oder mehrere Komponenten des Computersystems 1800 unter Verwendung von Compute-Express-Link-(CXL-)Zusammenschaltungen zusammengeschaltet.

[0121] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 in dem System aus **Fig. 18A** für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0122] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0123] **Fig. 19** ist ein Blockdiagramm, das eine elektronische Vorrichtung 1900 zum Verwenden eines Prozessors 1910 gemäß mindestens einer Ausführungsform veranschaulicht. In mindestens einer Ausführungsform kann die elektronische Vorrichtung 1900 zum Beispiel und ohne Einschränkung ein Notebook, ein Tower-Server, ein Rack-Server, ein Blade-Server, ein Laptop, ein Desktop, ein Tablet, eine mobile Vorrichtung, ein Telefon, ein eingebetteter Computer oder eine beliebige andere geeignete elektronische Vorrichtung sein.

[0124] In mindestens einer Ausführungsform kann das System 1900 ohne Einschränkung einen Prozessor 1910 beinhalten, der mit einer beliebigen geeigneten Anzahl oder Art von Komponenten, Peripheriegeräten, Modulen oder Vorrichtungen kommunikativ gekoppelt ist. In mindestens einer Ausführungsform ist der Prozessor 1910 unter Verwendung eines Busses oder einer Schnittstelle gekoppelt, wie etwa eines 1°C-Busses, eines Systemverwaltungsbus (System Management Bus - „SMBus“), eines Low-Pin-Count-(LPC-)Busses, einer seriellen peripheren Schnittstelle (Serial Peripheral Interface - „SPI“), eines High-Definition-Audio-(„HDA“-)Busses, eines Serial-Advance-Technology-Attachment-(„SATA“-)Busses, eines Universal Serial Bus („USB“) (Versionen 1, 2, 3) oder eines Universal-Asynchronous-Receiver/Transmitter-(„UART“-)Busses. In mindestens einer Ausführungsform veranschaulicht **Fig. 19** ein System, das zusammengesetzte Hardware-Vorrichtungen oder „Chips“ beinhaltet, wohingegen in anderen Ausführungsformen **Fig. 19** ein beispielhaftes System auf einem Chip („SoC“) veranschaulichen kann. In mindestens einer Ausführungsform können die in **Fig. 19** veranschaulichten Vorrichtungen mit proprietären Zusammenschaltungen, standardisierten Zusammenschaltungen (z. B. PCIe) oder einer Kombination davon zusammengesetzt sein. In mindestens einer Ausführungsform sind eine oder mehrere Komponenten aus **Fig. 19** unter Verwendung von Compute-Express-Link-(CXL-)Zusammenschaltungen zusammengesetzt.

[0125] In mindestens einer Ausführungsform kann **Fig. 19** eine Anzeige 1924, einen Touchscreen 1925, ein Touchpad 1930, eine Einheit zur Nahfeldkommunikation (Near Field Communications - „NFC“) 1945, einen Sensor-Hub 1940, einen Wärmesensor 1946, einen Express-Chipsatz („EC“) 1935, ein Trusted-Platform-Modul („TPM“) 1938, BIOS-/Firmware-/Flash-Speicher („BIOS, FW Flash“) 1922, einen DSP 1960, ein Laufwerk 1920, wie etwa ein Solid-State-Platten- (Solid State Disk - „SSD“) oder ein Festplattenlaufwerk (Hard Disk Drive - „HDD“), eine Einheit für ein drahtloses lokales Netzwerk (wireless local area network - „WLAN“) 1950, eine Bluetooth-Einheit 1952, eine Einheit für ein drahtloses Weitverkehrsnetzwerk (Wireless Wide Area Network - „WWAN“) 1956, ein globales Positionsbestimmungssystem (GPS) 1955, eine Kamera („USB-3.0-Kamera“) 1954, wie etwa eine USB-3.0-Kamera, und/oder eine Low-Power-Double-Data-Rate-(„LPDDR“-)Speichereinheit („LPDDR3“) 1915, die zum Beispiel im LPDDR3-Standard implementiert ist, beinhalten. Diese Komponenten können jeweils auf eine beliebige geeignete Weise implementiert sein.

[0126] In mindestens einer Ausführungsform können andere Komponenten durch die vorstehend erörterten Komponenten kommunikativ mit dem Prozessor 1910 gekoppelt sein. In mindestens einer Ausführungsform können ein Beschleunigungsmesser 1941, ein Umgebungslichtsensor (Ambient Light Sensor - „ALS“) 1942, ein Kompass 1943 und ein Gyroskop 1944 kommunikativ mit dem Sensor-Hub 1940 gekoppelt sein. In mindestens einer Ausführungsform können ein Wärmesensor 1939, ein Lüfter 1937, eine Tastatur 1946 und ein Touchpad 1930 kommunikativ mit dem EC 1935 gekoppelt sein. In mindestens einer Ausführungsform können ein Lautsprecher 1963, Kopfhörer 1964 und ein Mikrofon („Mic“) 1965 kommunikativ mit einer Audioeinheit („Audio-Codec und Klasse-D-Verst“) 1962 gekoppelt sein, die wiederum kommunikativ mit dem DSP 1960 gekoppelt sein kann. In mindestens einer Ausführungsform kann die Audioeinheit 1964 zum Beispiel und ohne Einschränkung einen Audiocodierer/-decodierer („Codec“) und einen Klasse-D-Verstärker beinhalten. In mindestens einer Ausführungsform kann eine SIM-Karte („SIM“) 1957 kommunikativ mit der WWAN-Einheit 1956 gekoppelt sein. In mindestens einer Ausführungsform können Komponenten, wie etwa die WLAN-Einheit 1950 und die Bluetooth-Einheit 1952 sowie die WWAN-Einheit 1956, in einem Next Generation Form Factor („NGFF“) implementiert sein.

[0127] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 in dem System aus **Fig. 19** für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0128] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0129] **Fig. 20** veranschaulicht ein Computersystem 2000 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist das Computersystem 2000 dazu konfiguriert, verschiedene Prozesse und Verfahren zu implementieren, die in dieser Offenbarung beschrieben sind.

[0130] In mindestens einer Ausführungsform umfasst das Computersystem 2000 ohne Einschränkung mindestens eine zentrale Verarbeitungseinheit („CPU“) 2002, die mit einem Kommunikationsbus 2010 verbunden ist, der unter Verwendung eines beliebigen geeigneten Protokolls implementiert ist, wie etwa PCI („Peripheral Component Interconnect“), Peripheral Component Interconnect Express („PCI-Express“), AGP („Accelerated Graphics Port“), HyperTransport oder beliebiger anderer Bus- oder Punkt-zu-Punkt-Kommunikationsprotokoll(e). In mindestens einer Ausführungsform beinhaltet das Computersystem 2000 ohne Einschränkung einen Hauptspeicher 2004 und eine Steuerlogik (z. B. implementiert als Hardware, Software oder eine Kombination davon) und Daten werden im Hauptspeicher 2004 gespeichert, der die Form eines Direktzugriffsspeichers (random access memory - „RAM“) annehmen kann. In mindestens einer Ausführungsform stellt ein Netzwerkschnittstellen-Teilsystem („Netzwerkschnittstelle“) 2022 eine Schnittstelle zu anderen Rechenvorrichtung und Netzwerken bereit, um Daten von anderen Systemen zu empfangen und von dem Computersystem 2000 an diese zu übertragen.

[0131] In mindestens einer Ausführungsform beinhaltet das Computersystem 2000 in mindestens einer Ausführungsform ohne Einschränkung Eingabevorrichtungen 2008, ein Parallelverarbeitungssystem 2012 und Anzeigevorrichtungen 2006, die unter Verwendung einer herkömmlichen Kathodenstrahlröhre (cathode ray tube „CRT“), Flüssigkristallanzeige (liquid crystal display - „LCD“), Leuchtdiode (light emitting diode - „LED“), Plasmaanzeige oder anderen geeigneten Anzeigetechnologien implementiert sein können. In mindestens einer Ausführungsform werden Benutzereingaben von Eingabevorrichtungen 2008, wie etwa Tastatur, Maus, Touchpad, Mikrofon und anderen, empfangen. In mindestens einer Ausführungsform kann sich jedes der vorgenannten Module auf einer einzelnen Halbleiterplattform befinden, um ein Verarbeitungssystem zu bilden.

[0132] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 in dem System aus **Fig. 20** für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0133] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0134] **Fig. 21** veranschaulicht ein Computersystem 2100 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform beinhaltet das Computersystem 2100 ohne Einschränkung einen Computer 2110 und einen USB-Stick 2120. In mindestens einer Ausführungsform kann der Computer 2110 ohne Einschränkung eine beliebige Anzahl und eine beliebige Art von Prozessor(en) (nicht gezeigt) und einen Speicher (nicht gezeigt) beinhalten. In mindestens einer Ausführungsform beinhaltet der Computer 2110 ohne Einschränkung einen Server, eine Cloud-Instanz, einen Laptop und einen Desktop-Computer.

[0135] In mindestens einer Ausführungsform beinhaltet der USB-Stick 2120 ohne Einschränkung eine Verarbeitungseinheit 2130, eine USB-Schnittstelle 2140 und eine USB-Schnittstellenlogik 2150. In mindestens einer Ausführungsform kann die Verarbeitungseinheit 2130 ein(e) beliebige(s) Anweisungsausführungssystem, -einrichtung oder -vorrichtung sein, das/die in der Lage ist, Anweisungen auszuführen. In mindestens einer Ausführungsform kann die Verarbeitungseinheit 2130 ohne Einschränkung eine beliebige Anzahl und eine beliebige Art von Verarbeitungskernen (nicht gezeigt) beinhalten. In mindestens einer Ausführungsform umfasst der Verarbeitungskern 2130 eine anwendungsspezifische integrierte Schaltung („ASIC“), die für die Durchführung beliebiger Mengen und Arten von Operationen im Zusammenhang mit maschinellem Lernen optimiert ist. Zum Beispiel ist in mindestens einer Ausführungsform der Verarbeitungskern 2130 eine Tensor-Verarbeitungseinheit (tensor processing unit - „TPU“), die für die Durchführung von Inferenzoperationen für maschinelles Lernen optimiert ist. In mindestens einer Ausführungsform ist der Verarbeitungskern 2130 eine Sichtverarbeitungseinheit (vision processing unit - „VPU“), die für die Durchführung von Inferenzoperationen für maschinelles Sehen und maschinelles Lernen optimiert ist.

[0136] In mindestens einer Ausführungsform kann die USB-Schnittstelle 2140 eine beliebige Art von USB-Stecker oder USB-Buchse sein. Zum Beispiel ist in mindestens einer Ausführungsform die USB-Schnittstelle 2140 eine USB-3.0-Typ-C-Buchse für Daten und Leistung. In mindestens einer Ausführungsform ist die USB-Schnittstelle 2140 ein USB-3.0-Typ-A-Stecker. In mindestens einer Ausführungsform kann die USB-Schnitt-

stellenlogik 2150 eine beliebige Menge und Art von Logik beinhalten, die es der Verarbeitungseinheit 2130 ermöglicht, über den USB-Stecker 2140 eine Schnittstelle mit Vorrichtungen (z. B. dem Computer 2110) zu bilden.

[0137] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 in dem System aus **Fig. 21** für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparemtern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0138] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0139] **Fig. 22A** veranschaulicht eine beispielhafte Architektur, in der eine Vielzahl von GPUs 2210-2213 mit einer Vielzahl von Mehrkern-Prozessoren 2205-2206 über Hochgeschwindigkeitsverbindungen 2240-2243 (z. B. Busse, Punkt-zu-Punkt-Zusammenschaltungen usw.) kommunikativ gekoppelt ist. In einer Ausführungsform unterstützen die Hochgeschwindigkeitsverbindungen 2240-2243 einen Kommunikationsdurchsatz von 4 GB/s, 30 GB/s, 80 GB/s oder höher. Verschiedene Zusammenschaltungsprotokolle können verwendet werden, einschließlich, aber nicht beschränkt auf, PCIe 4.0 oder 5.0 und NVLink 2.0.

[0140] Zusätzlich und in einer Ausführungsform sind zwei oder mehr der GPUs 2210-2213 über Hochgeschwindigkeitsverbindungen 2229-2230 zusammengeschaltet, die unter Verwendung von Protokollen/Verbindungen implementiert sein können, die gleich wie oder anders als diejenigen sind, die für die Hochgeschwindigkeitsverbindungen 2240-2243 verwendet werden. Auf ähnliche Weise können zwei oder mehr der Mehrkern-Prozessoren 2205-2206 über eine Hochgeschwindigkeitsverbindung 2228 verbunden sein, bei der es sich um symmetrische Multiprozessor-(SMP-)Busse handeln kann, die mit 20 GB/s, 30 GB/s, 120 GB/s oder höher arbeiten. Alternativ kann die gesamte Kommunikation zwischen den verschiedenen in **Fig. 22A** gezeigten Systemkomponenten unter Verwendung derselben Protokolle/Verbindungen erfolgen (z. B. über eine gemeinsame Zusammenschaltungsstruktur).

[0141] In einer Ausführungsform ist jeder Mehrkern-Prozessor 2205-2206 jeweils über Speicherzusammenschaltungen 2226-2227 kommunikativ mit einem Prozessorspeicher 2201-2202 gekoppelt und ist jede GPU 2210-2213 jeweils über GPU-Speicherzusammenschaltungen 2250-2253 kommunikativ mit dem GPU-Speicher 2220-2223 gekoppelt. Die Speicherzusammenschaltungen 2226-2227 und 2250-2253 können gleiche oder unterschiedliche Speicherzugriffstechnologien verwenden. Bei den Prozessorspeichern 2201-2202 und den GPU-Speichern 2220-2223 kann es sich beispielsweise um flüchtige Speicher, wie etwa dynamische Direktzugriffsspeicher (DRAMs) (einschließlich gestapelter DRAMs), Grafik-DDR-SDRAM (GDDR) (z. B. GDDR5, GDDR6) oder Speicher mit hoher Bandbreite (High Bandwidth Memory - HBM), und/oder um nicht flüchtige Speicher, wie etwa 3D XPoint oder Nano-Ram, handeln. In einer Ausführungsform kann ein Abschnitt der Prozessorspeicher 2201-2202 ein flüchtiger Speicher sein und kann ein anderer Abschnitt ein nicht flüchtiger Speicher sein (z. B. unter Verwendung einer Speicherhierarchie mit zwei Ebenen (two-level memory - 2LM)).

[0142] Wie nachstehend beschrieben, können verschiedene Prozessoren 2205-2206 und GPUs 2210-2213 zwar physisch mit einem konkreten Speicher 2201-2202 bzw. 2220-2223 gekoppelt sein, kann jedoch eine vereinheitlichte Speicherarchitektur implementiert werden, bei der ein und derselbe virtuelle Systemadressraum (auch als „effektiver Adressraum“ bezeichnet) auf verschiedene physische Speicher verteilt ist. Zum Beispiel können die Prozessorspeicher 2201-2202 jeweils 64 GB Systemadressraum umfassen und können die GPU-Speicher 2220-2223 jeweils 32 GB Systemadressraum umfassen (was in diesem Beispiel zu einem adressierbaren Speicher von insgesamt 256 GB führt).

[0143] **Fig. 22B** veranschaulicht zusätzliche Details für eine Zusammenschaltung zwischen einem Mehrkern-Prozessor 2207 und einem Grafikbeschleunigungsmodul 2246 gemäß einem Ausführungsbeispiel. Das Grafikbeschleunigungsmodul 2246 kann einen oder mehrere GPU-Chips beinhalten, die auf einer Leiterkarte integriert sind, die über eine Hochgeschwindigkeitsverbindung 2240 mit dem Prozessor 2207 gekoppelt

ist. Alternativ kann das Grafikbeschleunigungsmodul 2246 in demselben Gehäuse oder Chip wie der Prozessor 2207 integriert sein.

[0144] In mindestens einer Ausführungsform beinhaltet der veranschaulichte Prozessor 2207 eine Vielzahl von Kernen 2260A-2260D, jeder mit einem Adressübersetzungspuffer 2261A-2261D und einem oder mehreren Caches 2262A-2262D. In mindestens einer Ausführungsform können die Kerne 2260A-2260D verschiedene andere Komponenten zur Ausführung von Anweisungen und Verarbeitung von Daten beinhalten, die nicht veranschaulicht sind. Die Caches 2262A-2262D können Caches der Ebene 1 (L1) und der Ebene 2 (L2) umfassen. Außerdem können ein oder mehrere gemeinsam genutzte Caches 2256 in den Caches 2262A-2262D enthalten sein und von Sätzen von Kernen 2260A-2260D gemeinsam genutzt werden. Eine Ausführungsform des Prozessors 2207 beinhaltet zum Beispiel 24 Kerne, jeder mit einem eigenen L1-Cache, zwölf gemeinsam genutzten L2-Caches und zwölf gemeinsam genutzten L3-Caches. In dieser Ausführungsform werden ein oder mehrere L2- und L3-Caches von zwei benachbarten Kernen gemeinsam genutzt. Der Prozessor 2207 und das Grafikbeschleunigungsmodul 2246 sind mit dem Systemspeicher 2214 verbunden, der die Prozessorspeicher 2201-2202 aus **Fig. 22A** beinhalten kann.

[0145] Die Kohärenz wird für Daten und Anweisungen, die in verschiedenen Caches 2262A-2262D, 2256 und im Systemspeicher 2214 gespeichert sind, über eine Zwischenkernkommunikation über einen Kohärenzbus 2264 aufrechterhalten. Zum Beispiel kann in mindestens einer Ausführungsform jeder Cache eine Cache-Kohärenzlogik/-schaltung aufweisen, die ihm zugeordnet ist, um als Reaktion auf erfasste Lese- oder Schreibvorgänge in konkreten Cache-Zeilen über den Kohärenzbus 2264 zu kommunizieren. In einer Implementierung wird ein Cache-Snooping-Protokoll über den Kohärenzbus 2264 implementiert, um Cache-Zugriffe zu kontrollieren.

[0146] In einer Ausführungsform koppelt eine Proxy-Schaltung 2225 das Grafikbeschleunigungsmodul 2246 kommunikativ an den Kohärenzbus 2264, sodass das Grafikbeschleunigungsmodul 2246 an einem Cache-Kohärenzprotokoll als Peer der Kerne 2260A-2260D teilnehmen kann. Insbesondere stellt eine Schnittstelle 2235 Konnektivität zu der Proxy-Schaltung 2225 über eine Hochgeschwindigkeitsverbindung 2240 (z. B. einen PCIe-Bus, NVLink usw.) bereit und eine Schnittstelle 2237 verbindet das Grafikbeschleunigungsmodul 2246 mit der Verbindung 2240.

[0147] In einer Implementierung stellt eine Beschleuniger-Integrationsschaltung 2236 Cache-Verwaltungs-, Speicherzugriffs-, Kontextverwaltungs- und Unterbrechungsverwaltungsdienste im Auftrag einer Vielzahl von Grafikverarbeitungs-Engines 2231, 2232, N des Grafikbeschleunigungsmoduls 2246 bereit. Die Grafikverarbeitungs-Engines 2231, 2232, N können jeweils eine separate Grafikverarbeitungseinheit (GPU) umfassen. Alternativ können die Grafikverarbeitungs-Engines 2231, 2232, N unterschiedliche Arten von Grafikverarbeitungs-Engines innerhalb einer GPU umfassen, wie etwa Grafikausführungseinheiten, Medienverarbeitungs-Engines (z. B. Videocodierer/-decodierer), Abtaster und Blit-Engines. In mindestens einer Ausführungsform kann das Grafikbeschleunigungsmodul 2246 eine GPU mit einer Vielzahl von Grafikverarbeitungs-Engines 2231-2232, N sein, oder können die Grafikverarbeitungs-Engines 2231-2232, N einzelne GPUs sein, die in einem gemeinsamen Gehäuse, auf einer gemeinsamen Leitungskarte oder auf einem gemeinsamen Chip integriert sind.

[0148] In einer Ausführungsform beinhaltet die Beschleuniger-Integrationsschaltung 2236 eine Speicherverwaltungseinheit (memory management unit - MMU) 2239 zur Durchführung verschiedener Speicherverwaltungsfunktionen, wie etwa Übersetzungen von virtuellem in physischen Speicher (auch als Übersetzungen von effektivem in realen Speicher bezeichnet) und Speicherzugriffsprotokolle für den Zugriff auf den Systemspeicher 2214. Die MMU 2239 kann auch einen Adressübersetzungspuffer (translation lookaside buffer - TLB) (nicht gezeigt) zum Zwischenspeichern von Übersetzungen von virtuellen/effektiven in physische/reale Adressen beinhalten. In einer Implementierung speichert ein Cache 2238 Befehle und Daten für den effizienten Zugriff durch die Grafikverarbeitungs-Engines 2231-2232, N. In einer Ausführungsform werden die in dem Cache 2238 und in den Grafikspeichern 2233-2234, M gespeicherten Daten mit den Kern-Caches 2262A-2262D, 2256 und dem Systemspeicher 2214 kohärent gehalten. Wie vorstehend erwähnt, kann dies über die Proxy-Schaltung 2225 im Auftrag des Caches 2238 und der Speicher 2233-2234, M erfolgen (z. B. Senden von Aktualisierungen an den Cache 2238 in Bezug auf Modifikationen/Zugriffe auf Cache-Leitungen in den Prozessor-Caches 2262A-2262D, 2256 und Empfangen von Aktualisierungen von dem Cache 2238).

[0149] Ein Satz von Registern 2245 speichert Kontextdaten für Threads, die von den Grafikverarbeitungs-Engines 2231-2232, N ausgeführt werden, und eine Kontextverwaltungsschaltung 2248 verwaltet Thread-Kontexte. Zum Beispiel kann die Kontextverwaltungsschaltung 2248 Sicherungs- und Wiederherstellungso-

operationen durchführen, um Kontexte verschiedener Threads während Kontextumschaltungen zu sichern und wiederherzustellen (z. B., wenn ein erster Thread gesichert und ein zweiter Thread gespeichert wird, damit ein zweiter Thread von einer Grafikverarbeitungs-Engine ausgeführt werden kann). Zum Beispiel kann die Kontextverwaltungsschaltung 2248 bei einer Kontextumschaltung aktuelle Registerwerte in einer bezeichneten Region in dem Speicher speichern (z. B. identifiziert durch einen Kontextzeiger). Sie kann dann bei der Rückkehr zu einem Kontext die Registerwerte wiederherstellen. In einer Ausführungsform empfängt und verarbeitet eine Unterbrechungsverwaltungsschaltung 2247 von Systemvorrichtungen empfangene Unterbrechungen.

[0150] In einer Implementierung werden virtuelle/effektive Adressen von einer Grafikverarbeitungs-Engine 2231 durch die MMU 2239 in reale/physische Adressen in dem Systemspeicher 2214 übersetzt. Eine Ausführungsform der Beschleuniger-Integrationsschaltung 2236 unterstützt mehrere (z. B. 4, 8, 16) Grafikbeschleunigermodule 2246 und/oder andere Beschleunigervorrichtungen. Das Grafikbeschleunigermodul 2246 kann für eine einzelne Anwendung dediziert sein, die auf dem Prozessor 2207 ausgeführt wird, oder von mehreren Anwendungen gemeinsam genutzt werden. In einer Ausführungsform ist eine virtualisierte Grafikausführungsumgebung dargestellt, in der die Ressourcen der Grafikverarbeitungs-Engines 2231-2232, N mit mehreren Anwendungen oder virtuellen Maschinen (VMs) gemeinsam genutzt werden. In mindestens einer Ausführungsform können die Ressourcen in „Scheiben“ unterteilt werden, die unterschiedlichen VMs und/oder Anwendungen auf Grundlage von Verarbeitungsanforderungen und Prioritäten, die VMs und/oder Anwendungen zugeordnet sind, zugewiesen werden.

[0151] In mindestens einer Ausführungsform fungiert die Beschleuniger-Integrationsschaltung 2236 als Brücke zu einem System für das Grafikbeschleunigungsmodul 2246 und sie stellt Adressübersetzungs- und Systemspeicher-Cache-Dienste bereit. Darüber hinaus kann die Beschleuniger-Integrationsschaltung 2236 Virtualisierungseinrichtungen für einen Host-Prozessor bereitstellen, um die Virtualisierung der Grafikverarbeitungs-Engines 2231-2232, N, Unterbrechungen und Speicherverwaltung zu verwalten.

[0152] Da die Hardware-Ressourcen der Grafikverarbeitungs-Engines 2231-2232, N explizit auf einen realen Adressraum abgebildet sind, den der Host-Prozessor 2207 sieht, kann ein beliebiger Host-Prozessor diese Ressourcen unter Verwendung eines effektiven Adresswerts direkt adressieren. Eine Funktion der Beschleuniger-Integrationsschaltung 2236 ist in einer Ausführungsform die physische Trennung der Grafikverarbeitungs-Engines 2231-2232, N, sodass sie einem System als unabhängige Einheiten erscheinen.

[0153] In mindestens einer Ausführungsform sind ein oder mehrere Grafikspeicher 2233-2234, M mit jeder der Grafikverarbeitungs-Engines 2231-2232, N gekoppelt. Die Grafikspeicher 2233-2234, M speichern Anweisungen und Daten, die von jeder der Grafikverarbeitungs-Engines 2231-2232, N verarbeitet werden. Bei den Grafikspeichern 2233-2234, M kann es sich um flüchtige Speicher, wie zum Beispiel DRAMs (einschließlich gestapelter DRAMs), GDDR-Speicher (z. B. GDDR5, GDDR6) oder HBM, und/oder um nicht flüchtigen Speicher, wie zum Beispiel 3D XPoint oder Nano-Ram, handeln.

[0154] In einer Ausführungsform werden zum Reduzieren des Datenverkehrs über die Verbindung 2240 Verzerrungstechniken verwendet, um sicherzustellen, dass es sich bei den in den Grafikspeichern 2233-2234, M gespeicherten Daten um Daten handelt, die am häufigsten durch die Grafikverarbeitungs-Engines 2231-2232, N verwendet werden und vorzugsweise nicht durch die Kerne 2260A-2260D verwendet werden (zumindest nicht häufig). Auf ähnliche Weise versucht ein Verzerrungsmechanismus, Daten, die von den Kernen (und vorzugsweise nicht von den Grafikverarbeitungs-Engines 2231-2232, N) benötigt werden, innerhalb der Caches 2262A-2262D, 2256 der Kerne und des Systemspeichers 2214 zu behalten.

[0155] **Fig. 22C** veranschaulicht ein anderes Ausführungsbeispiel, in dem die Beschleuniger-Integrationsschaltung 2236 in den Prozessor 2207 integriert ist. Zumindest in dieser Ausführungsform kommunizieren die Grafikverarbeitungs-Engines 2231-2232, N direkt über die Hochgeschwindigkeitsverbindung 2240 mit der Beschleuniger-Integrationsschaltung 2236 über die Schnittstelle 2237 und die Schnittstelle 2235 (die wiederum eine beliebige Form von Bus- oder Schnittstellenprotokoll verwenden kann). Die Beschleuniger-Integrationsschaltung 2236 kann die gleichen Operationen durchführen wie diejenigen, die in Bezug auf **Fig. 22B** beschrieben sind, aber möglicherweise mit einem höheren Durchsatz, da sie sich in unmittelbarer Nähe zu dem Kohärenzbus 2264 und den Caches 2262A-2262D, 2256 befindet. Mindestens eine Ausführungsform unterstützt unterschiedliche Programmiermodelle, einschließlich eines Programmiermodells mit dediziertem Prozess (ohne Virtualisierung des Grafikbeschleunigungsmoduls) und gemeinsam genutzter Programmiermodelle (mit Virtualisierung), die Programmiermodelle beinhalten können, die durch die Beschleuniger-Integ-

rationsschaltung 2236 gesteuert werden, und Programmiermodelle, die durch das Grafikbeschleunigungsmodul 2246 gesteuert werden.

[0156] In mindestens einer Ausführungsform sind die Grafikverarbeitungs-Engines 2231-2232, N für eine einzelne Anwendung oder einen einzelnen Prozess unter einem einzelnen Betriebssystem dediziert. In mindestens einer Ausführungsform kann eine einzelne Anwendung andere Anwendungsanforderungen zu den Grafikverarbeitungs-Engines 2231-2232, N lenken und so eine Virtualisierung innerhalb einer VM/Partition bereitstellen.

[0157] In mindestens einer Ausführungsform können die Grafikverarbeitungs-Engines 2231-2232, N von mehreren VM-/Anwendungspartitionen gemeinsam genutzt werden. In mindestens einer Ausführungsform können gemeinsam genutzte Modelle einen System-Hypervisor verwenden, um die Grafikverarbeitungs-Engines 2231-2232, N zu virtualisieren, um Zugriff durch jedes Betriebssystem zu ermöglichen. Bei Systemen mit einzelner Partition ohne einen Hypervisor befinden sich die Grafikverarbeitungs-Engines 2231-2232, N im Besitz eines Betriebssystems. In mindestens einer Ausführungsform kann ein Betriebssystem die Grafikverarbeitungs-Engines 2231-2232, N virtualisieren, um Zugriff auf jeden Prozess oder jede Anwendung bereitzustellen.

[0158] In mindestens einer Ausführungsform wählt das Grafikbeschleunigungsmodul 2246 oder eine einzelne Grafikverarbeitungs-Engine 2231-2232, N ein Prozesselement unter Verwendung eines Prozessidentifikators aus. In mindestens einer Ausführungsform sind die Prozesselemente im Systemspeicher 2214 gespeichert und können unter Verwendung der in dieser Schrift beschriebenen Techniken zur Übersetzung von effektiven Adressen in reale Adressen adressiert werden. In mindestens einer Ausführungsform kann ein Prozessidentifikator ein implementierungsspezifischer Wert sein, der einem Host-Prozess bereitgestellt wird, wenn er seinen Kontext bei der Grafikverarbeitungs-Engine 2231-2232, N registriert (das heißt, die Systemsoftware aufruft, um ein Prozesselement zu einer mit dem Prozesselement verknüpften Liste hinzuzufügen). In mindestens einer Ausführungsform können die unteren 16 Bit eines Prozessidentifikators einen Versatz eines Prozesselements innerhalb einer mit dem Prozesselement verknüpften Liste sein.

[0159] Fig. 22D veranschaulicht eine beispielhafte Beschleuniger-Integrationsscheibe 2290. Im vorliegenden Zusammenhang umfasst eine „Scheibe“ einen vorgegebenen Abschnitt der Verarbeitungsressourcen der Beschleuniger-Integrationsschaltung 2236. Der effektive Adressraum 2282 der Anwendung innerhalb des Systemspeichers 2214 speichert Prozesselemente 2283. In einer Ausführungsform werden Prozesselemente 2283 als Reaktion auf GPU-Aufrufe 2281 von Anwendungen 2280, die auf dem Prozessor 2207 ausgeführt werden, gespeichert. Ein Prozesselement 2283 enthält den Prozesszustand für die entsprechende Anwendung 2280. Ein in dem Prozesselement 2283 enthaltener Arbeitsdeskriptor (work descriptor - WD) 2284 kann eine einzelne von einer Anwendung angeforderte Aufgabe sein oder einen Zeiger auf eine Warteschlange von Aufgaben enthalten. In mindestens einer Ausführungsform ist der WD 2284 ein Zeiger auf eine Aufgabeanforderungswarteschlange im Adressraum 2282 einer Anwendung.

[0160] Das Grafikbeschleunigungsmodul 2246 und/oder die einzelnen Grafikverarbeitungs-Engines 2231-2232, N können von allen oder einer Teilmenge der Prozesse in einem System gemeinsam genutzt werden. In mindestens einer Ausführungsform kann eine Infrastruktur zum Einrichten des Prozesszustands und zum Senden eines WD 2284 an ein Grafikbeschleunigungsmodul 2246 zum Starten einer Aufgabe in einer virtualisierten Umgebung enthalten sein.

[0161] In mindestens einer Ausführungsform ist ein Programmiermodell mit dediziertem Prozess implementierungsspezifisch. In diesem Modell besitzt ein einzelner Prozess das Grafikbeschleunigungsmodul 2246 oder eine einzelne Grafikverarbeitungs-Engine 2231. Da sich das Grafikbeschleunigungsmodul 2246 im Besitz eines einzelnen Prozesses befindet, initialisiert ein Hypervisor die Beschleuniger-Integrationsschaltung 2236 für eine besitzende Partition und initialisiert ein Betriebssystem die Beschleuniger-Integrationsschaltung 2236 für einen besitzenden Prozess, wenn das Grafikbeschleunigungsmodul 2246 zugewiesen ist.

[0162] Im Betrieb ruft eine WD-Abrufeinheit 2291 in der Beschleunigerintegrations-Scheibe 2290 den nächsten WD 2284 ab, der eine Angabe der Arbeit beinhaltet, die durch eine oder mehrere Grafikverarbeitungs-Engines des Grafikbeschleunigungsmoduls 2246 zu erledigen ist. Daten vom WD 2284 können in Registern 2245 gespeichert und von der MMU 2239, der Unterbrechungsverwaltungsschaltung 2247 und/oder der Kontextverwaltungsschaltung 2248 verwendet werden, wie veranschaulicht. Eine Ausführungsform der MMU 2239 beinhaltet zum Beispiel eine Segment-/Seitenlaufschaltung für den Zugriff auf Segment-/Seitentabellen 2286 innerhalb des virtuellen Adressraums 2285 des OS. Die Unterbrechungsverwaltungsschaltung 2247

kann von dem Grafikbeschleunigungsmodul 2246 empfangene Unterbrechungsereignisse 2292 verarbeiten. Beim Durchführen von Grafikoperationen wird eine effektive Adresse 2293, die durch eine Grafikverarbeitungs-Engine 2231-2232, N erzeugt wird, durch die MMU 2239 in eine reale Adresse übersetzt.

[0163] In einer Ausführungsform wird derselbe Satz von Registern 2245 für jede Grafikverarbeitungs-Engine 2231-2232, N und/oder jedes Grafikbeschleunigungsmodul 2246 dupliziert und kann von einem Hypervisor oder Betriebssystem initialisiert werden. Jedes dieser duplizierten Register kann in einer Beschleuniger-Integrationsscheibe 2290 enthalten sein. Beispielhafte Register, die durch einen Hypervisor initialisiert werden können, sind in Tabelle 1 gezeigt.

Tabelle 1 - Durch Hypervisor initialisierte Register

1	Scheibensteuerregister
2	Geplanter Prozessbereichszeiger für reale Adresse (RA)
3	Autoritätsmasken-Überschreibungsregister
4	Unterbrechungsvektor-Tabelleneintragsversatz
5	Unterbrechungsvektor-Tabelleneintragsbegrenzung
6	Zustandsregister
7	ID einer logischen Partition
8	Hypervisor-Beschleunigernutzungsaufzeichnungszeiger für reale Adresse (RA)
9	Speicherbeschreibungsregister

[0164] Beispielhafte Register, die von einem Betriebssystem initialisiert werden können, sind in Tabelle 2 gezeigt.

Tabelle 2 - Durch Betriebssystem initialisierte Register

1	Prozess- und Thread-Identifikation
2	Kontextsicherungs-/Wiederherstellungszeiger für effektive Adresse (EA)
3	Beschleunigernutzungsaufzeichnungszeiger für virtuelle Adresse (VA)
4	Speichersegmenttabellenzeiger für virtuelle Adresse (VA)
5	Autoritätsmaske
6	Arbeitsdeskriptor

[0165] In einer Ausführungsform ist jeder WD 2284 spezifisch für ein konkretes Grafikbeschleunigungsmodul 2246 und/oder die Grafikverarbeitungs-Engines 2231-2232, N. Er enthält alle Informationen, die von einer Grafikverarbeitungs-Engine 2231-2232, N benötigt werden, um Arbeit zu verrichten, oder er kann ein Zeiger auf einen Speicherort sein, an dem eine Anwendung eine Befehlswarteschlange von zu verrichtender Arbeit eingerichtet hat.

[0166] Fig. 22E veranschaulicht zusätzliche Details für ein Ausführungsbeispiel eines gemeinsam genutzten Modells. Diese Ausführungsform beinhaltet einen realen Hypervisor-Adressraum 2298, in dem eine Prozesselementliste 2299 gespeichert ist. Auf den realen Hypervisor-Adressraum 2298 kann über einen Hypervisor 2296 zugegriffen werden, der die Grafikbeschleunigungsmodul-Engines für das Betriebssystem 2295 virtualisiert.

[0167] In mindestens einer Ausführungsform ermöglichen gemeinsam genutzte Programmiermodelle, dass alle oder eine Teilmenge von Prozessen aus allen oder einer Teilmenge von Partitionen in einem System ein Grafikbeschleunigungsmodul 2246 verwenden. Es gibt zwei Programmiermodelle, bei denen das Grafikbeschleunigungsmodul 2246 von mehreren Prozessen und Partitionen gemeinsam genutzt wird: über Zeitscheiben gemeinsam genutzt (time-sliced shared) und über gerichtete Grafik gemeinsam genutzt (graphics-directed shared).

[0168] In diesem Modell besitzt der System-Hypervisor 2296 das Grafikbeschleunigungsmodul 2246 und stellt seine Funktion allen Betriebssystemen 2295 zur Verfügung. Damit ein Grafikbeschleunigungsmodul 2246 die Virtualisierung durch den System-Hypervisor 2296 unterstützen kann, muss das Grafikbeschleunigungsmodul 2246 Folgendes einhalten: 1) Die Aufgabenanforderung einer Anwendung muss autonom sein (das heißt, der Zustand muss zwischen den Aufgaben nicht beibehalten werden) oder das Grafikbeschleunigungsmodul 2246 muss einen Mechanismus zum Sichern und Wiederherstellen von Kontext bereitstellen. 2) Das Grafikbeschleunigungsmodul 2246 garantiert, dass die Aufgabenanforderung einer Anwendung innerhalb einer spezifizierten Zeitspanne abgeschlossen wird, einschließlich beliebiger Übersetzungsfehler, oder das Grafikbeschleunigungsmodul 2246 stellt eine Fähigkeit bereit, die Verarbeitung einer Aufgabe vorwegzunehmen. 3) Dem Grafikbeschleunigungsmodul 2246 muss Fairness zwischen den Prozessen garantiert werden, wenn es in einem gerichteten gemeinsam genutzten Programmiermodell arbeitet.

[0169] In mindestens einer Ausführungsform muss die Anwendung 2280 einen Systemaufruf des Betriebssystems 2295 mit einer Art des Grafikbeschleunigungsmoduls 2246, einem Arbeitsdeskriptor (WD), einem Autoritätsmaskenregister-(Authority Mask Register - AMR-)Wert und einem Kontextsicherungs-/Wiederherstellungsbereichszeiger (Context Save/Restore Area Pointer - CSRP) ausführen. In mindestens einer Ausführungsform beschreibt die Art des Grafikbeschleunigungsmoduls 2246 eine gezielte Beschleunigungsfunktion für einen Systemaufruf. In mindestens einer Ausführungsform kann die Art des Grafikbeschleunigungsmoduls 2246 ein systemspezifischer Wert sein. In mindestens einer Ausführungsform ist der WD spezifisch für das Grafikbeschleunigungsmodul 2246 formatiert und er kann in Form eines Befehls des Grafikbeschleunigungsmoduls 2246, eines effektiven Adresszeigers auf eine benutzerdefinierte Struktur, eines effektiven Adresszeigers auf eine Befehlswarteschlange oder einer beliebigen anderen Datenstruktur vorliegen, um durch das Grafikbeschleunigungsmodul 2246 zu verrichtende Arbeit zu beschreiben. In einer Ausführungsform ist ein AMR-Wert ein AMR-Zustand, der für einen aktuellen Prozess zu verwenden ist. In mindestens einer Ausführungsform ist ein Wert, der an ein Betriebssystem übergeben wird, ähnlich einer Anwendung, die einen AMR festlegt. Wenn Implementierungen der Beschleuniger-Integrationsschaltung 2236 und des Grafikbeschleunigungsmoduls 2246 kein Benutzer-Autoritätsmasken-Überschreibungsregister (User Authority Mask Override Register - UAMOR) unterstützen, kann ein Betriebssystem einen aktuellen UAMOR-Wert auf einen AMR-Wert anwenden, bevor ein AMR in einem Hypervisor-Aufruf übergeben wird. Der Hypervisor 2296 kann optional einen aktuellen Autoritätsmasken-Überschreibungsregister-(Authority Mask Override Register - AMOR-) Wert anwenden, bevor ein AMR in dem Prozesselement 2283 platziert wird. In mindestens einer Ausführungsform ist CSRP eines der Register 2245, die eine effektive Adresse eines Bereichs in dem effektiven Adressraum 2282 einer Anwendung für das Grafikbeschleunigungsmodul 2246 zum Sichern und Wiederherstellen des Kontextzustands enthalten. Dieser Zeiger ist optional, wenn zwischen den Aufgaben oder beim Vorwegnehmen einer Aufgabe kein Zustand gespeichert werden muss. In mindestens einer Ausführungsform kann der Kontextsicherungs-/Wiederherstellungsbereich gepinnter Systemspeicher sein.

[0170] Beim Empfangen eines Systemaufrufs kann das Betriebssystem 2295 verifizieren, ob die Anwendung 2280 registriert ist und die Autorität zum Verwenden des Grafikbeschleunigungsmoduls 2246 erhalten hat. Das Betriebssystem 2295 ruft dann den Hypervisor 2296 mit den in Tabelle 3 gezeigten Informationen auf.

Tabelle 3 - OS-zu-Hypervisor-Aufrufparameter

1	Ein Arbeitsdeskriptor (WD)
2	Ein Wert eines Autoritätsmaskenregisters (AMR) (möglicherweise maskiert)
3	Ein Kontextsicherungs-/Wiederherstellungsbereichszeiger (CSRP) für effektive Adressen (EA)
4	Eine Prozess-ID (PID) und optional eine Thread-ID (TID)
5	Ein Beschleunigernutzungsaufzeichnungszeiger (accelerator utilization record pointer - AURP) für virtuelle Adressen (virtual address - VA)
6	Virtuelle Adresse eines Speichersegmenttabellenzeigers (storage segment table pointer - SSTP)
7	Eine logische Unterbrechungsdienstnummer (logical interrupt service number - LISN)

[0171] Beim Empfang eines Hypervisor-Aufrufs verifiziert der Hypervisor 2296, dass das Betriebssystem 2295 registriert ist und die Autorität zur Verwendung des Grafikbeschleunigungsmoduls 2246 erhalten hat. Der Hypervisor 2296 setzt dann das Prozesselement 2283 in eine mit dem Prozesselement verknüpfte Liste

für eine entsprechende Art des Grafikbeschleunigungsmodul 2246 ein. Ein Prozesselement kann die in Tabelle 4 gezeigten Informationen enthalten.

Tabelle 4 - Prozesselementinformationen

1	Ein Arbeitsdeskriptor (WD)
2	Ein Wert eines Autoritätsmaskenregisters (AMR) (möglicherweise maskiert).
3	Ein Kontextsicherungs-/Wiederherstellungsbereichszeiger (CSRP) für effektive Adressen (EA)
4	Eine Prozess-ID (PID) und optional eine Thread-ID (TID)
5	Ein Beschleunigernutzungsaufzeichnungszeiger (accelerator utilization record pointer - AURP) für virtuelle Adressen (virtual address - VA)
6	Virtuelle Adresse eines Speichersegmenttabellenzeigers (storage segment table pointer - SSTP)
7	Eine logische Unterbrechungsdienstnummer (logical interrupt service number - LISN)
8	Unterbrechungsvektortabelle, abgeleitet aus Hypervisor-Aufrufparametern
9	Ein Wert des Zustandsregisters (state register - SR)
10	Eine ID einer logischen Partition (logical partition ID - LPID)
11	Ein Hypervisor-Beschleunigernutzungsaufzeichnungszeiger für reale Adressen (RA)
12	Speicherdeskriptorregister (Storage Descriptor Register - SDR)

[0172] In mindestens einer Ausführungsform initialisiert der Hypervisor eine Vielzahl von Registern 2245 der Beschleuniger-Integrationsscheibe 2290.

[0173] Wie in **Fig. 22F** veranschaulicht, wird in mindestens einer Ausführungsform ein vereinheitlichter Speicher verwendet, der über einen gemeinsamen virtuellen Speicheradressraum adressierbar ist, der für den Zugriff auf die physischen Prozessorspeicher 2201-2202 und die GPU-Speicher 2220-2223 verwendet wird. In dieser Implementierung verwenden Operationen, die auf den GPUs 2210-2213 ausgeführt werden, denselben virtuellen/effektiven Speicheradressraum für den Zugriff auf die Prozessorspeicher 2201-2202 und umgekehrt, was die Programmierbarkeit vereinfacht. In einer Ausführungsform wird ein erster Abschnitt eines virtuellen/effektiven Adressraums dem Prozessorspeicher 2201 zugewiesen, ein zweiter Abschnitt dem zweiten Prozessorspeicher 2202, ein dritter Abschnitt dem GPU-Speicher 2220 usw. In mindestens einer Ausführungsform wird dadurch ein gesamter virtueller/effektiver Speicherraum (mitunter als effektiver Adressraum bezeichnet) über jeden der Prozessorspeicher 2201-2202 und GPU-Speicher 2220-2223 verteilt, sodass ein beliebiger Prozessor oder eine beliebige GPU auf einen beliebigen physischen Speicher mit einer virtuellen Adresse zugreifen kann, die auf diesen Speicher abgebildet ist.

[0174] In einer Ausführungsform stellt die Verzerrungs-/Kohärenzverwaltungsschaltung 2294A-2294E innerhalb einer oder mehrerer MMUs 2239A-2239E die Cache-Kohärenz zwischen Caches eines oder mehrerer Host-Prozessoren (z. B. 2205) und GPUs 2210-2213 sicher und implementiert Verzerrungstechniken, die physische Speicher angeben, in denen bestimmte Arten von Daten gespeichert werden sollten. Wenngleich mehrere Instanzen der Verzerrungs-/Kohärenzverwaltungsschaltung 2294A-2294E in **Fig. 22F** veranschaulicht sind, kann die Verzerrungs-/Kohärenzschaltung innerhalb einer MMU eines oder mehrerer Host-Prozessoren 2205 und/oder innerhalb der Beschleuniger-Integrationsschaltung 2236 implementiert sein.

[0175] Eine Ausführungsform ermöglicht es, den GPU-zugewiesenen Speicher 2220-2223 als Teil des Systemspeichers zuzuordnen und unter Verwendung von gemeinsam genutzter virtueller Speicher-(Shared Virtual Memory SVM-)Technologie darauf zuzugreifen, ohne jedoch Rechenleistungsnachteile zu erleiden, die der vollständigen System-Cache-Kohärenz zugeordnet sind. In mindestens einer Ausführungsform bietet die Fähigkeit des GPU-gebundenen Speichers 2220-2223, auf den als Systemspeicher ohne lästigen Cache-Kohärenz-Overhead zugegriffen wird, eine vorteilhafte Betriebsumgebung für GPU-Auslagerung. Diese Anordnung ermöglicht es der Software des Host-Prozessors 2205, Operanden einzurichten und auf Berechnungsergebnisse zuzugreifen, und zwar ohne den Overhead der traditionellen E/A-DMA-Datenkopien. Derartige traditionelle Kopien beinhalten Treiberaufrufe, Unterbrechungen und speicherzugeordnete E/A-(memory mapped I/O - MMIO-)Zugriffe, die alle im Vergleich zu einfachen Speicherzugriffen ineffizient sind. In minde-

tens einer Ausführungsform kann eine Fähigkeit, ohne Cache-Kohärenz-Overheads auf den GPU-gebundenen Speicher 2220-2223 zuzugreifen, für die Ausführungszeit einer ausgelagerten Berechnung entscheidend sein. In Fällen mit erheblichem Streaming-Schreibspeicherverkehr kann zum Beispiel der Cache-Kohärenz-Overhead die effektive Schreibbandbreite, die von einer GPU 2210-2213 gesehen wird, erheblich reduzieren. In mindestens einer Ausführungsform können die Effizienz der Operandeneinrichtung, die Effizienz des Ergebniszugriffs und die Effizienz der GPU-Berechnung eine Rolle beim Bestimmen der Effektivität einer GPU-Auslagerung spielen.

[0176] In mindestens einer Ausführungsform wird die Auswahl von GPU-Verzerrung und Host-Prozessorverzerrung durch eine Verzerrungs-Tracker-Datenstruktur angetrieben. Es kann zum Beispiel eine Verzerrungstabelle verwendet werden, die eine seitengranulare Struktur sein kann (d. h. mit einer Granularität einer Speicherseite gesteuert), die 1 oder 2 Bit pro GPU-gebundener Speicherseite beinhaltet. In mindestens einer Ausführungsform kann eine Verzerrungstabelle in einem gestohlenen Speicherbereich eines oder mehrerer GPU-gebundener Speicher 2220-2223 implementiert werden, mit oder ohne Verzerrungs-Cache in einer GPU 2210-2213 (um z. B. häufig/kürzlich verwendete Einträge einer Verzerrungstabelle zwischenzuspeichern). Alternativ kann eine gesamte Verzerrungstabelle innerhalb einer GPU gepflegt werden.

[0177] In mindestens einer Ausführungsform wird vor dem tatsächlichen Zugriff auf einen GPU-Speicher auf einen Verzerrungstabelleneintrag zugegriffen, der jedem Zugriff auf den GPU-gebundenen Speicher 2220-2223 zugeordnet ist, wodurch die folgenden Operationen verursacht werden. Zunächst werden lokale Anforderungen von der GPU 2210-2213, die ihre Seite in der GPU-Verzerrung finden, direkt an einen entsprechenden GPU-Speicher 2220-2223 weitergeleitet. Lokale Anforderungen von einer GPU, die ihre Seite in der Host-Verzerrung findet, werden an den Prozessor 2205 weitergeleitet (z. B. über eine Hochgeschwindigkeitsverbindung, wie vorstehend erörtert). In einer Ausführungsform schließen Anforderungen von dem Prozessor 2205, die eine angeforderte Seite in der Host-Prozessor-Verzerrung finden, eine Anforderung wie ein normales Lesen des Speichers ab. Alternativ können Anforderungen, die an eine GPU-Verzerrungsseite gerichtet sind, an die GPU 2210-2213 weitergeleitet werden. In mindestens einer Ausführungsform kann eine GPU dann eine Seite in eine Host-Prozessor-Verzerrung umwandeln, falls sie derzeit keine Seite verwendet. In mindestens einer Ausführungsform kann der Verzerrungszustand einer Seite entweder durch einen softwarebasierten Mechanismus, einen hardwareunterstützten softwarebasierten Mechanismus oder, für einen begrenzten Satz von Fällen, einen rein hardwarebasierten Mechanismus geändert werden.

[0178] Ein Mechanismus zum Ändern des Verzerrungszustands setzt in mindestens einer Ausführungsform einen API-Aufruf (z. B. OpenCL) ein, der wiederum den Vorrichtungstreiber einer GPU aufruft, der wiederum eine Nachricht an eine GPU sendet (oder einen Befehlsdeskriptor in die Warteschlange stellt), die sie anweist, einen Verzerrungszustand zu ändern und bei einigen Übergängen eine Cache-Leerungsoperation in einem Host durchzuführen. In mindestens einer Ausführungsform wird eine Cache-Leerungsoperation für einen Übergang von der Verzerrung des Host-Prozessors 2205 zu der Verzerrung der GPU verwendet, jedoch nicht für einen entgegengesetzten Übergang.

[0179] In einer Ausführungsform wird die Cache-Kohärenz aufrechterhalten, indem bewirkt wird, dass GPU-verzerrte Seiten durch den Host-Prozessor 2205 vorübergehend nicht zwischengespeichert werden können. Um auf diese Seiten zuzugreifen, kann der Prozessor 2205 Zugriff von der GPU 2210 anfordern, die den Zugriff sofort gewähren kann oder auch nicht. Um die Kommunikation zwischen dem Prozessor 2205 und der GPU 2210 zu reduzieren, ist es daher vorteilhaft, sicherzustellen, dass GPU-verzerrte Seiten solche sind, die von einer GPU, aber nicht von dem Host-Prozessor 2205, benötigt werden und umgekehrt.

[0180] Inferenz- und/oder Trainingslogik 1615 wird verwendet, um eine oder mehrere Ausführungsformen durchzuführen. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt.

[0181] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0182] **Fig. 23** veranschaulicht beispielhafte integrierte Schaltungen und zugeordnete Grafikprozessoren, die unter Verwendung eines oder mehrerer IP-Kerne hergestellt werden können, gemäß verschiedenen in dieser Schrift beschriebenen Ausführungsformen. Zusätzlich zu dem, was veranschaulicht ist, können andere Logik und Schaltungen in mindestens einer Ausführungsform enthalten sein, einschließlich zusätzlicher Grafikprozessoren/-kerne, Peripherieschnittstellensteuerungen oder Universalprozessorkerne.

[0183] **Fig. 23** ist ein Blockdiagramm, das eine beispielhafte integrierte Schaltung 2300 als System auf einem Chip, die unter Verwendung eines oder mehrerer IP-Kerne hergestellt werden kann, gemäß mindestens einer Ausführungsform veranschaulicht. In mindestens einer Ausführungsform beinhaltet die integrierte Schaltung 2300 einen oder mehrere Anwendungsprozessor(en) 2305 (z. B. CPUs), mindestens einen Grafikprozessor 2310 und kann zusätzlich einen Bildprozessor 2315 und/oder einen Videoprozessor 2320 beinhalten, von denen jeder beliebige ein modularer IP-Kern sein kann. In mindestens einer Ausführungsform beinhaltet die integrierte Schaltung 2300 Peripherie- oder Buslogik, die eine USB-Steuerung 2325, eine UART-Steuerung 2330, eine SPI/FSPI-Steuerung 2335 und eine I²S/I²C-Steuerung 2340 beinhaltet. In mindestens einer Ausführungsform kann die integrierte Schaltung 2300 eine Anzeigevorrichtung 2345 beinhalten, die mit einer oder mehreren von einer High-Definition-Multimedia-Interface-(HDMI-)Steuerung 2350 und einer Mobile-Industry-Processor-Interface-(MIPI-)Anzeigeschnittstelle 2355 gekoppelt ist. In mindestens einer Ausführungsform kann der Speicher durch ein Flash-Speicherteilsystem 2360 bereitgestellt sein, das Flash-Speicher und eine Flash-Speichersteuerung beinhaltet. In mindestens einer Ausführungsform kann die Speicherschnittstelle über eine Speichersteuerung 2365 für den Zugriff auf SDRAM- oder SRAM-Speichervorrichtungen bereitgestellt sein. In mindestens einer Ausführungsform beinhalten einige integrierte Schaltungen zusätzlich eine eingebettete Sicherheits-Engine 2370.

[0184] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 in der integrierten Schaltung 2300 für Inferenz- oder Vorhersageoperationen verwendet werden, die zumindest teilweise auf Gewichtungsparemtern basieren, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0185] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0186] Die **Fig. 24A-24B** veranschaulichen beispielhafte integrierte Schaltungen und zugeordnete Grafikprozessoren, die unter Verwendung eines oder mehrerer IP-Kerne hergestellt werden können, gemäß verschiedenen in dieser Schrift beschriebenen Ausführungsformen. Zusätzlich zu dem, was veranschaulicht ist, können andere Logik und Schaltungen in mindestens einer Ausführungsform enthalten sein, einschließlich zusätzlicher Grafikprozessoren/-kerne, Peripherieschnittstellensteuerungen oder Universalprozessorkerne.

[0187] Die **Fig. 24A-24B** sind Blockdiagramme, die beispielhafte Grafikprozessoren zur Verwendung innerhalb eines SoC gemäß in dieser Schrift beschriebenen Ausführungsformen veranschaulichen. **Fig. 24A** veranschaulicht einen beispielhaften Grafikprozessor 2410 einer integrierte Schaltung als System auf einem Chip, die unter Verwendung eines oder mehrerer IP-Kerne hergestellt werden kann, gemäß mindestens einer Ausführungsform. **Fig. 24B** veranschaulicht einen zusätzlichen beispielhaften Grafikprozessor 2440 einer integrierte Schaltung als System auf einem Chip, die unter Verwendung eines oder mehrerer IP-Kerne hergestellt werden kann, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist der Grafikprozessor 2410 aus **Fig. 24A** ein Grafikprozessorkern mit niedriger Leistung. In mindestens einer Ausführungsform ist der Grafikprozessor 2440 aus **Fig. 24B** ein Grafikprozessorkern mit höherer Rechenleistung. In mindestens einer Ausführungsform kann jeder der Grafikprozessoren 2410, 2440 eine Variante des Grafikprozessors 2310 aus **Fig. 23** sein.

[0188] In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2410 einen Vertex-Prozessor 2405 und einen oder mehrere Fragmentprozessor(en) 2415A-2415N (z.B. 2415A, 2415B, 2415C, 2415D bis 2415N-1 und 2415N). In mindestens einer Ausführungsform kann der Grafikprozessor 2410 unterschiedliche Shader-Programme über separate Logik ausführen, sodass der Vertex-Prozessor 2405 optimiert ist, um Operationen für Vertex-Shader-Programme auszuführen, während ein oder mehrere Fragmentprozessor(en) 2415A-2415N Shading-Operationen für Fragmente (z. B. Pixel) für Fragment- oder Pixel-Shader-Programme ausführen. In mindestens einer Ausführungsform führt der Vertex-Prozessor 2405 eine Vertex-Verarbeitungsstufe einer 3D-Grafikpipeline durch und erzeugt Primitiv- und Vertexdaten. In mindestens einer Ausführungsform verwenden ein oder mehrere Fragmentprozessor(en) 2415A-2415N Primitiv- und Vertexdaten, die durch den Vertex-Prozessor 2405 erzeugt werden, um einen Bildpuffer zu erzeugen, der auf einer Anzeigevorrichtung angezeigt wird. In mindestens einer Ausführungsform sind ein oder mehrere Fragmentprozessor(en) 2415A-2415N optimiert, um Fragment-Shader-Programme auszuführen, wie in einer OpenGL-API bereitge-

stellt, die verwendet werden können, um ähnliche Operationen wie ein Pixel-Shader-Programm durchzuführen, wie es in einer Direct 3D-API bereitgestellt ist.

[0189] In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2410 zusätzlich eine(n) oder mehrere Speicherverwaltungseinheiten (MMUs) 2420A-2420B, Cache(s) 2425A-2425B und Schaltungszusammenschaltung(en) 2430A-2430B. In mindestens einer Ausführungsform stellen eine oder mehrere MMU(s) 2420A-2420B eine Abbildung von virtuellen auf physische Adressen für den Grafikprozessor 2410 bereit, einschließlich für den Vertexprozessor 2405 und/oder den/die Fragmentprozessor(en) 2415A-2415N, die auf in Speicher gespeicherte Vertex- oder Bild-/Texturdaten verweisen kann, zusätzlich zu den in einem oder mehreren Cache(s) 2425A-2425B gespeicherten Vertex- oder Bild-/Texturdaten. In mindestens einer Ausführungsform können eine oder mehrere MMU(s) 2420A-2420B mit anderen MMUs innerhalb des Systems synchronisiert werden, einschließlich einer oder mehrerer MMUs, die einem oder mehreren Anwendungsprozessor(en) 2305, Bildprozessoren 2315 und/oder Videoprocessoren 2320 aus **Fig. 23** zugeordnet sind, sodass jeder Prozessor 2305-2320 an einem gemeinsam genutzten oder vereinheitlichten virtuellen Speichersystem teilnehmen kann. In mindestens einer Ausführungsform ermöglichen es eine oder mehrere Schaltungszusammenschaltung(en) 2430A-2430B dem Grafikprozessor 2410, entweder über einen internen Bus des SoC oder über eine direkte Verbindung eine Schnittstelle mit anderen IP-Kernen innerhalb des SoC zu bilden.

[0190] In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2440 eine(n) oder mehrere MMU(s) 2420A-2420B, Cache(s) 2425A-2425B und Schaltungszusammenschaltung(en) 2430A-2430B des Grafikprozessors 2410 aus **Fig. 24A**. In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2440 einen oder mehrere Shader-Kern(e) 2455A-2455N (z. B. 2455A, 2455B, 2455C, 2455D, 2455E, 2455F bis 2455N-1 und 2455N), was eine vereinheitlichte Shader-Kernarchitektur bereitstellt, bei der ein einzelner Kern oder eine einzelne Art von Kern alle Arten von programmierbarem Shader-Code ausführen kann, einschließlich Shader-Programmcode zur Implementierung von Vertex-Shadern, Fragment-Shadern und/oder Rechen-Shadern. In mindestens einer Ausführungsform kann eine Anzahl der Shader-Kerne variieren. In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2440 einen Zwischenkern-Aufgaben-Manager 2445, der als Thread-Zuteiler fungiert, um Ausführungs-Threads einem oder mehreren Shader-Kernen 2455A-2455N zuzuteilen, sowie eine Kachelungseinheit 2458 zum Beschleunigen von Kachelungsoperationen für kachelbasiertes Rendering, bei dem Rendering-Operationen für eine Szene im Bildraum unterteilt werden, um zum Beispiel die lokale räumliche Kohärenz innerhalb einer Szene auszunutzen oder die Nutzung interner Caches zu optimieren.

[0191] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 in der integrierten Schaltung 24A und/oder 24B für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparemtern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0192] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0193] Die **Fig. 25A-25B** veranschaulichen zusätzliche beispielhafte Grafikprozessorlogik gemäß in dieser Schrift beschriebenen Ausführungsformen. **Fig. 25A** veranschaulicht einen Grafikkern 2500, der in mindestens einer Ausführungsform in dem Grafikprozessor 2310 aus **Fig. 23** enthalten sein kann und in mindestens einer Ausführungsform ein vereinheitlichter Shader-Kern 2455A-2455N, wie in **Fig. 24B**, sein kann. **Fig. 25B** veranschaulicht eine hochparallele Universal-Grafikverarbeitungseinheit 2530, die in mindestens einer Ausführungsform für den Einsatz auf einem Mehrchipmodul geeignet ist.

[0194] In mindestens einer Ausführungsform beinhaltet der Grafikkern 2500 einen gemeinsam genutzten Anweisungs-Cache 2502, eine Textureinheit 2518 und einen Cache/gemeinsam genutzten Speicher 2520, die den Ausführungsressourcen innerhalb des Grafikkerns 2500 gemeinsam sind. In mindestens einer Ausführungsform kann der Grafikkern 2500 mehrere Scheiben 2501A-2501N oder eine Partition für jeden Kern beinhalten und ein Grafikprozessor kann mehrere Instanzen des Grafikkerns 2500 beinhalten. Die Scheiben 2501A-2501N können eine Unterstützungslogik beinhalten, die einen lokalen Anweisungs-Cache 2504A-2504N, einen Thread-Planer 2506A-2506N, einen Thread-Zuteiler 2508A-2508N und einen Satz von

Registern 2510A-2510N beinhaltet. In mindestens einer Ausführungsform können die Scheiben 2501A-2501N einen Satz zusätzlicher Funktionseinheiten (additional function units - AFUs 2512A-2512N), Gleitkommaeinheiten (floating-point units - FPU 2514A-2514N), arithmetisch-logischer Einheiten (ALUs 2516-2516N) für Integer, Adressberechnungseinheiten (address computational units - ACUs 2513A-2513N), Gleitkommaeinheiten mit doppelter Genauigkeit (double-precision floating-point units - DPFPUs 2515A-2515N) und Matrixverarbeitungseinheiten (matrix processing units - MPUs 2517A-2517N) beinhalten.

[0195] In mindestens einer Ausführungsform können die FPU 2514A-2514N Gleitkommaoperationen mit einfacher Genauigkeit (32 Bit) und mit halber Genauigkeit (16 Bit) durchführen, während die DPFPUs 2515A-2515N Gleitkommaoperationen mit doppelter Genauigkeit (64 Bit) durchführen. In mindestens einer Ausführungsform können die ALUs 2516A-2516N ganzzahlige Operationen mit variabler Genauigkeit mit 8-Bit-, 16-Bit- und 32-Bit-Genauigkeit durchführen und für Operationen mit gemischter Genauigkeit konfiguriert sein. In mindestens einer Ausführungsform können die MPUs 2517A-2517N auch für Matrixoperationen mit gemischter Genauigkeit konfiguriert sein, einschließlich Gleitkomma- und ganzzahligen 8-Bit-Operationen mit halber Genauigkeit. In mindestens einer Ausführungsform können die MPUs 2517A-2517N eine Vielfalt von Matrixoperationen durchführen, um Anwendungsrahmen für maschinelles Lernen zu beschleunigen, einschließlich des Ermöglichens der Unterstützung für eine beschleunigte allgemeine Matrix-zu-Matrix-Multiplikation (general matrix to matrix multiplication - GEMM). In mindestens einer Ausführungsform können die AFUs 2512A-2512N zusätzliche logische Operationen ausführen, die nicht durch Gleitkomma- oder Ganzzahleinheiten unterstützt werden, einschließlich trigonometrischer Operationen (z. B. Sinus, Cosinus usw.).

[0196] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 in dem Grafikkern 2500 für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0197] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0198] **Fig. 25B** veranschaulicht in mindestens einer Ausführungsform eine Universalverarbeitungseinheit (GPGPU) 2530, die dazu konfiguriert sein kann, zu ermöglichen, dass hochparallele Rechenoperationen durch ein Array von Grafikverarbeitungseinheiten durchgeführt werden. In mindestens einer Ausführungsform kann die GPGPU 2530 direkt mit anderen Instanzen der GPGPU 2530 verknüpft sein, um ein Multi-GPU-Cluster zu erzeugen, um die Trainingsgeschwindigkeit für tiefe neuronale Netze zu verbessern. In mindestens einer Ausführungsform beinhaltet die GPGPU 2530 eine Host-Schnittstelle 2532, um eine Verbindung mit einem Host-Prozessor zu ermöglichen. In mindestens einer Ausführungsform ist die Host-Schnittstelle 2532 eine PCI-Express-Schnittstelle. In mindestens einer Ausführungsform kann die Host-Schnittstelle 2532 eine anbieterspezifische Kommunikationsschnittstelle oder Kommunikationsstruktur sein. In mindestens einer Ausführungsform empfängt die GPGPU 2530 Befehle von einem Host-Prozessor und sie verwendet einen globalen Planer 2534, um diesen Befehlen zugeordnete Ausführungs-Threads an einen Satz von Rechenclustern 2536A-2536H zu verteilen. In mindestens einer Ausführungsform nutzen die Rechencluster 2536A-2536H einen Cache-Speicher 2538 gemeinsam. In mindestens einer Ausführungsform kann der Cache-Speicher 2538 als Cache höherer Ebene für Cache-Speicher innerhalb der Rechencluster 2536A-2536H dienen.

[0199] In mindestens einer Ausführungsform beinhaltet die GPGPU 2530 Speicher 2544A-2544B, der über einen Satz von Speichersteuerungen 2542A-2542B mit den Rechenclustern 2536A-2536H gekoppelt ist. In mindestens einer Ausführungsform kann der Speicher 2544A-2544B verschiedene Arten von Speichervorrichtungen beinhalten, einschließlich dynamischen Direktzugriffsspeichers (DRAM) oder Grafik-Direktzugriffsspeichers, wie etwa synchronen Grafik-Direktzugriffsspeichers (SGRAM), einschließlich Grafik-Double-Data-Rate-(GDDR-)Speichers.

[0200] In mindestens einer Ausführungsform beinhalten die Rechencluster 2536A-2536H jeweils einen Satz von Grafikkernen, wie etwa den Grafikkern 2500 aus **Fig. 25A**, der mehrere Arten von logischen Einheiten für Integer und Gleitkommazahlen beinhalten kann, die Rechenoperationen mit einer Reihe von Genauigkei-

ten durchführen können, einschließlich solcher, die für Berechnungen für maschinelles Lernen geeignet sind. Zum Beispiel kann in mindestens einer Ausführungsform mindestens eine Teilmenge der Gleitkommaeinheiten in jedem der Rechencluster 2536A-2536H dazu konfiguriert sein, 16-Bit- oder 32-Bit-Gleitkommaoperationen durchzuführen, während eine andere Teilmenge der Gleitkommaeinheiten dazu konfiguriert sein kann, 64-Bit-Gleitkommaoperationen durchzuführen.

[0201] In mindestens einer Ausführungsform können mehrere Instanzen der GPGPU 2530 dazu konfiguriert sein, als Rechencluster zu arbeiten. In mindestens einer Ausführungsform variiert die Kommunikation, die durch die Rechencluster 2536A-2536H für die Synchronisation und den Datenaustausch verwendet wird, je nach Ausführungsform. In mindestens einer Ausführungsform kommunizieren mehrere Instanzen der GPGPU 2530 über die Host-Schnittstelle 2532. In mindestens einer Ausführungsform beinhaltet die GPGPU 2530 einen E/A-Hub 2539, der die GPGPU 2530 mit einer GPU-Verbindung 2540 koppelt, die eine direkte Verbindung zu anderen Instanzen der GPGPU 2530 ermöglicht. In mindestens einer Ausführungsform ist die GPU-Verbindung 2540 an eine dedizierte GPU-zu-GPU-Brücke gekoppelt, die Kommunikation und Synchronisation zwischen mehreren Instanzen der GPGPU 2530 ermöglicht. In mindestens einer Ausführungsform ist die GPU-Verbindung 2540 mit einer Hochgeschwindigkeitszusammenschaltung gekoppelt, um Daten an andere GPGPUs oder Parallelprozessoren zu senden und davon zu empfangen. In mindestens einer Ausführungsform befinden sich mehrere Instanzen der GPGPU 2530 in separaten Datenverarbeitungssystemen und kommunizieren über eine Netzwerkvorrichtung, auf die über die Host-Schnittstelle 2532 zugegriffen werden kann. In mindestens einer Ausführungsform kann die GPU-Verbindung 2540 dazu konfiguriert sein, eine Verbindung zu einem Host-Prozessor zusätzlich zu oder alternativ zu der Host-Schnittstelle 2532 zu ermöglichen.

[0202] In mindestens einer Ausführungsform kann die GPGPU 2530 dazu konfiguriert sein, neuronale Netze zu trainieren. In mindestens einer Ausführungsform kann die GPGPU 2530 innerhalb einer Inferenzplattform verwendet werden. In mindestens einer Ausführungsform, in der die GPGPU 2530 zum Inferenzieren verwendet wird, kann die GPGPU weniger Rechencluster 2536A-2536H beinhalten, als wenn die GPGPU zum Trainieren eines neuronalen Netzes verwendet wird. In mindestens einer Ausführungsform kann sich die dem Speicher 2544A-2544B zugeordnete Speichertechnologie zwischen Inferenz- und Trainingskonfigurationen unterscheiden, wobei den Trainingskonfigurationen Speichertechnologien mit höherer Bandbreite gewidmet sind. In mindestens einer Ausführungsform kann die Inferenzkonfiguration der GPGPU 2530 inferenzspezifische Anweisungen unterstützen. In mindestens einer Ausführungsform kann eine Inferenzkonfiguration zum Beispiel Unterstützung für eine oder mehrere 8-Bit-Integer-Skalarprodukt-Anweisungen bereitstellen, die während der Inferenzoperationen für eingesetzte neuronale Netze verwendet werden können.

[0203] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 in der GPGPU 2530 für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungssparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0204] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0205] **Fig. 26** ist ein Blockdiagramm, das ein Rechensystem 2600 gemäß mindestens einer Ausführungsform veranschaulicht. In mindestens einer Ausführungsform beinhaltet das Rechensystem 2600 ein Verarbeitungsteilsystem 2601, das einen oder mehrere Prozessor(en) 2602 und einen Systemspeicher 2604 aufweist, die über einen Zusammenschaltungspfad kommunizieren, der einen Speicher-Hub 2605 beinhalten kann. In mindestens einer Ausführungsform kann der Speicher-Hub 2605 eine separate Komponente innerhalb einer Chipsatzkomponente sein oder kann in einen oder mehrere Prozessor(en) 2602 integriert sein. In mindestens einer Ausführungsform ist der Speicher-Hub 2605 über eine Kommunikationsverbindung 2606 mit einem E/A-Teilsystem 2611 gekoppelt. In mindestens einer Ausführungsform beinhaltet das E/A-Teilsystem 2611 einen E/A-Hub 2607, der es dem Rechensystem 2600 ermöglichen kann, Eingaben von einer oder mehreren Eingabevorrichtung(en) 2608 zu empfangen. In mindestens einer Ausführungsform kann der E/A-Hub 2607 einer Anzeigesteuerung, die in einem oder mehreren Prozessor(en) 2602 enthalten sein kann, ermöglichen,

einer oder mehreren Anzeigevorrichtung(en) 2610A Ausgaben bereitzustellen. In mindestens einer Ausführungsform können eine oder mehrere mit dem E/A-Hub 2607 gekoppelte Anzeigevorrichtung(en) 2610A eine lokale, interne oder eingebettete Anzeigevorrichtung beinhalten.

[0206] In mindestens einer Ausführungsform beinhaltet das Verarbeitungsteilsystem 2601 einen oder mehrere Parallelprozessor(en) 2612, die über einen Bus oder eine andere Kommunikationsverbindung 2613 an den Speicher-Hub 2605 gekoppelt sind. In mindestens einer Ausführungsform kann die Kommunikationsverbindung 2613 eine von einer beliebigen Anzahl von standardbasierten Kommunikationsverbindungstechnologien oder -protokollen sein, wie etwa PCI Express, ohne darauf beschränkt zu sein, oder eine anbieterspezifische Kommunikationsschnittstelle oder Kommunikationsstruktur. In mindestens einer Ausführungsform bilden ein oder mehrere Parallelprozessor(en) 2612 ein rechnerisch fokussiertes Parallel- oder Vektorverarbeitungssystem, das eine große Anzahl von Verarbeitungskernen und/oder Verarbeitungsklustern beinhalten kann, wie etwa einen Prozessor mit vielen integrierten Kernen (Many Integrated Core - MIC). In mindestens einer Ausführungsform bilden einige oder alle Parallelprozessor(en) 2612 ein Grafikverarbeitungsteilsystem, das Pixel an eine oder mehrere Anzeigevorrichtung(en) 2610A ausgeben kann, die über den E/A-Hub 2607 gekoppelt sind. In mindestens einer Ausführungsform können ein oder mehrere Parallelprozessor(en) 2612 auch eine Anzeigesteuerung und eine Anzeigeschnittstelle (nicht gezeigt) beinhalten, um eine direkte Verbindung zu einer oder mehreren Anzeigevorrichtung(en) 2610B zu ermöglichen.

[0207] In mindestens einer Ausführungsform kann eine Systemspeichereinheit 2614 mit dem E/A-Hub 2607 verbunden sein, um einen Speichermechanismus für das Rechensystem 2600 bereitzustellen. In mindestens einer Ausführungsform kann ein E/A-Switch 2616 verwendet werden, um einen Schnittstellenmechanismus bereitzustellen, der Verbindungen zwischen dem E/A-Hub 2607 und anderen Komponenten ermöglicht, wie etwa ein Netzwerkadapter 2618 und/oder ein drahtloser Netzwerkadapter 2619, die in (eine) Plattform(en) integriert werden können, sowie verschiedenen anderen Vorrichtungen, die über eine oder mehrere Erweiterungsvorrichtung(en) 2620 hinzugefügt werden können. In mindestens einer Ausführungsform kann der Netzwerkadapter 2618 ein Ethernet-Adapter oder ein anderer drahtgebundener Netzwerkadapter sein. In mindestens einer Ausführungsform kann der drahtlose Netzwerkadapter 2619 eine oder mehrere Wi-Fi-, Bluetooth-, Nahfeldkommunikations- (NFC-) oder eine andere Netzwerkvorrichtung beinhalten, die ein oder mehrere drahtlose Funkgeräte beinhaltet.

[0208] In mindestens einer Ausführungsform kann das Rechensystem 2600 andere, nicht explizit gezeigte Komponenten beinhalten, z. B. USB- oder andere Portverbindungen, optische Speicherlaufwerke, Videoaufnahmevorrichtungen usw., die ebenfalls mit dem E/A-Hub 2607 verbunden sein können. In mindestens einer Ausführungsform können Kommunikationspfade, die verschiedene Komponenten in **Fig. 26** zusammenschalten, unter Verwendung beliebiger geeigneter Protokolle implementiert werden, wie etwa auf PCI (Peripheral Component Interconnect) basierender Protokolle (z. B. PCI-Express) oder anderer Bus- oder Punkt-zu-Punkt-Kommunikationsschnittstellen und/oder -protokoll(e), wie etwa NV-Link-Hochgeschwindigkeitszusammenschaltung, oder Zusammenschaltungsprotokolle.

[0209] In mindestens einer Ausführungsform beinhalten ein oder mehrere Parallelprozessor(en) 2612 Schaltungen, die für die Grafik- und Videoverarbeitung optimiert sind, einschließlich zum Beispiel Videoausgabeschaltungen, und stellen eine Grafikverarbeitungseinheit (GPU) dar. In mindestens einer Ausführungsform beinhalten ein oder mehrere Parallelprozessor(en) 2612 Schaltungen, die für Universalverarbeitung optimiert sind. In mindestens einer Ausführungsform können die Komponenten des Rechensystems 2600 in ein oder mehrere andere Systemelemente auf einer einzelnen integrierten Schaltung integriert sein. Zum Beispiel können in mindestens einer Ausführungsform ein oder mehrere Parallelprozessor(en) 2612, der Speicher-Hub 2605, der/die Prozessor(en) 2602 und der E/A-Hub 2607 in eine integrierte Schaltung als System auf einem Chip (SoC) integriert sein. In mindestens einer Ausführungsform können die Komponenten des Rechensystems 2600 in ein einzelnes Gehäuse integriert sein, um eine Konfiguration mit einem System in einem Gehäuse (system in package - SIP) zu bilden. In mindestens einer Ausführungsform kann mindestens ein Abschnitt der Komponenten des Rechensystems 2600 in ein Mehrchipmodul (multi-chip module - MCM) integriert sein, das mit anderen Mehrchipmodulen Modulen zu einem modularen Rechensystem zusammengeschaltet sein kann.

[0210] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 in dem System aus **FIG. 2600** für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von

Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0211] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

PROZESSOREN

[0212] Fig. 27A veranschaulicht einen Parallelprozessor 2700 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform können verschiedene Komponenten des Parallelprozessors 2700 unter Verwendung einer oder mehrerer integrierter Schaltungen implementiert werden, wie etwa programmierbare Prozessoren, anwendungsspezifische integrierte Schaltungen (ASICs) oder feldprogrammierbare Gate-Arrays (FPGAs). In mindestens einer Ausführungsform ist der veranschaulichte Parallelprozessor 2700 eine Variante eines oder mehrerer der in Fig. 26 gezeigten Parallelprozessor(en) 2612, gemäß einem Ausführungsbeispiel.

[0213] In mindestens einer Ausführungsform beinhaltet der Parallelprozessor 2700 eine Parallelverarbeitungseinheit 2702. In mindestens einer Ausführungsform beinhaltet die Parallelverarbeitungseinheit 2702 eine E/A-Einheit 2704, die Kommunikation mit anderen Vorrichtungen ermöglicht, einschließlich anderer Instanzen der Parallelverarbeitungseinheit 2702. In mindestens einer Ausführungsform kann die E/A-Einheit 2704 direkt mit anderen Vorrichtungen verbunden sein. In mindestens einer Ausführungsform ist die E/A-Einheit 2704 über die Verwendung einer Hub- oder Switch-Schnittstelle, wie etwa des Speicher-Hubs 2605, mit anderen Vorrichtungen verbunden. In mindestens einer Ausführungsform bilden Verbindungen zwischen dem Speicher-Hub 2605 und der E/A-Einheit 2704 eine Kommunikationsverbindung 2613. In mindestens einer Ausführungsform ist die E/A-Einheit 2704 mit einer Host-Schnittstelle 2706 und einer Speicherkreuzschiene 2716 verbunden, wobei die Host-Schnittstelle 2706 Befehle empfängt, die auf das Durchführen von Verarbeitungsoperationen gerichtet sind, und die Speicherkreuzschiene 2716 Befehle empfängt, die auf das Durchführen von Speicheroperationen gerichtet sind.

[0214] In mindestens einer Ausführungsform kann, wenn die Host-Schnittstelle 2706 einen Befehlspuffer über die E/A-Einheit 2704 empfängt, die Host-Schnittstelle 2706 Arbeitsoperationen zum Durchführen dieser Befehle an ein Frontend 2708 richten. In mindestens einer Ausführungsform ist das Frontend 2708 mit einem Planer 2710 gekoppelt, der dazu konfiguriert ist, Befehle oder andere Arbeitselemente an ein Verarbeitungsclusterarray 2712 zu verteilen. In mindestens einer Ausführungsform stellt der Planer 2710 sicher, dass das Verarbeitungsclusterarray 2712 ordnungsgemäß konfiguriert ist und sich in einem gültigen Zustand befindet, bevor Aufgaben an das Verarbeitungsclusterarray 2712 verteilt werden. In mindestens einer Ausführungsform ist der Planer 2710 über Firmware-Logik implementiert, die auf einem Mikrocontroller ausgeführt wird. In mindestens einer Ausführungsform ist der Mikrocontroller-implementierte Planer 2710 dazu konfigurierbar, komplexe Planungs- und Arbeitsverteilungsoperationen mit grober und feiner Granularität durchzuführen, was eine schnelle Präemption und Kontextumschaltung von Threads ermöglicht, die auf dem Verarbeitungsclusterarray 2712 ausgeführt werden. In mindestens einer Ausführungsform kann die Host-Software Arbeitslasten für die Planung auf dem Verarbeitungsclusterarray 2712 über eine von mehreren Grafikverarbeitungs-Doorbells prüfen. In mindestens einer Ausführungsform können die Arbeitslasten dann automatisch auf das Verarbeitungsclusterarray 2712 durch die Logik des Planers 2710 innerhalb eines Mikrocontrollers, einschließlich des Planers 2710, verteilt werden.

[0215] In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2712 bis zu „N“ Verarbeitungscluster beinhalten (z. B. Cluster 2714A, Cluster 2714B bis Cluster 2714N). In mindestens einer Ausführungsform kann jedes Cluster 2714A-2714N des Verarbeitungsclusterarrays 2712 eine große Anzahl von gleichzeitigen Threads ausführen. In mindestens einer Ausführungsform kann der Planer 2710 den Clustern 2714A-2714N des Verarbeitungsclusterarrays 2712 Arbeit zuweisen, indem er verschiedene Planungs- und/oder Arbeitsverteilungsalgorithmen verwendet, die in Abhängigkeit von der Arbeitslast variieren können, die für jede Art von Programm oder Berechnung entsteht. In mindestens einer Ausführungsform kann die Planung dynamisch durch den Planer 2710 gehandhabt werden oder teilweise durch Compiler-Logik während der Kompilierung der Programmlogik unterstützt werden, die für die Ausführung durch das Verarbeitungsclusterarray 2712 konfiguriert ist. In mindestens einer Ausführungsform können unterschiedliche Cluster 2714A-2714N des Verarbeitungsclusterarrays 2712 zum Verarbeiten unterschiedlicher Arten von Programmen oder zum Durchführen unterschiedlicher Arten von Berechnungen zugewiesen sein.

[0216] In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2712 dazu konfiguriert sein, verschiedene Arten von Parallelverarbeitungsoperationen durchzuführen. In mindestens einer Ausführungsform ist das Verarbeitungsclusterarray 2712 dazu konfiguriert, Universal-Parallelberechnungsoperationen durchzuführen. In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2712 zum Beispiel Logik zum Ausführen von Verarbeitungsaufgaben beinhalten, einschließlich des Filterns von Video- und/oder Audiodaten, des Durchführens von Modellierungsoperationen, einschließlich Physikoperationen, und des Durchführens von Datentransformationen.

[0217] In mindestens einer Ausführungsform ist das Verarbeitungsclusterarray 2712 dazu konfiguriert, Parallelgrafikverarbeitungsoperationen durchzuführen. In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2712 zusätzliche Logik beinhalten, um die Ausführung solcher Grafikverarbeitungsoperationen zu unterstützen, einschließlich, aber nicht beschränkt auf, Texturabastlogik, um Texturoperationen durchzuführen, sowie Tessellationslogik und andere Vertex-Verarbeitungslogik. In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2712 dazu konfiguriert sein, grafikverarbeitungsbezogene Shader-Programme auszuführen, wie etwa ohne Einschränkung Vertex-Shader, Tessellations-Shader, Geometrie-Shader und Pixel-Shader. In mindestens einer Ausführungsform kann die Parallelverarbeitungseinheit 2702 Daten aus dem Systemspeicher über die E/A-Einheit 2704 zur Verarbeitung übertragen. In mindestens einer Ausführungsform können die übertragenen Daten während der Verarbeitung in einem chipinternen Speicher (z. B. dem Parallelprozessorspeicher 2722) während der Verarbeitung gespeichert und dann wieder in den Systemspeicher geschrieben werden.

[0218] Wenn die Parallelverarbeitungseinheit 2702 zum Durchführen der Grafikverarbeitung verwendet wird, kann der Planer 2710 in mindestens einer Ausführungsform dazu konfiguriert sein, eine Verarbeitungsarbeitslast in annähernd gleich große Aufgaben aufzuteilen, um eine bessere Verteilung der Grafikverarbeitungsoperationen auf mehrere Cluster 2714A-2714N des Verarbeitungsclusterarrays 2712 zu ermöglichen. In mindestens einer Ausführungsform können Abschnitte des Verarbeitungsclusterarrays 2712 dazu konfiguriert sein, verschiedene Arten von Verarbeitung durchzuführen. Zum Beispiel kann in mindestens einer Ausführungsform ein erster Abschnitt dazu konfiguriert sein, Vertex-Shading und Topologieerzeugung durchzuführen, ein zweiter Abschnitt dazu konfiguriert sein, Tessellations- und Geometrie-Shading durchzuführen und ein dritter Abschnitt dazu konfiguriert sein, Pixel-Shading oder andere Bildschirmraumoperationen durchzuführen, um ein gerendertes Bild zur Anzeige zu produzieren. In mindestens einer Ausführungsform können Zwischendaten, die durch eines oder mehrere der Cluster 2714A-2714N erzeugt werden, in Puffern gespeichert werden, um zu ermöglichen, dass Zwischendaten zwischen den Clustern 2714A-2714N zur weiteren Verarbeitung übertragen werden.

[0219] In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2712 auszuführende Verarbeitungsaufgaben über den Planer 2710 empfangen, der von dem Frontend 2708 Befehle empfängt, die Verarbeitungsaufgaben definieren. In mindestens einer Ausführungsform können Verarbeitungsaufgaben Indizes von zu verarbeitenden Daten beinhalten, z. B. Oberflächen-(Patch-)Daten, Primitivdaten, Vertexdaten und/oder Pixeldaten sowie Zustandsparameter und Befehle, die definieren, wie Daten verarbeitet werden sollen (z. B. welches Programm ausgeführt werden soll). In mindestens einer Ausführungsform kann der Planer 2710 dazu konfiguriert sein, Indizes abzurufen, die Aufgaben entsprechen, oder kann Indizes von dem Frontend 2708 empfangen. In mindestens einer Ausführungsform kann das Frontend 2708 dazu konfiguriert sein, sicherzustellen, dass das Verarbeitungsclusterarray 2712 in einen gültigen Zustand konfiguriert wird, bevor eine durch eingehende Befehlspuffer (z. B. Batch-Puffer, Push-Puffer usw.) vorgegebene Arbeitslast initiiert wird.

[0220] In mindestens einer Ausführungsform kann jede von einer oder mehreren Instanzen der Parallelverarbeitungseinheit 2702 mit dem Parallelprozessorspeicher 2722 gekoppelt sein. In mindestens einer Ausführungsform kann auf den Parallelprozessorspeicher 2722 über die Speicherkreuzschiene 2716 zugegriffen werden, die Speicheranforderungen von dem Verarbeitungsclusterarray 2712 sowie von der E/A-Einheit 2704 empfangen kann. In mindestens einer Ausführungsform kann die Speicherkreuzschiene 2716 über eine Speicherschnittstelle 2718 auf den Parallelprozessorspeicher 2722 zugreifen. In mindestens einer Ausführungsform kann die Speicherschnittstelle 2718 mehrere Partitionseinheiten (z. B. Partitionseinheit 2720A, Partitionseinheit 2720B bis Partitionseinheit 2720N) beinhalten, die jeweils an einen Abschnitt (z. B. Speichereinheit) des Parallelprozessorspeichers 2722 gekoppelt werden können. In mindestens einer Ausführungsform ist eine Anzahl der Partitionseinheiten 2720A-2720N dazu konfiguriert, gleich einer Anzahl von Speichereinheiten zu sein, sodass eine erste Partitionseinheit 2720A eine entsprechende erste Speichereinheit 2724A aufweist, eine zweite Partitionseinheit 2720B eine entsprechende Speichereinheit 2724B aufweist und eine N-te Partitionseinheit 2720N eine entsprechende N-te Speichereinheit 2724N aufweist. In mindes-

tens einer Ausführungsform darf eine Anzahl der Partitionseinheiten 2720A-2720N nicht gleich einer Anzahl der Speichervorrichtungen sein.

[0221] In mindestens einer Ausführungsform können die Speichereinheiten 2724A-2724N verschiedene Arten von Speichervorrichtungen beinhalten, einschließlich dynamischen Direktzugriffsspeichers (DRAM) oder Grafik-Direktzugriffsspeichers, wie etwa synchronen Grafik-Direktzugriffsspeichers (SGRAM), einschließlich Grafik-Double-Data-Rate-(GDDR-)Speichers. In mindestens einer Ausführungsform können die Speichereinheiten 2724A-2724N auch 3D-Stapelspeicher beinhalten, einschließlich, aber nicht beschränkt auf, Speicher mit hoher Bandbreite (high bandwidth memory - HBM). In mindestens einer Ausführungsform können Rendering-Ziele, wie etwa Bildpuffer oder Texturkarten, über die Speichereinheiten 2724A-2724N hinweg gespeichert werden, was es den Partitionseinheiten 2720A-2720N ermöglicht, Abschnitte jedes Rendering-Ziels parallel zu schreiben, um die verfügbare Bandbreite des Parallelprozessorspeichers 2722 effizient zu nutzen. In mindestens einer Ausführungsform kann eine lokale Instanz des Parallelprozessorspeichers 2722 zugunsten einer vereinheitlichten Speicherausgestaltung ausgeschlossen werden, die Systemspeicher in Verbindung mit lokalem Cache-Speicher verwendet.

[0222] In mindestens einer Ausführungsform kann ein beliebiges der Cluster 2714A-2714N des Verarbeitungsclusterarrays 2712 Daten verarbeiten, die in beliebige der Speichereinheiten 2724A-2724N innerhalb des Parallelprozessorspeichers 2722 geschrieben werden. In mindestens einer Ausführungsform kann die Speicherkreuzschiene 2716 dazu konfiguriert sein, eine Ausgabe jedes Clusters 2714A-2714N an eine beliebige Partitionseinheit 2720A-2720N oder an ein anderes Cluster 2714A-2714N zu übertragen, das zusätzliche Verarbeitungsoperationen an einer Ausgabe durchführen kann. In mindestens einer Ausführungsform kann jedes Cluster 2714A-2714N durch die Speicherkreuzschiene 2716 mit der Speicherschnittstelle 2718 kommunizieren, um aus verschiedenen externen Speichervorrichtungen zu lesen oder in diese zu schreiben. In mindestens einer Ausführungsform weist die Speicherkreuzschiene 2716 eine Verbindung mit der Speicherschnittstelle 2718 auf, um mit der E/A-Einheit 2704 zu kommunizieren, sowie eine Verbindung mit einer lokalen Instanz des Parallelprozessorspeichers 2722, was es den Verarbeitungseinheiten innerhalb der unterschiedlichen Verarbeitungscluster 2714A-2714N ermöglicht, mit Systemspeicher oder anderem Speicher zu kommunizieren, der nicht lokal zu der Parallelverarbeitungseinheit 2702 ist. In mindestens einer Ausführungsform kann die Speicherkreuzschiene 2716 virtuelle Kanäle verwenden, um Verkehrsströme zwischen Clustern 2714A-2714N und Partitionseinheiten 2720A-2720N zu trennen.

[0223] In mindestens einer Ausführungsform können mehrere Instanzen der Parallelverarbeitungseinheit 2702 auf einer einzelnen Erweiterungskarte bereitgestellt sein oder mehrere Erweiterungskarten zusammengeschaltet sein. In mindestens einer Ausführungsform können unterschiedliche Instanzen der Parallelverarbeitungseinheit 2702 dazu konfiguriert sein, zusammenzuarbeiten, auch wenn die unterschiedlichen Instanzen unterschiedliche Anzahlen von Verarbeitungskernen, unterschiedliche Mengen von lokalem Parallelprozessorspeicher und/oder andere Konfigurationsunterschiede aufweisen. Zum Beispiel können in mindestens einer Ausführungsform einige Instanzen der Parallelverarbeitungseinheit 2702 Gleitkommaeinheiten mit höherer Genauigkeit in Bezug auf andere Instanzen beinhalten. In mindestens einer Ausführungsform können Systeme, die eine oder mehrere Instanzen der Parallelverarbeitungseinheit 2702 oder des Parallelprozessors 2700 beinhalten, in einer Vielfalt von Konfigurationen und Formfaktoren implementiert sein, einschließlich, aber nicht beschränkt auf, Desktop-, Laptop- oder tragbarer Personalcomputer, Server, Arbeitsstationen, Spielekonsolen und/oder eingebetteter Systeme.

[0224] **Fig. 27B** ist ein Blockdiagramm einer Partitionseinheit 2720 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist die Partitionseinheit 2720 eine Instanz einer der Partitionseinheiten 2720A-2720N aus **Fig. 27A**. In mindestens einer Ausführungsform beinhaltet die Partitionseinheit 2720 einen L2-Cache 2721, eine Bildpufferschnittstelle 2725 und eine Rasteroperationseinheit (raster operations unit-„ROP“) 2726. Der L2-Cache 2721 ist ein Lese-/Schreib-Cache, der dazu konfiguriert ist, Lade- und Sicherungsoperationen durchzuführen, die von der Speicherkreuzschiene 2716 und der ROP 2726 empfangen werden. In mindestens einer Ausführungsform werden Lesefehler und dringende Rückschreibenanforderungen durch den L2-Cache 2721 an die Bildpufferschnittstelle 2725 zur Verarbeitung ausgegeben. In mindestens einer Ausführungsform können Aktualisierungen auch über die Bildpufferschnittstelle 2725 zur Verarbeitung an einen Bildpuffer gesendet werden. In mindestens einer Ausführungsform bildet die Bildpufferschnittstelle 2725 eine Schnittstelle mit einer der Speichereinheiten in dem Parallelprozessorspeicher, wie etwa mit den Speichereinheiten 2724A-2724N aus **Fig. 27** (z. B. in dem Parallelprozessorspeicher 2722).

[0225] In mindestens einer Ausführungsform ist die ROP 2726 eine Verarbeitungseinheit, die Rasteroperationen, wie etwa Stencil, Z-Test, Blending usw., durchführt. In mindestens einer Ausführungsform gibt die

ROP 2726 dann verarbeitete Grafikdaten aus, die im Grafikspeicher gespeichert werden. In mindestens einer Ausführungsform beinhaltet die ROP 2726 Komprimierungslogik, um Tiefen- oder Farbdaten, die in den Speicher geschrieben werden, zu komprimieren und Tiefen- oder Farbdaten, die aus dem Speicher gelesen werden, zu dekomprimieren. In mindestens einer Ausführungsform kann die Komprimierungslogik verlustfreie Komprimierungslogik sein, die einen oder mehrere von mehreren Komprimierungsalgorithmen verwendet. Die Komprimierungslogik, die von der ROP 2726 ausgeführt wird, kann auf Grundlage von statistischen Eigenschaften der zu komprimierenden Daten variieren. Zum Beispiel wird in mindestens einer Ausführungsform die Delta-Farbkomprimierung an Tiefen- und Farbdaten auf Kachelbasis ausgeführt.

[0226] In mindestens einer Ausführungsform ist die ROP 2726 innerhalb jedes Verarbeitungsclusters (z. B. Cluster 2714A-2714N aus **Fig. 27A**) anstatt innerhalb der Partitionseinheit 2720 enthalten. In mindestens einer Ausführungsform werden Lese- und Schreibsanforderungen für Pixeldaten anstelle von Pixelfragmentdaten über die Speicherkreuzschiene 2716 übertragen. In mindestens einer Ausführungsform können verarbeitete Grafikdaten auf einer Anzeigevorrichtung angezeigt werden, wie etwa einer der einen oder der mehreren Anzeigevorrichtung(en) 2610 aus **Fig. 26**, zur weiteren Verarbeitung durch den/die Prozessor(en) 2602 geleitet werden oder zur weiteren Verarbeitung durch eine der Verarbeitungsentitäten innerhalb des Parallelprozessors 2700 aus **Fig. 27A** geleitet werden.

[0227] **Fig. 27C** ist ein Blockdiagramm eines Verarbeitungsclusters 2714 innerhalb einer Parallelverarbeitungseinheit gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist ein Verarbeitungscluster eine Instanz eines der Verarbeitungscluster 2714A-2714N aus **Fig. 27A**. In mindestens einer Ausführungsform können einer oder mehrere des/der Verarbeitungscluster(s) 2714 dazu konfiguriert sein, viele Threads parallel auszuführen, wobei sich „Thread“ auf eine Instanz eines konkreten Programms bezieht, das an einem konkreten Satz von Eingabedaten ausgeführt wird. In mindestens einer Ausführungsform werden Ausgabetechniken für Single-Instruction-Multiple-Data-(SIMD-)Anweisungen verwendet, um die parallele Ausführung einer großen Anzahl von Threads zu unterstützen, ohne mehrere unabhängige Anweisungseinheiten bereitzustellen. In mindestens einer Ausführungsform werden Single-Instruction-Multiple-Thread-(SIMT-)Techniken verwendet, um die parallele Ausführung einer großen Anzahl von im Allgemeinen synchronisierten Threads zu unterstützen, wobei eine gemeinsame Anweisungseinheit verwendet wird, die dazu konfiguriert ist, Anweisungen an einen Satz von Verarbeitungs-Engines innerhalb jedes der Verarbeitungscluster auszugeben.

[0228] In mindestens einer Ausführungsform kann der Betrieb des Verarbeitungsclusters 2714 über einen Pipeline-Manager 2732 gesteuert werden, der die Verarbeitungsaufgaben auf die SIMT-Parallelprozessoren verteilt. In mindestens einer Ausführungsform empfängt der Pipeline-Manager 2732 Anweisungen von dem Planer 2710 aus **Fig. 34A** und verwaltet die Ausführung dieser Anweisungen über einen Grafik-Multiprozessor 2734 und/oder eine Textureinheit 2736. In mindestens einer Ausführungsform ist der Grafik-Multiprozessor 2734 eine beispielhafte Instanz eines SIMT-Parallelprozessors. In mindestens einer Ausführungsform können jedoch verschiedene Arten von SIMT-Parallelprozessoren mit unterschiedlichen Architekturen innerhalb des Verarbeitungsclusters 2714 enthalten sein. In mindestens einer Ausführungsform können eine oder mehrere Instanzen des Grafik-Multiprozessors 2734 innerhalb eines Verarbeitungsclusters 2714 enthalten sein. In mindestens einer Ausführungsform kann der Grafik-Multiprozessor 2734 Daten verarbeiten und kann eine Datenkreuzschiene 2740 verwendet werden, um verarbeitete Daten an eines von mehreren möglichen Zielen, einschließlich anderer Shader-Einheiten, zu verteilen. In mindestens einer Ausführungsform kann der Pipeline-Manager 2732 die Verteilung der verarbeiteten Daten ermöglichen, indem er Ziele für die zu verteilenden verarbeiteten Daten gegenüber der Datenkreuzschiene 2740 angibt.

[0229] In mindestens einer Ausführungsform kann jeder Grafik-Multiprozessor 2734 innerhalb des Verarbeitungsclusters 2714 einen identischen Satz von funktioneller Ausführungslogik (z. B. arithmetisch-logische Einheiten, Lade-Speicher-Einheiten usw.) beinhalten. In mindestens einer Ausführungsform kann funktionelle Ausführungslogik pipelineartig konfiguriert sein, wobei neue Anweisungen ausgegeben werden können, bevor vorherige Anweisungen abgeschlossen sind. In mindestens einer Ausführungsform unterstützt die funktionelle Ausführungslogik eine Vielfalt von Operationen, einschließlich Ganzzahl- und Gleitkommaarithmetik, Vergleichsoperationen, Boolescher Operationen, Bitverschiebung und Berechnung verschiedener algebraischer Funktionen. In mindestens einer Ausführungsform kann dieselbe Hardware einer funktionellen Einheit genutzt werden, um unterschiedliche Operationen auszuführen, und eine beliebige Kombination von funktionellen Einheiten kann vorhanden sein.

[0230] In mindestens einer Ausführungsform stellen die an das Verarbeitungscluster 2714 übertragenen Anweisungen einen Thread dar. In mindestens einer Ausführungsform ist ein Satz von Threads, der über

einen Satz von Parallelverarbeitungs-Engines ausgeführt wird, eine Thread-Gruppe. In mindestens einer Ausführungsform führt die Thread-Gruppe ein Programm an unterschiedlichen Eingabedaten aus. In mindestens einer Ausführungsform kann jeder Thread innerhalb einer Thread-Gruppe einer anderen Verarbeitungs-Engine innerhalb eines Grafik-Multiprozessors 2734 zugeordnet sein. In mindestens einer Ausführungsform kann eine Thread-Gruppe weniger Threads beinhalten als eine Anzahl von Verarbeitungs-Engines innerhalb des Grafik-Multiprozessors 2734. In mindestens einer Ausführungsform, wenn eine Thread-Gruppe weniger Threads beinhaltet als eine Anzahl von Verarbeitungs-Engines, können eine oder mehrere Verarbeitungs-Engines während der Zyklen, in denen diese Thread-Gruppe verarbeitet wird, inaktiv sein. In mindestens einer Ausführungsform kann eine Thread-Gruppe auch mehr Threads beinhalten als eine Anzahl von Verarbeitungs-Engines innerhalb des Grafik-Multiprozessors 2734. In mindestens einer Ausführungsform, wenn eine Thread-Gruppe mehr Threads als Verarbeitungs-Engines innerhalb des Grafik-Multiprozessors 2734 beinhaltet, kann die Verarbeitung über aufeinanderfolgende Taktzyklen durchgeführt werden. In mindestens einer Ausführungsform können mehrere Thread-Gruppen gleichzeitig auf einem Grafik-Multiprozessor 2734 ausgeführt werden.

[0231] In mindestens einer Ausführungsform beinhaltet der Grafik-Multiprozessor 2734 einen internen Cache-Speicher zum Durchführen von Lade- und Speicheroperationen. In mindestens einer Ausführungsform kann der Grafik-Multiprozessor 2734 auf einen internen Cache verzichten und einen Cache-Speicher (z.B. L1-Cache 2748) innerhalb des Verarbeitungsclusters 2714 verwenden. In mindestens einer Ausführungsform hat jeder Grafik-Multiprozessor 2734 auch Zugriff auf L2-Caches innerhalb von Partitionseinheiten (z. B. Partitionseinheiten 2720A-2720N aus **Fig. 27A**), die von allen Verarbeitungsclustern 2714 gemeinsam genutzt werden und zum Übertragen von Daten zwischen Threads verwendet werden können. In mindestens einer Ausführungsform kann der Grafik-Multiprozessor 2734 auch auf den globalen chipexternen Speicher zugreifen, der einen oder mehrere von lokalem Parallelprozessorspeicher und/oder Systemspeicher beinhalten kann. In mindestens einer Ausführungsform kann ein beliebiger Speicher, der zu der Parallelverarbeitungseinheit 2702 extern ist, als globaler Speicher verwendet werden. In mindestens einer Ausführungsform beinhaltet das Verarbeitungscluster 2714 mehrere Instanzen des Grafik-Multiprozessors 2734, die gemeinsame Anweisungen und Daten gemeinsam nutzen können, die im L1-Cache 2748 gespeichert sein können.

[0232] In mindestens einer Ausführungsform kann jedes Verarbeitungscluster 2714 eine Speicherverwaltungseinheit („MMU“) 2745 beinhalten, die dazu konfiguriert ist, virtuelle Adressen auf physische Adressen abzubilden. In mindestens einer Ausführungsform können sich eine oder mehrere Instanzen der MMU 2745 innerhalb der Speicherschnittstelle 2718 aus **Fig. 27A** befinden. In mindestens einer Ausführungsform beinhaltet die MMU 2745 einen Satz von Seitentabelleneinträgen (page table entries - PTEs), der dazu verwendet wird, eine virtuelle Adresse auf eine physische Adresse einer Kachel abzubilden, sowie optional einen Cache-Zeilenindex. In mindestens einer Ausführungsform kann die MMU 2745 Adressübersetzungspuffer (TLB) oder Caches beinhalten, die sich in dem Grafik-Multiprozessor 2734 oder in dem L1-Cache oder Verarbeitungscluster 2714 befinden können. In mindestens einer Ausführungsform wird die physische Adresse verarbeitet, um die Oberflächendaten-Zugriffslokalität zu verteilen, um eine effiziente Anforderungsver-schachtelung zwischen den Partitionseinheiten zu ermöglichen. In mindestens einer Ausführungsform kann der Cache-Zeilenindex verwendet werden, um zu bestimmen, ob eine Anforderung für eine Cache-Zeile ein Treffer oder eine Auslassung ist.

[0233] In mindestens einer Ausführungsform kann ein Verarbeitungscluster 2714 derart konfiguriert sein, dass jeder Grafik-Multiprozessor 2734 an eine Textureinheit 2736 zum Durchführen von Texturabbildungsoperationen gekoppelt ist, z. B. zum Bestimmen von Texturabtastpositionen, Lesen von Texturdaten und Filtern von Texturdaten. In mindestens einer Ausführungsform werden die Texturdaten aus einem internen Textur-L1-Cache (nicht gezeigt) oder aus einem L1-Cache innerhalb des Grafik-Multiprozessors 2734 gelesen und je nach Bedarf aus einem L2-Cache, dem lokalen Parallelprozessorspeicher oder dem Systemspeicher abgerufen. In mindestens einer Ausführungsform gibt jeder Grafik-Multiprozessor 2734 verarbeitete Aufgaben an die Datenkreuzschiene 2740 aus, um (eine) verarbeitete Aufgabe(n) einem anderen Verarbeitungscluster 2714 zur weiteren Verarbeitung bereitzustellen oder um (eine) verarbeitete Aufgabe(n) über die Speicherkreuzschiene 2716 in einem L2-Cache, lokalen Parallelprozessorspeicher oder Systemspeicher zu speichern. In mindestens einer Ausführungsform ist eine preROP 2742 (pre-raster operations unit - Vor-Rasteroperationeneinheit) dazu konfiguriert, Daten von dem Grafik-Multiprozessor 2734 zu empfangen und Daten an ROP-Einheiten zu leiten, die sich in Partitionseinheiten, wie in dieser Schrift beschrieben, befinden können (z. B. Partitionseinheiten 2720A-2720N aus **Fig. 27A**). In mindestens einer Ausführungsform kann die PreROP-Einheit 2742 Optimierungen für die Farbmischung durchführen, Pixelfarbdaten organisieren und Adressübersetzungen durchführen.

[0234] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 in dem Grafikverarbeitungscluster 2714 für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0235] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0236] **Fig. 27D** zeigt einen Grafik-Multiprozessor 2734 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist der Grafik-Multiprozessor 2734 mit dem Pipelinemanager 2732 des Verarbeitungsclusters 2714 gekoppelt. In mindestens einer Ausführungsform weist der Grafik-Multiprozessor 2734 eine Ausführungspipeline auf, die einen Anweisungs-Cache 2752, eine Anweisungseinheit 2754, eine Adressabbildungseinheit 2756, eine Registerdatei 2758, einen oder mehrere Kerne 2762 einer Universal-Grafikverarbeitungseinheit (GPGPU) und eine oder mehrere Lade-/Speichereinheiten 2766 beinhaltet, ohne darauf beschränkt zu sein. Der/die GPGPU-Kern(e) 2762 und die Lade-/Speichereinheit(en) 2766 sind über eine Speicher- und Cache-Zusammenschaltung 2768 mit dem Cache-Speicher 2772 und dem gemeinsam genutzten Speicher 2770 gekoppelt.

[0237] In mindestens einer Ausführungsform empfängt der Anweisungs-Cache 2752 einen Strom aus auszuführenden Anweisungen von dem Pipeline-Manager 2732. In mindestens einer Ausführungsform werden die Anweisungen im Anweisungs-Cache 2752 zwischengespeichert und von der Anweisungseinheit 2754 zur Ausführung zugeteilt. In mindestens einer Ausführungsform kann die Anweisungseinheit 2754 Anweisungen als Thread-Gruppen (z. B. Warps) zuteilen, wobei jede Thread-Gruppe einer anderen Ausführungseinheit innerhalb des/der GPGPU-Kerns/- Kerne 2762 zugewiesen ist. In mindestens einer Ausführungsform kann eine Anweisung auf einen beliebigen von einem lokalen, gemeinsam genutzten oder globalen Adressraum zugreifen, indem sie eine Adresse innerhalb eines vereinheitlichten Adressraums vorgibt. In mindestens einer Ausführungsform kann die Adressabbildungseinheit 2756 verwendet werden, um Adressen in einem vereinheitlichten Adressraum in eine eindeutige Speicheradresse zu übersetzen, auf die durch die Lade-/Speichereinheit(en) 2766 zugegriffen werden kann.

[0238] In mindestens einer Ausführungsform stellt die Registerdatei 2758 einen Satz von Registern für funktionelle Einheiten des Grafik-Multiprozessors 2734 bereit. In mindestens einer Ausführungsform stellt die Registerdatei 2758 einen temporären Datenspeicher für Operanden bereit, die mit Datenpfaden von funktionellen Einheiten (z. B. GPGPU-Kernen 2762, Lade-/Speichereinheiten 2766) des Grafik-Multiprozessors 2734 verbunden sind. In mindestens einer Ausführungsform ist die Registerdatei 2758 derart zwischen den einzelnen funktionellen Einheiten aufgeteilt, dass jeder funktionellen Einheit ein dedizierter Abschnitt der Registerdatei 2758 zugewiesen ist. In mindestens einer Ausführungsform ist die Registerdatei 2758 auf unterschiedliche Warps aufgeteilt, die durch den Grafik-Multiprozessor 2734 ausgeführt werden.

[0239] In mindestens einer Ausführungsform können die GPGPU-Kerne 2762 jeweils Gleitkommaeinheiten (floating point units - FPUs) und/oder arithmetisch-logische Einheiten (ALUs) für Integer beinhalten, die zum Ausführen von Anweisungen des Grafik-Multiprozessors 2734 verwendet werden. Die GPGPU-Kerne 2762 können eine ähnliche Architektur aufweisen oder sich bezüglich der Architektur unterscheiden. In mindestens einer Ausführungsform beinhaltet ein erster Abschnitt der GPGPU-Kerne 2762 eine FPU mit einfacher Genauigkeit und eine Integer-ALU, während ein zweiter Abschnitt der GPGPU-Kerne eine FPU mit doppelter Genauigkeit beinhaltet. In mindestens einer Ausführungsform können FPUs den Standard IEEE 754-2008 für Gleitkommaarithmetik implementieren oder Gleitkommaarithmetik mit variabler Genauigkeit ermöglichen. In mindestens einer Ausführungsform kann der Grafik-Multiprozessor 2734 zusätzlich eine oder mehrere Festfunktions- oder Spezialfunktionseinheiten beinhalten, um spezifische Funktionen, wie etwa Operationen zum Kopieren von Rechtecken oder Pixel-Blending, durchzuführen. In mindestens einer Ausführungsform können eine oder mehrere der GPGPU-Kerne auch Fest- oder Spezialfunktionslogik beinhalten.

[0240] In mindestens einer Ausführungsform beinhalten die GPGPU-Kerne 2762 SIMD-Logik, die in der Lage ist, eine einzelne Anweisung an mehreren Datensätzen durchzuführen. In mindestens einer Ausführungsform können die GPGPU-Kerne 2762 SIMD4-, SIMD8- und SIMD16-Anweisungen physisch ausführen

und SIMD1-, SIMD2- und SIMD32-Anweisungen logisch ausführen. In mindestens einer Ausführungsform können SIMD-Anweisungen für GPGPU-Kerne zur Kompilierzeit durch einen Shader-Compiler erzeugt werden oder automatisch erzeugt werden, wenn Programme ausgeführt werden, die für Single-Program-Multiple-Data-(SPMD-) oder SIMT-Architekturen geschrieben und kompiliert wurden. In mindestens einer Ausführungsform können mehrere Threads eines für ein SIMT-Ausführungsmodell konfigurierten Programms über eine einzelne SIMD-Anweisung ausgeführt werden. Zum Beispiel können in mindestens einer Ausführungsform acht SIMT-Threads, die gleiche oder ähnliche Operationen durchführen, parallel über eine einzelne SIMD8-Logikeinheit ausgeführt werden.

[0241] In mindestens einer Ausführungsform ist die Speicher- und Cache-Zusammenschaltung 2768 ein Zusammenschaltungsnetz, das jede funktionelle Einheit des Grafik-Multiprozessors 2734 mit der Registerdatei 2758 und dem gemeinsam genutzten Speicher 2770 verbindet. In mindestens einer Ausführungsform ist die Speicher- und Cache-Zusammenschaltung 2768 eine Kreuzschienen-Zusammenschaltung, die es der Lade-/Speichereinheit 2766 ermöglicht, Lade- und Speicheroperationen zwischen dem gemeinsam genutzten Speicher 2770 und der Registerdatei 2758 zu implementieren. In mindestens einer Ausführungsform kann die Registerdatei 2758 mit der gleichen Frequenz wie die GPGPU-Kerne 2762 arbeiten, sodass die Datenübertragung zwischen den GPGPU-Kernen 2762 und der Registerdatei 2758 eine sehr geringe Latenz aufweist. In mindestens einer Ausführungsform kann der gemeinsam genutzte Speicher 2770 verwendet werden, um die Kommunikation zwischen Threads zu ermöglichen, die auf funktionellen Einheiten innerhalb des Grafik-Multiprozessors 2734 ausgeführt werden. In mindestens einer Ausführungsform kann der Cache-Speicher 2772 zum Beispiel als Daten-Cache verwendet werden, um Texturdaten zwischenspeichern, die zwischen funktionellen Einheiten und der Textureinheit 2736 kommuniziert werden. In mindestens einer Ausführungsform kann der gemeinsam genutzte Speicher 2770 auch als programmverwalteter Cache verwendet werden. In mindestens einer Ausführungsform können Threads, die auf den GPGPU-Kernen 2762 ausgeführt werden, zusätzlich zu den automatisch zwischengespeicherten Daten, die innerhalb des Cache-Speichers 2772 gespeichert sind, programmatisch Daten innerhalb des gemeinsam genutzten Speichers speichern.

[0242] In mindestens einer Ausführungsform ist ein Parallelprozessor oder eine GPGPU, wie in dieser Schrift beschrieben, kommunikativ an Host-/Prozessorkerne gekoppelt, um Grafikoperationen, Operationen für maschinelles Lernen, Musteranalyseoperationen und verschiedene Funktionen einer Universal-GPU (GPGPU) zu beschleunigen. In mindestens einer Ausführungsform kann die GPU über einen Bus oder eine andere Zusammenschaltung (z. B. eine Hochgeschwindigkeitszusammenschaltung wie PCIe oder NVLink) mit dem Host-Prozessor/den Kernen kommunikativ gekoppelt sein. In mindestens einer Ausführungsform kann die GPU in demselben Gehäuse oder Chip wie die Kerne integriert sein und kommunikativ über einen internen Prozessorbus/eine interne Zusammenschaltung (d. h. intern zum Gehäuse oder Chip) mit den Kernen gekoppelt sein. In mindestens einer Ausführungsform können die Prozessorkerne unabhängig von der Weise, auf welche die GPU verbunden ist, der GPU Arbeit in Form von Sequenzen von Befehlen/Anweisungen zuweisen, die in einem Arbeitsdeskriptor enthalten sind. In mindestens einer Ausführungsform verwendet die GPU dann dedizierte Schaltungen/Logik zur effizienten Verarbeitung dieser Befehle/Anweisungen.

[0243] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 in dem Grafik-Multiprozessor 2734 für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0244] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0245] **Fig. 28** veranschaulicht ein Multi-GPU-Rechensystem 2800 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann das Multi-GPU-Rechensystem 2800 einen Prozessor 2802 beinhalten, der über einen Host-Schnittstellen-Switch 2804 an mehrere Universal-Grafikverarbeitungseinheiten (GPGPUs) 2806A-D gekoppelt ist. In mindestens einer Ausführungsform ist der Host-Schnittstellen-Switch 2804 eine PCI-Express-Switch-Vorrichtung, die den Prozessor 2802 an einen PCI-Express-Bus koppelt, über den der Prozessor 2802 mit den GPGPUs 2806A-D kommunizieren kann. Die GPGPUs 2806A-D können über einen Satz von Hochgeschwindigkeits-Punkt-zu-Punkt-GPU-zu-GPU-Verbindungen 2816

zusammengeschaltet sein. In mindestens einer Ausführungsform sind die GPU-zu-GPU-Verbindungen 2816 mit jeder der GPGPUs 2806A-D über eine dedizierte GPU-Verbindung verbunden. In mindestens einer Ausführungsform ermöglichen die P2P-GPU-Verbindungen 2816 direkte Kommunikation zwischen jeder der GPGPUs 2806A-D, ohne dass Kommunikation über den Host-Schnittstellenbus 2804 erforderlich ist, mit dem der Prozessor 2802 verbunden ist. In mindestens einer Ausführungsform, bei der GPU-zu-GPU-Verkehr zu den P2P-GPU-Verbindungen 2816 geleitet wird, bleibt der Host-Schnittstellenbus 2804 für den System-speicherzugriff oder zum Kommunizieren mit anderen Instanzen des Multi-GPU-Rechensystems 2800 verfügbar, zum Beispiel über eine oder mehrere Netzwerkvorrichtungen. Während in mindestens einer Ausführungsform die GPGPUs 2806A-D mit dem Prozessor 2802 über den Host-Schnittstellen-Switch 2804 verbunden sind, beinhaltet der Prozessor 2802 in mindestens einer Ausführungsform eine direkte Unterstützung für P2P-GPU-Verbindungen 2816 und kann direkt mit den GPGPUs 2806A-D verbunden sein.

[0246] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 in dem Multi-GPU-Rechensystem 2800 für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparemtern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0247] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0248] **Fig. 29** ist ein Blockdiagramm eines Grafikprozessors 2900 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2900 eine Ringzusammenschaltung 2902, ein Pipeline-Frontend 2904, eine Medien-Engine 2937 und Grafikkerne 2980A-2980N. In mindestens einer Ausführungsform koppelt die Ringzusammenschaltung 2902 den Grafikprozessor 2900 an andere Verarbeitungseinheiten, einschließlich anderer Grafikprozessoren oder eines oder mehrerer Universalprozessorkerne. In mindestens einer Ausführungsform ist der Grafikprozessor 2900 einer von vielen Prozessoren, die in ein Mehrkern-Verarbeitungssystem integriert sind.

[0249] In mindestens einer Ausführungsform empfängt der Grafikprozessor 2900 Batches von Befehlen über die Ringzusammenschaltung 2902. In mindestens einer Ausführungsform werden eingehende Befehle durch einen Befehls-Streamer 2903 in dem Pipeline-Frontend 2904 interpretiert. In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2900 skalierbare Ausführungslogik zum Durchführen von 3D-Geometrieverarbeitung und Medienverarbeitung über die Grafikkerne(e) 2980A-2980N. In mindestens einer Ausführungsform führt der Befehls-Streamer 2903 der Geometriepipeline 2936 Befehle für 3D-Geometrieverarbeitungsbefehle zu. In mindestens einer Ausführungsform führt der Befehls-Streamer 2903 für mindestens einige Medienverarbeitungsbefehle einem mit einer Medien-Engine 2937 gekoppelten Video-Frontend 2934 Befehle zu. In mindestens einer Ausführungsform beinhaltet die Medien-Engine 2937 eine Videoqualitäts-Engine (Video Quality Engine - VQE) 2930 für die Video- und Bildnachbearbeitung und eine Multiformat-Codier/Decodier-(multi-format encode/decode - MFX-)Engine 2933 zum Bereitstellen von hardwarebeschleunigter Codierung und Decodierung von Mediendaten. In mindestens einer Ausführungsform erzeugen die Geometriepipeline 2936 und die Medien-Engine 2937 jeweils Ausführungs-Threads für Thread-Ausführungsressourcen, die von mindestens einem Grafikkern 2980A bereitgestellt werden.

[0250] In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2900 skalierbare Thread-Ausführungsressourcen mit modularen Kernen 2980A-2980N (mitunter als Kernscheiben bezeichnet), die jeweils mehrere Teilkernen 2950A-2950N, 2960A-2960N (mitunter als Kernteilscheiben bezeichnet) aufweisen. In mindestens einer Ausführungsform kann der Grafikprozessor 2900 eine beliebige Anzahl von Grafikkernen 2980A bis 2980N aufweisen. In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2900 einen Grafikkern 2980A, der mindestens einen ersten Teilkern 2950A und einen zweiten Teilkern 2960A aufweist. In mindestens einer Ausführungsform ist der Grafikprozessor 2900 ein Prozessor mit niedriger Leistung mit einem einzelnen Teilkern (z. B. 2950A). In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2900 mehrere Grafikkerne 2980A-2980N, von denen jeder einen Satz von ersten Teilkernen 2950A-2950N und einen Satz von zweiten Teilkernen 2960A-2960N beinhaltet. In mindestens einer Ausführungsform beinhaltet jeder Teilkern in den ersten Teilkernen 2950A-2950N mindestens einen ersten Satz von Ausführungseinheiten 2952A-2952N und Medien-/Texturabtastern 2954A-2954N. In mindestens einer

Ausführungsform beinhaltet jeder Teilkern in den zweiten Teilkernen 2960A-2960N mindestens einen zweiten Satz von Ausführungseinheiten 2962A-2962N und Abtastern 2964A-2964N. In mindestens einer Ausführungsform nutzen die Teilkern 2950A-2950N, 2960A-2960N jeweils einen Satz von gemeinsam genutzten Ressourcen 2970A-2970N gemeinsam. In mindestens einer Ausführungsform beinhalten die gemeinsam genutzten Ressourcen gemeinsam genutzten Cache-Speicher und Pixeloperationslogik.

[0251] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 1615 in dem Grafikprozessor 2900 für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparemtern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0252] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0253] **Fig. 30** ist ein Blockdiagramm, das die Mikroarchitektur für einen Prozessor 3000, der Logikschaltungen zum Durchführen von Anweisungen beinhalten kann, gemäß mindestens einer Ausführungsform veranschaulicht. In mindestens einer Ausführungsform kann der Prozessor 3000 Anweisungen durchführen, einschließlich x86-Anweisungen, ARM-Anweisungen, spezialisierter Anweisungen für anwendungsspezifische integrierte Schaltungen (ASICs) usw. In mindestens einer Ausführungsform kann der Prozessor 3000 Register zum Speichern von gepackten Daten beinhalten, wie etwa 64 Bit breite MMX™-Register in Mikroprozessoren, die mit der MMX-Technologie der Intel Corporation aus Santa Clara, Kalifornien, ausgestattet sind. In mindestens einer Ausführungsform können MMX-Register, die sowohl in Integer- als auch in Gleitkommaform verfügbar sind, mit Elementen mit gepackten Daten arbeiten, die mit Single-Instruction-Multiple-Data- („SIMD“-) und Streaming-SIMD-Erweiterungs- („SSE“-) Anweisungen einhergehen. In mindestens einer Ausführungsform können 128 Bit breite XMM-Register, die sich auf die SSE2-, SSE3-, SSE4-, AVX- oder darüber hinausgehende Technologie beziehen (allgemein als „SSEx“ bezeichnet), derartige Operanden mit gepackten Daten aufbewahren. In mindestens einer Ausführungsform kann der Prozessor 3000 Anweisungen zum Beschleunigen von Algorithmen für maschinelles Lernen oder Deep Learning, Training oder Inferenzieren durchführen.

[0254] In mindestens einer Ausführungsform beinhaltet der Prozessor 3000 ein In-Order-Frontend („Frontend“) 3001 zum Abrufen von auszuführenden Anweisungen und Vorbereiten von Anweisungen, die später in der Prozessorphipeline verwendet werden sollen. In mindestens einer Ausführungsform kann das Frontend 3001 mehrere Einheiten beinhalten. In mindestens einer Ausführungsform ruft ein Anweisungsvorababrufer 3026 Anweisungen aus dem Speicher ab und führt die Anweisungen einem Anweisungsdecodierer 3028 zu, der die Anweisungen wiederum decodiert oder interpretiert. Zum Beispiel decodiert in mindestens einer Ausführungsform der Anweisungsdecodierer 3028 eine empfangene Anweisung in einen oder mehrere Operationen, die als „Mikroanweisungen“ oder „Mikrooperationen“ bezeichnet werden (auch als „Mikro-Ops“ oder „µOps“ bezeichnet) und von der Maschine ausgeführt werden können. In mindestens einer Ausführungsform parst der Anweisungsdecodierer 3028 die Anweisung in einen Opcode und entsprechende Daten- und Steuerfelder, die von der Mikroarchitektur verwendet werden können, um Operationen gemäß mindestens einer Ausführungsform durchzuführen. In mindestens einer Ausführungsform kann ein Abfolge-Cache 3030 decodierte µOps in programmgeordnete Sequenzen oder Abfolgen in einer µOp-Warteschlange 3034 zur Ausführung zusammenstellen. Wenn der Abfolge-Cache 3030 auf eine komplexe Anweisung stößt, stellt in mindestens einer Ausführungsform ein Mikrocode-ROM 3032 die für den Abschluss einer Operation benötigten µOps bereit.

[0255] In mindestens einer Ausführungsform können einige Anweisungen in eine einzelne Mikro-Op umgewandelt werden, während andere mehrere Mikro-Ops benötigen, um eine vollständige Operation abzuschließen. In mindestens einer Ausführungsform kann der Anweisungsdecodierer 3028 auf den Mikrocode-ROM 3032 zugreifen, um die Anweisung durchzuführen, wenn mehr als vier Mikro-Ops benötigt werden, um eine Anweisung abzuschließen. In mindestens einer Ausführungsform kann eine Anweisung in eine kleine Anzahl von Mikro-Ops zur Verarbeitung in dem Anweisungsdecodierer 3028 decodiert werden. In mindestens einer Ausführungsform kann eine Anweisung im Mikrocode-ROM 3032 gespeichert werden, wenn eine Reihe von Mikro-Ops zur Ausführung der Operation erforderlich ist. In mindestens einer Ausführungsform bezieht sich

der Abfolge-Cache 3030 auf ein programmierbares logisches Array (programmable logic array - „PLA“) für den Einsprungpunkt, um einen korrekten Mikroanweisungszeiger zum Lesen von Mikrocodesequenzen zu bestimmen, um eine oder mehrere Anweisungen aus dem Mikrocode-ROM 3032 gemäß mindestens einer Ausführungsform abzuschließen. In mindestens einer Ausführungsform kann das Frontend 3001 der Maschine, nachdem der Mikrocode-ROM 3032 die Sequenzierung von Mikro-Ops für eine Anweisung fertiggestellt hat, das Abrufen von Mikro-Ops aus dem Abfolge-Cache 3030 wieder aufnehmen.

[0256] In mindestens einer Ausführungsform kann eine Engine zur Ausführung in einer anderen Reihenfolge („Out-of-Order-Engine“) 3003 Anweisungen zur Ausführung vorbereiten. In mindestens einer Ausführungsform verfügt die Out-of-Order-Ausführungslogik über eine Reihe von Puffern, um den Fluss der Anweisungen zu glätten und neu zu ordnen, um die Rechenleistung zu optimieren, während sie eine Pipeline durchlaufen und zur Ausführung geplant werden. In mindestens einer Ausführungsform beinhaltet die Out-of-Order-Ausführungs-Engine 3003 ohne Einschränkung einen Zuweiser/Registerumbenenner 3040, eine Speicher- μ Op-Warteschlange 3042, eine Integer-/Gleitkomma- μ Op-Warteschlange 3044, einen Speicherplaner 3046, einen schnellen Planer 3002, einen langsamen/allgemeinen Gleitkommaplaner („langsamer/allgemeiner FP-Planer“) 3004 und einen einfachen Gleitkommaplaner („einfacher FP-Planer“) 3006. In mindestens einer Ausführungsform werden der schnelle Planer 3002, der langsame/allgemeine Gleitkommaplaner 3004 und der einfache Gleitkommaplaner 3006 in dieser Schrift auch zusammen als „ μ Op-Planer 3002, 3004, 3006“ bezeichnet. In mindestens einer Ausführungsform weist der Zuweiser/Registerumbenenner 3040 Maschinenpuffer und Ressourcen zu, die jede μ Op für die Ausführung benötigt. In mindestens einer Ausführungsform benennt der Zuweiser/Registerumbenenner 3040 logische Register in Einträge in einer Registerdatei um. In mindestens einer Ausführungsform weist der Zuweiser/Registerumbenenner 3040 auch einen Eintrag für jede μ Op in einer von zwei μ Op-Warteschlangen zu, und zwar in der Speicher- μ Op-Warteschlange 3042 für Speicheroperationen und der Integer-/Gleitkomma- μ Op-Warteschlange 3044 für Nicht-Speicheroperationen, vor dem Speicherplaner 3046 und den μ Op-Planern 3002, 3004, 3006. In mindestens einer Ausführungsform bestimmen die μ Op-Planer 3002, 3004, 3006 auf Grundlage der Bereitschaft ihrer abhängigen Eingaberegister-Operandenquellen und der Verfügbarkeit der Ausführungsressourcen, die μ Ops benötigen, um ihre Operation abzuschließen, wann eine μ Op zur Ausführung bereit ist. In mindestens einer Ausführungsform kann der schnelle Planer 3002 mindestens einer Ausführungsform auf jeder Hälfte des Haupttaktzyklus planen, während der langsame/allgemeine Gleitkomma-Planer 3004 und der einfache Gleitkomma-Planer 3006 einmal pro Hauptprozessortaktzyklus planen können. In mindestens einer Ausführungsform vermitteln die μ Op-Planer 3002, 3004, 3006 Zuteilungssports, um μ Ops zur Ausführung zu planen.

[0257] In mindestens einer Ausführungsform beinhaltet der Ausführungsblock 3011 ohne Einschränkung eine Integerregisterdatei/ein Umgehungsnetz 3008, eine Gleitkommaregisterdatei/ein Umgehungsnetz („FP-Registerdatei/Umgehungsnetz“) 3010, Adresserzeugungseinheiten (address generation units - „AGUs“) 3012 und 3014, schnelle arithmetisch-logische Einheiten (ALUs) („schnelle ALUs“) 3016 und 3018, eine langsame arithmetisch-logische Einheit („langsame ALU“) 3020, eine Gleitkomma-ALU („FP“) 3022 und eine Gleitkomma-Bewegungseinheit („FP-Bewegung“) 3024. In mindestens einer Ausführungsform werden die Integerregisterdatei/das Umgehungsnetz 3008 und die Gleitkommaregisterdatei/das Umgehungsnetz 3010 in dieser Schrift auch als „Registerdateien 3008, 3010“ bezeichnet. In mindestens einer Ausführungsform werden die AGUs 3012 und 3014, die schnellen ALUs 3016 und 3018, die langsame ALU 3020, die Gleitkomma-ALU 3022 und die Gleitkomma-Bewegungseinheit 3024 in dieser Schrift auch als „Ausführungseinheiten 3012, 3014, 3016, 3018, 3020, 3022 und 3024“ bezeichnet. In mindestens einer Ausführungsform kann der Ausführungsblock b11 ohne Einschränkung eine beliebige Anzahl (einschließlich null) und Art von Registerdateien, Umgehungsnetzen, Adresserzeugungseinheiten und Ausführungseinheiten in beliebiger Kombination beinhalten.

[0258] In mindestens einer Ausführungsform können die Registerdateien 3008, 3010 zwischen den μ Op-Planern 3002, 3004, 3006 und den Ausführungseinheiten 3012, 3014, 3016, 3018, 3020, 3022 und 3024 angeordnet sein. In mindestens einer Ausführungsform führt die Integerregisterdatei/das Umgehungsnetz 3008 Integeroperationen durch. In mindestens einer Ausführungsform führt die Gleitkommaregisterdatei/das Umgehungsnetz 3010 Gleitkommaoperationen durch. In mindestens einer Ausführungsform kann jede der Registerdateien 3008, 3010 ohne Einschränkung ein Umgehungsnetz beinhalten, das gerade abgeschlossene Ergebnisse, die noch nicht in die Registerdatei geschrieben wurden, zu neuen abhängigen μ Ops umgehen oder weiterleiten kann. In mindestens einer Ausführungsform können die Registerdateien 3008, 3010 miteinander Daten kommunizieren. In mindestens einer Ausführungsform kann die Integerregisterdatei/das Umgehungsnetz 3008 ohne Einschränkung zwei separate Registerdateien beinhalten, eine Registerdatei für Daten niedriger Ordnung mit zweiunddreißig Bit und eine zweite Registerdatei für Daten hoher Ordnung mit zweiunddreißig Bit. In mindestens einer Ausführungsform kann die Gleitkommaregisterdatei/das Umge-

hungsnetz 3010 ohne Einschränkung 128 Bit breite Einträge beinhalten, da Gleitkommaanweisungen typischerweise Operanden mit einer Breite von 64 bis 128 Bit aufweisen.

[0259] In mindestens einer Ausführungsform können die Ausführungseinheiten 3012, 3014, 3016, 3018, 3020, 3022, 3024 Anweisungen ausführen. In mindestens einer Ausführungsform speichern die Registerdateien 3008, 3010 Integer- und Gleitkommadaten-Operandenwerte, die Mikroanweisungen für die Ausführung benötigen. In mindestens einer Ausführungsform kann der Prozessor 3000 ohne Einschränkung eine beliebige Anzahl und Kombination der Ausführungseinheiten 3012, 3014, 3016, 3018, 3020, 3022, 3024 beinhalten. In mindestens einer Ausführungsform können die Gleitkomma-ALU 3022 und die Gleitkomma-Bewegungseinheit 3024 Gleitkomma-, MMX-, SIMD-, AVX- und SSE- oder andere Operationen ausführen, einschließlich spezialisierter Anweisungen für maschinelles Lernen. In mindestens einer Ausführungsform kann die Gleitkomma-ALU 3022 ohne Einschränkung einen 64 Bit mal 64 Bit großen Gleitkommadividierer zum Ausführen von Divisions-, Quadratwurzel- und Rest-Mikro-Ops beinhalten. In mindestens einer Ausführungsform können Anweisungen, an denen ein Gleitkommawert beteiligt ist, mit Gleitkomma-Hardware gehandhabt werden. In mindestens einer Ausführungsform können ALU-Operationen an die schnellen ALUs 3016, 3018 übergeben werden. In mindestens einer Ausführungsform können die schnellen ALUs 3016, 3018 schnelle Operationen mit einer effektiven Latenz von einem halben Taktzyklus ausführen. In mindestens einer Ausführungsform gehen die meisten komplexen Integeroperationen an die langsame ALU 3020, da die langsame ALU 3020 ohne Einschränkung Integerausführungs-Hardware für Operationen vom Typ mit langer Latenz beinhalten kann, wie etwa einen Multiplikator, Verschiebungen, Flag-Logik und Verzweungsverarbeitung. In mindestens einer Ausführungsform können die Lade-/Speicheroperationen eines Speichers durch die AGUs 3012, 3014 ausgeführt werden. In mindestens einer Ausführungsform können die schnelle ALU 3016, die schnelle ALU 3018 und die langsame ALU 3020 Integeroperationen an 64-Bit-Datenoperanden durchführen. In mindestens einer Ausführungsform können die schnelle ALU 3016, die schnelle ALU 3018 und die langsame ALU 3020 dazu implementiert sein, eine Vielfalt von Datenbitgrößen zu unterstützen, einschließlich sechzehn, zweiunddreißig, 128, 256 usw. In mindestens einer Ausführungsform können die Gleitkomma-ALU 3022 und die Gleitkomma-Bewegungseinheit 3024 dazu implementiert sein, einen Bereich von Operanden zu unterstützen, die Bits mit verschiedenen Breiten aufweisen. In mindestens einer Ausführungsform können die Gleitkomma-ALU 3022 und die Gleitkomma-Bewegungseinheit 3024 an 128 Bit breiten Paket-Datenoperanden in Verbindung mit SIMD- und Multimedia-Anweisungen arbeiten.

[0260] In mindestens einer Ausführungsform teilen die μ Op-Planer 3002, 3004, 3006 abhängige Operationen zu, bevor die Ausführung einer übergeordneten Last beendet ist. In mindestens einer Ausführungsform kann der Prozessor 3000, da μ Ops in dem Prozessor 3000 spekulativ geplant und ausgeführt werden können, auch Logik zum Handhaben von Speicherfehlern beinhalten. In mindestens einer Ausführungsform können, falls eine Datenlast in einem Daten-Cache einen Fehler verursacht, abhängige Operationen in einer Pipeline im Gange sein, die einen Planer mit temporär inkorrekten Daten verlassen haben. In mindestens einer Ausführungsform verfolgt ein Wiederholungsmechanismus Anweisungen, die falsche Daten verwenden, und führt sie erneut aus. In mindestens einer Ausführungsform könnte es sein, dass abhängige Operationen wiederholt werden müssen, und es kann unabhängigen Operationen ermöglicht werden, abgeschlossen zu werden. In mindestens einer Ausführungsform können die Planer und ein Wiederholungsmechanismus mindestens einer Ausführungsform eines Prozessors auch dafür ausgestaltet sein, Anweisungssequenzen für Zeichenfolgenvergleichsoperationen abzufangen.

[0261] In mindestens einer Ausführungsform kann sich der Ausdruck „Register“ auf prozessorinterne Speicherorte beziehen, die als Teil von Anweisungen zur Identifizierung von Operanden verwendet werden können. In mindestens einer Ausführungsform kann es sich bei den Registern um diejenigen handeln, die von außerhalb des Prozessors (aus der Sicht eines Programmierers) nutzbar sind. In mindestens einer Ausführungsform sind die Register möglicherweise nicht auf einen konkreten Schaltungstyp beschränkt. Vielmehr kann ein Register in mindestens einer Ausführungsform Daten speichern, Daten bereitstellen und die in dieser Schrift beschriebenen Funktionen durchführen. In mindestens einer Ausführungsform können die in dieser Schrift beschriebenen Register durch Schaltungen innerhalb eines Prozessors unter Verwendung einer beliebigen Anzahl unterschiedlicher Techniken implementiert werden, wie etwa dedizierter physischer Register, dynamisch zugewiesener physischer Register unter Verwendung von Registerumbenennung, Kombinationen aus dedizierten und dynamisch zugewiesenen physischen Registern usw. In mindestens einer Ausführungsform speichern Integerregister 32-Bit-Integerdaten. Eine Registerdatei aus mindestens einer Ausführungsform beinhaltet zudem acht Multimedia-SIMD-Register für gepackte Daten.

[0262] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Infe-

renz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform können Abschnitte der oder die gesamte Inferenz- und/oder Trainingslogik 1615 in den Ausführungsblock 3011 und andere gezeigte oder nicht gezeigte Speicher oder Register integriert sein. Zum Beispiel können in mindestens einer Ausführungsform die in dieser Schrift beschriebenen Trainings- und/oder Inferenztechniken eine oder mehrere der in dem Ausführungsblock 3011 veranschaulichten ALUs verwenden. Darüber hinaus können Gewichtungsparmeter im chipinternem oder chipexternem Speicher und/oder in Registern (gezeigt oder nicht gezeigt) gespeichert werden, die ALUs des Ausführungsblocks 3011 konfigurieren, um eine(n) oder mehrere in dieser Schrift beschriebene Algorithmen des maschinellen Lernens, neuronale Netzarchitekturen, Anwendungsfälle oder Trainingstechniken durchzuführen.

[0263] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0264] **Fig. 31** veranschaulicht einen Deep-Learning-Anwendungsprozessor 3100 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform verwendet der Deep-Learning-Anwendungsprozessor 3100 Anweisungen, die bei Ausführung durch den Deep-Learning-Anwendungsprozessor 3100 den Deep-Learning-Anwendungsprozessor 3100 dazu veranlassen, einige oder alle der in dieser gesamten Offenbarung beschriebenen Prozesse und Techniken durchzuführen. In mindestens einer Ausführungsform ist der Deep-Learning-Anwendungsprozessor 3100 eine anwendungsspezifische integrierte Schaltung (ASIC). In mindestens einer Ausführungsform führt der Anwendungsprozessor 3100 Matrixmultiplikationsoperationen entweder „fest verdrahtet“ in Hardware als Ergebnis des Durchführens einer oder mehrerer Anweisungen oder beides durch. In mindestens einer Ausführungsform beinhaltet der Deep-Learning-Anwendungsprozessor 3100 ohne Einschränkung Verarbeitungscluster 3110(1)-3110(12), chipübergreifende Verbindungen (Inter-Chip Links - „ICLs“) 3120(1)-3120(12), chipübergreifende Steuerungen (Inter-Chip Controllers - „ICCs“) 3130(1)-3130(2), Speichersteuerungen (memory controllers - „Mem Ctrlrs“) 3142(1)-3142(4), eine physische Schicht mit Speicher mit hoher Bandbreite (high bandwidth memory physical layer - „HBM PHY“) 3144(1)-3144(4), eine zentrale Verwaltungssteuerungs-Verarbeitungseinheit („Verwaltungssteuerungs-CPU“) 3150, eine serielle Peripherieschnittstelle, eine zwischenintegrierte Schaltung und einen Universal-Eingabe/Ausgabe-Block („SPI, I2C, GPIO“), eine Interconnect-Express-Steuerung für Peripheriekomponenten und einen Direktspeicherzugriffsblock („PCIe-Steuerung und DMA“) 3170 und einen sechzehnspurigen Interconnect-Express-Port für Peripheriekomponenten („PCI-Express x 16“) 3180.

[0265] In mindestens einer Ausführungsform können die Verarbeitungscluster 3110 Deep-Learning-Operationen durchführen, einschließlich Inferenz- oder Vorhersageoperationen auf Grundlage von Gewichtungsparmetern, die mit einer oder mehreren Trainingstechniken, einschließlich der in dieser Schrift beschriebenen, berechnet wurden. In mindestens einer Ausführungsform kann jedes Verarbeitungscluster 3110 ohne Einschränkung eine beliebige Anzahl und einen beliebigen Typ von Prozessoren beinhalten. In mindestens einer Ausführungsform kann der Deep-Learning-Anwendungsprozessor 3100 eine beliebige Anzahl und eine beliebige Art von Verarbeitungsclustern 3110 beinhalten. In mindestens einer Ausführungsform sind die chipübergreifenden Verbindungen 3120 bidirektional. In mindestens einer Ausführungsform ermöglichen die chipübergreifenden Verbindungen 3120 und die chipübergreifenden Steuerungen 3130 mehreren Deep-Learning-Anwendungsprozessoren 3100 den Austausch von Informationen, einschließlich Aktivierungsinformationen, die aus dem Durchführen eines oder mehrerer Algorithmen des maschinellen Lernens resultieren, die in einem oder mehreren neuronalen Netzen verkörpert sind. In mindestens einer Ausführungsform kann der Deep-Learning-Anwendungsprozessor 3100 eine beliebige Anzahl (einschließlich null) und eine beliebige Art von ICLs 3120 und ICCs 3130 beinhalten.

[0266] In mindestens einer Ausführungsform stellen die HBM2s 3140 insgesamt 32 Gigabyte (GB) Speicher bereit. Die HBM2 3140(i) ist sowohl der Speichersteuerung 3142(i) als auch der HBM PHY 3144(i) zugeordnet. In mindestens einer Ausführungsform kann eine beliebige Anzahl von HBM2s 3140 eine beliebige Art und eine beliebige Gesamtmenge an Speicher mit hoher Bandbreite bereitstellen und einer beliebigen Anzahl (einschließlich null) und einer beliebigen Art von Speichersteuerungen 3142 und HBM PHYs 3144 zugeordnet sein. In mindestens einer Ausführungsform können die SPI, die I2C, der GPIO 3160, die PCIe-Steuerung und der DMA 3170 und/oder PCIe 3180 durch eine beliebige Anzahl und Art von Blöcken ersetzt werden, die eine beliebige Anzahl und Art von Kommunikationsstandards auf eine beliebige technisch durchführbare Weise ermöglichen.

[0267] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform wird der Deep-Learning-Anwendungsprozessor 3100 verwendet, um ein Modell für maschinelles Lernen, wie etwa ein neuronales Netz, zu trainieren, um die dem Deep-Learning-Anwendungsprozessor 3100 bereitgestellten Informationen vorherzusagen oder abzuleiten. In mindestens einer Ausführungsform wird der Deep-Learning-Anwendungsprozessor 3100 verwendet, um Informationen auf Grundlage eines trainierten Modells für maschinelles Lernen (z. B. eines neuronalen Netzes) abzuleiten oder vorherzusagen, das durch einen anderen Prozessor oder ein anderes System oder durch den Deep-Learning-Anwendungsprozessor 3100 trainiert wurde. In mindestens einer Ausführungsform kann der Prozessor 3100 verwendet werden, um einen oder mehrere der in dieser Schrift beschriebenen Anwendungsfälle für neuronale Netze durchzuführen.

[0268] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0269] **Fig. 32** ist ein Blockdiagramm eines neuromorphen Prozessors 3200 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann der neuromorphe Prozessor 3200 eine oder mehrere Eingaben von Quellen, die extern zu dem neuromorphen Prozessor 3200 sind, empfangen. In mindestens einer Ausführungsform können diese Eingaben an ein oder mehrere Neuronen 3202 innerhalb des neuromorphen Prozessors 3200 übertragen werden. In mindestens einer Ausführungsform können die Neuronen 3202 und Komponenten davon unter Verwendung von Schaltungen oder Logik implementiert sein, einschließlich einer oder mehrerer arithmetisch-logischer Einheiten (ALUs). In mindestens einer Ausführungsform kann der neuromorphe Prozessor 3200 ohne Einschränkung Tausende oder Millionen von Instanzen von Neuronen 3202 beinhalten, aber es kann eine beliebige geeignete Anzahl von Neuronen 3202 verwendet werden. In mindestens einer Ausführungsform kann jede Instanz des Neurons 3202 einen Neuroneneingang 3204 und einen Neuronenausgang 3206 beinhalten. In mindestens einer Ausführungsform können die Neuronen 3202 Ausgaben erzeugen, die an Eingänge anderer Instanzen von Neuronen 3202 übertragen werden können. Zum Beispiel können in mindestens einer Ausführungsform die Neuroneneingänge 3204 und die Neuronenausgänge 3206 über Synapsen 3208 zusammengeschaltet sein.

[0270] In mindestens einer Ausführungsform können die Neuronen 3202 und die Synapsen 3208 derart zusammengeschaltet sein, dass der neuromorphe Prozessor 3200 arbeitet, um die durch den neuromorphen Prozessor 3200 empfangenen Informationen zu verarbeiten oder zu analysieren. In mindestens einer Ausführungsform können die Neuronen 3202 einen Ausgabeimpuls (oder „Fire“ oder „Spike“) übertragen, wenn die über den Neuroneneingang 3204 empfangenen Eingaben einen Schwellenwert überschreiten. In mindestens einer Ausführungsform können die Neuronen 3202 die an den Neuroneneingängen 3204 empfangenen Signale summieren oder integrieren. Zum Beispiel können in mindestens einer Ausführungsform die Neuronen 3202 als leckende Integrate-and-Fire-Neuronen implementiert sein, wobei, falls eine Summe (als „Membranpotential“ bezeichnet) einen Schwellenwert überschreitet, das Neuron 3202 eine Ausgabe (oder „Feuern“) unter Verwendung einer Übertragungsfunktion, wie etwa einer Sigmoid- oder Schwellenwertfunktion, erzeugen kann. In mindestens einer Ausführungsform kann ein leckendes Integrate-and-Fire-Neuron Signale, die an den Neuroneneingängen 3204 empfangen werden, zu einem Membranpotential summieren und auch einen Abklingfaktor (oder ein Leck) anwenden, um ein Membranpotential zu reduzieren. In mindestens einer Ausführungsform kann ein leckendes Integrate-and-Fire-Neuron feuern, falls mehrere Eingabesignale an den Neuroneneingängen 3204 schnell genug empfangen werden, um einen Schwellenwert zu überschreiten (d. h., bevor ein Membranpotential zu weit nach unten abklingt, um zu feuern). In mindestens einer Ausführungsform können die Neuronen 3202 unter Verwendung von Schaltungen oder Logik implementiert sein, die Eingaben empfangen, Eingaben in ein Membranpotential integrieren und ein Membranpotential abklingen lassen. In mindestens einer Ausführungsform können Eingaben gemittelt werden oder kann eine beliebige andere geeignete Übertragungsfunktion verwendet werden. Darüber hinaus können die Neuronen 3202 in mindestens einer Ausführungsform ohne Einschränkung Komparatorschaltungen oder -logik beinhalten, die einen Ausgabe-Spike an dem Neuronenausgang 3206 erzeugen, wenn das Ergebnis des Anwendens einer Übertragungsfunktion auf den Neuroneneingang 3204 einen Schwellenwert überschreitet. In mindestens einer Ausführungsform kann das Neuron 3202, sobald es feuert, zuvor empfangene Eingabeinformationen ignorieren, indem es zum Beispiel ein Membranpotential auf 0 oder einen anderen geeigneten Standardwert zurücksetzt. In mindestens einer Ausführungsform kann das Neuron 3202, sobald das Membranpotential auf 0 zurückgesetzt wurde, nach einer geeigneten Zeitspanne (oder Refraktärzeit) den normalen Betrieb wieder aufnehmen.

[0271] In mindestens einer Ausführungsform können die Neuronen 3202 durch die Synapsen 3208 zusammengeschaltet sein. In mindestens einer Ausführungsform können die Synapsen 3208 arbeiten, um Signale von einer Ausgabe eines ersten Neurons 3202 an eine Eingabe eines zweiten Neurons 3202 zu übertragen. In mindestens einer Ausführungsform können die Neuronen 3202 Informationen über mehr als eine Instanz der Synapse 3208 übertragen. In mindestens einer Ausführungsform können eine oder mehrere Instanzen des Neuronenausgangs 3206 über eine Instanz der Synapse 3208 mit einer Instanz des Neuroneneingangs 3204 in dem gleichen Neuron 3202 verbunden sein. In mindestens einer Ausführungsform kann eine Instanz des Neurons 3202, die eine über eine Instanz der Synapse 3208 zu übertragende Ausgabe erzeugt, als „präsynaptisches Neuron“ in Bezug auf diese Instanz der Synapse 3208 bezeichnet werden. In mindestens einer Ausführungsform kann eine Instanz des Neurons 3202, die eine über eine Instanz der Synapse 3208 übertragene Eingabe empfängt, als „postsynaptisches Neuron“ in Bezug auf diese Instanz der Synapse 3208 bezeichnet werden. Da eine Instanz des Neurons 3202 Eingaben von einer oder mehreren Instanzen der Synapse 3208 empfangen kann und auch Ausgaben über eine oder mehrere Instanzen der Synapse 3208 übertragen kann, kann in mindestens einer Ausführungsform eine einzelne Instanz des Neurons 3202 daher sowohl ein „präsynaptisches Neuron“ als auch ein „postsynaptisches Neuron“ in Bezug auf verschiedene Instanzen der Synapsen 3208 sein.

[0272] In mindestens einer Ausführungsform können die Neuronen 3202 in eine oder mehrere Schichten organisiert sein. Jede Instanz des Neurons 3202 kann einen Neuronenausgang 3206 aufweisen, der sich über eine oder mehrere Synapsen 3208 zu einem oder mehreren Neuroneneingängen 3204 auffächern kann. In mindestens einer Ausführungsform können die Neuronenausgänge 3206 der Neuronen 3202 in einer ersten Schicht 3210 mit den Neuroneneingängen 3204 der Neuronen 3202 in einer zweiten Schicht 3212 verbunden sein. In mindestens einer Ausführungsform kann die Schicht 3210 als „vorwärtsgekoppelte Schicht“ bezeichnet werden. In mindestens einer Ausführungsform kann sich jede Instanz des Neurons 3202 in einer Instanz der ersten Schicht 3210 zu jeder Instanz des Neurons 3202 in der zweiten Schicht 3212 auffächern. In mindestens einer Ausführungsform kann die erste Schicht 3210 als „vollständig verbundene vorwärtsgekoppelte Schicht“ bezeichnet werden. In mindestens einer Ausführungsform kann sich jede Instanz des Neurons 3202 in einer Instanz der zweiten Schicht 3212 zu weniger als allen Instanzen des Neurons 3202 in einer dritten Schicht 3214 auffächern. In mindestens einer Ausführungsform kann die zweite Schicht 3212 als „spärlich verbundene vorwärtsgekoppelte Schicht“ bezeichnet werden. In mindestens einer Ausführungsform können sich Neuronen 3202 in der zweiten Schicht 3212 zu Neuronen 3202 in mehreren anderen Schichten auffächern, einschließlich zu Neuronen 3202 in (derselben) zweiten Schicht 3212. In mindestens einer Ausführungsform kann die zweite Schicht 3212 als „rekurrente Schicht“ bezeichnet werden. In mindestens einer Ausführungsform kann der neuromorphe Prozessor 3200 ohne Einschränkung eine beliebige geeignete Kombination von rekurrenten Schichten und vorwärtsgekoppelten Schichten beinhalten, einschließlich ohne Einschränkung sowohl spärlich verbundener vorwärtsgekoppelter Schichten als auch vollständig verbundener vorwärtsgekoppelter Schichten.

[0273] In mindestens einer Ausführungsform kann der neuromorphe Prozessor 3200 ohne Einschränkung eine rekonfigurierbare Zusammenschaltungsarchitektur oder dedizierte fest verdrahtete Zusammenschaltungen beinhalten, um die Synapse 3208 mit den Neuronen 3202 zu verbinden. In mindestens einer Ausführungsform kann der neuromorphe Prozessor 3200 ohne Einschränkung Schaltungen oder Logik beinhalten, die es ermöglicht, Synapsen je nach Bedarf auf Grundlage der Topologie des neuronalen Netzes und des Neuronen-Fan-In/-Out unterschiedlichen Neuronen 3202 zuzuweisen. Zum Beispiel können in mindestens einer Ausführungsform die Synapsen 3208 mit den Neuronen 3202 unter Verwendung einer Zusammenschaltungsstruktur, wie etwa eines Netzes auf einem Chip, oder mit dedizierten Verbindungen verbunden sein. In mindestens einer Ausführungsform können die Synapsenzusammenschaltungen und Komponenten davon unter Verwendung von Schaltungen oder Logik implementiert sein.

[0274] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0275] **Fig. 33** ist ein Blockdiagramm eines Verarbeitungssystems gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform beinhaltet das System 3300 einen oder mehrere Prozessoren 3302 und einen oder mehrere Grafikprozessoren 3308 und kann ein Einzelprozessor-Desktopsystem, ein Multiprozessor-Arbeitsstationssystem oder ein Serversystem sein, das eine große Anzahl von Prozessoren 3302 oder Prozessorkernen 3307 aufweist. In mindestens einer Ausführungsform ist das System 3300 eine Verarbeitungsplattform, die in eine integrierte Schaltung als System auf einem Chip (SoC) zur Verwendung in mobilen, tragbaren oder eingebetteten Vorrichtungen integriert ist.

[0276] In mindestens einer Ausführungsform kann das System 3300 eine serverbasierte Spieleplattform, eine Spielekonsole, einschließlich einer Spiele- und Medienkonsole, eine mobile Spielekonsole, eine tragbare Spielekonsole oder eine Online-Spielekonsole beinhalten oder in diese integriert sein. In mindestens einer Ausführungsform ist das System 3300 ein Mobiltelefon, ein Smartphone, eine Tablet-Rechenvorrichtung oder eine mobile Internet-Vorrichtung. In mindestens einer Ausführungsform kann das Verarbeitungssystem 3300 auch eine tragbare Vorrichtung beinhalten, mit dieser gekoppelt oder in diese integriert sein, wie z. B. eine tragbare Smartwatch-Vorrichtung, eine Smart-Eyewear-Vorrichtung, eine Augmented-Reality-Vorrichtung oder eine Virtual-Reality-Vorrichtung. In mindestens einer Ausführungsform ist das Verarbeitungssystem 3300 eine Fernseh- oder Set-Top-Box-Vorrichtung, die einen oder mehrere Prozessoren 3302 und eine grafische Schnittstelle aufweist, die durch einen oder mehrere Grafikprozessoren 3308 erzeugt wird.

[0277] In mindestens einer Ausführungsform beinhalten ein oder mehrere Prozessoren 3302 jeweils einen oder mehrere Prozessorkerne 3307 zum Verarbeiten von Anweisungen, die bei ihrer Ausführung Operationen für System- und Benutzer-Software durchführen. In mindestens einer Ausführungsform ist jeder von einem oder mehreren Prozessorkernen 3307 dazu konfiguriert, einen spezifischen Anweisungssatz 3309 zu verarbeiten. In mindestens einer Ausführungsform kann der Anweisungssatz 3309 Complex Instruction Set Computing („CISC“), Reduced Instruction Set Computing („RISC“) oder Rechnen über Very Long Instruction Word („VLIW“) ermöglichen. In mindestens einer Ausführungsform können die Prozessorkerne 3307 jeweils einen anderen Anweisungssatz 3309 verarbeiten, der Anweisungen enthalten kann, um die Emulation anderer Anweisungssätze zu erleichtern. In mindestens einer Ausführungsform kann der Prozessorkern 3307 auch andere Verarbeitungsvorrichtungen beinhalten, wie etwa einen digitalen Signalprozessor (DSP).

[0278] In mindestens einer Ausführungsform beinhaltet der Prozessor 3302 einen Cache-Speicher 3304. In mindestens einer Ausführungsform kann der Prozessor 3302 einen einzelnen internen Cache oder mehrere Ebenen von internem Cache aufweisen. In mindestens einer Ausführungsform wird der Cache-Speicher von verschiedenen Komponenten des Prozessors 3302 gemeinsam genutzt. In mindestens einer Ausführungsform verwendet der Prozessor 3302 außerdem einen externen Cache (z. B. einen Cache der Ebene 3 („L3“) oder einen Cache der letzten Ebene (Last Level Cache - „LLC“)) (nicht gezeigt), der von den Prozessorkernen 3307 unter Verwendung bekannter Cache-Kohärenztechniken gemeinsam genutzt werden kann. In mindestens einer Ausführungsform ist in dem Prozessor 3302 zusätzlich eine Registerdatei 3306 enthalten, die unterschiedliche Arten von Registern zum Speichern unterschiedlicher Datenarten beinhalten kann (z. B. Ganzzahlregister, Gleitkommaregister, Statusregister und ein Anweisungszeigerregister). In mindestens einer Ausführungsform kann die Registerdatei 3306 Universalregister oder andere Register beinhalten.

[0279] In mindestens einer Ausführungsform sind ein oder mehrere Prozessor(en) 3302 mit einem oder mehreren Schnittstellenbus(sen) 3310 gekoppelt, um Kommunikationssignale, wie etwa Adress-, Daten- oder Steuersignale, zwischen dem Prozessor 3302 und anderen Komponenten in dem System 3300 zu übertragen. In mindestens einer Ausführungsform kann der Schnittstellenbus 3310 in einer Ausführungsform ein Prozessorbus sein, wie etwa eine Version eines Direct-Media-Interface-(„DMI“-)Busses. In mindestens einer Ausführungsform ist die Schnittstelle 3310 nicht auf einen DMI-Bus beschränkt, sondern kann einen oder mehrere Peripheral-Component-Interconnect-Busse (z. B. PCI, PCI Express), Speicherbusse oder andere Arten von Schnittstellenbussen beinhalten. In mindestens einer Ausführungsform beinhalten der/die Prozessor(en) 3302 eine integrierte Speichersteuerung 3316 und einen Plattformsteuerungs-Hub 3330. In mindestens einer Ausführungsform ermöglicht die Speichersteuerung 3316 die Kommunikation zwischen einer Speichervorrichtung und anderen Komponenten des Systems 3300, während der Plattformsteuerungs-Hub (platform controller hub - PCH) 3330 Verbindungen mit E/A-Vorrichtungen über einen lokalen E/A-Bus bereitstellt.

[0280] In mindestens einer Ausführungsform kann eine Speichervorrichtung 3320 eine Vorrichtung mit dynamischem Direktzugriffsspeicher (DRAM), eine Vorrichtung mit statischem Direktzugriffsspeicher (SRAM), eine Flash-Speichervorrichtung, eine Phasenwechsel-Speichervorrichtung oder eine andere Speichervorrichtung sein, die eine geeignete Rechenleistung aufweist, um als Prozessspeicher zu dienen. In mindestens einer Ausführungsform kann die Speichervorrichtung 3320 als Systemspeicher für das System 3300 arbeiten, um Daten 3322 und Anweisungen 3321 zur Verwendung zu speichern, wenn ein oder mehrere Prozessoren 3302 eine Anwendung oder einen Prozess ausführen. In mindestens einer Ausführungsform ist die Speichersteuerung 3316 zudem an einen optionalen externen Grafikprozessor 3312 gekoppelt, der mit einem oder mehreren Grafikprozessoren 3308 in den Prozessoren 3302 kommunizieren kann, um Grafik- und Medienoperationen durchzuführen. In mindestens einer Ausführungsform kann eine Anzeigevorrichtung 3311 mit dem/den Prozessor(en) 3302 verbunden sein. In mindestens einer Ausführungsform kann die Anzeigevorrichtung 3311 eine oder mehrere von einer internen Anzeigevorrichtung, wie in einer mobilen elektronischen

Vorrichtung oder einer Laptop-Vorrichtung, oder einer externen Anzeigevorrichtung beinhalten, die über eine Anzeigeschnittstelle (z. B. DisplayPort usw.) angeschlossen ist. In mindestens einer Ausführungsform kann die Anzeigevorrichtung 3311 eine am Kopf befestigte Anzeige (head mounted display - „HMD“) beinhalten, wie etwa eine stereoskopische Anzeigevorrichtung zur Verwendung bei Virtual-Reality-(„VR“-)Anwendungen oder Augmented-Reality-(„AR“-)Anwendungen.

[0281] In mindestens einer Ausführungsform ermöglicht der Plattformsteuerungs-Hub 3330, dass Peripheriegeräte mit der Speichervorrichtung 3320 und dem Prozessor 3302 über einen Hochgeschwindigkeits-E/A-Bus verbunden werden. In mindestens einer Ausführungsform beinhalten die E/A-Peripheriegeräte eine Audiosteuerung 3346, eine Netzwerksteuerung 3334, eine Firmware-Schnittstelle 3328, einen drahtlosen Transceiver 3326, Berührungssensoren 3325 und eine Datenspeichervorrichtung 3324 (z. B. Festplattenlaufwerk, Flash-Speicher usw.). In mindestens einer Ausführungsform kann die Datenspeichervorrichtung 3324 über eine Speicherschnittstelle (z. B. SATA) oder über einen Peripheriebus, wie etwa einen Peripheral-Component-Interconnect-Bus (z. B. PCI, PCI Express), verbunden sein. In mindestens einer Ausführungsform können die Berührungssensoren 3325 Touchscreen-Sensoren, Drucksensoren oder Fingerabdrucksensoren beinhalten. In mindestens einer Ausführungsform kann der drahtlose Transceiver 3326 ein Wi-Fi-Transceiver, ein Bluetooth-Transceiver oder ein Mobilfunknetz-Transceiver, wie etwa ein 3G-, 4G- oder Long-Term-Evolution-(„LTE“-)Transceiver, sein. In mindestens einer Ausführungsform ermöglicht die Firmware-Schnittstelle 3328 die Kommunikation mit der System-Firmware und kann zum Beispiel eine Unified Extensible Firmware Interface (UEFI) sein. In mindestens einer Ausführungsform kann die Netzwerksteuerung 3334 eine Netzwerkverbindung zu einem drahtgebundenen Netzwerk ermöglichen. In mindestens einer Ausführungsform ist eine Hochleistungsnetzwerksteuerung (nicht gezeigt) mit dem Schnittstellenbus 3310 gekoppelt. In mindestens einer Ausführungsform ist die Audiosteuerung 3346 eine Mehrkanal-High-Definition-Audiosteuerung. In mindestens einer Ausführungsform beinhaltet das System 3300 eine optionale Legacy-E/A-Steuerung 3340 zur Kopplung von Legacy-Vorrichtungen (z. B. Personal System 2 (PS/2)) mit dem System. In mindestens einer Ausführungsform kann der Plattformsteuerungs-Hub 3330 auch mit einer oder mehreren Universal-Serial-Bus-(„USB“-)Steuerungen 3342 verbunden sein, die mit Eingabevorrichtungen, wie etwa Kombinationen aus Tastatur und Maus 3343, einer Kamera 3344 oder anderen USB-Eingabevorrichtungen, verbunden sind.

[0282] In mindestens einer Ausführungsform kann eine Instanz der Speichersteuerung 3316 und des Plattformsteuerungs-Hubs 3330 in einen diskreten externen Grafikprozessor, wie etwa den externen Grafikprozessor 3312, integriert sein. In mindestens einer Ausführungsform können der Plattformsteuerungs-Hub 3330 und/oder die Speichersteuerung 3316 extern zu einem oder mehreren Prozessor(en) 3302 sein. Zum Beispiel kann das System 3300 in mindestens einer Ausführungsform eine externe Speichersteuerung 3316 und einen Plattformsteuerungs-Hub 3330 beinhalten, der als Speichersteuerungs-Hub und Peripheriesteuerungs-Hub innerhalb eines Systemchipsatzes konfiguriert sein kann, der mit den Prozessor(en) 3302 in Kommunikation steht.

[0283] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform können Abschnitte oder die gesamte Inferenz- und/oder Trainingslogik 1615 in den Grafikprozessor 3300 integriert sein. Zum Beispiel können in mindestens einer Ausführungsform die in dieser Schrift beschriebenen Trainings- und/oder Inferenztechniken eine oder mehrere der in dem Grafikprozessor 3312 verkörpert ALUs verwenden. Darüber hinaus können in mindestens einer Ausführungsform die in dieser Schrift beschriebenen Inferenz- und/oder Trainingsoperationen unter Verwendung von anderer Logik als der in den **Fig. 16A** oder **Fig. 16B** veranschaulichten Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtungsparmeter in chipinternem oder chipexternem Speicher und/oder in Registern (gezeigt oder nicht gezeigt) gespeichert werden, die ALUs des Grafikprozessors 3300 konfigurieren, um eine(n) oder mehrere in dieser Schrift beschriebene Algorithmen für maschinelles Lernen, neuronale Netzarchitekturen, Anwendungsfälle oder Trainingstechniken durchzuführen.

[0284] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0285] **Fig. 34** ist ein Blockdiagramm eines Prozessors 3400, der einen oder mehrere Prozessorkerne 3402A-3402N, eine integrierte Speichersteuerung 3414 und einen integrierten Grafikprozessor 3408 auf-

weist, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann der Prozessor 3400 zusätzliche Kerne bis einschließlich des zusätzlichen Kerns 3402N beinhalten, die durch Kästen mit gestrichelten Linien dargestellt sind. In mindestens einer Ausführungsform beinhaltet jeder der Prozessorkerne 3402A-3402N eine oder mehrere interne Cache-Einheiten 3404A-3404N. In mindestens einer Ausführungsform hat jeder Prozessorkern auch Zugriff auf eine oder mehrere gemeinsam genutzte Cache-Einheiten 3406.

[0286] In mindestens einer Ausführungsform stellen die internen Cache-Einheiten 3404A-3404N und die gemeinsam genutzten Cache-Einheiten 3406 eine Cache-Speicherhierarchie innerhalb des Prozessors 3400 dar. In mindestens einer Ausführungsform können die Cache-Speichereinheiten 3404A-3404N mindestens eine Ebene von Anweisungs- und Daten-Cache innerhalb jedes Prozessorkerns und eine oder mehrere Ebenen von gemeinsam genutztem Cache der mittleren Ebene, wie etwa einen Cache der Ebene 2 (L2), Ebene 3 (L3), Ebene 4 (L4) oder anderer Ebenen, beinhalten, wobei ein höchstes Cache-Ebene vor dem externen Speicher als LLC klassifiziert ist. In mindestens einer Ausführungsform hält die Cache-Kohärenzlogik die Kohärenz zwischen verschiedenen Cache-Einheiten 3406 und 3404A-3404N aufrecht.

[0287] In mindestens einer Ausführungsform kann der Prozessor 3400 auch einen Satz von einer oder mehreren Bussteuerungseinheiten 3416 und einen Systemagentenkern 3410 beinhalten. In mindestens einer Ausführungsform verwalten eine oder mehrere Bussteuereinheiten 3416 einen Satz von peripheren Bussen, wie zum Beispiel einen oder mehrere PCI- oder PCI-Express-Busse. In mindestens einer Ausführungsform stellt der Systemagentenkern 3410 eine Verwaltungsfunktionalität für verschiedene Prozessorkomponenten bereit. In mindestens einer Ausführungsform beinhaltet der Systemagentenkern 3410 eine oder mehrere integrierte Speichersteuerungen 3414, um den Zugriff auf verschiedene externe Speichervorrichtungen (nicht gezeigt) zu verwalten.

[0288] In mindestens einer Ausführungsform beinhalten einer oder mehrere der Prozessorkerne 3402A-3402N Unterstützung für simultanes Multi-Threading. In mindestens einer Ausführungsform beinhaltet der Systemagentenkern 3410 Komponenten zum Koordinieren und Betreiben der Kerne 3402A-3402N während der Multi-Thread-Verarbeitung. In mindestens einer Ausführungsform kann der Systemagentenkern 3410 zusätzlich eine Leistungssteuereinheit (power control unit - PCU) beinhalten, die Logik und Komponenten zum Regulieren eines oder mehrerer Leistungszustände der Prozessorkerne 3402A-3402N und des Grafikprozessors 3408 beinhaltet.

[0289] In mindestens einer Ausführungsform beinhaltet der Prozessor 3400 zusätzlich den Grafikprozessor 3408 zum Ausführen von Grafikverarbeitungsoperationen. In mindestens einer Ausführungsform ist der Grafikprozessor 3408 mit gemeinsam genutzten Cache-Einheiten 3406 und dem Systemagentenkern 3410 gekoppelt, der eine oder mehrere integrierte Speichersteuerungen 3414 beinhaltet. In mindestens einer Ausführungsform beinhaltet der Systemagentenkern 3410 zudem eine Anzeigesteuerung 3411, um die Grafikprozessorausgabe zu einer oder mehreren gekoppelten Anzeigen zu treiben. In mindestens einer Ausführungsform kann die Anzeigesteuerung 3411 auch ein separates Modul sein, das über mindestens eine Zusammenschaltung mit dem Grafikprozessor 3408 gekoppelt ist, oder sie kann in den Grafikprozessor 3408 integriert sein.

[0290] In mindestens einer Ausführungsform wird eine ringbasierte Zusammenschaltungseinheit 3412 zur Kopplung interner Komponenten des Prozessors 3400 verwendet. In mindestens einer Ausführungsform kann eine alternative Zusammenschaltungseinheit verwendet werden, wie etwa eine Punkt-zu-Punkt-Zusammenschaltung, eine geschaltete Zusammenschaltung oder andere Techniken. In mindestens einer Ausführungsform ist der Grafikprozessor 3408 über eine E/A-Verbindung 3413 mit der Ringzusammenschaltung 3412 gekoppelt.

[0291] In mindestens einer Ausführungsform stellt die E/A-Verbindung 3413 mindestens eine von mehreren Varietäten von E/A-Verbindungen dar, einschließlich einer gehäuseinternen E/A-Verbindung, die Kommunikation zwischen verschiedenen Prozessorkomponenten und einem eingebetteten Speichermodul 3418 mit hoher Rechenleistung, wie etwa einem eDRAM-Modul, ermöglicht. In mindestens einer Ausführungsform verwenden jeder der Prozessorkerne 3402A-3402N und der Grafikprozessor 3408 eingebettete Speichermodule 3418 als gemeinsam genutzten Cache der letzten Ebene.

[0292] In mindestens einer Ausführungsform sind die Prozessorkerne 3402A-3402N homogene Kerne, die eine gemeinsame Anweisungssatzarchitektur ausführen. In mindestens einer Ausführungsform sind die Prozessorkerne 3402A-3402N bezüglich der Anweisungssatzarchitektur (instruction set architecture - ISA) hete-

rogen, wobei einer oder mehrere der Prozessorkerne 3402A-3402N einen gemeinsamen Anweisungssatz ausführen, während ein oder mehrere andere Kerne der Prozessorkerne 3402A-3402N eine Teilmenge eines gemeinsamen Anweisungssatzes oder eines anderen Anweisungssatzes ausführen. In mindestens einer Ausführungsform sind die Prozessorkerne 3402A-3402N bezüglich der Mikroarchitektur heterogen, wobei ein oder mehrere Kerne, die einen relativ betrachtet höheren Leistungsverbrauch aufweisen, mit einem oder mehreren Kernen, die einen niedrigeren Leistungsverbrauch aufweisen, gekoppelt sind. In mindestens einer Ausführungsform kann der Prozessor 3400 auf einem oder mehreren Chips oder als SoC-integrierte Schaltung implementiert sein.

[0293] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform können Abschnitte oder die gesamte Inferenz- und/oder Trainingslogik 1615 in den Prozessor 3400 integriert sein. Zum Beispiel können in mindestens einer Ausführungsform die in dieser Schrift beschriebenen Trainings- und/oder Inferenztechniken eine oder mehrere der ALUs verwenden, die in dem Grafikprozessor 3312, in dem/den Grafikkern(en) 3402A-3402N oder in anderen Komponenten in **Fig. 34** verkörpert sind. Darüber hinaus können in mindestens einer Ausführungsform die in dieser Schrift beschriebenen Inferenz- und/oder Trainingsoperationen unter Verwendung von anderer Logik als der in den **Fig. 16A** oder **Fig. 16B** veranschaulichten Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtungssparameter in chipinternem oder chipexternem Speicher und/oder in Registern (gezeigt oder nicht gezeigt) gespeichert werden, die ALUs des Grafikprozessors 3400 konfigurieren, um eine(n) oder mehrere in dieser Schrift beschriebene Algorithmen für maschinelles Lernen, neuronale Netzarchitekturen, Anwendungsfälle oder Trainingstechniken durchzuführen.

[0294] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0295] **Fig. 35** ist ein Blockdiagramm von Hardware-Logik eines Grafikprozessorkerns 3500 gemäß mindestens einer in dieser Schrift beschriebenen Ausführungsform. In mindestens einer Ausführungsform ist der Grafikprozessorkern 3500 in einem Grafikkernarray enthalten. In mindestens einer Ausführungsform kann es sich bei dem Grafikprozessorkern 3500, mitunter als Kernscheibe bezeichnet, um einen oder mehrere Grafikkerne innerhalb eines modularen Grafikprozessors handeln. In mindestens einer Ausführungsform ist der Grafikprozessorkern 3500 beispielhaft für eine Grafikkernscheibe und ein Grafikprozessor, wie in dieser Schrift beschrieben, kann auf Grundlage der angestrebten Leistungs- und Rechenleistungshüllkurven mehrere Grafikkernscheiben beinhalten. In mindestens einer Ausführungsform kann jeder Grafikkern 3500 einen Festfunktionsblock 3530 beinhalten, der mit mehreren Teilkernen 3501A-3501F gekoppelt ist, die auch als Teilscheiben bezeichnet werden und modulare Blöcke von Universal- und Festfunktionslogik beinhalten.

[0296] In mindestens einer Ausführungsform beinhaltet der Festfunktionsblock 3530 eine Geometrie-/Festfunktionspipeline 3536, die von allen Teilkernen im Grafikprozessor 3500 gemeinsam genutzt werden kann, zum Beispiel in Implementierungen mit Grafikprozessoren mit niedrigerer Rechenleistung und/oder niedrigerer Leistung. In mindestens einer Ausführungsform beinhaltet die Geometrie-/Festfunktionspipeline 3536 eine 3D-Festfunktionspipeline, eine Video-Frontend-Einheit, einen Thread-Spawner und Thread-Zuteiler sowie einen Manager für vereinheitlichten Rückgabepuffer, der den vereinheitlichten Rückgabepuffer verwaltet.

[0297] In mindestens einer Ausführungsform beinhaltet der Festfunktionsblock 3530 auch eine Grafik-SoC-Schnittstelle 3537, einen Grafik-Mikrocontroller 3538 und eine Medienpipeline 3539. In mindestens einer festen Ausführungsform stellt die Grafik-SoC-Schnittstelle 3537 eine Schnittstelle zwischen dem Grafikkern 3500 und anderen Prozessorkernen innerhalb einer integrierten Schaltung als System auf einem Chip bereit. In mindestens einer Ausführungsform ist der Grafik-Mikrocontroller 3538 ein programmierbarer Teilprozessor, der dazu konfiguriert werden kann, er verschiedene Funktionen des Grafikprozessors 3500 zu verwalten, einschließlich Thread-Zuteilung, -Planung und -Präemption. In mindestens einer Ausführungsform beinhaltet die Medienpipeline 3539 Logik zum Erleichtern der Decodierung, Codierung, Vorverarbeitung und/oder Nachverarbeitung von Multimedia-Daten, einschließlich Bild- und Videodaten. In mindestens einer Ausführungsform implementiert die Medienpipeline 3539 Medienoperationen über Anforderungen an Rechen- oder Abtastlogik innerhalb der Teilkerne 3501A-3501F.

[0298] In mindestens einer Ausführungsform ermöglicht die SoC-Schnittstelle 3537 es dem Grafikkern 3500, mit Universal-Anwendungsprozessorkernen (z. B. CPUs) und/oder anderen Komponenten innerhalb eines SoC zu kommunizieren, einschließlich Speicherhierarchieelementen, wie etwa einem gemeinsam genutzten Cache-Speicher der letzten Ebene, System-RAM und/oder eingebettetem chipinternem oder gehäuseinternem DRAM. In mindestens einer Ausführungsform kann die SoC-Schnittstelle 3537 auch Kommunikation mit Festfunktionsvorrichtungen innerhalb eines SoC ermöglichen, wie etwa Kamera-Bildgebungs pipelines, und sie ermöglicht die Verwendung von globalem atomarem Speicher und/oder implementiert diesen, der von dem Grafikkern 3500 und den CPUs innerhalb eines SoCs gemeinsam genutzt werden kann. In mindestens einer Ausführungsform kann die SoC-Schnittstelle 3537 auch Leistungsverwaltungssteuerungen für den Grafikkern 3500 implementieren und eine Schnittstelle zwischen einer Taktdomäne des Grafikkerns 3500 und anderen Taktdomänen innerhalb eines SoCs ermöglichen. In mindestens einer Ausführungsform ermöglicht die SoC-Schnittstelle 3537 den Empfang von Befehlspuffern von einem Befehls-Streamer und einem globalen Thread-Zuteiler, die dazu konfiguriert sind, jedem von einem oder mehreren Grafikkernen innerhalb eines Grafikprozessors Befehle und Anweisungen bereitzustellen. In mindestens einer Ausführungsform können Befehle und Anweisungen an die Medienpipeline 3539 gesendet werden, wenn Medienoperationen durchgeführt werden sollen, oder an eine Geometrie- und Festfunktionspipeline (z. B. Geometrie- und Festfunktionspipeline 3536, Geometrie- und Festfunktionspipeline 3514), wenn Grafikverarbeitungsoperationen durchgeführt werden sollen.

[0299] In mindestens einer Ausführungsform kann der Grafik-Mikrocontroller 3538 dazu konfiguriert sein, verschiedene Planungs- und Verwaltungsaufgaben für den Grafikkern 3500 durchzuführen. In mindestens einer Ausführungsform kann der Grafik-Mikrocontroller 3538 die Planung der Grafik- und/oder Rechenarbeitslast auf verschiedenen Grafikparallel-Engines innerhalb der Arrays 3502A-3502F, 3504A-3504F der Ausführungseinheiten (execution unit - EU) innerhalb der Teilkerns 3501A-3501F durchführen. In mindestens einer Ausführungsform kann Host-Software, die auf einem CPU-Kern eines SoC, einschließlich des Grafikkerns 3500, ausgeführt wird, Arbeitslasten an eine von mehreren Grafikprozessor-Doorbells übermitteln, die eine Planungsoperation auf einer geeigneten Grafik-Engine aufruft. In mindestens einer Ausführungsform beinhalten die Planungsoperationen das Bestimmen, welche Arbeitslast als Nächstes laufen soll, das Absenden einer Arbeitslast an einen Befehls-Streamer, das Vorwegnehmen bestehender Arbeitslasten, die auf einer Engine laufen, das Überwachen des Fortschritts einer Arbeitslast und das Benachrichtigen der Host-Software, wenn eine Arbeitslast abgeschlossen ist. In mindestens einer Ausführungsform kann der Grafik-Mikrocontroller 3538 auch Zustände mit niedriger Leistung oder inaktive Zustände für den Grafikkern 3500 erleichtern, wobei dem Grafikkern 3500 eine Fähigkeit bereitgestellt wird, Register innerhalb des Grafikkerns 3500 über Zustandsübergänge mit niedriger Leistung unabhängig von einem Betriebssystem und/oder einer Grafiktreiber-Software auf einem System zu sichern und wiederherzustellen.

[0300] In mindestens einer Ausführungsform kann der Grafikkern 3500 mehr oder weniger als die veranschaulichten Teilkerns 3501A-3501F aufweisen, bis zu N modulare Teilkerns. Für jeden Satz von N Teilkernen kann der Grafikkern 3500 in mindestens einer Ausführungsform auch gemeinsam genutzte Funktionslogik 3510, gemeinsam genutzten und/oder Cache-Speicher 3512, eine Geometrie-/Festfunktionspipeline 3514 sowie zusätzliche Festfunktionslogik 3516 zum Beschleunigen verschiedener Grafik- und Rechenverarbeitungsoperationen beinhalten. In mindestens einer Ausführungsform kann die gemeinsam genutzte Funktionslogik 3510 Logikeinheiten (z. B. Abtaster-, Mathematik- und/oder Zwischen-Thread-Kommunikationslogik) beinhalten, die von N Teilkernen innerhalb des Grafikkerns 3500 gemeinsam genutzt werden können. In mindestens einer Ausführungsform kann der feste, gemeinsam genutzte und/oder Cache-Speicher 3512 ein Cache der letzten Ebene für N Teilkerns 3501A-3501F innerhalb des Grafikkerns 3500 sein und kann auch als gemeinsam genutzter Speicher dienen, auf den mehrere Teilkerns zugreifen können. In mindestens einer Ausführungsform kann die Geometrie-/Festfunktionspipeline 3514 anstelle der Geometrie-/Festfunktionspipeline 3536 innerhalb des Festfunktionsblocks 3530 enthalten sein und gleiche oder ähnliche Logikeinheiten beinhalten.

[0301] In mindestens einer Ausführungsform beinhaltet der Grafikkern 3500 zusätzliche Festfunktionslogik 3516, die verschiedene Festfunktionsbeschleunigungslogik zur Verwendung durch den Grafikkern 3500 beinhalten kann. In mindestens einer Ausführungsform beinhaltet die zusätzliche Festfunktionslogik 3516 eine zusätzliche Geometriepipeline für die Verwendung beim Shading von nur der Position. Beim Shading von nur der Position existieren mindestens zwei Geometriepipelines, während eine vollständige Geometriepipeline innerhalb der Geometrie-/Festfunktionspipeline 3516, 3536 und eine Culling-Pipeline, die eine zusätzliche Geometriepipeline ist, die innerhalb der zusätzlichen Festfunktionslogik 3516 enthalten sein kann. In mindestens einer Ausführungsform ist die Culling-Pipeline eine abgespeckte Version einer vollständigen Geometriepipeline. In mindestens einer Ausführungsform können eine vollständige Pipeline und eine Cul-

ling-Pipeline unterschiedliche Instanzen einer Anwendung ausführen, wobei jede Instanz einen separaten Kontext aufweist. In mindestens einer Ausführungsform kann das Shading von nur der Position lange Culling-Läufe von verworfenen Dreiecken verbergen, sodass ermöglicht wird, dass das Shading in einigen Fällen früher abgeschlossen wird. Zum Beispiel kann in mindestens einer Ausführungsform die Culling-Pipeline-Logik innerhalb der zusätzlichen Festfunktionslogik 3516 Positions-Shader parallel zu einer Hauptanwendung ausführen und erzeugt im Allgemeinen kritische Ergebnisse schneller als eine vollständige Pipeline, da die Culling-Pipeline das Positionsattribut von Vertices abrufen und schattiert, ohne eine Rasterung und ein Rendering von Pixeln in einem Bildspeicher durchzuführen. In mindestens einer Ausführungsform kann die Culling-Pipeline erzeugte kritische Ergebnisse verwenden, um Sichtbarkeitsinformationen für alle Dreiecke zu berechnen, ohne Rücksicht darauf, ob diese Dreiecke Culling unterzogen werden. In mindestens einer Ausführungsform kann die vollständige Pipeline (die in diesem Fall als Wiederholungspipeline bezeichnet werden kann) Sichtbarkeitsinformationen verbrauchen, um Culling unterzogene Dreiecke zu überspringen, um nur sichtbare Dreiecke zu schattieren, die schließlich an eine Rasterungsphase übergeben werden.

[0302] In mindestens einer Ausführungsform kann die zusätzliche Festfunktionslogik 3516 auch Logik zur Beschleunigung des maschinellen Lernens wie etwa Festfunktions-Matrixmultiplikationslogik für Implementierungen beinhalten, die Optimierungen für das Training oder das Inferenzieren des maschinellen Lernens beinhalten.

[0303] In mindestens einer Ausführungsform beinhaltet jeder Grafikeilkern 3501A-3501F einen Satz von Ausführungsressourcen, die verwendet werden können, um Grafik-, Medien- und Rechenoperationen als Reaktion auf Anforderungen durch Grafikpipeline-, Medienpipeline- oder Shader-Programme durchzuführen. In mindestens einer Ausführungsform beinhalten die Grafikeilkerne 3501A-3501F mehrere EU-Arrays 3502A-3502F, 3504A-3504F, Logik 3503A-3503F zur Thread-Zuteilung und Zwischen-Thread-Kommunikation (thread dispatch/inter-thread communication - TD/IC), einen 3D-(z. B. Textur-)Abtaster 3505A-3505F, einen Medienabtaster 3506A-3506F, einen Shader-Prozessor 3507A-3507F und gemeinsam genutzten lokalen Speicher (shared local memory - SLM) 3508A-3508F. Die EU-Arrays 3502A-3502F, 3504A-3504F beinhalten jeweils mehrere Ausführungseinheiten, bei denen es sich um Universal-Grafikverarbeitungseinheiten handelt, die in der Lage sind, Gleitkomma- und Integer-/Festkomma-Logikoperationen im Dienste einer Grafik-, Medien- oder Rechenoperation, einschließlich Grafik-, Medien- oder Rechen-Shader-Programmen, durchzuführen. In mindestens einer Ausführungsform führt die TD/IC-Logik 3503A-3503F lokale Thread-Zuteilungs- und Thread-Steueroperationen für Ausführungseinheiten innerhalb eines Teilkerns durch und erleichtert die Kommunikation zwischen Threads, die auf Ausführungseinheiten eines Teilkerns ausgeführt werden. In mindestens einer Ausführungsform können die 3D-Abtaster 3505A-3505F Textur- oder andere mit 3D-Grafik verwandte Daten in Speicher lesen. In mindestens einer Ausführungsform kann der 3D-Abtaster Texturdaten auf Grundlage eines konfigurierten Abtastzustands und eines Texturformats, das einer gegebenen Textur zugeordnet ist, unterschiedlich lesen. In mindestens einer Ausführungsform kann der Medien-Abtaster 3506A-3506F ähnliche Leseoperationen auf Grundlage einer und eines Formats durchführen, die den Mediendaten zugeordnet sind. In mindestens einer Ausführungsform kann jeder Grafikeilkern 3501A-3501F alternativ einen vereinheitlichten 3D- und Medienabtaster beinhalten. In mindestens einer Ausführungsform können Threads, die auf Ausführungseinheiten innerhalb jedes der Teilkerne 3501A-3501F ausgeführt werden, den gemeinsam genutzten lokalen Speicher 3508A-3508F innerhalb jedes Teilkerns verwenden, um zu ermöglichen, dass Threads, die innerhalb einer Thread-Gruppe ausgeführt werden, unter Verwendung eines gemeinsamen Pools von chipinternem Speicher ausgeführt werden.

[0304] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform können Abschnitte oder die gesamte Inferenz- und/oder Trainingslogik 1615 in den Grafikprozessor 3510 integriert sein. Zum Beispiel können in mindestens einer Ausführungsform die in dieser Schrift beschriebenen Trainings- und/oder Inferenztechniken eine oder mehrere der ALUs verwenden, die in dem Grafikprozessor 3312, in dem Grafik-Mikrocontroller 3538, in der Geometrie- und Festfunktionspipeline 3514 und 3536 oder einer anderen Logik in **Fig. 34** verkörpert sind. Darüber hinaus können in mindestens einer Ausführungsform die in dieser Schrift beschriebenen Inferenz- und/oder Trainingsoperationen unter Verwendung von anderer Logik als der in den **Fig. 16A** oder **Fig. 16B** veranschaulichten Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtungssparameter in chipinternem oder chipexternem Speicher und/oder in Registern (gezeigt oder nicht gezeigt) gespeichert werden, die ALUs des Grafikprozessors 3500 konfigurieren, um eine(n) oder mehrere in dieser Schrift beschriebene Algorithmen für maschinelles Lernen, neuronale Netzarchitekturen, Anwendungsfälle oder Trainingstechniken durchzuführen.

[0305] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0306] Die **Fig. 36A-36B** veranschaulichen die Thread-Ausführungslogik 3600, die ein Array von Verarbeitungselementen eines Grafikprozessorkerns beinhaltet, gemäß mindestens einer Ausführungsform. **Fig. 36A** veranschaulicht mindestens eine Ausführungsform, in der die Thread-Ausführungslogik 3600 verwendet wird. **Fig. 36B** veranschaulicht beispielhafte interne Details einer Ausführungseinheit gemäß mindestens einer Ausführungsform.

[0307] Wie in **Fig. 36A** veranschaulicht, beinhaltet die Thread-Ausführungslogik 3600 in mindestens einer Ausführungsform einen Shader-Prozessor 3602, einen Thread-Zuteiler 3604, einen Anweisungs-Cache 3606, ein skalierbares Ausführungseinheitenarray einschließlich einer Vielzahl von Ausführungseinheiten 3608A-3608N, (einen) Abtaster 3610, einen Daten-Cache 3612 und einen Datenport 3614. In mindestens einer Ausführungsform kann ein skalierbares Ausführungseinheitenarray dynamisch skalieren, indem eine oder mehrere Ausführungseinheiten (z. B. beliebige der Ausführungseinheiten 3608A, 3608B, 3608C, 3608D bis 3608N-1 und 3608N) zum Beispiel auf Grundlage von Rechenanforderungen einer Arbeitslast aktiviert oder deaktiviert werden. In mindestens einer Ausführungsform sind die skalierbaren Ausführungseinheiten über eine Zusammenschaltungsstruktur miteinander verbunden, die mit allen Ausführungseinheiten verbunden ist. In mindestens einer Ausführungsform umfasst die Thread-Ausführungslogik 3600 eine oder mehrere Verbindungen zu einem Speicher, z. B. einem Systemspeicher oder einem Cache-Speicher, durch eines oder mehrere des Anweisungs-Caches 3606, des Datenports 3614, des Abtasters 3610 und der Ausführungseinheiten 3608A-3608N. In mindestens einer Ausführungsform ist jede Ausführungseinheit (z. B. 3608A) eine eigenständige programmierbare Universalrecheneinheit, die dazu in der Lage ist, mehrere simultane Hardware-Threads auszuführen, während mehrere Datenelemente parallel für jeden Thread verarbeitet werden. In mindestens einer Ausführungsform ist das Array der Ausführungseinheiten 3608A-3608N und/oder 3308 so skalierbar, dass es eine beliebige Anzahl einzelner Ausführungseinheiten beinhaltet.

[0308] In mindestens einer Ausführungsform werden die Ausführungseinheiten 3608A-3608N hauptsächlich zur Ausführung von Shader-Programmen verwendet. In mindestens einer Ausführungsform kann der Shader-Prozessor 3602 verschiedene Shader-Programme verarbeiten und den Shader-Programmen zugeordnete Ausführungs-Threads über einen Thread-Zuteiler 3604 zuteilen. In mindestens einer Ausführungsform beinhaltet der Thread-Zuteiler 3604 Logik zur Vermittlung von Thread-Initiierungsanforderungen von Grafik- und Medienpipelines und zur Instanziierung angeforderter Threads auf einer oder mehreren Ausführungseinheiten in den Ausführungseinheiten 3608A-3608N. Zum Beispiel kann in mindestens einer Ausführungsform eine Geometriepipeline Vertex-, Tesselations- oder Geometrie-Shader der Thread-Ausführungslogik zum Verarbeiten zuteilen. In mindestens einer Ausführungsform kann der Thread-Zuteiler 3604 auch Laufzeit-Thread-Erzeugungsanforderungen von ausführenden Shader-Programmen verarbeiten.

[0309] In mindestens einer Ausführungsform unterstützen die Ausführungseinheiten 3608A-3608N einen Anweisungssatz, der systemeigene Unterstützung für viele Standard-3D-Grafik-Shader-Anweisungen beinhaltet, sodass Shader-Programme aus Grafikbibliotheken (z. B. Direct 3D und OpenGL) mit einer minimalen Übersetzung ausgeführt werden. In mindestens einer Ausführungsform unterstützen die Ausführungseinheiten die Vertex- und Geometrieverarbeitung (z. B. Vertex-Programme, Geometrieprogramme, Vertex-Shader), die Pixelverarbeitung (z. B. Pixel-Shader, Fragment-Shader) und die Universalverarbeitung (z. B. Rechen- und Medien-Shader). In mindestens einer Ausführungsform ist jede der Ausführungseinheiten 3608A-3608N, die eine oder mehrere arithmetisch-logische Einheiten (ALUs) beinhalten, zur Multi-Issue-Single-Instruction-Multiple-Data-(SIMD-)Ausführung in der Lage und der Mehr-Thread-Betrieb ermöglicht eine effiziente Ausführungsumgebung trotz Speicherzugriffen mit höherer Latenz. In mindestens einer Ausführungsform weist jeder Hardware-Thread innerhalb jeder Ausführungseinheit eine dedizierte Registerdatei mit hoher Bandbreite und einen zugeordneten unabhängigen Thread-Zustand auf. In mindestens einer Ausführungsform erfolgt die Ausführung mit mehreren Ausgaben pro Takt an Pipelines, die zu Integer- und Gleitkommaoperationen mit einfacher oder doppelter Genauigkeit, SIMD-Verzweigungsfähigkeit, logischen Operationen, transzendentalen Operationen und anderen sonstigen Operationen in der Lage sind. In mindestens einer Ausführungsform bewirkt die Abhängigkeitslogik in den Ausführungseinheiten 3608A-3608N, dass ein wartender Thread schläft, bis die angeforderten Daten zurückgegeben wurden, während er auf Daten aus dem Speicher oder einer der gemeinsam genutzten Funktionen wartet. In mindestens einer Ausführungsform können, während ein wartender Thread schläft, Hardware-Ressourcen für die Verarbeitung anderer Threads verwendet werden. Zum Beispiel kann in mindestens einer Ausführungsform eine Ausführungseinheit während einer Verzögerung, die einer Vertex-Shader-Operation zugeordnet ist, Operationen für einen Pixel-Shader, Frag-

ment-Shader oder eine andere Art von Shader-Programm, einschließlich eines anderen Vertex-Shaders, durchführen.

[0310] In mindestens einer Ausführungsform arbeitet jede Ausführungseinheit in den Ausführungseinheiten 3608A-3608N an Arrays von Datenelementen. In mindestens einer Ausführungsform ist die Anzahl der Datenelemente die „Ausführungsgröße“ oder die Anzahl der Kanäle für eine Anweisung. In mindestens einer Ausführungsform ist ein Ausführungskanal eine logische Ausführungseinheit für den Zugriff auf Datenelemente, die Maskierung und die Flusssteuerung innerhalb von Anweisungen. In mindestens einer Ausführungsform kann die Anzahl der Kanäle unabhängig von einer Anzahl der physischen arithmetisch-logischen Einheiten (ALUs) oder Gleitkommaeinheiten (FPUs) für einen konkreten Grafikprozessor sein. In mindestens einer Ausführungsform unterstützen die Ausführungseinheiten 3608A-3608N Integer- und Gleitkommadatenarten.

[0311] In mindestens einer Ausführungsform beinhaltet ein Anweisungssatz einer Ausführungseinheit SIMD-Anweisungen. In mindestens einer Ausführungsform können verschiedene Datenelemente als gepackter Datentyp in einem Register gespeichert werden und eine Ausführungseinheit verarbeitet verschiedene Elemente auf Grundlage der Datengröße der Elemente. Zum Beispiel werden in mindestens einer Ausführungsform beim Betreiben an einem 256 Bit breiten Vektor 256 Bit eines Vektors in einem Register gespeichert und eine Ausführungseinheit wird an einem Vektor als vier separate 64 Bit große gepackte Datenelemente (Datenelemente der Größe Vierfachwort (Quad-Word - QW)), acht separate 32 Bit große gepackte Datenelemente (Datenelemente der Größe Doppelwort (Double Word - DW)), sechzehn separate 16 Bit große gepackte Datenelemente (Datenelemente der Größe Wort (Word - W)) oder zweiunddreißig separate 8 Bit große Datenelemente (Datenelemente der Größe Byte (B)) betrieben. In mindestens einer Ausführungsform sind jedoch andere Vektorbreiten und Registergrößen möglich.

[0312] In mindestens einer Ausführungsform können eine oder mehrere Ausführungseinheiten zu einer fusionierten Ausführungseinheit 3609A-3609N kombiniert werden, die Thread-Steuerlogik (3607A-3607N) aufweist, die fusionierten EUs gemeinsam ist. In mindestens einer Ausführungsform können mehrere EUs zu einer EU-Gruppe fusioniert werden. In mindestens einer Ausführungsform kann jede EU in der fusionierten EU-Gruppe dazu konfiguriert sein, einen separaten SIMD-Hardware-Thread auszuführen. Die Anzahl der EUs in einer fusionierten EU-Gruppe kann gemäß verschiedenen Ausführungsformen variieren. In mindestens einer Ausführungsform können verschiedene SIMD-Breiten pro EU durchgeführt werden, einschließlich, aber nicht beschränkt auf, SIMD8, SIMD16 und SIMD32. In mindestens einer Ausführungsform beinhaltet jede fusionierte Grafikausführungseinheit 3609A-3609N mindestens zwei Ausführungseinheiten. Zum Beispiel beinhaltet in mindestens einer Ausführungsform die fusionierte Ausführungseinheit 3609A eine erste EU 3608A, eine zweite EU 3608B und Thread-Steuerlogik 3607A, die der ersten EU 3608A und der zweiten EU 3608B gemeinsam ist. In mindestens einer Ausführungsform steuert die Thread-Steuerlogik 3607A Threads, die auf der fusionierten Grafikausführungseinheit 3609A ausgeführt werden, sodass jede EU innerhalb der fusionierten Ausführungseinheiten 3609A-3609N unter Verwendung eines gemeinsamen Anweisungszeigerregisters ausgeführt werden kann.

[0313] In mindestens einer Ausführungsform sind ein oder mehrere interne Anweisungs-Caches (z. B. 3606) in der Thread-Ausführungslogik 3600 enthalten, um Thread-Anweisungen für Ausführungseinheiten zwischenspeichern. In mindestens einer Ausführungsform sind ein oder mehrere Daten-Caches (z. B. 3612) enthalten, um Thread-Daten während der Thread-Ausführung zwischenspeichern. In mindestens einer Ausführungsform ist ein Abtaster 3610 enthalten, der Texturabtastung für 3D-Operationen und Medienabtastung für Medienoperationen bereitstellt. In mindestens einer Ausführungsform beinhaltet der Abtaster 3610 eine spezielle Textur- oder Medienabtastfunktionalität, um Textur- oder Mediendaten während eines Abtastprozesses zu verarbeiten, bevor die abgetasteten Daten einer Ausführungseinheit bereitgestellt werden.

[0314] Während der Ausführung senden Grafik- und Medienpipelines in mindestens einer Ausführungsform Thread-Initiierungsanforderungen über die Thread-Erzeugungs- und -Zuteilungslogik an die Thread-Ausführungslogik 3600. In mindestens einer Ausführungsform wird, sobald eine Gruppe geometrischer Objekte verarbeitet und in Pixeldaten gerastert wurde, die Pixelprozessorlogik (z. B. Pixel-Shader-Logik, Fragment-Shader-Logik usw.) innerhalb des Shader-Prozessors 3602 aufgerufen, um Ausgabeinformationen weiter zu berechnen und zu veranlassen, dass die Ergebnisse in Ausgabeblöcken (z. B. Farbpuffer, Tiefenpuffer, Schablonenpuffer usw.) geschrieben werden. In mindestens einer Ausführungsform berechnet ein Pixel-Shader oder Fragment-Shader die Werte verschiedener Vertex-Attribute, die über ein gerastertes Objekt interpoliert werden sollen. In mindestens einer Ausführungsform führt die Pixelprozessorlogik innerhalb des Shader-Prozessors 3602 dann ein über eine Anwendungsprogrammierschnittstelle (API) zugeführtes Pixel- oder

Fragment-Shader-Programm aus. In mindestens einer Ausführungsform teilt der Shader-Prozessor 3602 zum Ausführen eines Shader-Programms Threads über den Thread-Zuteiler 3604 einer Ausführungseinheit (z. B. 3608A) zu. In mindestens einer Ausführungsform verwendet der Shader-Prozessor 3602 die Texturab-tastlogik in dem Abtaster 3610, um auf Texturdaten in Texturkarten zuzugreifen, die in Speicher gespeichert sind. In mindestens einer Ausführungsform werden durch arithmetische Operationen an Texturdaten und ein-gegebenen Geometriedaten Pixelfarbdaten für jedes geometrische Fragment berechnet oder ein oder meh-rere Pixel von der weiteren Verarbeitung ausgeschlossen.

[0315] In mindestens einer Ausführungsform stellt der Datenport 3614 einen Speicherzugriffsmechanismus für die Thread-Ausführungslogik 3600 bereit, um verarbeitete Daten zur weiteren Verarbeitung an einer Gra-fikprozessor-Ausgabepipeline an Speicher auszugeben. In mindestens einer Ausführungsform beinhaltet der Datenport 3614 einen oder mehrere Cache-Speicher (z. B. den Daten-Cache 3612) oder ist an diesen gekop-pelt, um Daten für den Speicherzugriff über einen Datenport zwischenzuspeichern.

[0316] Wie in **Fig. 36B** veranschaulicht, kann eine Grafikausführungseinheit 3608 in mindestens einer Aus-führungsform eine Anweisungsabrufeinheit 3637, ein Array von allgemeinen Registerdateien (general regis-ter file - GRF) 3624, eine Anordnung von architektonischen Registerdateien (architectural register file - ARF) 3626, einen Thread-Vermittler 3622, eine Sendeeinheit 3630, eine Verzweigungseinheit 3632, einen Satz SIMD-Gleitkommaeinheiten (FPUs) 3634 und in mindestens einer Ausführungsform einen Satz dedizierter Integer-SIMD-ALUs 3635 beinhalten. In mindestens einer Ausführungsform beinhalten die GRF 3624 und die ARF 3626 einen Satz allgemeiner Registerdateien und Architekturregisterdateien, die jedem simultanen Hardware-Thread zugeordnet sind, der in der Grafikausführungseinheit 3608 aktiv sein kann. In mindestens einer Ausführungsform wird der architektonische Zustand pro Thread in der ARF 3626 verwaltet, während die während der Thread-Ausführung verwendeten Daten in der GRF 3624 gespeichert werden. In mindes-tens einer Ausführungsform kann der Ausführungszustand jedes Threads, einschließlich der Anweisungszei-ger für jeden Thread, in Thread-spezifischen Registern in der ARF 3626 aufbewahrt werden.

[0317] In mindestens einer Ausführungsform weist die Grafikausführungseinheit 3608 eine Architektur auf, die eine Kombination aus simultanem Multi-Threading (Simultaneous Multi-Threading - SMT) und feinkörnig-em verschachteltem Multi-Threading (Interleaved Multi-Threading - IMT) ist. In mindestens einer Ausführ-ungsform weist die Architektur eine modulare Konfiguration auf, die zur Ausgestaltungszeit auf Grundlage einer Zielanzahl von simultanen Threads und Anzahl von Registern pro Ausführungseinheit fein abgestimmt werden kann, wobei die Ressourcen der Ausführungseinheit auf die Logik aufgeteilt werden, die zum Ausfüh-ren mehrerer simultaner Threads verwendet wird.

[0318] In mindestens einer Ausführungsform kann die Grafikausführungseinheit 3608 mehrere Anweisungen gleichzeitig ausgeben, die jeweils unterschiedliche Anweisungen sein können. In mindestens einer Ausführ-ungsform kann der Thread-Vermittler 3622 des Grafikausführungseinheits-Threads 3608 Anweisungen einer der Sendeeinheit 3630, der Verzweigungseinheit 3642 oder der SIMD-FPU(s) 3634 zur Ausführung zuteilen. In mindestens einer Ausführungsform kann jeder Ausführungs-Thread auf 128 Universalregister innerhalb der GRF 3624 zugreifen, wobei jedes Register 32 Byte speichern kann, die als SIMD-8-Element-vektor von 32-Bit-Datenelementen zugänglich sind. In mindestens einer Ausführungsform weist jeder Thread der Ausführungseinheit Zugriff auf 4 KByte innerhalb der GRF 3624 auf, obwohl Ausführungsformen nicht derart beschränkt sind und in anderen Ausführungsformen mehr oder weniger Registerressourcen bereitge-stellt sein können. In mindestens einer Ausführungsform können bis zu sieben Threads simultan ausgeführt werden, obwohl eine Anzahl der Threads pro Ausführungseinheit gemäß Ausführungsformen auch variieren kann. In mindestens einer Ausführungsform, in der sieben Threads auf 4 KByte zugreifen können, kann die GRF 3624 insgesamt 28 KByte speichern. In mindestens einer Ausführungsform können flexible Adressie-rungsmodi erlauben, dass Register zusammen adressiert werden, um effektiv breitere Register zu bilden oder um geschichtete rechteckige Blockdatenstrukturen darzustellen.

[0319] In mindestens einer Ausführungsform werden Speicheroperationen, Abtasteroperationen und andere Systemkommunikationen mit längerer Latenz über „Sende“-Anweisungen zugeteilt, die durch Nachrichten-weitergabe an die Sendeeinheit 3630 ausgeführt werden. In mindestens einer Ausführungsform werden Ver-zweigungsanweisungen einer dedizierten Verzweigungseinheit 3632 zugeteilt, um SIMD-Divergenz und schließlich -Konvergenz zu ermöglichen.

[0320] In mindestens einer Ausführungsform beinhaltet die Grafikausführungseinheit 3608 eine oder meh-rere SIMD-Gleitkommaeinheiten (FPU(s)) 3634 zur Durchführung von Gleitkommaoperationen. In mindes-tens einer Ausführungsform unterstützen die FPU(s) 3634 auch Integerberechnung. In mindestens einer Aus-

führungsform können die FPU(s) 3634 bis zur Anzahl von M 32-Bit-Gleitkomma(oder -Integer)operationen über SMID ausführen oder bis zu 2M 16-Bit-Integer- oder 16-Bit-Gleitkommaoperationen über SMID ausführen. In mindestens einer Ausführungsform stellt mindestens eine der FPU(s) erweiterte mathematische Fähigkeiten bereit, um transzendente mathematische Funktionen mit hohem Durchsatz und 64-Bit-Gleitkomma mit doppelter Genauigkeit zu unterstützen. In mindestens einer Ausführungsform ist auch ein Satz von 8-Bit-Integer-SIMD-ALUs 3635 vorhanden, der spezifisch zum Durchführen von Operationen, die Berechnungen für maschinelles Lernen zugeordnet sind, optimiert sein kann.

[0321] In mindestens einer Ausführungsform können Arrays von mehreren Instanzen der Grafikausführungseinheit 3608 in einer Grafikteilkern-Gruppierung (z. B. einer Teilscheibe) instanziiert sein. In mindestens einer Ausführungsform kann die Ausführungseinheit 3608 Anweisungen über eine Vielzahl von Ausführungskanälen hinweg ausführen. In mindestens einer Ausführungsform wird jeder Thread, der auf der Grafikausführungseinheit 3608 ausgeführt wird, auf einem anderen Kanal ausgeführt.

[0322] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform können Abschnitte oder die gesamte Inferenz- und/oder Trainingslogik 1615 in die Ausführungslogik 3600 integriert sein. Darüber hinaus können in mindestens einer Ausführungsform die in dieser Schrift beschriebenen Inferenz- und/oder Trainingsoperationen unter Verwendung von anderer Logik als der in den **Fig. 16A** oder **Fig. 16B** veranschaulichten Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtungsparemeter im chipinternen oder chipexternen Speicher und/oder in Registern (gezeigt oder nicht gezeigt) gespeichert werden, die ALUs der Ausführungslogik 3600 konfigurieren, um eine(n) oder mehrere in dieser Schrift beschriebene Algorithmen für maschinelles Lernen, neuronale Netzarchitekturen, Anwendungsfälle oder Trainingstechniken durchzuführen.

[0323] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0324] **Fig. 37** veranschaulicht eine Parallelverarbeitungseinheit („PPU“) 3700 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist die PPU 3700 mit maschinenlesbarem Code konfiguriert, der bei Ausführung durch die PPU 3700 die PPU 3700 dazu veranlasst, einige oder alle der in dieser Offenbarung beschriebenen Prozesse und Techniken durchzuführen. In mindestens einer Ausführungsform ist die PPU 3700 ein Multi-Thread-Prozessor, der auf einer oder mehreren Vorrichtungen mit integrierter Schaltung implementiert ist und der Multi-Threading als Technik zur Latenzverbergung nutzt, die dazu ausgestaltet ist, computerlesbare Anweisungen (auch als maschinenlesbare Anweisungen oder einfach Anweisungen bezeichnet) auf mehreren Threads parallel zu verarbeiten. In mindestens einer Ausführungsform bezieht sich ein Thread auf einen Ausführungs-Thread und er ist eine Instanziierung eines Satzes von Anweisungen, der dazu konfiguriert ist, durch die PPU 3700 ausgeführt zu werden. In mindestens einer Ausführungsform ist die PPU 3700 eine Grafikverarbeitungseinheit („GPU“), die dazu konfiguriert ist, eine Grafik-Rendering-Pipeline zum Verarbeiten dreidimensionaler („3D“) Grafikdaten zu implementieren, um zweidimensionale („2D“) Bilddaten zur Anzeige auf einer Anzeigevorrichtung, wie etwa einer Flüssigkristallanzeige („LCD“-)Vorrichtung, zu erzeugen. In mindestens einer Ausführungsform wird die PPU 3700 verwendet, um Berechnungen, wie etwa lineare Algebraoperationen und Operationen für maschinelles Lernen, durchzuführen. **Fig. 37** veranschaulicht ein Beispiel für einen Parallelprozessor, der nur zu Veranschaulichungszwecken dient und als nicht einschränkendes Beispiel für Prozessorarchitekturen ausgelegt werden sollte, die innerhalb des Umfangs dieser Offenbarung in Betracht gezogen werden, und dass ein beliebiger geeigneter Prozessor zur Ergänzung und/oder als Ersatz für diesen eingesetzt werden kann.

[0325] In mindestens einer Ausführungsform sind eine oder mehrere PPU 3700 dazu konfiguriert, Anwendungen für Hochleistungsrechnen (High Performance Computing - „HPC“), für ein Rechenzentrum und für maschinelles Lernen zu beschleunigen. In mindestens einer Ausführungsform ist die PPU 3700 dazu konfiguriert, Deep-Learning-Systeme und -Anwendungen zu beschleunigen, einschließlich der folgenden nicht einschränkenden Beispiele: Plattformen für autonome Fahrzeuge, Deep Learning, Sprach-, Bild- und Texterkennungssysteme mit hoher Genauigkeit, intelligente Videoanalytik, Molekularsimulationen, Arzneimittelforschung, Krankheitsdiagnose, Wettervorhersage, Big-Data-Analytik, Astronomie, Molekulardynamiksimulation, Finanzmodellierung, Robotik, Fabrikautomatisierung, Echtzeit-Sprachübersetzung, Online-Suchoptimierungen und personalisierte Benutzerempfehlungen und andere.

[0326] In mindestens einer Ausführungsform beinhaltet die PPU 3700 ohne Einschränkung eine Eingabe-/Ausgabe- („E/A“-)Einheit 3706, eine Frontend-Einheit 3710, eine Planer-Einheit 3712, eine Arbeitsverteilungseinheit 3714, einen Hub 3716, eine Kreuzschiene („Xbar“) 3720, ein oder mehrere Universalverarbeitungskluster („GPCs“) 3718 und eine oder mehrere Partitionseinheiten („Speicherpartitionseinheiten“) 3722. In mindestens einer Ausführungsform ist die PPU 3700 mit einem Host-Prozessor oder anderen PPUs 3700 über eine oder mehrere Hochgeschwindigkeits-GPU-Zusammenschaltungen („GPU-Zusammenschaltungen“) 3708 verbunden. In mindestens einer Ausführungsform ist die PPU 3700 über eine Zusammenschaltung 3702 mit einem Host-Prozessor oder anderen Peripherievorrichtungen verbunden. In mindestens einer Ausführungsform ist die PPU 3700 mit einem lokalen Speicher verbunden, der eine oder mehrere Speichervorrichtungen („Speicher“) 3704 umfasst. In mindestens einer Ausführungsform beinhalten die Speichervorrichtungen 3704 ohne Einschränkung eine oder mehrere Vorrichtungen mit dynamischem Direktzugriffsspeicher („DRAM“). In mindestens einer Ausführungsform sind eine oder mehrere DRAM-Vorrichtungen als Teilsysteme mit Speicher mit hoher Bandbreite („HBM“) konfiguriert und/oder konfigurierbar, wobei mehrere DRAM-Dies innerhalb jeder Vorrichtung gestapelt sind.

[0327] In mindestens einer Ausführungsform kann sich die Hochgeschwindigkeits-GPU-Zusammenschaltung 3708 auf eine drahtbasierte mehrspurige Kommunikationsverbindung beziehen, die durch Systeme zum Skalieren verwendet wird und eine oder mehrere PPUs 3700 in Kombination mit einer oder mehreren zentralen Verarbeitungseinheiten („CPUs“) beinhaltet und die Cache-Kohärenz zwischen PPUs 3700 und CPUs sowie CPU-Mastering unterstützt. In mindestens einer Ausführungsform werden Daten und/oder Befehle durch die Hochgeschwindigkeits-GPU-Zusammenschaltung 3708 über den Hub 3716 zu/von anderen Einheiten der PPU 3700 übertragen, wie etwa einer/einem oder mehreren Kopier-Engines, Videocodierern, Videodecodierern, Leistungsverwaltungseinheiten und anderen Komponenten, die in **Fig. 37** möglicherweise nicht explizit veranschaulicht sind.

[0328] In mindestens einer Ausführungsform ist die E/A-Einheit 3706 dazu konfiguriert, Kommunikationen (z. B. Befehle, Daten) von einem Host-Prozessor (in **Fig. 37** nicht veranschaulicht) über den Systembus 3702 zu übertragen und zu empfangen. In mindestens einer Ausführungsform kommuniziert die E/A-Einheit 3706 mit dem Host-Prozessor direkt über den Systembus 3702 oder durch eine oder mehrere Zwischenvorrichtungen, wie etwa eine Speicherbrücke. In mindestens einer Ausführungsform kann die E/A-Einheit 3706 über den Systembus 3702 mit einem oder mehreren anderen Prozessoren kommunizieren, wie etwa einer oder mehreren der PPUs 3700. In mindestens einer Ausführungsform implementiert die E/A-Einheit 3706 eine Peripheral-Component-Interconnect-Express- („PCIe“-)Schnittstelle für die Kommunikation über einen PCIe-Bus. In mindestens einer Ausführungsform implementiert die E/A-Einheit 3706 Schnittstellen zum Kommunizieren mit externen Vorrichtungen.

[0329] In mindestens einer Ausführungsform decodiert die E/A-Einheit 3706 über den Systembus 3702 empfangene Pakete. In mindestens einer Ausführungsform stellen mindestens einige Pakete Befehle dar, die dazu konfiguriert sind, die PPU 3700 dazu zu veranlassen, verschiedene Operationen durchzuführen. In mindestens einer Ausführungsform überträgt die E/A-Einheit 3706 decodierte Befehle an verschiedene andere Einheiten der PPU 3700, wie durch Befehle vorgegeben. In mindestens einer Ausführungsform werden Befehle an die Frontend-Einheit 3710 übertragen und/oder an den Hub 3716 oder andere Einheiten der PPU 3700 übertragen, wie etwa eine oder mehrere Kopier-Engines, einen Videocodierer, einen Videodecodierer, eine Leistungsverwaltungseinheit usw. (in **Fig. 37** nicht explizit veranschaulicht). In mindestens einer Ausführungsform ist die E/A-Einheit 3706 dazu konfiguriert, Kommunikation zwischen und unter verschiedenen logischen Einheiten der PPU 3700 zu leiten.

[0330] In mindestens einer Ausführungsform codiert ein durch den Host-Prozessor ausgeführtes Programm einen Befehlsstrom in einem Puffer, der der PPU 3700 Arbeitslasten zur Verarbeitung bereitstellt. In mindestens einer Ausführungsform umfasst eine Arbeitslast Anweisungen und Daten, die durch diese Anweisungen verarbeitet werden sollen. In mindestens einer Ausführungsform ist der Puffer eine Region in einem Speicher, auf die sowohl der Host-Prozessor als auch die PPU 3700 zugreifen können (z. B. Lesen/Schreiben) - eine Host-Schnittstelleneinheit kann dazu konfiguriert sein, auf den Puffer in einem Systemspeicher zuzugreifen, der mit dem Systembus 3702 verbunden ist, und zwar über Speicheranforderungen, die über den Systembus 3702 von der E/A-Einheit 3706 übertragen werden. In mindestens einer Ausführungsform schreibt der Host-Prozessor den Befehlsstrom in den Puffer und überträgt dann einen Zeiger für den Start des Befehlsstroms an die PPU 3700, sodass die Frontend-Einheit 3710 Zeiger für einen oder mehrere Befehlsströme empfängt und einen oder mehrere Befehlsströme verwaltet, indem sie Befehle aus den Befehlsströmen liest und Befehle an verschiedene Einheiten der PPU 3700 weiterleitet.

[0331] In mindestens einer Ausführungsform ist die Frontend-Einheit 3710 an die Planer-Einheit 3712 gekoppelt, die verschiedene GPCs 3718 zum Verarbeiten von Aufgaben konfiguriert, die durch einen oder mehrere Befehlsströme definiert sind. In mindestens einer Ausführungsform ist die Planer-Einheit 3712 dazu konfiguriert, Zustandsinformationen in Bezug auf verschiedene durch die Planer-Einheit 3712 verwaltete Aufgaben zu verfolgen, wobei die Zustandsinformationen angeben können, welchem der GPCs 3718 eine Aufgabe zugewiesen ist, ob die Aufgabe Task aktiv oder inaktiv ist, welche Prioritätsstufe der Aufgabe zugeordnet ist usw. In mindestens einer Ausführungsform verwaltet die Planer-Einheit 3712 die Ausführung einer Vielzahl von Aufgaben auf einem oder mehreren GPCs 3718.

[0332] In mindestens einer Ausführungsform ist die Planer-Einheit 3712 an die Arbeitsverteilungseinheit 3714 gekoppelt, die dazu konfiguriert ist, Aufgaben zur Ausführung auf den GPCs 3718 zuzuteilen. In mindestens einer Ausführungsform verfolgt die Arbeitsverteilungseinheit 3714 eine Anzahl geplanter Aufgaben nach, die von der Planer-Einheit 3712 empfangen wurde, und die Arbeitsverteilungseinheit 3714 verwaltet einen Pool ausstehender Aufgaben und einen Pool aktiver Aufgaben für jeden der GPCs 3718. In mindestens einer Ausführungsform umfasst der Pool ausstehender Aufgaben eine Anzahl von Schlitzen (z. B. 32 Schlitze), die Aufgaben enthalten, die zur Verarbeitung durch einen konkreten GPC 3718 zugewiesen sind; der Pool aktiver Aufgaben kann eine Anzahl von Schlitzen (z. B. 4 Schlitze) für Aufgaben umfassen, die aktiv von den GPCs 3718 verarbeitet werden, sodass, wenn einer der GPCs 3718 die Ausführung einer Aufgabe abschließt, diese Aufgabe aus dem Pool aktiver Aufgaben für den GPC 3718 entfernt wird und eine der anderen Aufgaben aus dem Pool ausstehender Aufgaben ausgewählt und zur Ausführung auf dem GPC 3718 geplant wird. In mindestens einer Ausführungsform wird, wenn eine aktive Aufgabe auf dem GPC 3718 inaktiv ist, wie etwa während sie darauf wartet, dass eine Datenabhängigkeit aufgelöst wird, dann die aktive Aufgabe aus dem GPC 3718 entfernt und in den Pool ausstehender Aufgaben zurückgeführt, während eine andere Aufgabe im Pool ausstehender Aufgaben ausgewählt und zur Ausführung auf dem GPC 3718 geplant wird.

[0333] In mindestens einer Ausführungsform kommuniziert die Arbeitsverteilungseinheit 3714 mit einem oder mehreren GPCs 3718 über die XBar 3720. In mindestens einer Ausführungsform ist die XBar 3720 ein Zusammenschaltungsnetz, das viele Einheiten der PPU 3700 an andere Einheiten der PPU 3700 koppelt und dazu konfiguriert sein kann, die Arbeitsverteilungseinheit 3714 an einen konkreten GPC 3718 zu koppeln. In mindestens einer Ausführungsform können auch eine oder mehrere andere Einheiten der PPU 3700 über den Hub 3716 mit der XBar 3720 verbunden sein.

[0334] In mindestens einer Ausführungsform werden Aufgaben durch die Planer-Einheit 3712 verwaltet und durch die Arbeitsverteilungseinheit 3714 einem der GPCs 3718 zugeteilt. Der GPC 3718 ist dazu konfiguriert, die Aufgabe zu verarbeiten und Ergebnisse zu erzeugen. In mindestens einer Ausführungsform können die Ergebnisse durch andere Aufgaben innerhalb des GPC 3718 verbraucht, über die XBar 3720 an einen anderen GPC 3718 geleitet oder in dem Speicher 3704 gespeichert werden. In mindestens einer Ausführungsform können die Ergebnisse über die Partitionseinheiten 3722, die eine Speicherschnittstelle zum Lesen und Schreiben von Daten in den/aus dem Speicher 3704 implementieren, in den Speicher 3704 geschrieben werden. In mindestens einer Ausführungsform können die Ergebnisse über die Hochgeschwindigkeits-GPU-Zusammenschaltung 3708 an eine andere PPU 3700 oder CPU übertragen werden. In mindestens einer Ausführungsform beinhaltet die PPU 3700 ohne Einschränkung eine Anzahl U von Partitionseinheiten 3722, die gleich der Anzahl von separaten und unterschiedlichen Speichervorrichtungen 3704 ist, die mit der PPU 3700 verbunden sind. In mindestens einer Ausführungsform wird die Partitionseinheit 3722 nachstehend in Verbindung mit **Fig. 39** detaillierter beschrieben.

[0335] In mindestens einer Ausführungsform führt ein Host-Prozessor einen Treiberkernel aus, der eine Anwendungsprogrammierschnittstelle („API“) implementiert, die es einer oder mehreren auf dem Host-Prozessor ausgeführten Anwendungen ermöglicht, Operationen zur Ausführung auf der PPU 3700 zu planen. In mindestens einer Ausführungsform werden mehrere Rechenanwendungen simultan durch die PPU 3700 ausgeführt und die PPU 3700 stellt Isolierung, Dienstgüte (quality of service - „QoS“) und unabhängige Adressräume für mehrere Rechenanwendungen bereit. In mindestens einer Ausführungsform erzeugt eine Anwendung Anweisungen (z. B. in Form von API-Aufrufen), die den Treiberkernel dazu veranlassen, eine oder mehrere Tasks zur Ausführung durch die PPU 3700 zu erzeugen, und der Treiberkernel gibt Tasks an einen oder mehrere Datenströme aus, die von der PPU 3700 verarbeitet werden. In mindestens einer Ausführungsform umfasst jede Aufgabe eine oder mehrere Gruppen von verwandten Threads, die als Warp bezeichnet werden können. In mindestens einer Ausführungsform umfasst ein Warp eine Vielzahl von zugehörigen Threads (z. B. 32 Threads), die parallel ausgeführt werden können. In mindestens einer Ausführungsform können sich kooperierende Threads auf eine Vielzahl von Threads beziehen, die Anweisungen

zum Durchführen von Aufgaben beinhalten und die Daten durch einen gemeinsam genutzten Speicher austauschen. In mindestens einer Ausführungsform werden Threads und kooperierende Threads gemäß mindestens einer Ausführungsform in Verbindung mit **Fig. 39** detaillierter beschrieben.

[0336] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform wird der Deep-Learning-Anwendungsprozessor verwendet, um ein Modell des maschinellen Lernens, wie etwa ein neuronales Netz, zu trainieren, um der PPU 3700 bereitgestellte Informationen vorherzusagen oder abzuleiten. In mindestens einer Ausführungsform wird die PPU 3700 verwendet, um Informationen auf der Grundlage eines trainierten Modells des maschinellen Lernens (z. B. eines neuronalen Netzes) abzuleiten oder vorherzusagen, das von einem anderen Prozessor oder System oder von der PPU 3700 trainiert wurde. In mindestens einer Ausführungsform kann die PPU 3700 verwendet werden, um einen oder mehrere der in dieser Schrift beschriebenen Anwendungsfälle für neuronale Netze durchzuführen.

[0337] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0338] **Fig. 38** veranschaulicht ein Universalverarbeitungscluster („GPC“) 3800 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform handelt es sich bei dem GPC 3800 um den GPC 3718 aus **Fig. 37**. In mindestens einer Ausführungsform beinhaltet jeder GPC 3800 ohne Einschränkung eine Anzahl von Hardware-Einheiten zur Verarbeitung von Aufgaben und beinhaltet jeder GPC 3800 ohne Einschränkung einen Pipeline-Manager 3802, eine Vor-Rasteroperationeneinheit (pre-raster operation - „PROP“) 3804, eine Raster-Engine 3808, eine Arbeitsverteilungskreuzschiene (work distribution crossbar - „WDX“) 3816, eine Speicherverwaltungseinheit („MMU“) 3818, ein oder mehrere Datenverarbeitungscluster (Data Processing Clusters - „DPCs“) 3806 und eine beliebige geeignete Kombination von Teilen.

[0339] In mindestens einer Ausführungsform wird der Betrieb des GPC 3800 durch den Pipeline-Manager 3802 gesteuert. In mindestens einer Ausführungsform verwaltet der Pipeline-Manager 3802 die Konfiguration eines oder mehrerer DPCs 3806 für die Verarbeitung von Aufgaben, die dem GPC 3800 zugewiesen sind. In mindestens einer Ausführungsform konfiguriert der Pipeline-Manager 3802 mindestens einen von einem oder mehreren DPCs 3806 dazu, mindestens einen Abschnitt einer Grafik-Rendering-Pipeline zu implementieren. In mindestens einer Ausführungsform ist der DPC 3806 dazu konfiguriert, ein Vertex-Shader-Programm auf einem programmierbaren Streaming-Multiprozessor (streaming multi-processor - „SM“) 3814 auszuführen. In mindestens einer Ausführungsform ist der Pipeline-Manager 3802 dazu konfiguriert, die von einer Arbeitsverteilungseinheit empfangenen Pakete in mindestens einer Ausführungsform an die entsprechenden logischen Einheiten innerhalb des GPC 3800 zu leiten und einige Pakete können an Festfunktions-Hardwareeinheiten in der PROP 3804 und/oder die Raster-Engine 3808 geleitet werden, während andere Pakete an die DPCs 3806 zur Verarbeitung durch eine Primitiv-Engine 3812 oder den SM 3814 geleitet werden können. In mindestens einer Ausführungsform konfiguriert der Pipeline-Manager 3802 mindestens einen der DPCs 3806 zum Implementieren eines Modells eines neuronalen Netzes und/oder einer Rechenpipeline.

[0340] In mindestens einer Ausführungsform ist die PROP-Einheit 3804 dazu konfiguriert, in mindestens einer Ausführungsform die von der Raster-Engine 3808 und den DPCs 3806 erzeugten Daten an eine Rasteroperations-(„ROP“-)Einheit in der Partitionseinheit 3722 zu leiten, die vorstehend in Verbindung mit **Fig. 37**. In mindestens einer Ausführungsform ist die PROP-Einheit 3804 dazu konfiguriert, Optimierungen für die Farbmischung durchzuführen, Pixeldaten zu organisieren, Adressübersetzungen durchzuführen usw. In mindestens einer Ausführungsform beinhaltet die Raster-Engine 3808 ohne Einschränkung eine Anzahl von Festfunktions-Hardwareeinheiten, die dazu konfiguriert sind, verschiedene Rasteroperationen durchzuführen, und in mindestens einer Ausführungsform beinhaltet die Raster-Engine 3808 ohne Einschränkung eine Einrichtungs-Engine, eine Grobraster-Engine, eine Culling-Engine, eine Clipping-Engine, eine Feinraster-Engine, eine Kachelverschmelzungs-Engine und eine beliebige geeignete Kombination davon. In mindestens einer Ausführungsform empfängt die Einrichtungs-Engine transformierte Vertices und erzeugt Ebenengleichungen, die dem durch die Vertices definierten geometrischen Primitiv zugeordnet sind; die Ebenengleichungen werden an die Grobraster-Engine übertragen, um Abdeckungsinformationen (z. B. eine x-, y-Abdeckungsmaske für eine Kachel) für das Primitiv zu erzeugen; die Ausgabe der Grobraster-Engine wird an die Culling-Engine übertragen, in der Fragmente, die dem Primitiv zugeordnet sind und einen z-Test nicht bestehen, Culling unterzogen werden und an eine Clipping-Engine übertragen werden, in der Frag-

mente, die außerhalb eines Sichtkegelstumpfes liegen, Clipping unterzogen werden. In mindestens einer Ausführungsform werden Fragmente, die das Clipping und Culling überstehen, an die Feinraster-Engine übergeben, um Attribute für Pixelfragmente basierend auf Ebenengleichungen zu erzeugen, die von der Einrichtungs-Engine erzeugt werden. In mindestens einer Ausführungsform umfasst die Ausgabe der Raster-Engine 3808 Fragmente, die von einer beliebigen geeigneten Einheit, wie etwa von einem in dem DPC 3806 implementierten Fragment-Shader, verarbeitet werden.

[0341] In mindestens einer Ausführungsform umfasst jeder DPC 3806, der in dem GPC 3800 enthalten ist, ohne Einschränkung eine M-Pipe-Steuerung (M-Pipe Controller - „MPC“) 3810; eine Primitiv-Engine 3812; einen oder mehrere SMs 3814 und eine beliebige geeignete Kombination davon. In mindestens einer Ausführungsform steuert die MPC 3810 den Betrieb des DPC 3806 und leitet von dem Pipeline-Manager 3802 empfangene Pakete an die entsprechenden Einheiten in dem DPC 3806. In mindestens einer Ausführungsform werden Pakete, die einem Vertex zugeordnet sind, an die Primitiv-Engine 3812 geleitet, die dazu konfiguriert ist, Vertexattribute, die dem Vertex zugeordnet sind, aus dem Speicher abzurufen; im Gegensatz dazu können Pakete, die einem Shader-Programm zugeordnet sind, an den SM 3814 übertragen werden.

[0342] In mindestens einer Ausführungsform umfasst der SM 3814 ohne Einschränkung einen programmierbaren Streaming-Prozessor, der dazu konfiguriert ist, Aufgaben zu verarbeiten, die durch eine Anzahl von Threads dargestellt sind. In mindestens einer Ausführungsform weist der SM 3814 mehrere Threads auf und ist dazu konfiguriert, eine Vielzahl von Threads (z.B. 32 Threads) aus einer konkreten Gruppe von Threads gleichzeitig auszuführen und eine Single-Instruction-Multiple-Data- („SIMD“-) Architektur zu implementieren, bei der jeder Thread in einer Gruppe von Threads (z. B. ein Warp) dazu konfiguriert ist, einen anderen Datensatz auf Grundlage des gleichen Anweisungssatzes zu verarbeiten. In mindestens einer Ausführungsform führen alle Threads in einer Gruppe von Threads dieselben Anweisungen aus. In mindestens einer Ausführungsform implementiert der SM 3814 eine Single-Instruction-Multiple-Thread- („SIMT“-) Architektur, bei der jeder Thread in einer Gruppe von Threads dazu konfiguriert ist, einen anderen Datensatz auf Grundlage desselben Anweisungssatzes zu verarbeiten, wobei jedoch die einzelnen Threads in der Gruppe von Threads während der Ausführung divergieren dürfen. In mindestens einer Ausführungsform werden ein Programmzähler, ein Aufrufstapel und ein Ausführungszustand für jeden Warp beibehalten, wodurch Nebenläufigkeit zwischen Warps und serielle Ausführung innerhalb von Warps ermöglicht wird, wenn Threads innerhalb eines Warps divergieren. In einer anderen Ausführungsform werden ein Programmzähler, ein Aufrufstapel und ein Ausführungszustand für jeden einzelnen Thread beibehalten, wodurch gleichwertige Nebenläufigkeit zwischen allen Threads, innerhalb und zwischen Warps, ermöglicht wird. In mindestens einer Ausführungsform wird der Ausführungszustand für jeden einzelnen Thread beibehalten und können Threads, die dieselben Anweisungen ausführen, zur besseren Effizienz konvergieren und parallel ausgeführt werden. Mindestens eine Ausführungsform des SM 3814 wird nachstehend detaillierter beschrieben.

[0343] In mindestens einer Ausführungsform stellt die MMU 3818 eine Schnittstelle zwischen dem GPC 3800 und der Speicherpartitionseinheit (z. B. der Partitionseinheit 3722 aus **Fig. 37**) bereit und stellt die MMU 3818 die Übersetzung virtueller Adressen in physische Adressen, den Speicherschutz und die Vermittlung von Speicheranforderungen bereit. In mindestens einer Ausführungsform stellt die MMU 3818 einen oder mehrere Adressübersetzungspuffer („TLBs“) zum Durchführen der Übersetzung von virtuellen Adressen in physische Adressen in Speicher bereit.

[0344] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform wird der Deep-Learning-Anwendungsprozessor verwendet, um ein Modell des maschinellen Lernens, wie etwa ein neuronales Netz, zu trainieren, um dem GPC 3800 bereitgestellte Informationen vorherzusagen oder abzuleiten. In mindestens einer Ausführungsform wird der GPC 3800 verwendet, um Informationen auf Grundlage eines trainierten Modells des maschinellen Lernens (z. B. eines neuronalen Netzes) abzuleiten oder vorherzusagen, das durch einen anderen Prozessor oder ein anderes System oder durch den GPC 3800 trainiert wurde. In mindestens einer Ausführungsform kann der GPC 3800 verwendet werden, um einen oder mehrere der in dieser Schrift beschriebenen Anwendungsfälle für neuronale Netze durchzuführen.

[0345] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0346] Fig. 39 veranschaulicht eine Speicherpartitionseinheit 3900 einer Parallelverarbeitungseinheit („PPU“) gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform beinhaltet die Speicherpartitionseinheit 3900 ohne Einschränkung eine Rasteroperations- („ROP“-)Einheit 3902, einen Cache 3904 der Ebene zwei („L2“-), eine Speicherschnittstelle 3906 und eine beliebige geeignete Kombination davon. In mindestens einer Ausführungsform ist die Speicherschnittstelle 3906 an Speicher gekoppelt. In mindestens einer Ausführungsform kann die Speicherschnittstelle 3906 32-, 64-, 128-, 1024-Bit-Datenbusse oder ähnliche Implementierungen für Hochgeschwindigkeitsdatenübertragung implementieren. In mindestens einer Ausführungsform beinhaltet die PPU U Speicherschnittstellen 3906, eine Speicherschnittstelle 3906 pro Paar von Partitionseinheiten 3900, wobei jedes Paar von Partitionseinheiten 3900 mit einer entsprechenden Speichervorrichtung verbunden ist. In mindestens einer Ausführungsform kann die PPU zum Beispiel mit bis zu Y Speichervorrichtungen verbunden sein, wie etwa mit Speicherstapeln mit hoher Bandbreite oder mit einem synchronen dynamischen Direktzugriffsspeicher mit Grafik-Double-Data-Rate, Version 5 („GDDR5-SDRAM“).

[0347] In mindestens einer Ausführungsform implementiert die Speicherschnittstelle 3906 eine Speicherschnittstelle mit Speicher mit hoher Bandbreite der zweiten Generation (high bandwidth memory second generation - „HBM2“) und ist Y gleich der Hälfte von U. In mindestens einer Ausführungsform befinden sich HBM2-Speicherstapel in demselben physischen Gehäuse wie die PPU, was im Vergleich zu herkömmlichen GDDR5-SDRAM-Systemen erhebliche Leistungs- und Flächeneinsparungen bereitstellt. In mindestens einer Ausführungsform beinhaltet jeder HBM2-Stapel ohne Einschränkung vier Speicherchips und ist Y gleich 4, wobei jeder HBM2-Stapel zwei 128-Bit-Kanäle pro Chip für insgesamt 8 Kanäle und eine Datenbusbreite von 1024 Bit beinhaltet. In mindestens einer Ausführungsform unterstützt der Speicher Single-Error-Correcting-Double-Error-Detecting- („SECCED“-)Fehlerkorrekturcode (Error Correction Code - „ECC“) zum Schützen von Daten. In mindestens einer Ausführungsform stellt der ECC eine höhere Zuverlässigkeit für Rechenanwendungen bereit, die empfindlich auf Datenkorruption reagieren.

[0348] In mindestens einer Ausführungsform implementiert die PPU eine Speicherhierarchie mit mehreren Ebenen. In mindestens einer Ausführungsform unterstützt die Speicherpartitionseinheit 3900 einen vereinheitlichten Speicher, um einen einzelnen vereinheitlichten virtuellen Adressraum für den Speicher der zentralen Verarbeitungseinheit („CPU“) und der PPU bereitzustellen, wodurch die gemeinsame Nutzung von Daten zwischen virtuellen Speichersystemen ermöglicht wird. In mindestens einer Ausführungsform wird die Häufigkeit von Zugriffen von einer PPU auf Speicher, der sich auf anderen Prozessoren befindet, verfolgt, um sicherzustellen, dass Speicherseiten in den physischen Speicher der PPU verschoben werden, die häufiger auf Seiten zugreift. In mindestens einer Ausführungsform unterstützt die Hochgeschwindigkeits-GPU-Zusammenschaltung 3708 Adressübersetzungsdienste, die es der PPU ermöglichen, direkt auf die Seitentabellen der CPU zuzugreifen, und der PPU vollen Zugriff auf den CPU-Speicher bereitzustellen.

[0349] In mindestens einer Ausführungsform übertragen Kopier-Engines Daten zwischen mehreren PPUs oder zwischen PPUs und CPUs. In mindestens einer Ausführungsform können Kopier-Engines Seitenfehler für Adressen erzeugen, die nicht in Seitentabellen zugeordnet sind, und die Speicherpartitionseinheit 3900 bedient dann Seitenfehler, indem sie die Adressen in der Seitentabelle zuordnet, woraufhin die Kopier-Engine die Übertragung durchführt. In mindestens einer Ausführungsform ist der Speicher für mehrere Kopier-Engine-Operationen zwischen mehreren Prozessoren gepinnt (d. h. nicht auslagerbar), was den verfügbaren Speicher erheblich reduziert. In mindestens einer Ausführungsform können bei Hardware-Seitenfehlern Adressen an Kopier-Engines übergeben werden, ohne Rücksicht darauf, ob Speicherseiten speicherresident sind und der Kopierprozess transparent ist.

[0350] Daten aus dem Speicher 3704 aus Fig. 37 oder einem anderen Systemspeicher werden gemäß mindestens einer Ausführungsform durch die Speicherpartitionseinheit 3900 abgerufen und in L2-Cache 3904 gespeichert, der sich auf dem Chip befindet und von verschiedenen GPCs gemeinsam genutzt wird. Jede Speicherpartitionseinheit 3900 beinhaltet in mindestens einer Ausführungsform ohne Einschränkung mindestens einen Abschnitt des L2-Caches, der einer entsprechenden Speichervorrichtung zugeordnet ist. In mindestens einer Ausführungsform sind die Caches der unteren Ebene in verschiedenen Einheiten innerhalb der GPCs implementiert. In mindestens einer Ausführungsform kann jeder der SMs 3814 einen Cache der Ebene eins („L1“) implementieren, wobei der L1-Cache ein privater Speicher ist, der für einen konkreten SM 3814 dediziert ist, und Daten aus dem L2-Cache 3904 abgerufen und in jedem der L1-Caches zur Verarbeitung in funktionellen Einheiten der SMs 3814 gespeichert werden. In mindestens einer Ausführungsform ist der L2-Cache 3904 an die Speicherschnittstelle 3906 und die XBar 3720 gekoppelt.

[0351] In mindestens einer Ausführungsform führt die ROP-Einheit 3902 Grafikrasteroperationen durch, die mit der Pixelfarbe in Bezug stehen, wie etwa Farbkompromierung, Pixel-Blending und andere. In mindestens einer Ausführungsform implementiert die ROP-Einheit 3902 die Tiefenprüfung in Verbindung mit der Raster-Engine 3808, wobei sie eine Tiefe für eine Abtastposition, die einem Pixelfragment zugeordnet ist, von der Culling-Engine der Raster-Engine 3808 empfängt. In mindestens einer Ausführungsform wird die Tiefe gegen eine entsprechende Tiefe in einem Tiefenpuffer für eine dem Fragment zugeordnete Abtastposition geprüft. In mindestens einer Ausführungsform, wenn das Fragment die Tiefenprüfung für die Abtastposition besteht, aktualisiert die ROP-Einheit 3902 dann den Tiefenpuffer und überträgt ein Ergebnis der Tiefenprüfung an die Raster-Engine 3808. Es versteht sich, dass sich die Anzahl der Partitionseinheiten 3900 von der Anzahl der GPCs unterscheiden kann, und daher kann jede ROP-Einheit 3902 in mindestens einer Ausführungsform mit jedem der GPCs gekoppelt sein. In mindestens einer Ausführungsform verfolgt die ROP-Einheit 3902 die von verschiedenen GPCs empfangenen Pakete und bestimmt, ob ein durch die ROP-Einheit 3902 erzeugtes Ergebnis an die XBar 3720 geleitet werden soll.

[0352] **Fig. 40** veranschaulicht einen Streaming-Multiprozessor („SM“) 4000 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist der SM 4000 der SM 3814 aus **Fig. 38**. In mindestens einer Ausführungsform beinhaltet der SM 4000 ohne Einschränkung einen Anweisungs-Cache 4002; eine oder mehrere Planer-Einheiten 4004; eine Registerdatei 4008; einen oder mehrere Verarbeitungskerne („Kerne“) 4010; eine oder mehrere Spezialfunktionseinheiten (special function units - „SFUs“) 4012; eine oder mehrere Lade-/Speichereinheiten (load/store units - „LSUs“) 4014; ein Zusammenschaltungsnetz 4016; einen gemeinsam genutzten Speicher/Cache der Ebene eins („L1“) 4018 und eine beliebige geeignete Kombination davon. In mindestens einer Ausführungsform teilt eine Arbeitsverteilungseinheit Aufgaben zur Ausführung auf Universalverarbeitungsclustern („GPCs“) von Parallelverarbeitungseinheiten („PPUs“) zu und wird jede Aufgabe einem konkreten Datenverarbeitungscluster („DPC“) innerhalb eines GPCs zugewiesen und wird, wenn die Aufgabe einem Shader-Programm zugeordnet ist, die Aufgabe einem der SMs 4000 zugewiesen. In mindestens einer Ausführungsform empfängt die Planer-Einheit 4004 Aufgaben von der Arbeitsverteilungseinheit und verwaltet die Anweisungsplanung für einen oder mehrere Thread-Blöcke, die dem SM 4000 zugewiesen sind. In mindestens einer Ausführungsform plant die Planer-Einheit 4004 Thread-Blöcke für die Ausführung als Warps aus parallelen Threads, wobei jedem Thread-Block mindestens ein Warp zugewiesen ist. In mindestens einer Ausführungsform führt jeder Warp Threads aus. In mindestens einer Ausführungsform verwaltet die Planer-Einheit 4004 eine Vielzahl von unterschiedlichen Thread-Blöcken, indem sie unterschiedlichen Thread-Blöcken Warps zuweist und dann während jedes Taktzyklus Anweisungen aus einer Vielzahl von unterschiedlichen kooperativen Gruppen verschiedenen funktionellen Einheiten (z. B. Verarbeitungskernen 4010, SFUs 4012 und LSUs 4014) zuteilt.

[0353] In mindestens einer Ausführungsform können sich kooperative Gruppen auf ein Programmiermodell zum Organisieren von Gruppen kommunizierender Threads beziehen, das es Entwicklern erlaubt, die Granularität auszudrücken, mit der Threads kommunizieren, und so den Ausdruck satterer, effizienterer paralleler Dekompositionen ermöglicht. In mindestens einer Ausführungsform unterstützen kooperative Launch-APIs die Synchronisation zwischen Thread-Blöcken zur Ausführung paralleler Algorithmen. In mindestens einer Ausführungsform bieten Anwendungen herkömmlicher Programmiermodelle ein einzelnes, einfaches Konstrukt für die Synchronisation kooperierender Threads: eine Sperre über alle Threads eines Thread-Blocks (z. B. die Funktion `syncthreads()`). In mindestens einer Ausführungsform können Programmierer jedoch Gruppen von Threads mit einer geringeren als Thread-Block-Granularität definieren und innerhalb definierter Gruppen synchronisieren, um eine höhere Rechenleistung, Ausgestaltungsflexibilität und Software-Wiederverwendung in Form von kollektiven gruppenweiten Funktionsschnittstellen zu ermöglichen. In mindestens einer Ausführungsform ermöglichen es kooperative Gruppen Programmierern, Gruppen von Threads explizit mit Teilblock- (d. h. so klein wie ein einzelner Thread) und Mehrblock-Granularität zu definieren und kollektive Operationen, wie etwa Synchronisation, an Threads in einer kooperativen Gruppe durchzuführen. In mindestens einer Ausführungsform unterstützt das Programmiermodell eine saubere Komposition über Software-Grenzen hinweg, sodass sich Bibliotheken und Nutzenfunktionen innerhalb ihres lokalen Kontexts sicher synchronisieren können, ohne Annahmen über Konvergenz treffen zu müssen. In mindestens einer Ausführungsform ermöglichen Primitive der kooperativen Gruppen neue Muster der kooperativen Parallelität, einschließlich, aber nicht beschränkt auf, Produzent-Konsument-Parallelität, opportunistischer Parallelität und globaler Synchronisation über ein gesamtes Gitter von Thread-Blöcken.

[0354] In mindestens einer Ausführungsform ist eine Zuteilungseinheit 4006 dazu konfiguriert, Anweisungen an eine oder mehrere der funktionellen Einheiten zu übertragen und die Planer-Einheit 4004 beinhaltet ohne Einschränkung zwei Zuteilungseinheiten 4006, die ermöglichen, dass zwei unterschiedliche Anweisungen aus demselben Warp während jedes Taktzyklus versendet werden. In mindestens einer Ausführungsform

beinhaltet jede Planer-Einheit 4004 eine einzelne Zuteilungseinheit 4006 oder zusätzliche Zuteilungseinheiten 4006.

[0355] In mindestens einer Ausführungsform beinhaltet jeder SM 4000 in mindestens einer Ausführungsform ohne Einschränkung die Registerdatei 4008, die einen Satz von Registern für funktionelle Einheiten des SM 4000 bereitstellt. In mindestens einer Ausführungsform ist die Registerdatei 4008 derart zwischen den einzelnen funktionellen Einheiten aufgeteilt, dass jeder funktionellen Einheit ein dedizierter Abschnitt der Registerdatei 4008 zugewiesen ist. In mindestens einer Ausführungsform ist die Registerdatei 4008 auf unterschiedliche Warps aufgeteilt, die durch den SM 4000 ausgeführt werden, und die Registerdatei 4008 stellt temporären Datenspeicher für Operanden bereit, die mit Datenpfaden von funktionellen Einheiten verbunden sind. In mindestens einer Ausführungsform umfasst jeder SM 4000 ohne Einschränkung eine Vielzahl von L Verarbeitungskernen 4010. In mindestens einer Ausführungsform beinhaltet der SM 4000 ohne Einschränkung eine große Anzahl (z. B. 128 oder mehr) von unterschiedlichen Verarbeitungskernen 4010. In mindestens einer Ausführungsform beinhaltet jeder Verarbeitungskern 4010 in mindestens einer Ausführungsform ohne Einschränkung eine Vollpipelineverarbeitungseinheit mit einfacher Genauigkeit, doppelter Genauigkeit und/oder gemischter Genauigkeit, die ohne Einschränkung eine arithmetisch-logische Einheit für Gleitkommazahlen und eine arithmetisch-logische Einheit für Integer beinhaltet. In mindestens einer Ausführungsform implementieren die arithmetisch-logischen Einheiten für Gleitkommazahlen den Standard IEEE 754-2008 für Gleitkommaarithmetik. In mindestens einer Ausführungsform beinhalten die Verarbeitungskerne 4010 ohne Einschränkung 64 Gleitkommakerne mit einfacher Genauigkeit (32 Bit), 64 Integerkerne, 32 Gleitkommakerne mit doppelter Genauigkeit (64 Bit) und 8 Tensorkerne.

[0356] Tensorkerne sind gemäß mindestens einer Ausführungsform dazu konfiguriert, Matrixoperationen durchzuführen. In mindestens einer Ausführungsform sind ein oder mehrere Tensorkerne in den Verarbeitungskernen 4010 enthalten. In mindestens einer Ausführungsform sind die Tensorkerne dazu konfiguriert, Deep-Learning-Matrixarithmetik durchzuführen, wie etwa Faltungsoperationen zum Trainieren und Inferenzieren neuronaler Netze. In mindestens einer Ausführungsform arbeitet jeder Tensorkern an einer 4x4-Matrix und er führt eine Matrixmultiplikations- und -akkumulationsoperation $D = A \times B + C$ durch, wobei A, B, C und D 4x4-Matrizen sind.

[0357] In mindestens einer Ausführungsform sind die Matrixmultiplikationseingaben A und B 16-Bit-Gleitkommamatrizen und sind die Akkumulationsmatrizen C und D 16-Bit-Gleitkomma- oder 32-Bit-Gleitkommamatrizen. In mindestens einer Ausführungsform arbeiten die Tensorkerne an 16-Bit-Gleitkommaeingabedaten mit 32-Bit-Gleitkommaakkumulation. In mindestens einer Ausführungsform verwendet die 16-Bit-Gleitkommamultiplikation 64 Operationen und ergibt ein Produkt mit voller Genauigkeit, das dann unter Verwendung von 32-Bit-Gleitkommaaddition mit anderen Zwischenprodukten für eine 4x4x4-Matrixmultiplikation akkumuliert wird. In mindestens einer Ausführungsform werden die Tensorkerne verwendet, um viel größere zweidimensionale oder höherdimensionale Matrixoperationen durchzuführen, die aus diesen kleineren Elementen aufgebaut sind. In mindestens einer Ausführungsform macht eine API, wie etwa eine CUDA 9 C++ API, spezialisierte Matrixlade-, Matrixmultiplikations- und -akkumulations- sowie Matrixspeicheroperationen verfügbar, um die Tensorkerne anhand eines Programms mit CUDA-C++ effizient zu verwenden. In mindestens einer Ausführungsform nimmt die Schnittstelle auf Warp-Ebene auf der CUDA-Ebene Matrizen der Größe 16x16 an, die sich über alle 32 Threads des Warp erstrecken.

[0358] In mindestens einer Ausführungsform umfasst jeder SM 4000 ohne Einschränkung M SFUs 4012, die Spezialfunktionen durchführen (z. B. Attributauswertung, reziproke Quadratwurzel usw.). In mindestens einer Ausführungsform beinhalten die SFUs 4012 ohne Einschränkung eine Baumtraversierungseinheit, die dazu konfiguriert ist, eine hierarchische Baumdatenstruktur zu traversieren. In mindestens einer Ausführungsform beinhalten die SFUs 4012 ohne Einschränkung eine Textureinheit, die dazu konfiguriert ist, Texturkartenfilteroperationen durchzuführen. In mindestens einer Ausführungsform sind die Textureinheiten dazu konfiguriert, Texturkarten (z. B. ein 2D-Array von Texeln) aus Speicher zu laden und Texturkarten abzutasten, um abgetastete Texturwerte zur Verwendung in Shader-Programmen zu erzeugen, die durch den SM 4000 ausgeführt werden. In mindestens einer Ausführungsform werden die Texturkarten in dem gemeinsam genutzten Speicher/L1-Cache 4018 gespeichert. In mindestens einer Ausführungsform implementieren die Textureinheiten Texturoperationen, wie etwa Filteroperationen unter Verwendung von Mip-Karten (z. B. Texturkarten mit variierenden Detailgraden), gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform beinhaltet jeder SM 4000 ohne Einschränkung zwei Textureinheiten.

[0359] Jeder SM 4000 umfasst in mindestens einer Ausführungsform ohne Einschränkung N LSUs 4014, die Lade- und Speicheroperationen zwischen dem gemeinsam genutzten Speicher/L1-Cache 4018 und der

Registerdatei 4008 implementieren. Jeder SM 4000 beinhaltet in mindestens einer Ausführungsform ohne Einschränkung ein Zusammenschaltungsnetz 4016, das jede der funktionellen Einheiten mit der Registerdatei 4008 und der LSU 4014 mit der Registerdatei 4008 und dem gemeinsam genutzten Speicher/L1-Cache 4018 verbindet. In mindestens einer Ausführungsform ist das Zusammenschaltungsnetz 4016 eine Kreuzschiene, die dazu konfiguriert sein kann, beliebige funktionelle Einheiten mit beliebigen Registern in der Registerdatei 4008 zu verbinden und LSUs 4014 mit der Registerdatei 4008 und Speicherorten im gemeinsam genutzten Speicher/L1-Cache 4018 zu verbinden.

[0360] In mindestens einer Ausführungsform ist der gemeinsam genutzte Speicher/L1-Cache 4018 ein Array von chipinternem Speicher, das in mindestens einer Ausführungsform die Datenspeicherung und die Kommunikation zwischen dem SM 4000 und der Primitiv-Engine sowie zwischen Threads in dem SM 4000 ermöglicht. In mindestens einer Ausführungsform umfasst der gemeinsam genutzte Speicher/L1-Cache 4018 ohne Einschränkung eine Speicherkapazität von 128 KB und befindet sich in einem Pfad von dem SM 4000 zu einer Partitionseinheit. In mindestens einer Ausführungsform wird der gemeinsam genutzte Speicher/L1-Cache 4018 in mindestens einer Ausführungsform zum Zwischenspeichern von Lese- und Schreibvorgängen verwendet. In mindestens einer Ausführungsform sind einer oder mehrere von dem gemeinsam genutzten Speicher/L1-Cache 4018, L2-Cache und Speicher Ergänzungsspeicher.

[0361] Das Kombinieren der Funktionalität des Daten-Cache und des gemeinsam genutzten Speichers in einem einzelnen Speicherblock stellt in mindestens einer Ausführungsform eine verbesserte Rechenleistung für beide Typen von Speicherzugriffen bereit. In mindestens einer Ausführungsform wird die Kapazität durch Programme, die keinen gemeinsam genutzten Speicher verwenden, als Cache verwendet oder sie kann so verwendet werden, wie etwa, falls der gemeinsam genutzte Speicher dazu konfiguriert ist, die Hälfte einer Kapazität verwendet Textur- und Lade-/Speicheroperationen die verbleibende Kapazität verwenden können. Die Integration in den gemeinsam genutzten Speicher/L1-Cache 4018 ermöglicht dem gemeinsam genutzten Speicher/L1-Cache 4018 gemäß mindestens einer Ausführungsform als Leitung mit hohem Durchsatz für Streaming-Daten zu fungieren, während simultan Zugriff mit hoher Bandbreite und niedriger Latenz auf häufig wiederverwendete Daten bereitgestellt wird. In mindestens einer Ausführungsform kann bei einer Konfiguration für Universalparallelberechnung eine einfachere Konfiguration im Vergleich zur Grafikverarbeitung verwendet werden. In mindestens einer Ausführungsform werden Festfunktions-Grafikverarbeitungseinheiten umgangen, wodurch ein wesentlich einfacheres Programmiermodell entsteht. Bei der Konfiguration für Universalparallelberechnungen weist die Arbeitsverteilungseinheit in mindestens einer Ausführungsform Blöcke von Threads direkt DPCs zu und verteilt sie an diese. In mindestens einer Ausführungsform führen Threads in einem Block dasselbe Programm aus, wobei eine eindeutige Thread-ID bei der Berechnung verwendet wird, um sicherzustellen, dass jeder Thread eindeutige Ergebnisse erzeugt, wobei der SM 4000 zur Ausführung des Programms und zur Durchführung von Berechnungen, der gemeinsam genutzte Speicher/L1-Cache 4018 zur Kommunikation zwischen den Threads und die LSU 4014 zum Lesen des und Schreiben in den globalen Speicher durch den gemeinsam genutzten Speicher/L1-Cache 4018 und die Speicherpartitions-einheit verwendet werden. Bei der Konfiguration für Universalparallelberechnungen schreibt der SM 4000 in mindestens einer Ausführungsform Befehle, die durch die Planer-Einheit 4004 verwendet werden können, um neue Arbeit in den DPCs zu starten.

[0362] In mindestens einer Ausführungsform ist die PPU in einem Desktop-Computer, einem Laptop-Computer, einem Tablet-Computer, Servern, Supercomputern, einem Smartphone (z. B. einer drahtlosen tragbaren Vorrichtung), einem persönlichen digitalen Assistenten („PDA“), einer Digitalkamera, einem Fahrzeug, einer am Kopf befestigten Anzeige, einer elektronischen tragbaren Vorrichtung und anderen enthalten oder damit gekoppelt. In mindestens einer Ausführungsform ist die PPU auf einem einzelnen Halbleitersubstrat verkörpert. In mindestens einer Ausführungsform ist die PPU in einem System auf einem Chip („SoC“) zusammen mit einer oder mehreren anderen Vorrichtungen enthalten, wie etwa zusätzlichen PPUs, Speicher, einer Reduced-Instruction-Set-Computer- („RISC“-)CPU, einer Speicherverwaltungseinheit („MMU“), einem Digital-Analog-Wandler (digital-to-analog converter - „DAC“) und dergleichen.

[0363] In mindestens einer Ausführungsform kann die PPU auf einer Grafikkarte enthalten sein, die eine oder mehrere Speichervorrichtungen beinhaltet. Eine Grafikkarte kann dazu konfiguriert sein, mit einem PCIe-Steckplatz auf einer Hauptplatine eines Desktop-Computers eine Schnittstelle zu bilden. In mindestens einer Ausführungsform kann die PPU eine integrierte Grafikverarbeitungseinheit („iGPU“) sein, die im Chipsatz der Hauptplatine enthalten ist.

[0364] Die Inferenz- und/oder Trainingslogik 1615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Infe-

renz- und/oder Trainingslogik 1615 sind nachstehend in Verbindung mit den **Fig. 16A** und/oder **Fig. 16B** bereitgestellt. In mindestens einer Ausführungsform wird der Deep-Learning-Anwendungsprozessor verwendet, um ein Modell für maschinelles Lernen, wie etwa ein neuronales Netz, zu trainieren, um dem SM 4000 bereitgestellte Informationen vorherzusagen oder abzuleiten. In mindestens einer Ausführungsform wird der SM 4000 verwendet, um Informationen auf Grundlage eines trainierten Modells für maschinelles Lernen (z. B. eines neuronalen Netzes) abzuleiten oder vorherzusagen, das durch einen anderen Prozessor oder ein anderes System oder durch den SM 4000 trainiert wurde. In mindestens einer Ausführungsform kann der SM 4000 verwendet werden, um einen oder mehrere der in dieser Schrift beschriebenen Anwendungsfälle für neuronale Netze durchzuführen.

[0365] In mindestens einer Ausführungsform können solche Komponenten verwendet werden, um einem oder mehreren Spielern eines Spiels Feedback zu geben, um ihre Leistung beim Spielen des Spiels zu verbessern.

[0366] In mindestens einer Ausführungsform kann sich eine einzelne Halbleiterplattform auf eine einzige einheitliche halbleiterbasierte integrierte Schaltung oder einen solchen Chip beziehen. In mindestens einer Ausführungsform können Mehrchipmodule mit erhöhter Konnektivität verwendet werden, die den chipinternen Betrieb simulieren und wesentliche Verbesserungen gegenüber der Verwendung einer herkömmlichen zentralen Verarbeitungseinheit („CPU“) und einer Busimplementierung bieten. In mindestens einer Ausführungsform können verschiedene Module je nach Wunsch des Benutzers auch separat oder in verschiedenen Kombinationen von Halbleiterplattformen untergebracht sein.

[0367] In mindestens einer Ausführungsform sind Computerprogramme in Form von maschinenlesbarem ausführbarem Code oder Computersteuerlogik-Algorithmen im Hauptspeicher 2004 und/oder im Sekundärspeicher gespeichert. Bei Ausführung durch einen oder mehrere Prozessoren ermöglichen es Computerprogramme dem System 2000, verschiedene Funktionen gemäß mindestens einer Ausführungsform durchzuführen. In mindestens einer Ausführungsform sind der Arbeitsspeicher 2004, der Datenspeicher und/oder ein beliebiger anderer Datenspeicher mögliche Beispiele für computerlesbare Medien. In mindestens einer Ausführungsform kann sich der Sekundärspeicher auf ein(e) beliebige(s) geeignete(s) Speichervorrichtung oder -system beziehen, wie etwa ein Festplattenlaufwerk und/oder ein Wechselspeicherlaufwerk, das ein Diskettenlaufwerk, ein Magnetbandlaufwerk, ein Compact-Disk-Laufwerk, ein Digital-Versatile-Disk- („DVD“-) Laufwerk, eine Aufzeichnungsvorrichtung, einen Universal-Serial-Bus- („USB“-) Flash-Speicher usw. darstellt. In mindestens einer Ausführungsform werden die Architektur und/oder die Funktionalität verschiedener vorhergehender Figuren im Kontext der CPU 2002; des Parallelverarbeitungssystems 2012; einer integrierten Schaltung, die mindestens zu einem Abschnitt der Fähigkeiten sowohl der CPU 2002 als auch des Parallelverarbeitungssystems 2012 in der Lage ist; eines Chipsatzes (z. B. einer Gruppe integrierter Schaltungen, die dazu ausgestaltet ist, als Einheit zur Durchführung zugehöriger Funktionen zu arbeiten und verkauft zu werden usw.); und einer beliebigen geeigneten Kombination aus (einer) integrierten Schaltung(en) implementiert.

[0368] In mindestens einer Ausführungsform werden die Architektur und/oder die Funktionalität verschiedener vorhergehender Figuren im Kontext eines allgemeinen Computersystems, eines Leiterplattensystems, eines für Unterhaltungszwecke dedizierten Spielekonsolensystems, eines anwendungsspezifischen Systems und anderer implementiert. In mindestens einer Ausführungsform kann das Computersystem 2000 die Form eines Desktop-Computers, eines Laptop-Computers, eines Tablet-Computers, von Servern, von Supercomputern, eines Smartphones (z. B. einer drahtlosen tragbaren Vorrichtung), eines persönlichen digitalen Assistenten („PDA“), einer Digitalkamera, eines Fahrzeugs, einer am Kopf befestigten Anzeige, einer tragbaren elektronischen Vorrichtung, einer Mobiltelefonvorrichtung, eines Fernsehgeräts, einer Arbeitsstation, von Spielekonsolen, eines eingebetteten Systems und/oder eines beliebigen anderen Typs von Logik annehmen.

[0369] In mindestens einer Ausführungsform beinhaltet das Parallelverarbeitungssystem 2012 ohne Einschränkung eine Vielzahl von Parallelverarbeitungseinheiten („PPUs“) 2014 und zugeordneten Speichern 2016. In mindestens einer Ausführungsform sind die PPU's 2014 über eine Zusammenschaltung 2018 und einen Switch 2020 oder Multiplexer mit einem Host-Prozessor oder anderen Peripherievorrichtungen verbunden. In mindestens einer Ausführungsform verteilt das Parallelverarbeitungssystem 2012 Rechenaufgaben auf PPU's 2014, die parallelisierbar sein können - zum Beispiel als Teil der Verteilung von Rechenaufgaben auf mehrere Thread-Blöcke einer Grafikverarbeitungseinheit („GPU“). In mindestens einer Ausführungsform wird der Speicher von einigen oder allen PPU's 2014 gemeinsam genutzt und ist er für diese zugänglich (z. B. für Lese- und/oder Schreibzugriff), obwohl ein derartiger gemeinsam genutzter Speicher Einbußen bei der Rechenleistung in Bezug auf die Verwendung von lokalem Speicher und Registern, die in einer PPU 2014

resident sind, mit sich bringen kann. In mindestens einer Ausführungsform wird der Betrieb der PPU's 2014 durch Verwendung eines Befehls wie etwa `syncthreads()` synchronisiert, wobei alle Threads in einem Block (z. B. über mehrere PPU's 2014 hinweg ausgeführt) einen bestimmten Punkt der Codeausführung erreichen müssen, bevor sie fortfahren.

[0370] Andere Variationen liegen innerhalb des Wesens der vorliegenden Offenbarung. Somit können zwar bezüglich der offenbarten Techniken diverse Modifikationen und alternative Konstruktionen vorgenommen werden, bestimmte veranschaulichte Ausführungsformen davon werden jedoch in den Zeichnungen gezeigt und wurden vorangehend ausführlich beschrieben. Allerdings versteht es sich, dass nicht die Absicht verfolgt wird, die Offenbarung auf die konkrete(n) offenbarte(n) Form oder Formen einzuschränken, sondern die Absicht ganz im Gegenteil darin besteht, sämtliche Modifikationen, alternativen Konstruktionen und Äquivalente abzudecken, die in den Geist und Umfang der wie in den beigefügten Ansprüchen definierten Offenbarung fallen.

[0371] Die Verwendung der Ausdrücke „ein“ und „eine“ und „der/die/das“ und ähnlicher Bezeichnungen im Kontext mit der Beschreibung offenbarer Ausführungsformen (insbesondere im Kontext mit den folgenden Ansprüchen) ist so auszulegen, dass sie sowohl den Singular als auch den Plural abdeckt, sofern in dieser Schrift nicht anders angegeben oder durch den Kontext eindeutig widerlegt, und nicht als Definition eines Ausdrucks. Die Ausdrücke „umfassend“, „aufweisend“, „beinhaltend“ und „enthaltend“ sind als offene Ausdrücke auszulegen (d. h. „einschließlich, aber nicht begrenzt auf“), sofern nicht anderweitig angegeben. Der Ausdruck „verbunden“ ist als teilweise oder vollständig ineinander enthalten, aneinander befestigt oder aneinander angefügt auszulegen, wenn er unmodifiziert vorliegt und sich auf physische Verbindungen bezieht, selbst, wenn ein Element dazwischen eingefügt ist. Die Nennung von Wertebereichen in dieser Schrift soll lediglich als schnelles Verfahren des einzelnen Bezugnehmens auf jeden separaten Wert dienen, der in den Bereich fällt, es sei denn, in dieser Schrift ist etwas anderes angegeben, und jeder separate Wert ist in die Beschreibung eingeschlossen, als ob er einzeln in dieser Schrift wiedergegeben wäre. Die Verwendung des Ausdrucks „Satz“ (z. B. „ein Satz von Objekten“) oder „Teilmenge“ ist als eine nicht leere Zusammenstellung auszulegen, die ein oder mehrere Elemente umfasst, es sei denn, es ist etwas anderes angemerkt oder dies widerspricht dem Kontext. Sofern nichts anderes angegeben ist oder der Kontext dem widerspricht, bezeichnet ferner der Ausdruck „Teilmenge“ eines entsprechenden Satzes nicht notwendigerweise eine richtige Teilmenge des entsprechenden Satzes, sondern die Teilmenge und der entsprechende Satz können gleich sein.

[0372] Sofern nicht spezifisch etwas anderes genannt ist oder der Kontext dem eindeutig widerspricht, ist verbindende Sprache, wie etwa Formulierungen der Form „wenigstens eines von A, B und C“ oder „mindestens eines von A, B und C“, andernfalls in dem Kontext zu verstehen, in dem sie allgemein verwendet werden, um darzustellen, dass ein Gegenstand, ein Ausdruck usw. entweder A oder B oder C oder eine beliebige nicht leere Teilmenge des Satzes aus A und B und C sein kann. Zum Beispiel beziehen sich in dem veranschaulichenden Beispiel für einen Satz, der drei Elemente aufweist, die verbindenden Formulierungen „wenigstens eines von A, B und C“ und „mindestens eines von A, Bund C“ auf einen beliebigen der folgenden Sätze: {A}, {B}, {C}, {A, B}, {A, C}, {B, C}, {A, B, C}. Somit soll solche verbindende Sprache im Allgemeinen nicht ausdrücken, dass bestimmte Ausführungen erforderlich machen, dass zumindest eines von A, zumindest eines von B und zumindest eines von C vorhanden ist. Außerdem, sofern nicht anders angemerkt oder durch den Kontext widerlegt, bezeichnet der Ausdruck „Vielzahl“ einen Zustand der Pluralität (z. B. gibt „eine Vielzahl von Objekten“ mehrere Objekte an). Bei einer Vielzahl handelt es sich um mindestens zwei Objekte, es können aber auch mehr sein, wenn dies entweder explizit oder durch den Kontext angegeben wird. Sofern nicht anders angegeben oder anderweitig aus dem Kontext ersichtlich, bedeutet „basierend auf“ „mindestens teilweise basierend auf“ und nicht „ausschließlich basierend auf“.

[0373] In dieser Schrift beschriebene Operationen von Prozessen können in einer beliebigen geeigneten Reihenfolge durchgeführt werden, sofern in dieser Schrift nicht anders angegeben oder der Kontext dem anderweitig eindeutig widerspricht. In mindestens einer Ausführungsform wird ein Prozess, wie zum Beispiel die in dieser Schrift beschriebenen Prozesse (oder Variationen und/oder Kombinationen davon), unter der Kontrolle von einem oder mehreren Computersystemen ausgeführt, die mit ausführbaren Anweisungen konfiguriert sind, und ist als Code (z. B. ausführbare Anweisungen, ein oder mehrere Computerprogramme oder eine oder mehrere Anwendungen), die kollektiv auf einem oder mehreren Prozessoren ausgeführt werden, durch Hardware oder Kombinationen davon implementiert. In mindestens einer Ausführungsform ist der Code auf einem computerlesbaren Speichermedium gespeichert, zum Beispiel in Form eines Computerprogramms, das eine Vielzahl von Anweisungen umfasst, die durch einen oder mehrere Prozessoren ausgeführt werden können. In mindestens einer Ausführungsform ist ein computerlesbares Speichermedium ein nicht transitorisches computerlesbares Speichermedium, das transitorische Signale (z. B. eine sich ausbreitende

transiente elektrische oder elektromagnetische Übertragung) ausschließt, aber nicht transitorische Datenspeicherschaltungen (z. B. Puffer, Cache und Warteschlangen) innerhalb der Transceiver von transitorischen Signalen einschließt. In mindestens einer Ausführungsform ist der Code (z. B. ausführbarer Code oder Quellcode) auf einem Satz von einem oder mehreren nicht transitorischen computerlesbaren Speichermedien gespeichert, auf denen ausführbare Anweisungen gespeichert sind (oder ein anderer Speicher zum Speichern ausführbarer Anweisungen), die bei Ausführung (d. h. als Ergebnis der Ausführung) durch einen oder mehrere Prozessoren eines Computersystems das Computersystem dazu veranlassen, in dieser Schrift beschriebene Operationen durchzuführen. Ein Satz von nicht transitorischen computerlesbaren Speichermedien umfasst in mindestens einer Ausführungsform mehrere nicht transitorische computerlesbare Speichermedien und einem oder mehreren der einzelnen nicht transitorischen Speichermedien mehrerer nicht transitorischer computerlesbarer Speichermedien fehlt der gesamte Code, während mehrere nicht transitorische computerlesbare Speichermedien kollektiv den gesamten Code speichern. In mindestens einer Ausführungsform werden die ausführbaren Anweisungen so ausgeführt, dass unterschiedliche Anweisungen durch unterschiedliche Prozessoren ausgeführt werden - zum Beispiel speichert ein nichttransitorisches computerlesbares Speichermedium Anweisungen und eine hauptsächliche zentrale Verarbeitungseinheit („CPU“) führt einige der Anweisungen aus, während eine Grafikverarbeitungseinheit („GPU“) andere Anweisungen ausführt. In mindestens einer Ausführungsform weisen unterschiedliche Komponenten eines Computersystems separate Prozessoren auf und führen unterschiedliche Prozessoren unterschiedliche Teilmengen von Anweisungen aus.

[0374] Dementsprechend sind in mindestens einer Ausführungsform Computersysteme dazu konfiguriert, einen oder mehrere Dienste zu implementieren, die einzeln oder zusammen Operationen der in dieser Schrift beschriebenen Prozesse durchführen, und sind derartige Computersysteme mit geeigneter Hardware und/oder Software konfiguriert, die eine Durchführung der Operationen ermöglicht. Ferner ist ein Computersystem, das mindestens eine Ausführungsform der vorliegenden Offenbarung implementiert, eine einzelne Vorrichtung und in einer anderen Ausführungsform ein verteiltes Computersystem, das mehrere Vorrichtungen umfasst, die unterschiedlich arbeiten, sodass das verteilte Computersystem die in dieser Schrift beschriebenen Operationen durchführt und sodass eine einzelne Vorrichtung nicht alle Operationen durchführt.

[0375] Die Verwendung jeglicher Beispiele oder beispielhafter Wortwahl (z. B. „wie etwa“), die in dieser bereitgestellt sind, soll lediglich die Ausführungsformen der Offenbarung besser veranschaulichen und stellt keine Einschränkung des Umfangs der Offenbarung dar, es sei denn, es ist etwas anderes beansprucht. Keinerlei Wortwahl in der Beschreibung sollte so ausgelegt werden, dass sie ein beliebiges nicht beanspruchtes Element als für die Umsetzung der Offenbarung wesentlich angibt.

[0376] Jegliche Referenzen, einschließlich Veröffentlichungen, Patentanmeldungen und Patenten, die in dieser Schrift erwähnt werden, sind hiermit durch Bezugnahme in demselben Maße aufgenommen, als wäre jede Referenz einzeln und spezifisch als durch Referenz eingeschlossen angegeben und in ihrer Gesamtheit in dieser Schrift ausgeführt.

[0377] In der Beschreibung und den Ansprüchen können die Ausdrücke „gekoppelt“ und „verbunden“ sowie deren Ableitungen verwendet werden. Es versteht sich, dass diese Ausdrücke nicht als Synonyme füreinander bestimmt sein können. Vielmehr kann in bestimmten Beispielen „verbunden“ oder „gekoppelt“ verwendet werden, um anzuzeigen, dass zwei oder mehr Elemente in direktem oder indirektem physischen oder elektrischen Kontakt miteinander stehen. Mit „gekoppelt“ kann auch gemeint sein, dass zwei oder mehr Elemente nicht in direktem Kontakt miteinander stehen, jedoch trotzdem miteinander zusammenwirken oder interagieren.

[0378] Sofern nicht ausdrücklich anders angegeben, versteht es sich, dass sich Ausdrücke wie „Verarbeitung“, „Berechnung“, „Berechnen“, „Bestimmen“ oder dergleichen in der gesamten Beschreibung auf Handlungen und/oder Prozesse eines Computers oder Rechensystems oder einer ähnlichen elektronischen Rechenvorrichtung, die Daten, die als physische, wie etwa elektronische, Größen in den Registern und/oder Speichern des Rechensystems dargestellt sind, manipulieren und/oder in andere Daten umwandeln, die auf ähnliche Weise als physische Größen in den Speichern, Registern oder anderen derartigen Informationsspeicher-, -übertragungs- oder -anzeigevorrichtungen des Rechensystems dargestellt sind.

[0379] Auf ähnliche Weise kann sich der Ausdruck „Prozessor“ auf eine beliebige Vorrichtung oder einen Abschnitt einer Vorrichtung beziehen, die/der elektronische Daten aus Registern und/oder dem Speicher verarbeitet und diese elektronischen Daten in andere elektronische Daten umwandelt, die in Registern und/oder im Speicher gespeichert werden können. Als nicht einschränkende Beispiele kann der „Prozessor“ eine CPU

oder eine GPU sein. Eine „Rechenplattform“ kann einen oder mehrere Prozessoren umfassen. Im vorliegenden Zusammenhang können „Software“-Prozesse zum Beispiel Software- und/oder Hardware-Entitäten beinhalten, die im Laufe der Zeit Arbeit durchführen, wie etwa Aufgaben, Threads und intelligente Agenten. Außerdem kann sich jeder Prozess auf mehrere Prozesse beziehen, um Anweisungen nacheinander oder parallel, kontinuierlich oder intermittierend auszuführen. Die Ausdrücke „System“ und „Verfahren“ werden in dieser Schrift insofern austauschbar verwendet, dass ein System ein oder mehrere Verfahren verkörpern kann und die Verfahren als System betrachtet werden können.

[0380] Im vorliegenden Dokument kann auf Erhalten, Erfassen, Empfangen oder Eingeben von analogen oder digitalen Daten in ein Teilsystem, Computersystem oder eine computerimplementierte Maschine Bezug genommen werden. Das Erhalten, Erfassen, Empfangen oder Eingeben von analogen und digitalen Daten kann auf vielfältige Weise erfolgen, wie etwa durch Empfangen von Daten als Parameter eines Funktionsaufrufs oder eines Aufrufs einer Anwendungsprogrammierschnittstelle. In einigen Implementierungen kann der Prozess des Erhaltens, Erfassens, Empfangens oder Eingebens von analogen oder digitalen Daten durch Übertragen von Daten über eine serielle oder parallele Schnittstelle erfolgen. In einer anderen Implementierung kann der Prozess des Erhaltens, Erfassens, Empfangens oder Eingebens von analogen oder digitalen Daten durch Übertragen von Daten über ein Computernetzwerk von der bereitstellenden Entität zur erfassenden Entität erfolgen. Es kann auch auf Bereitstellen, Ausgeben, Übertragen, Senden oder Darstellen von analogen oder digitalen Daten Bezug genommen werden. In verschiedenen Beispielen kann der Prozess des Bereitstellens, Ausgebens, Übertragens, Sendens oder Darstellens von analogen oder digitalen Daten durch Übertragen von Daten als Eingabe- oder Ausgabeparameter eines Funktionsaufrufs, Parameter einer Anwendungsprogrammierschnittstelle oder eines Interprozess-Kommunikationsmechanismus erfolgen.

[0381] Obwohl die vorstehende Erörterung beispielhafte Implementierung der beschriebenen Techniken darlegt, können auch andere Architekturen verwendet werden, um die beschriebene Funktionalität zu implementieren, und sie sollen im Umfang dieser Offenbarung liegen. Darüber hinaus könnten, obwohl spezifische Verteilungen von Zuständigkeiten vorstehend zum Zwecke der Erörterung definiert sind, verschiedene Funktionen und Zuständigkeiten in Abhängigkeit von den Umständen anders verteilt und aufgeteilt werden.

[0382] Obwohl der Gegenstand in für Strukturmerkmale und/oder Verfahrenshandlungen spezifischer Sprache beschrieben wurde, versteht es sich ferner, dass der in den beigefügten Ansprüchen beanspruchte Gegenstand nicht unbedingt auf die beschriebenen konkreten Merkmale oder Handlungen beschränkt ist. Vielmehr werden spezifische Merkmale und Handlungen als beispielhafte Formen zum Implementieren der Ansprüche offenbart.

ZITATE ENTHALTEN IN DER BESCHREIBUNG

Zitierte Patentliteratur

- US 16/581969 [0001]

Patentansprüche

1. Spielsystem, umfassend:
einen oder mehrere Prozessoren, um ein oder mehrere neuronale Netze zu verwenden, um zu bewirken, dass einem oder mehreren Spielern eines Spiels Feedback bereitgestellt wird, wobei das Feedback dazu dient, dem einen oder mehreren Spielern zu helfen, ihre Leistung beim Spielen des Spiels zu verbessern.
2. Spielsystem nach Anspruch 1, wobei das eine oder die mehreren neuronalen Netze als Eingabe Daten in Bezug auf Handlungen und Belohnungen für den einen oder die mehreren Spieler während des Spiels akzeptieren, um Fähigkeitswerte für den einen oder die mehreren Spieler zu bestimmen.
3. Spielsystem nach Anspruch 2, wobei die Handlungen in eine Anzahl von Kategorien eingeteilt sind und wobei die Fähigkeitswerte unter Verwendung von Gewichtungen von Handlungen in verschiedenen Kategorien bestimmt werden, wobei die Gewichtungen zum Teil von den zugehörigen Belohnungen abhängen.
4. Spielsystem nach Anspruch 1, wobei das eine oder die mehreren neuronalen Netze unter Verwendung von unüberwachtem Lernen mit Dimensionalitätsreduktion und Datenclustering trainiert werden.
5. Spielsystem nach Anspruch 1, wobei sich das Feedback auf einen aktuellen Kontext des Spiels bezieht, das von dem einen oder den mehreren Spielern gespielt wird.
6. Spielsystem nach Anspruch 1, wobei das eine oder die mehreren neuronalen Netze verwendet werden, um ferner zu bewirken, dass ein Feedback bereitgestellt wird, das sich auf die Müdigkeit und/oder die Reaktionszeit des einen oder der mehreren Spieler des Spiels bezieht.
7. Prozessor, umfassend:
eine oder mehrere Schaltungen, um ein oder mehrere neuronale Netze zu veranlassen, ein Feedback zu veranlassen, die einem oder mehreren Spielern eines Spiels bereitgestellt wird, wobei das Feedback dazu dient, dem einen oder mehreren Spielern zu helfen, ihre Leistung beim Spielen des Spiels zu verbessern.
8. Prozessor nach Anspruch 7, wobei das eine oder die mehreren neuronalen Netze als Eingabe Daten bezüglich Handlungen und Belohnungen für den einen oder die mehreren Spieler während des Spiels akzeptieren, um Fähigkeitswerte für den einen oder die mehreren Spieler zu bestimmen.
9. Prozessor nach Anspruch 8, wobei die Handlungen in eine Anzahl von Kategorien eingeteilt sind, und wobei die Fähigkeitswerte unter Verwendung von Gewichtungen von Handlungen in verschiedenen Kategorien bestimmt werden, wobei die Gewichtungen zum Teil von den zugehörigen Belohnungen abhängen.
10. Prozessor nach Anspruch 7, wobei das eine oder die mehreren neuronalen Netze unter Verwendung von unüberwachtem Lernen mit Dimensionalitätsreduktion und Datenclustering trainiert werden.
11. Prozessor nach Anspruch 7, wobei sich das Feedback auf einen aktuellen Kontext des Spiels bezieht, das von dem einen oder den mehreren Spielern gespielt wird.
12. Prozessor nach Anspruch 7, wobei das eine oder die mehreren neuronalen Netze verwendet werden, um ferner zu bewirken, dass ein Feedback in Bezug auf mindestens eines von Müdigkeit oder Reaktionszeit des einen oder der mehreren Spieler des Spiels bereitgestellt wird.
13. Nicht-transitorisches maschinenlesbares Medium, auf dem ein Satz von Anweisungen gespeichert ist, die, wenn sie von einem oder mehreren Prozessoren ausgeführt werden, den einen oder die mehreren Prozessoren veranlassen, mindestens:
ein oder mehrere neuronale Netze zu veranlassen, einem oder mehreren Spielern eines Spiels Feedback bereitzustellen, wobei das Feedback dazu dient, dem einen oder mehreren Spielern zu helfen, ihre Leistung beim Spielen des Spiels zu verbessern.
14. Nicht-transitorisches maschinenlesbares Medium nach Anspruch 13, wobei das eine oder die mehreren neuronalen Netze als Eingabe Daten bezüglich Handlungen und Belohnungen für den einen oder die mehreren Spieler während des Spiels akzeptieren, um Fähigkeitswerte für den einen oder die mehreren Spieler zu bestimmen.

15. Nicht-transitorisches maschinenlesbares Medium nach Anspruch 14, wobei die Handlungen in eine Anzahl von Kategorien eingeteilt sind und wobei die Fähigkeitswerte unter Verwendung von Gewichtungen von Handlungen in verschiedenen Kategorien bestimmt werden, wobei die Gewichtungen zum Teil von den zugehörigen Belohnungen abhängen.

16. Nicht-transitorisches maschinenlesbares Medium nach Anspruch 13, wobei das eine oder die mehreren neuronalen Netze unter Verwendung von unüberwachtem Lernen mit Dimensionalitätsreduktion und Datenclustering trainiert werden.

17. Nicht-transitorisches maschinenlesbares Medium nach Anspruch 13, wobei sich das Feedback auf einen aktuellen Kontext des Spiels bezieht, das von dem einen oder den mehreren Spielern gespielt wird.

18. Nicht-transitorisches maschinenlesbares Medium nach Anspruch 13, wobei das eine oder die mehreren neuronalen Netze verwendet werden, um ferner zu bewirken, dass ein Feedback bereitgestellt wird, das sich auf mindestens eines von Müdigkeit oder Reaktionszeit des einen oder der mehreren Spieler des Spiels bezieht.

19. Prozessor, umfassend:
eine oder mehrere Schaltungen, die so konfiguriert sind, dass sie ein oder mehrere neuronale Netze mindestens zum Teil trainieren, indem sie Handlungen analysieren, die von einem oder mehreren Spielern eines Spiels ausgeführt werden, um Feedback zu erzeugen, das dem einen oder den mehreren Spielern helfen, ihre Leistung beim Spielen des Spiels zu verbessern.

20. Prozessor nach Anspruch 19, wobei das eine oder die mehreren neuronalen Netze als Eingabe Daten bezüglich Handlungen und Belohnungen für den einen oder die mehreren Spieler während des Spiels akzeptieren, um Fähigkeitswerte für den einen oder die mehreren Spieler zu bestimmen.

21. Prozessor nach Anspruch 20, wobei die Aktionen in eine Anzahl von Kategorien eingeteilt sind und wobei die Fähigkeitswerte unter Verwendung von Gewichtungen von Handlungen in verschiedenen Kategorien bestimmt werden, wobei die Gewichtungen zum Teil von den zugehörigen Belohnungen abhängen.

22. Prozessor nach Anspruch 19, wobei das eine oder die mehreren neuronalen Netze unter Verwendung von unüberwachtem Lernen mit Dimensionalitätsreduktion und Datenclustering trainiert werden.

23. Prozessor nach Anspruch 19, wobei sich das Feedback auf einen aktuellen Kontext des Spiels bezieht, das von dem einen oder den mehreren Spielern gespielt wird.

24. Prozessor nach Anspruch 19, wobei das eine oder die mehreren neuronalen Netze verwendet werden, um ferner zu bewirken, dass ein Feedback bereitgestellt wird, das sich auf mindestens eines von Müdigkeit oder Reaktionszeit des einen oder der mehreren Spieler des Spiels bezieht.

25. System, umfassend:
einen oder mehrere Prozessoren, die so konfiguriert sind, dass sie ein oder mehrere neuronale Netze mindestens zum Teil trainieren, indem sie Handlungen analysieren, die von einem oder mehreren Spielern eines Spiels ausgeführt werden, um Feedback zu erzeugen, das dem einen oder mehreren Spielern helfen, ihre Leistung beim Spielen des Spiels zu verbessern; und
einen oder mehrere Speicher zum Speichern des einen oder der mehreren neuronalen Netze.

26. System nach Anspruch 25, wobei das eine oder die mehreren neuronalen Netze als Eingabe Daten bezüglich Handlungen und Belohnungen für den einen oder die mehreren Spieler während des Spiels akzeptieren, um Fähigkeitswerte für den einen oder die mehreren Spieler zu bestimmen.

27. System nach Anspruch 26, wobei die Handlungen in eine Anzahl von Kategorien eingeteilt sind und wobei die Fähigkeitswerte unter Verwendung von Gewichtungen von Handlungen in verschiedenen Kategorien bestimmt werden, wobei die Gewichtungen zum Teil von den zugehörigen Belohnungen abhängen.

28. System nach Anspruch 25, wobei das eine oder die mehreren neuronalen Netze unter Verwendung von unüberwachtem Lernen mit Dimensionalitätsreduktion und Datenclustering trainiert werden.

29. System nach Anspruch 25, wobei sich das Feedback auf einen aktuellen Kontext des Spiels bezieht, das von dem einen oder den mehreren Spielern gespielt wird.

30. System nach Anspruch 25, wobei das eine oder die mehreren neuronalen Netze verwendet werden, um ferner zu bewirken, dass ein Feedback bereitgestellt wird, das sich auf mindestens eines von Müdigkeit oder Reaktionszeit des einen oder der mehreren Spieler des Spiels bezieht.

31. Nicht-transitorisches maschinenlesbares Medium, auf dem ein Satz von Anweisungen gespeichert ist, die, wenn sie von einem oder mehreren Prozessoren ausgeführt werden, den einen oder die mehreren Prozessoren veranlassen, mindestens:

eines oder mehrere neuronale Netze, mindestens zum Teil, durch Analysieren von Handlungen, die von einem oder mehreren Spielern eines Spiels ausgeführt werden, zu trainieren, um Feedback zu erzeugen, das dem einen oder mehreren Spielern hilft, ihre Leistung beim Spielen des Spiels zu verbessern.

32. Nicht-transitorisches maschinenlesbares Medium nach Anspruch 31, wobei das eine oder die mehreren neuronalen Netze als Eingabe Daten bezüglich Handlungen und Belohnungen für den einen oder die mehreren Spieler während des Spiels akzeptieren, um Fähigkeitswerte für den einen oder die mehreren Spieler zu bestimmen.

33. Nicht-transitorisches maschinenlesbares Medium nach Anspruch 32, wobei die Handlungen in eine Anzahl von Kategorien eingeteilt sind, und wobei die Fähigkeitswerte unter Verwendung von Gewichtungen von Handlungen in verschiedenen Kategorien bestimmt werden, wobei die Gewichtungen zum Teil von den zugehörigen Belohnungen abhängen.

34. Nicht-transitorisches maschinenlesbares Medium nach Anspruch 31, wobei das eine oder die mehreren neuronalen Netze unter Verwendung von unüberwachtem Lernen mit Dimensionalitätsreduktion und Datenclustering trainiert werden.

35. Nicht-transitorisches maschinenlesbares Medium nach Anspruch 31, wobei sich das Feedback auf einen aktuellen Kontext des Spiels bezieht, das von dem einen oder den mehreren Spielern gespielt wird.

36. Nicht-transitorisches maschinenlesbares Medium nach Anspruch 31, wobei das eine oder die mehreren neuronalen Netze verwendet werden, um ferner zu bewirken, dass ein Feedback bereitgestellt wird, die sich auf mindestens eines von Müdigkeit oder Reaktionszeit des einen oder der mehreren Spieler des Spiels bezieht.

37. Verfahren, umfassend:

Veranlassen, unter Verwendung eines oder mehrerer neuronaler Netze, dass einem oder mehreren Spielern eines Spiels Feedback bereitgestellt wird, wobei das Feedback dazu dient, dem einen oder mehreren Spielern zu helfen, ihre Leistung beim Spielen des Spiels zu verbessern.

38. Verfahren nach Anspruch 37, wobei das eine oder die mehreren neuronalen Netze als Eingabe Daten bezüglich Handlungen und Belohnungen für den einen oder die mehreren Spieler während des Spiels akzeptieren, um Fähigkeitswerte für den einen oder die mehreren Spieler zu bestimmen.

39. Verfahren nach Anspruch 38, wobei die Aktionen in eine Anzahl von Kategorien eingeteilt werden, und wobei die Fähigkeitswerte unter Verwendung von Gewichtungen von Handlungen in verschiedenen Kategorien bestimmt werden, wobei die Gewichtungen zum Teil von den zugehörigen Belohnungen abhängen.

40. Verfahren nach Anspruch 37, wobei das eine oder die mehreren neuronalen Netze unter Verwendung von unüberwachtem Lernen mit Dimensionalitätsreduktion und Datenclustering trainiert werden.

41. Verfahren nach Anspruch 37, wobei sich das Feedback auf einen aktuellen Kontext des Spiels bezieht, das von dem einen oder den mehreren Spielern gespielt wird.

42. Verfahren nach Anspruch 37, wobei das eine oder die mehreren neuronalen Netze verwendet werden, um ferner zu bewirken, dass ein Feedback bereitgestellt wird, das sich auf mindestens eines von Müdigkeit oder Reaktionszeit des einen oder der mehreren Spieler des Spiels bezieht.

Es folgen 51 Seiten Zeichnungen

Anhängende Zeichnungen

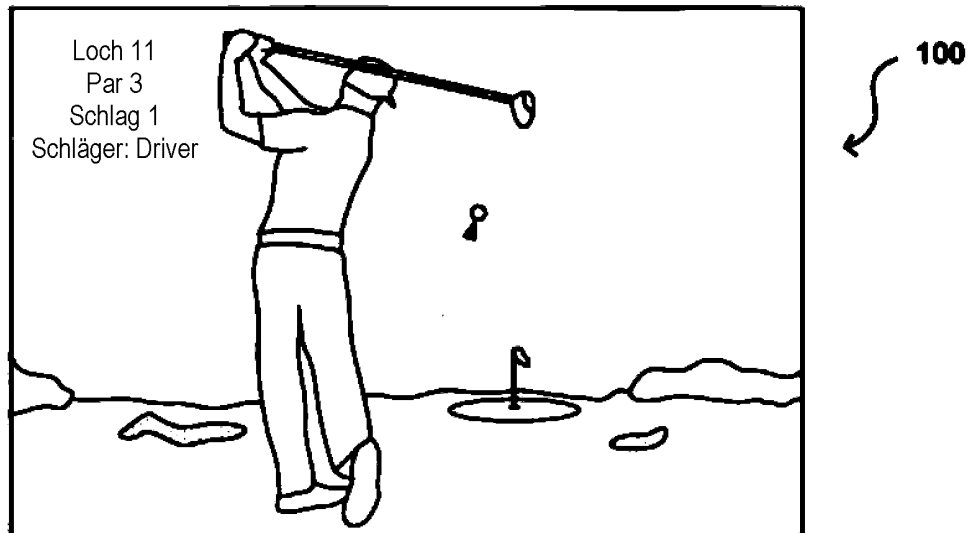


FIG. 1A

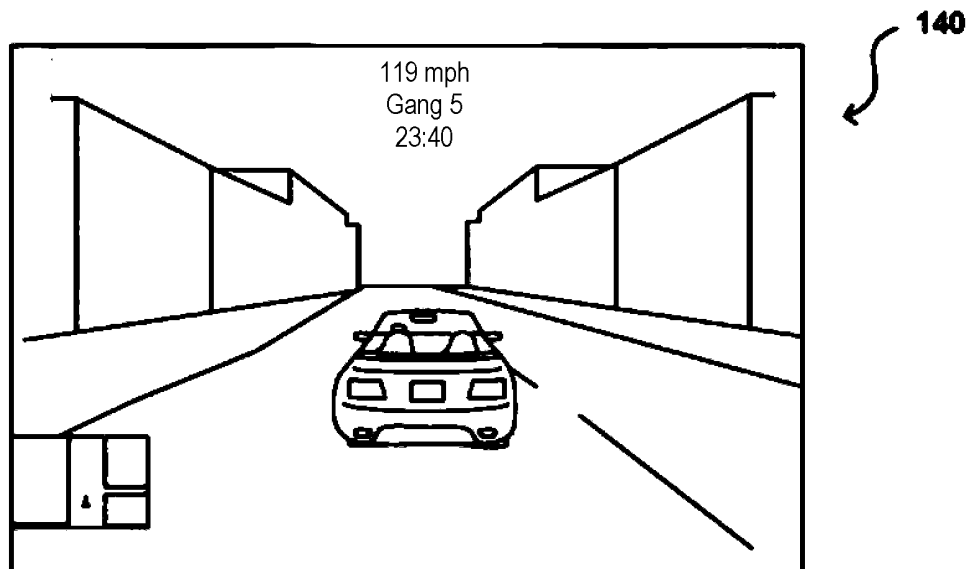


FIG. 1B

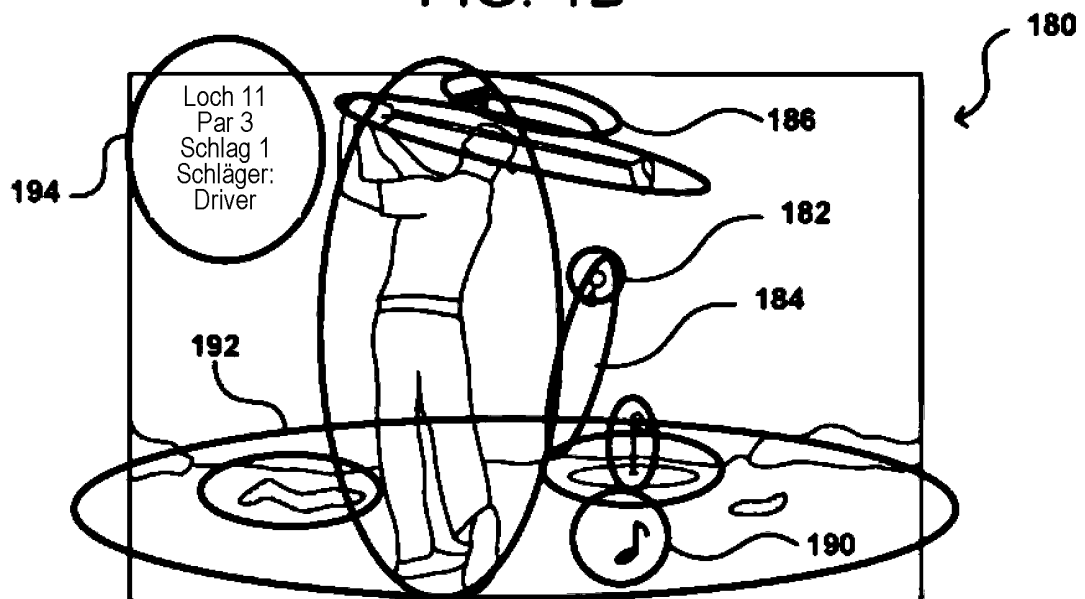


FIG. 1C

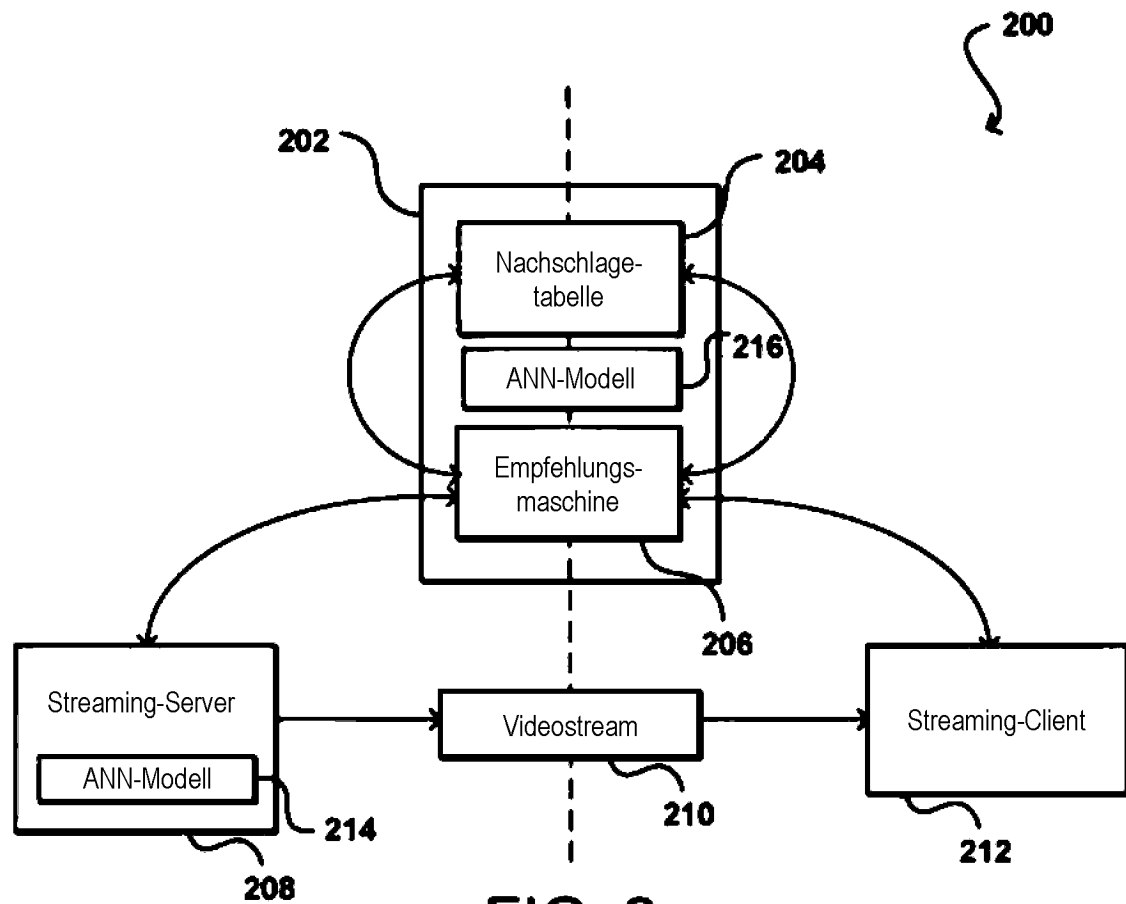


FIG. 2

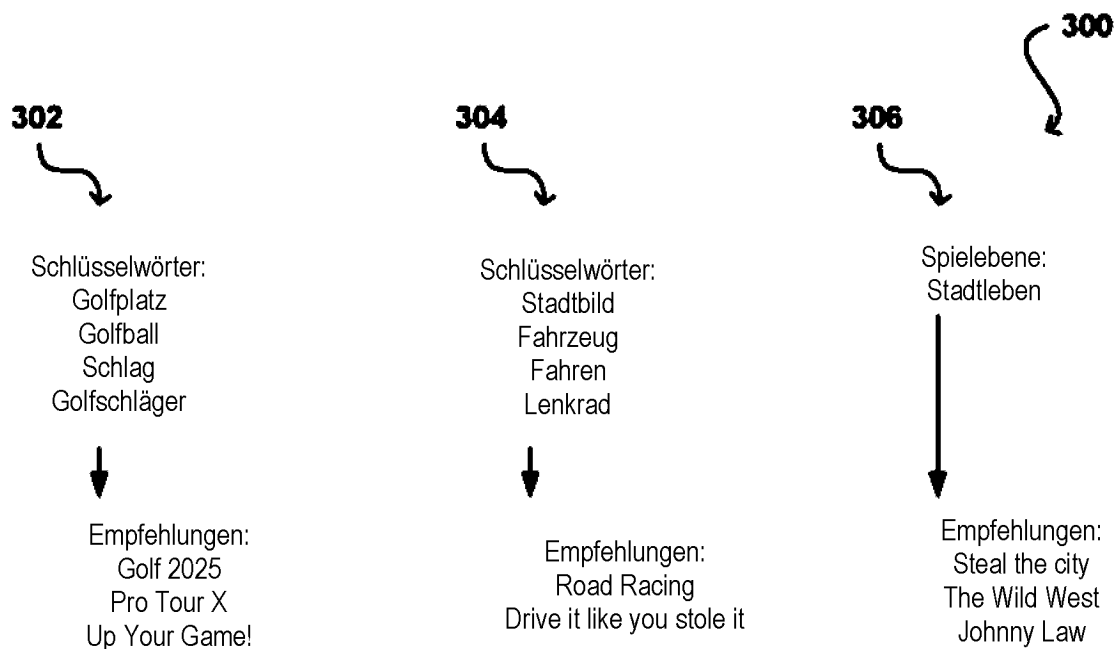
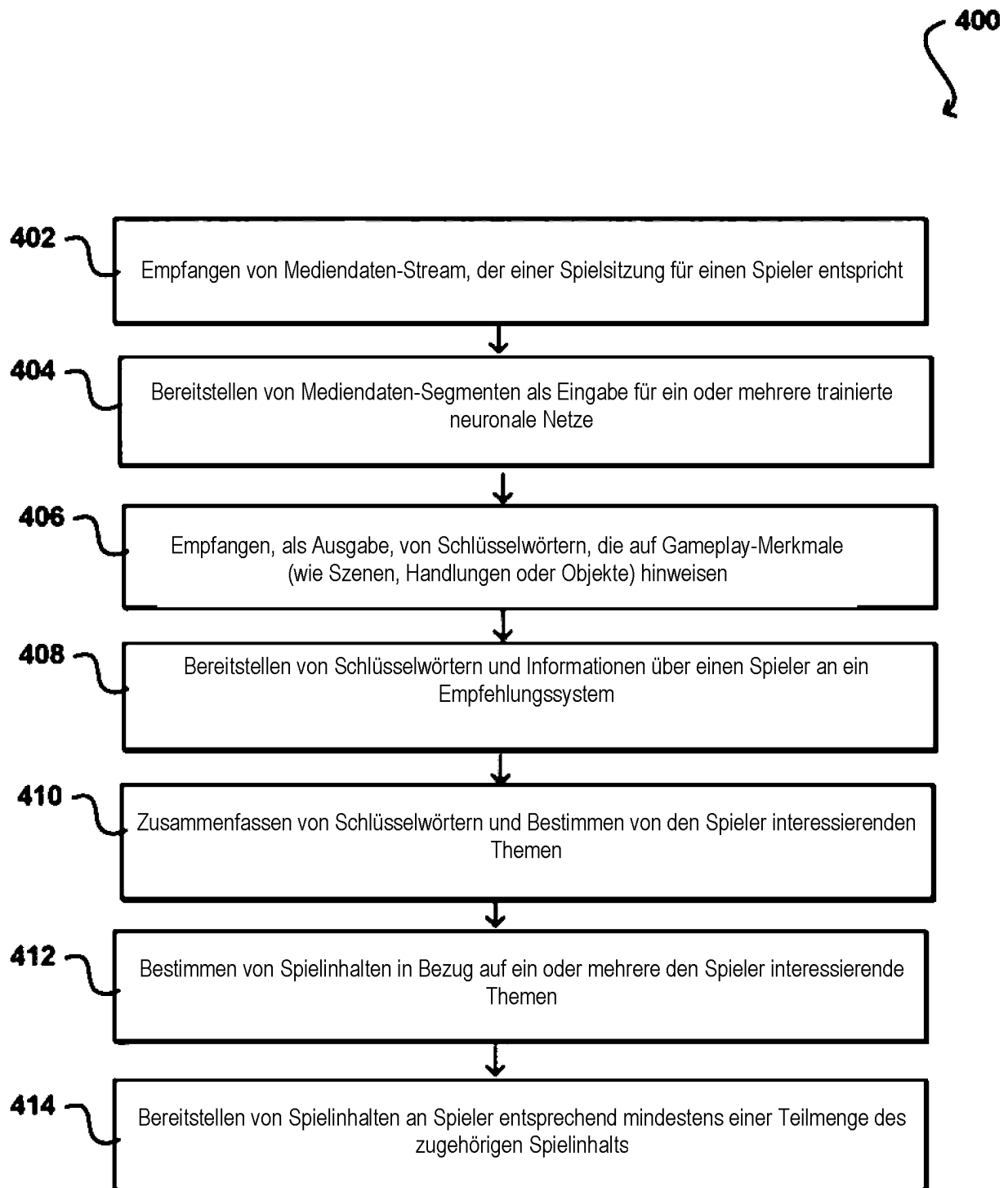


FIG. 3

**FIG. 4**

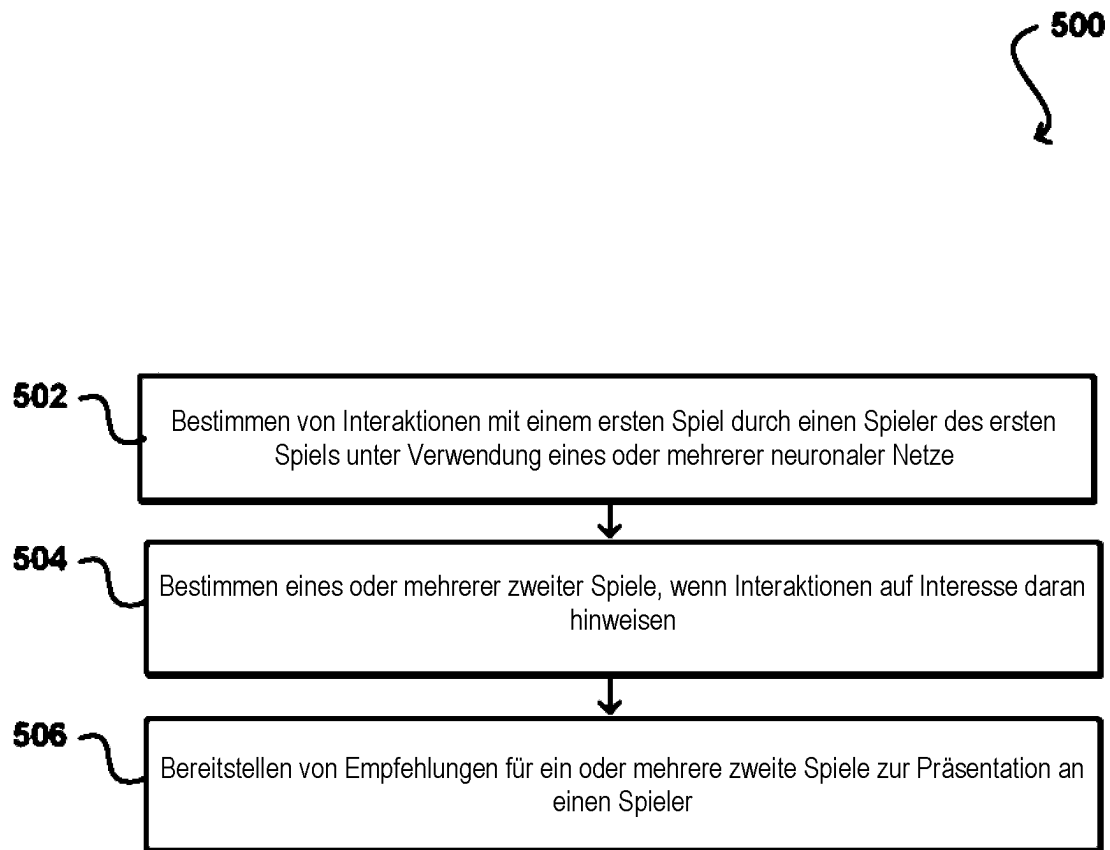


FIG. 5

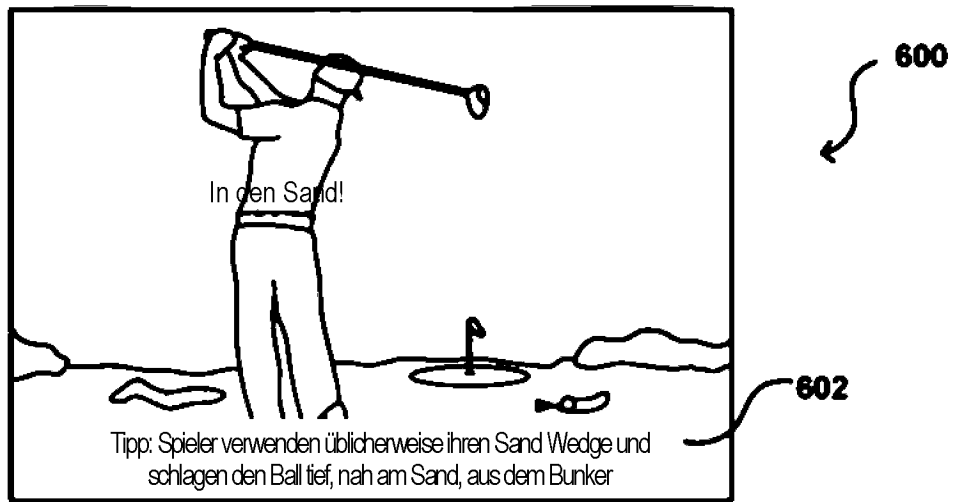


FIG. 6A

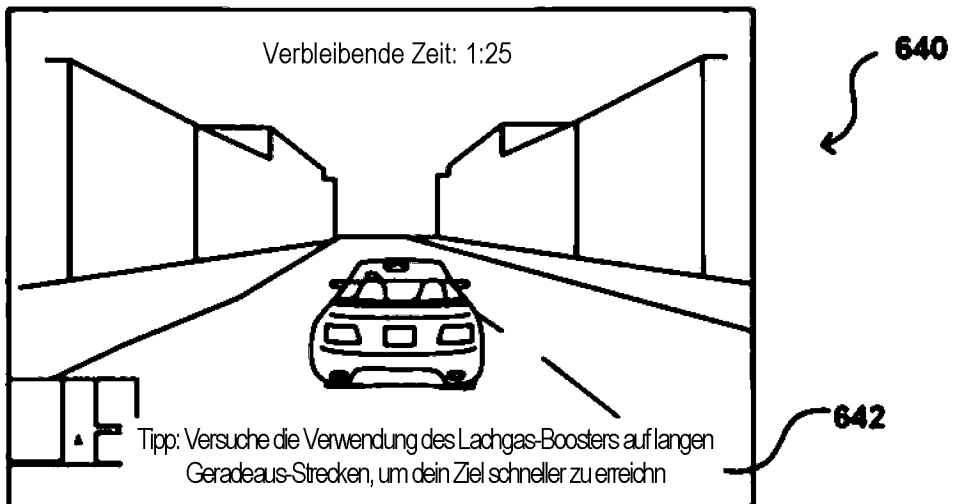


FIG. 6B

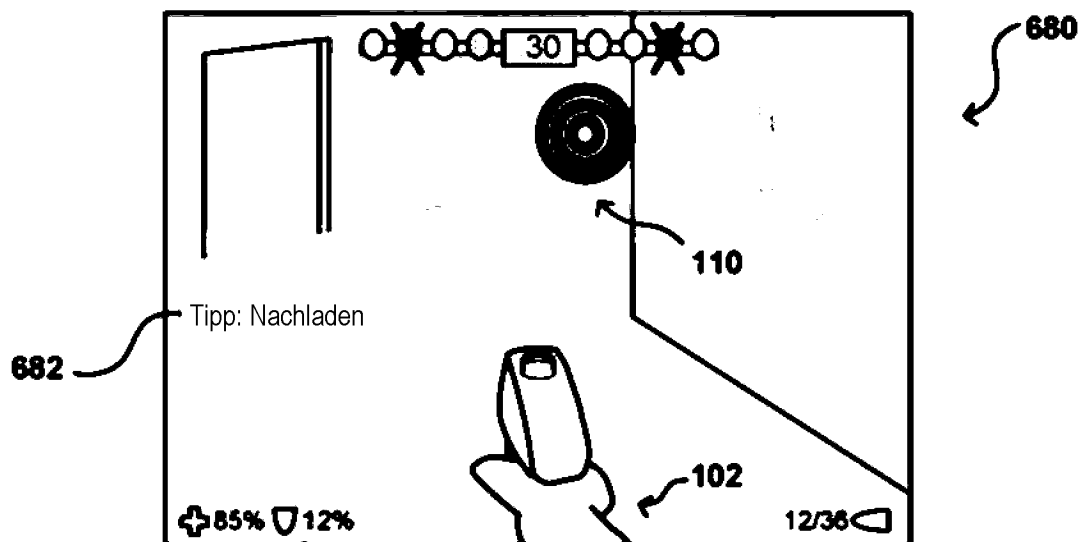


FIG. 6C

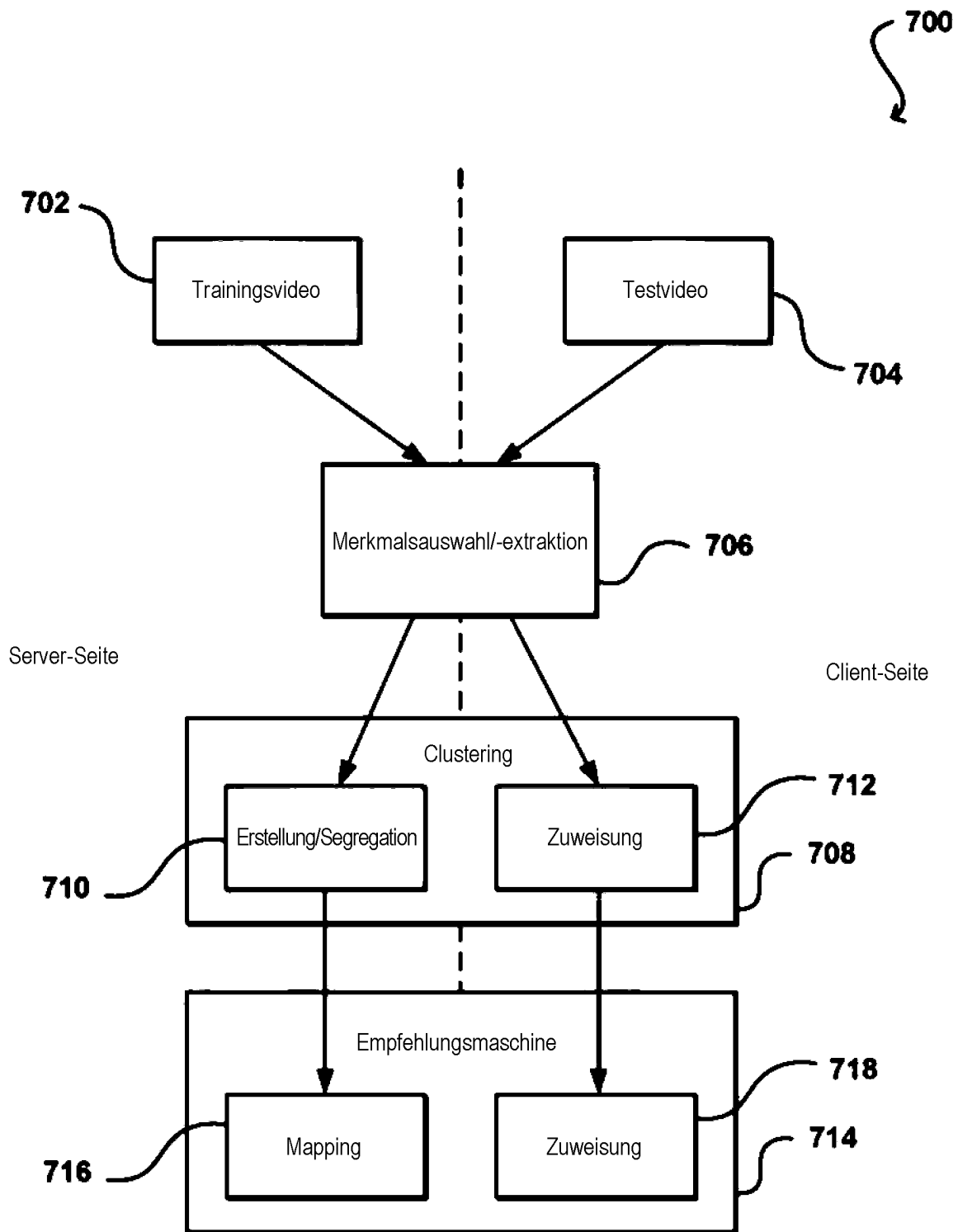


FIG. 7

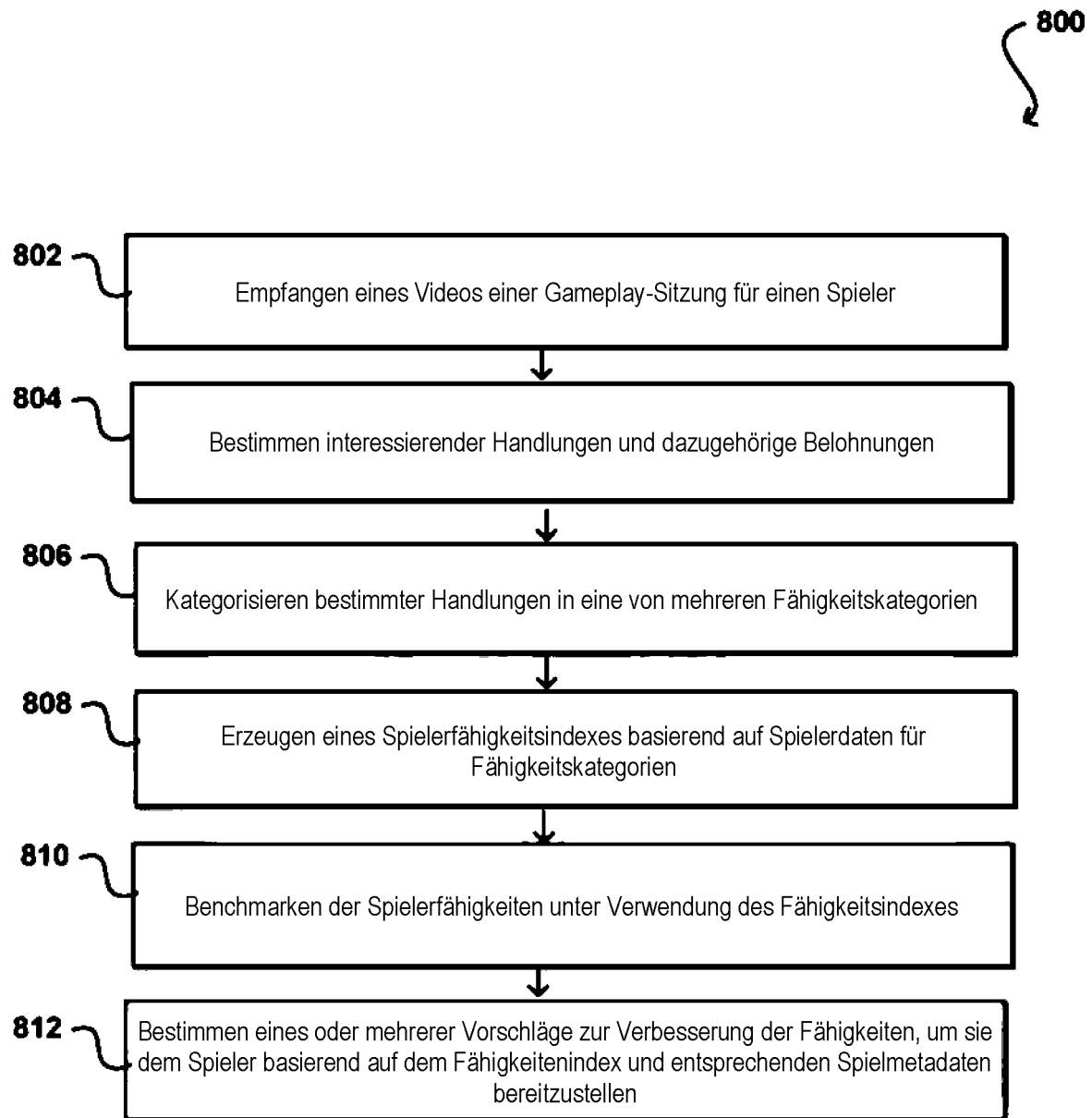


FIG. 8

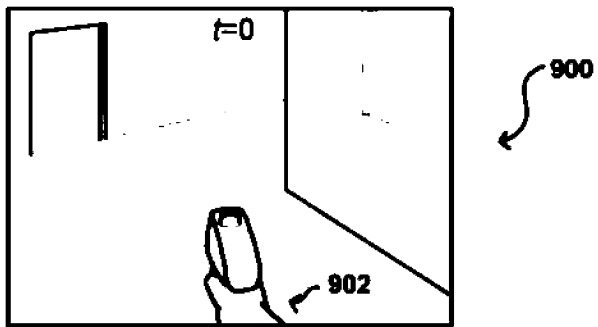


FIG. 9A

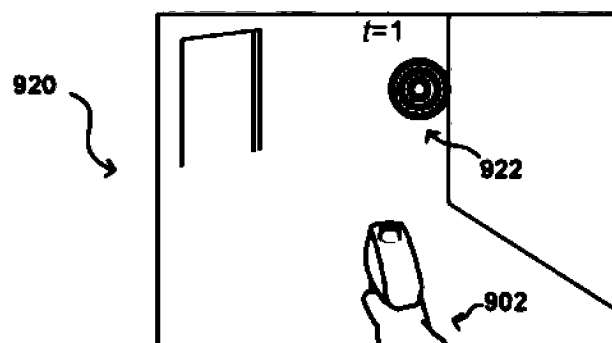


FIG. 9B

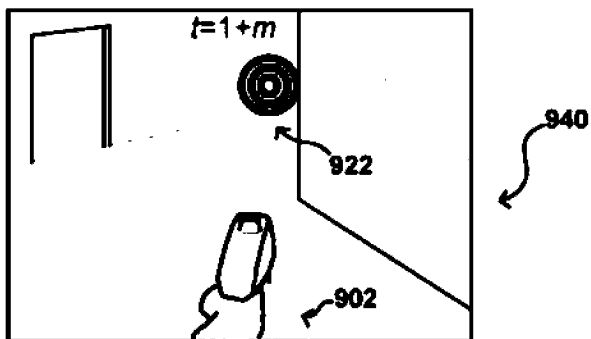


FIG. 9C

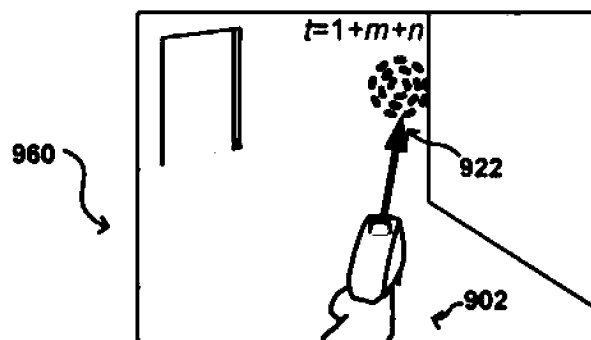


FIG. 9D

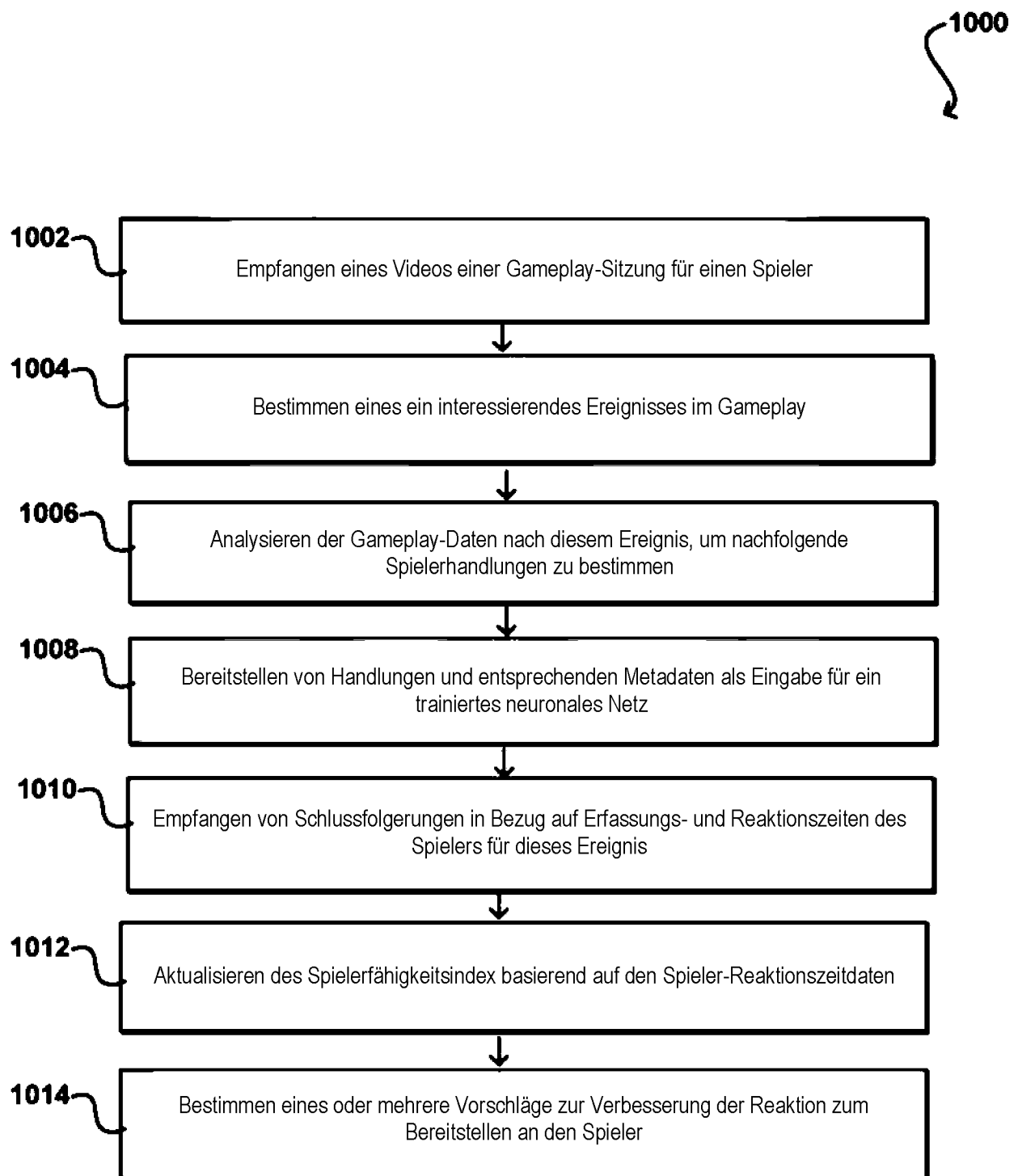


FIG. 10

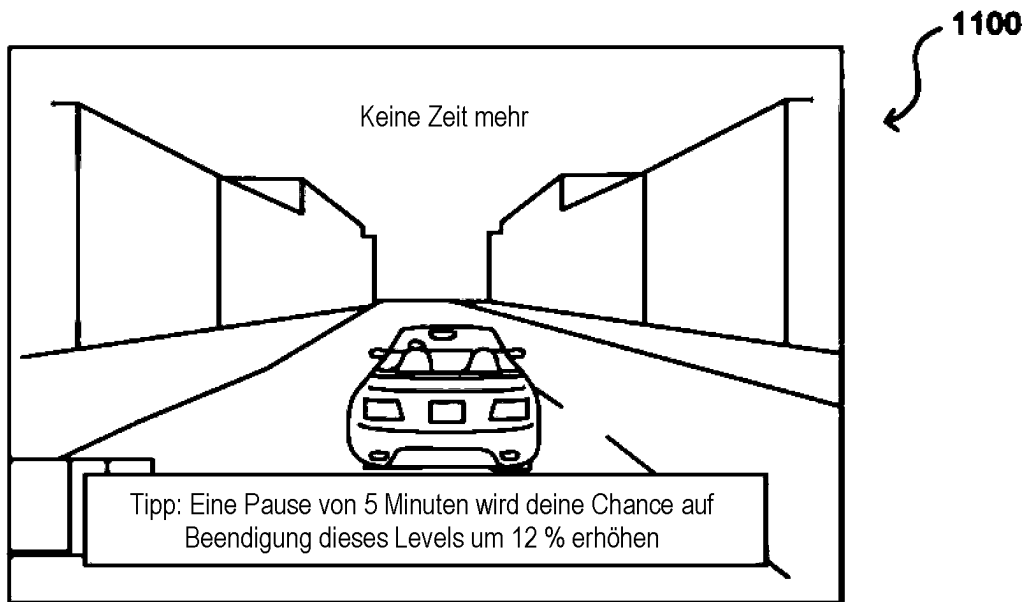


FIG. 11A



FIG. 11B

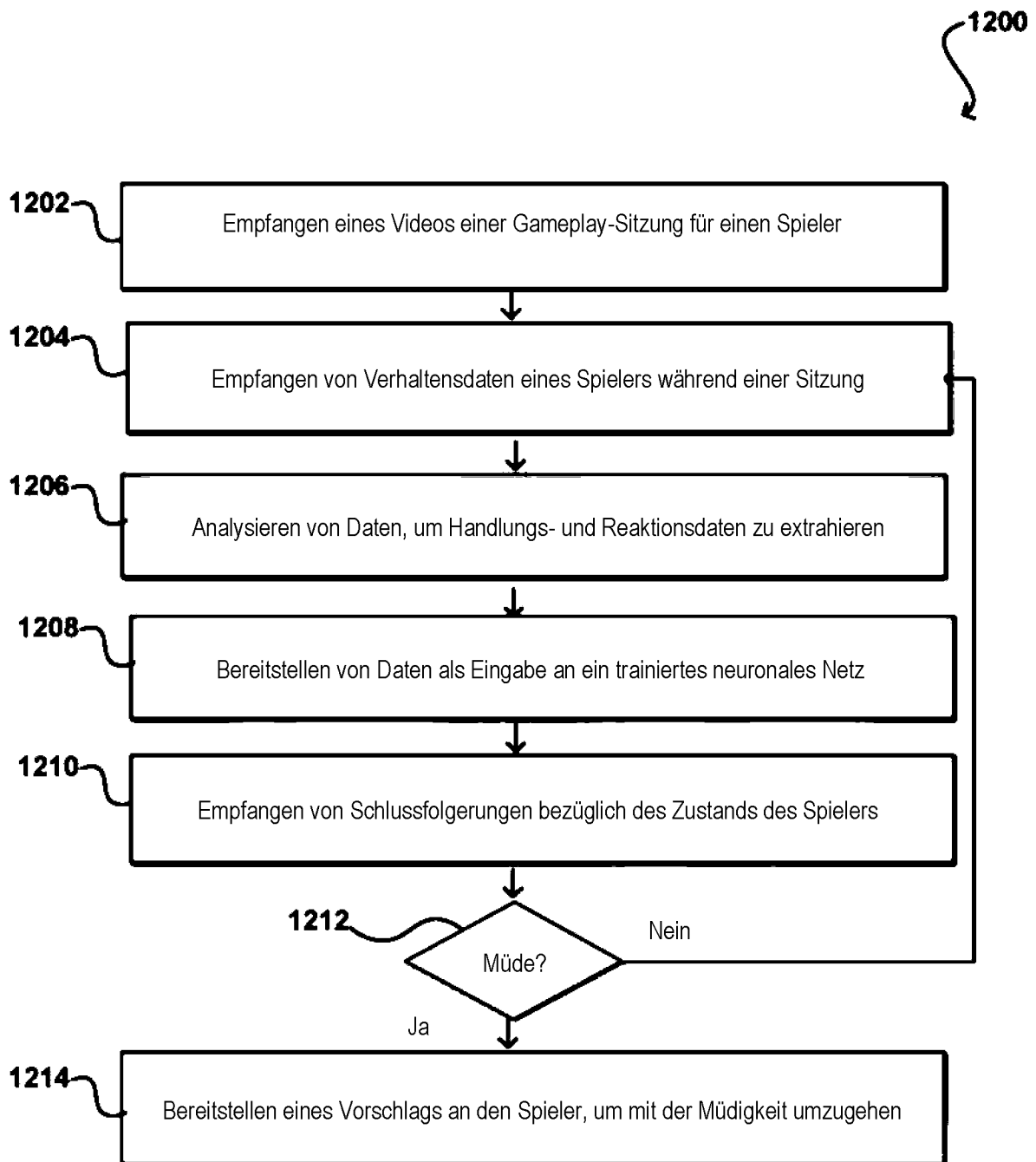


FIG. 12

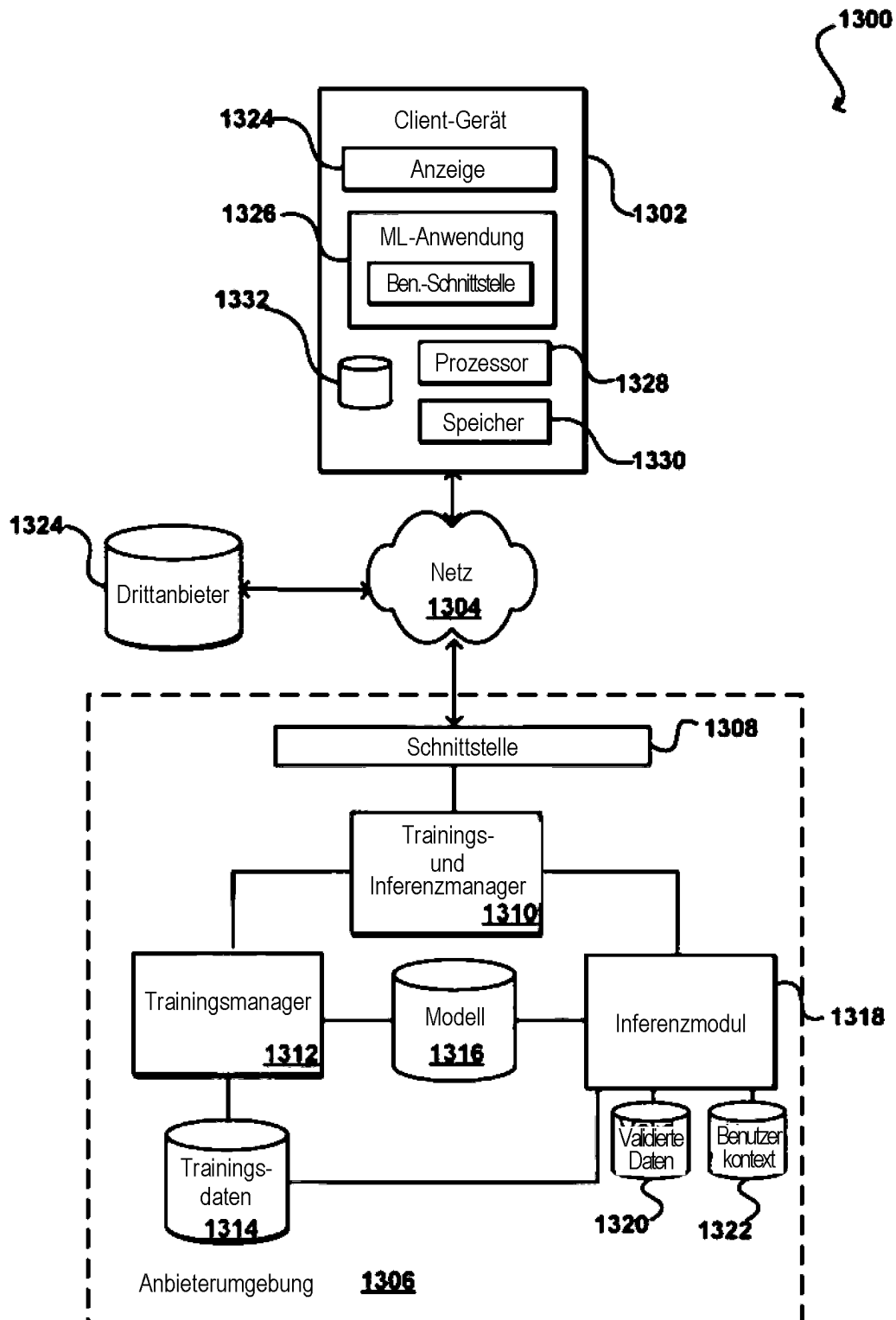


FIG. 13

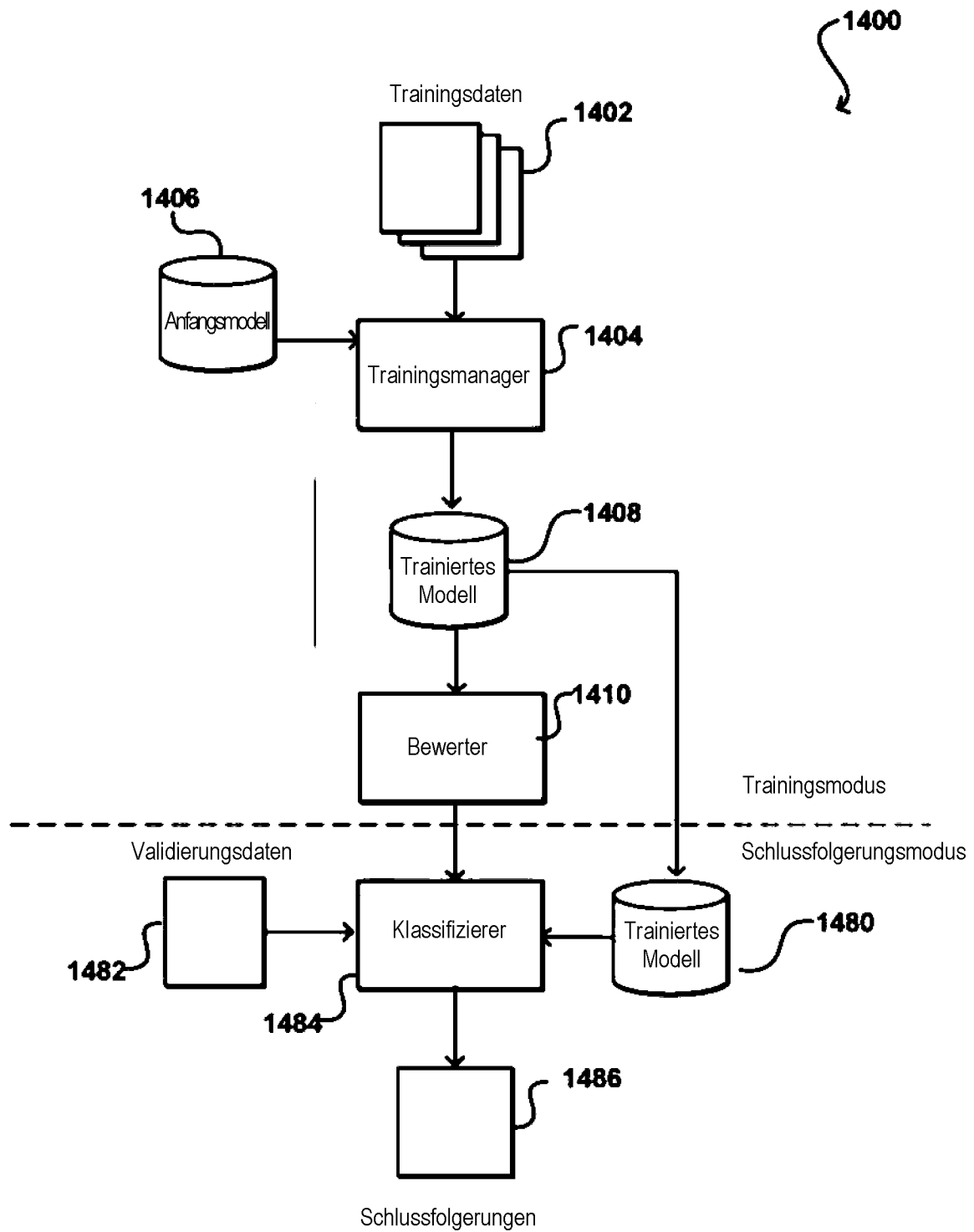


FIG. 14

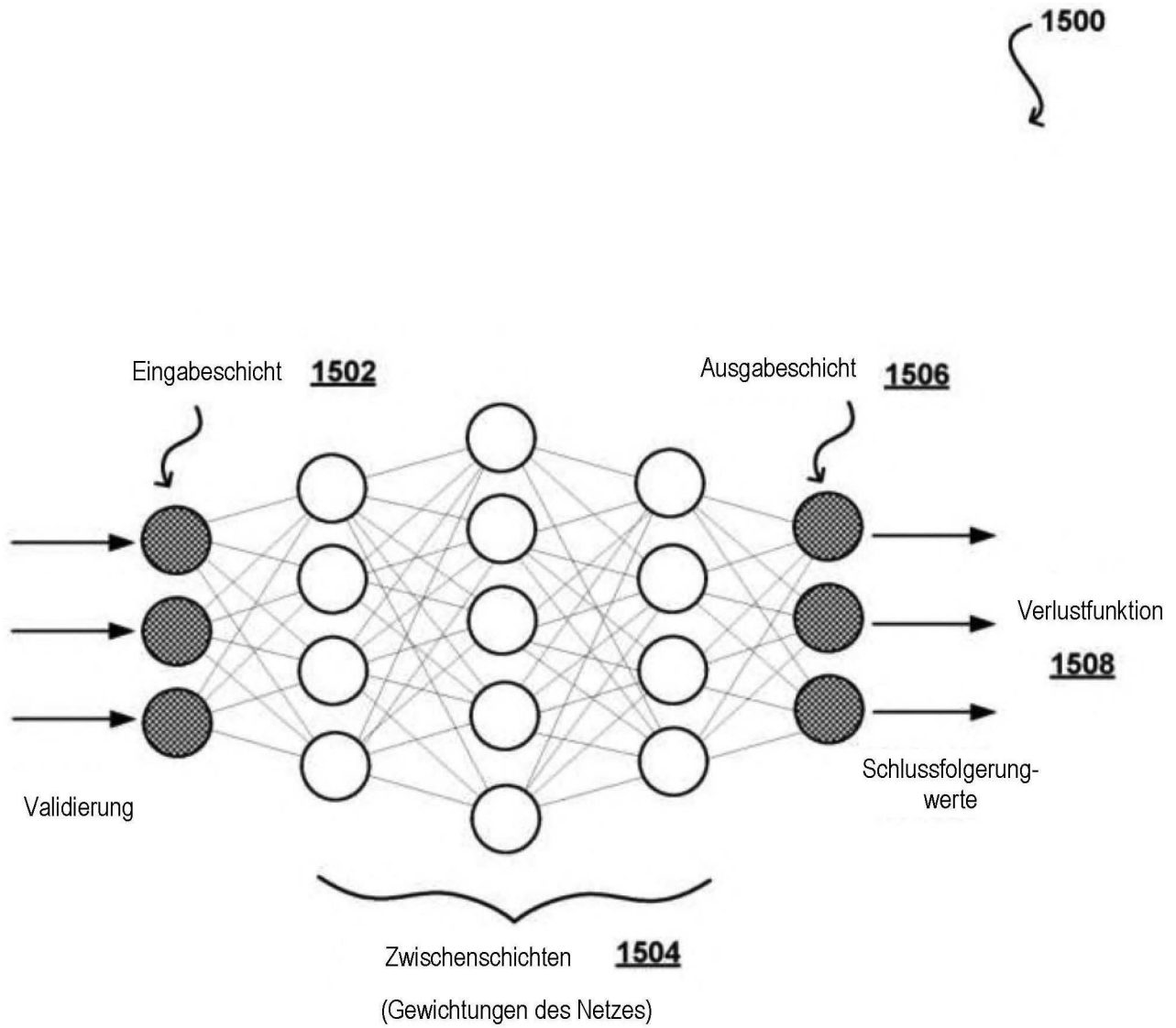


FIG. 15

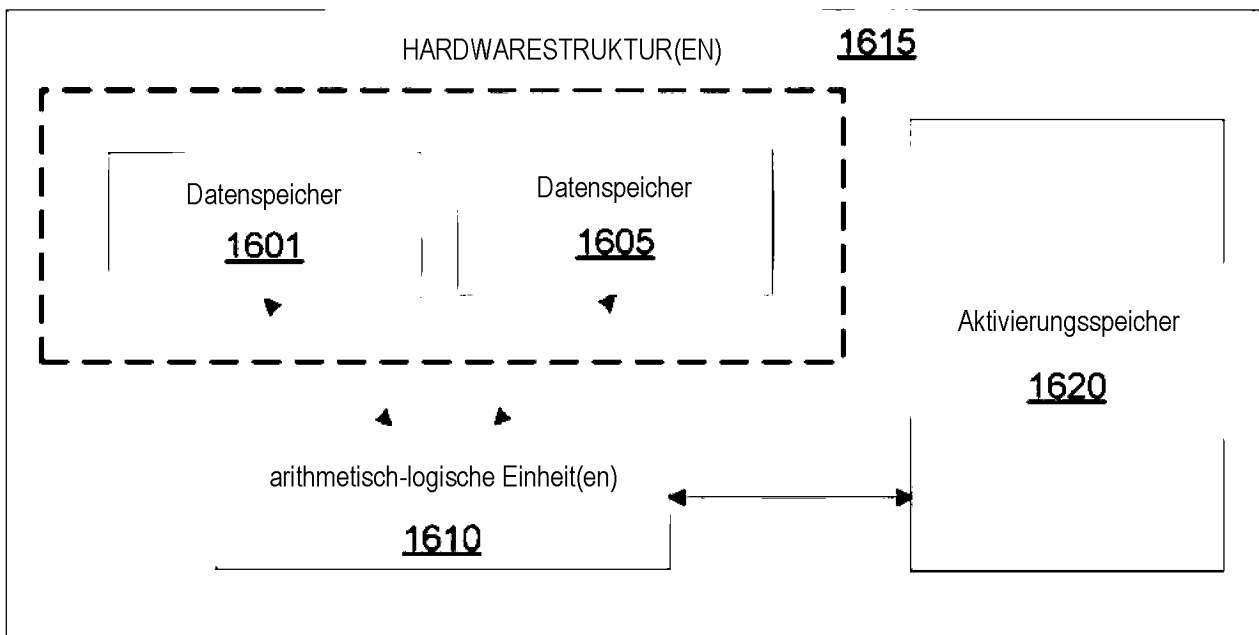


FIG. 16A

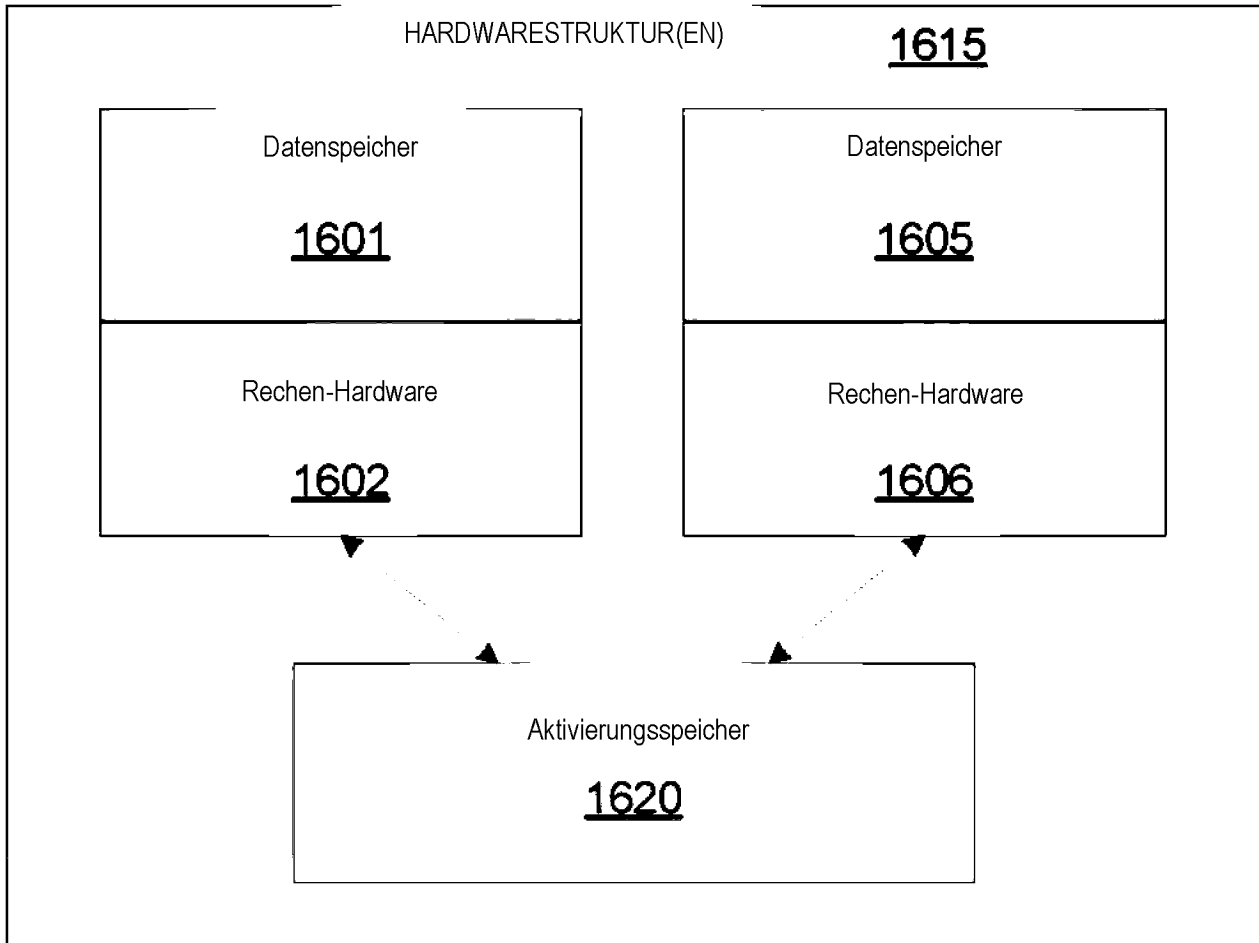


FIG. 16B

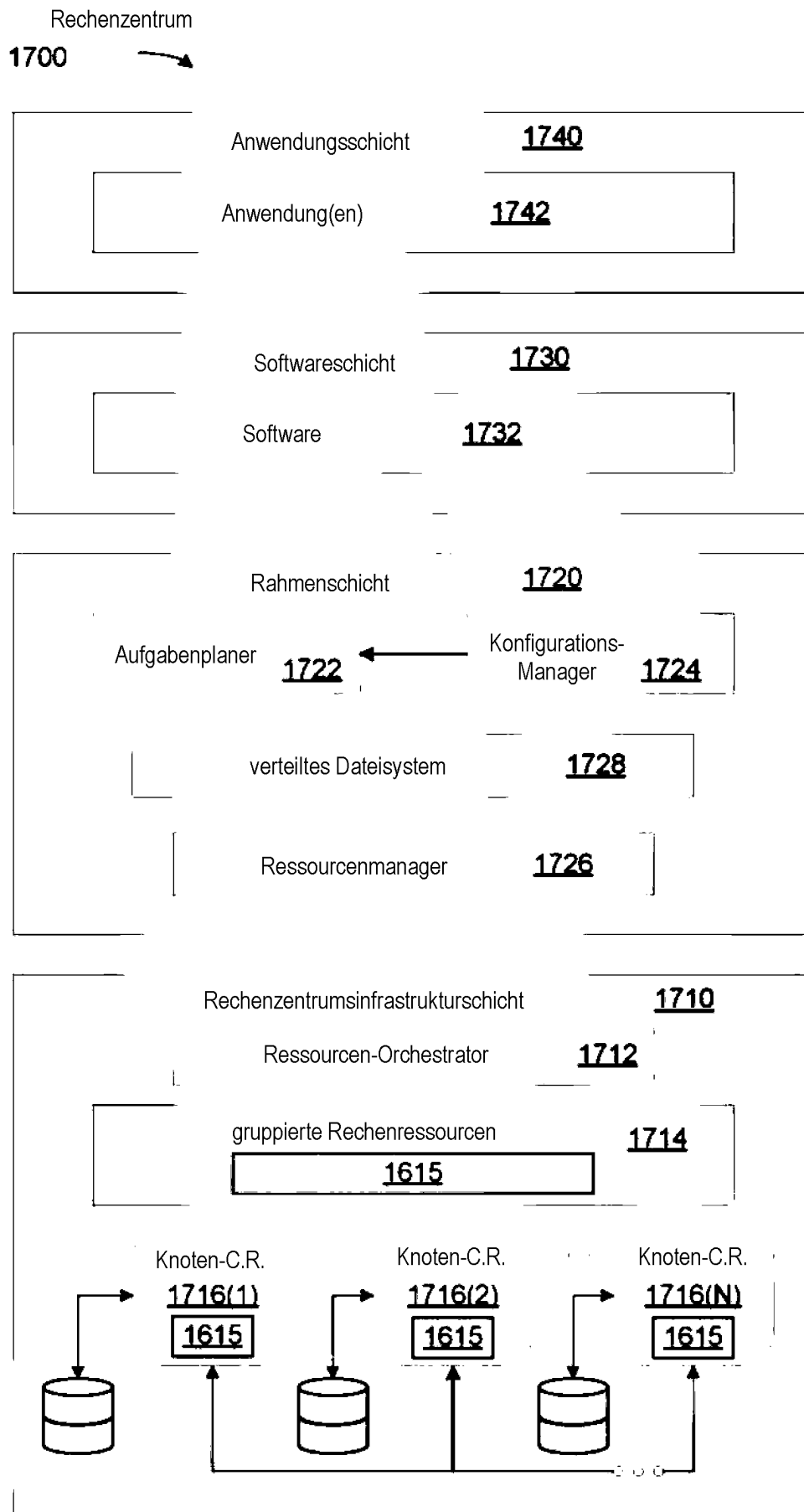


FIG. 17

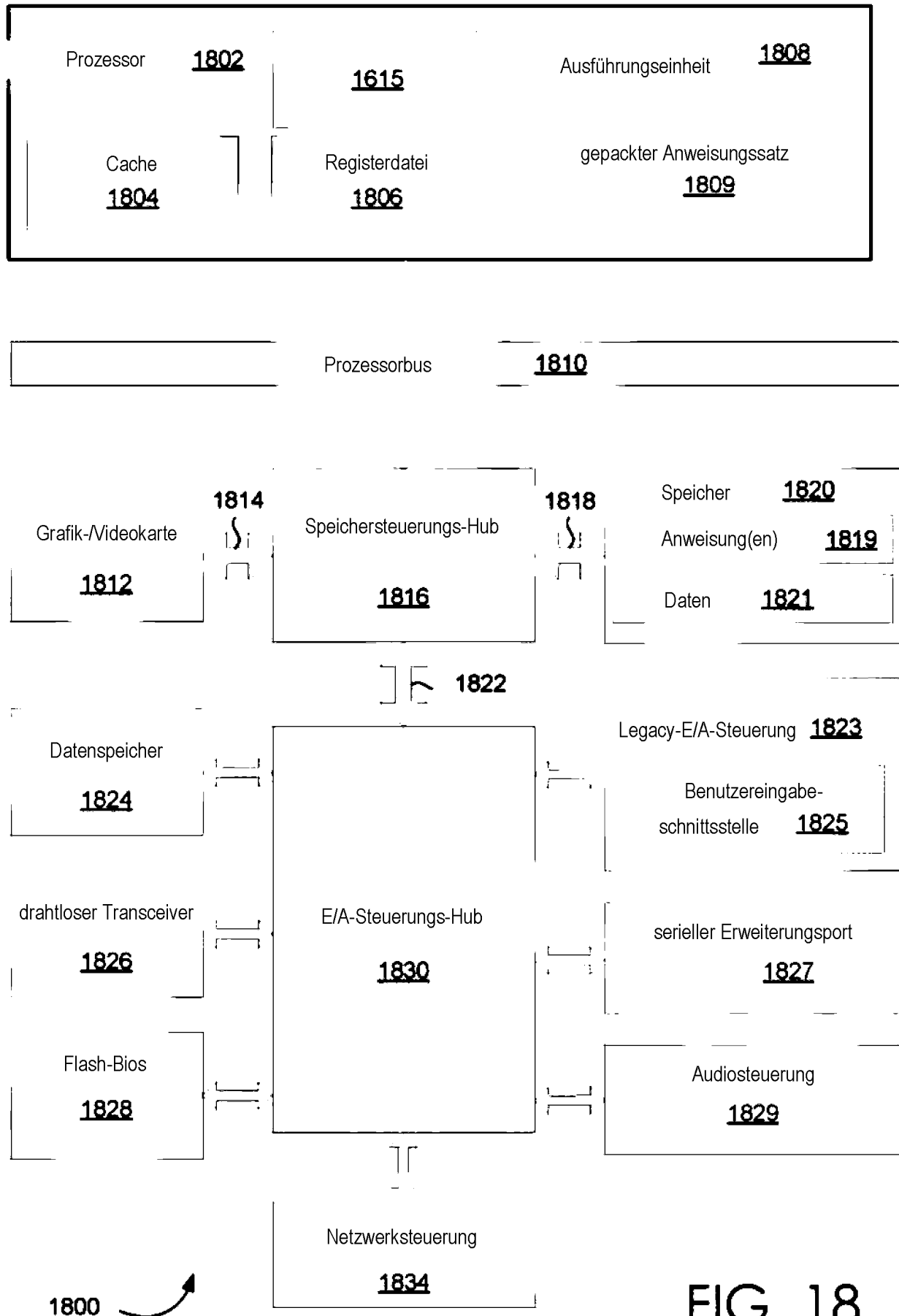


FIG. 18

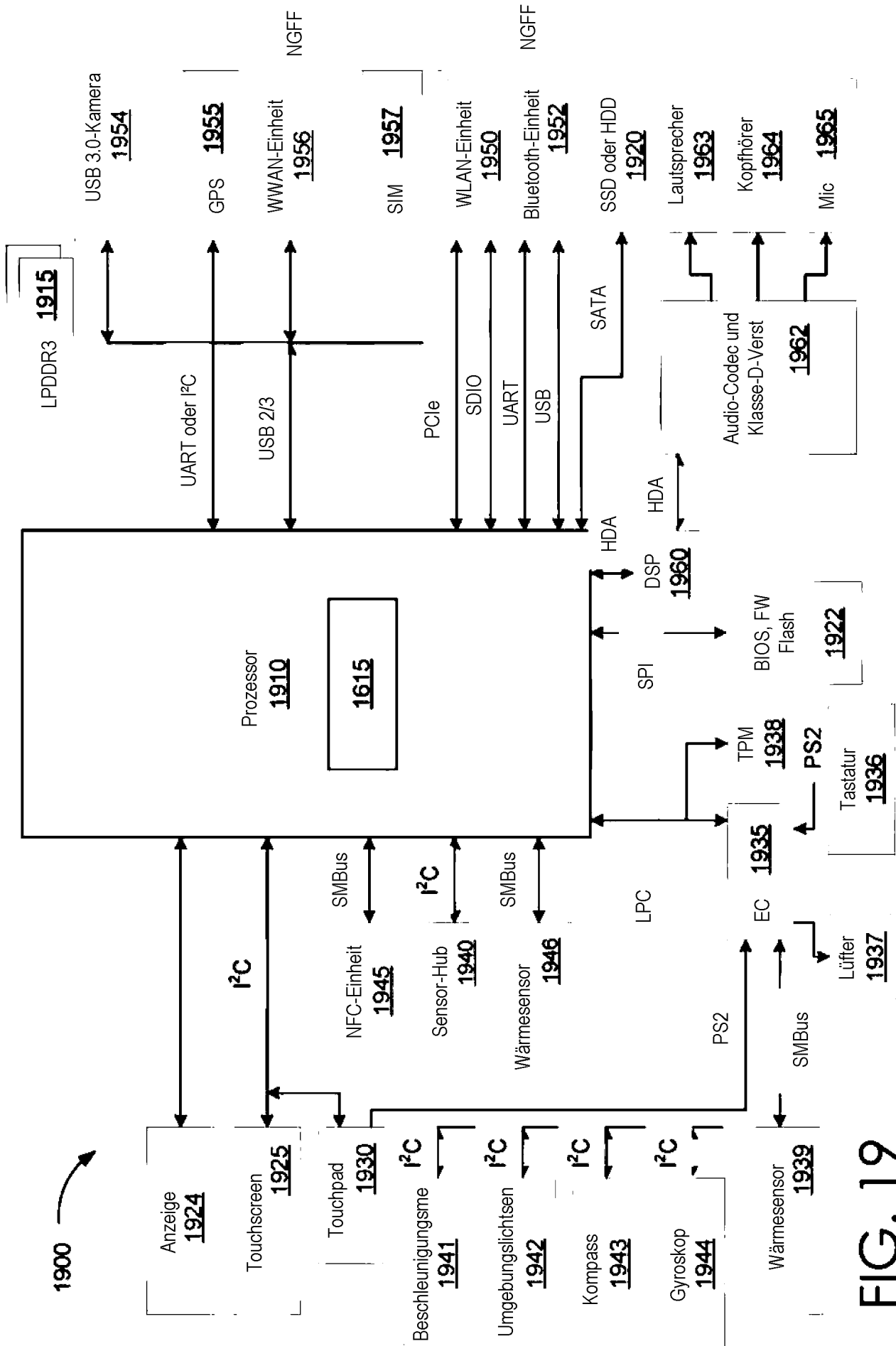


FIG. 19

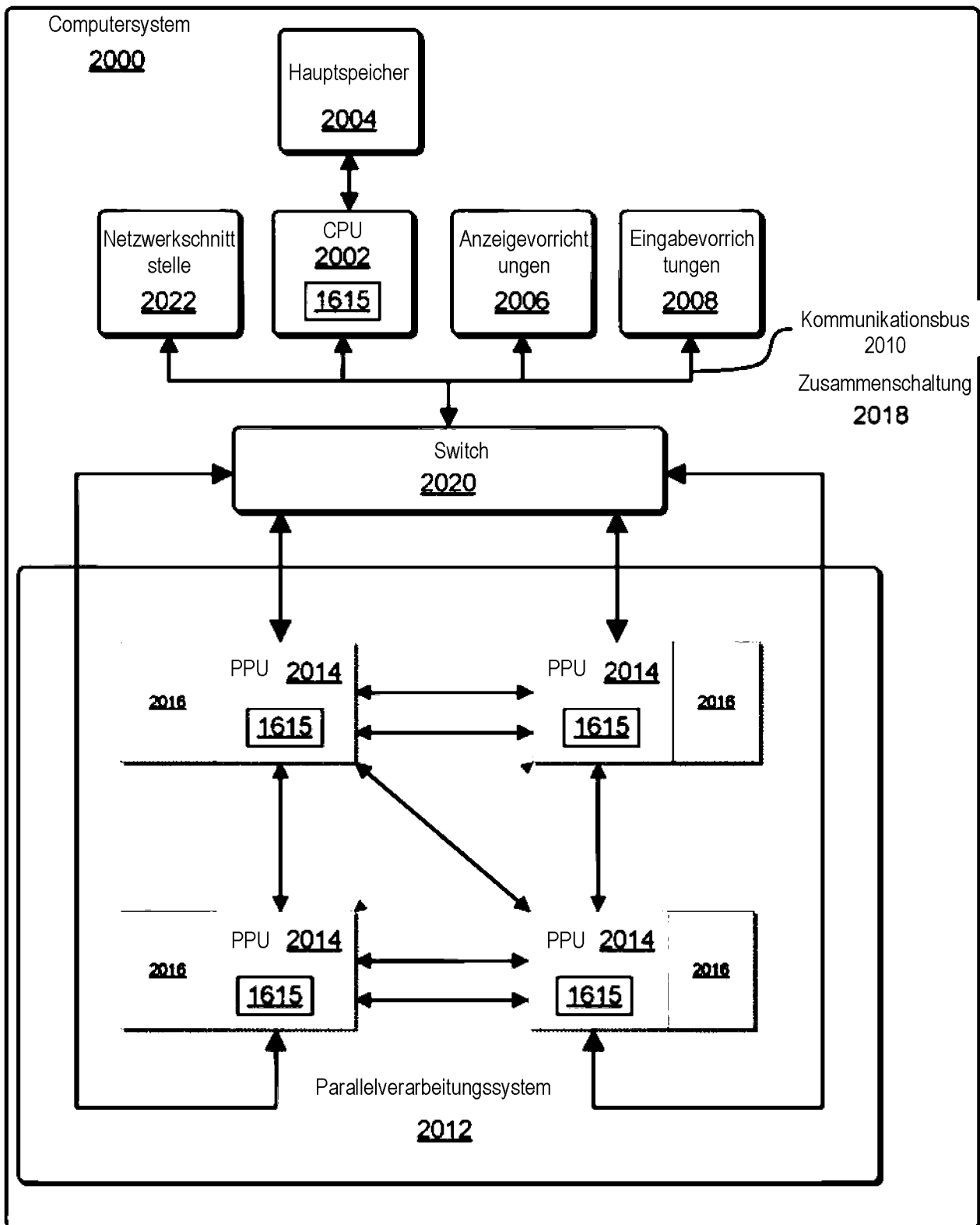


FIG. 20

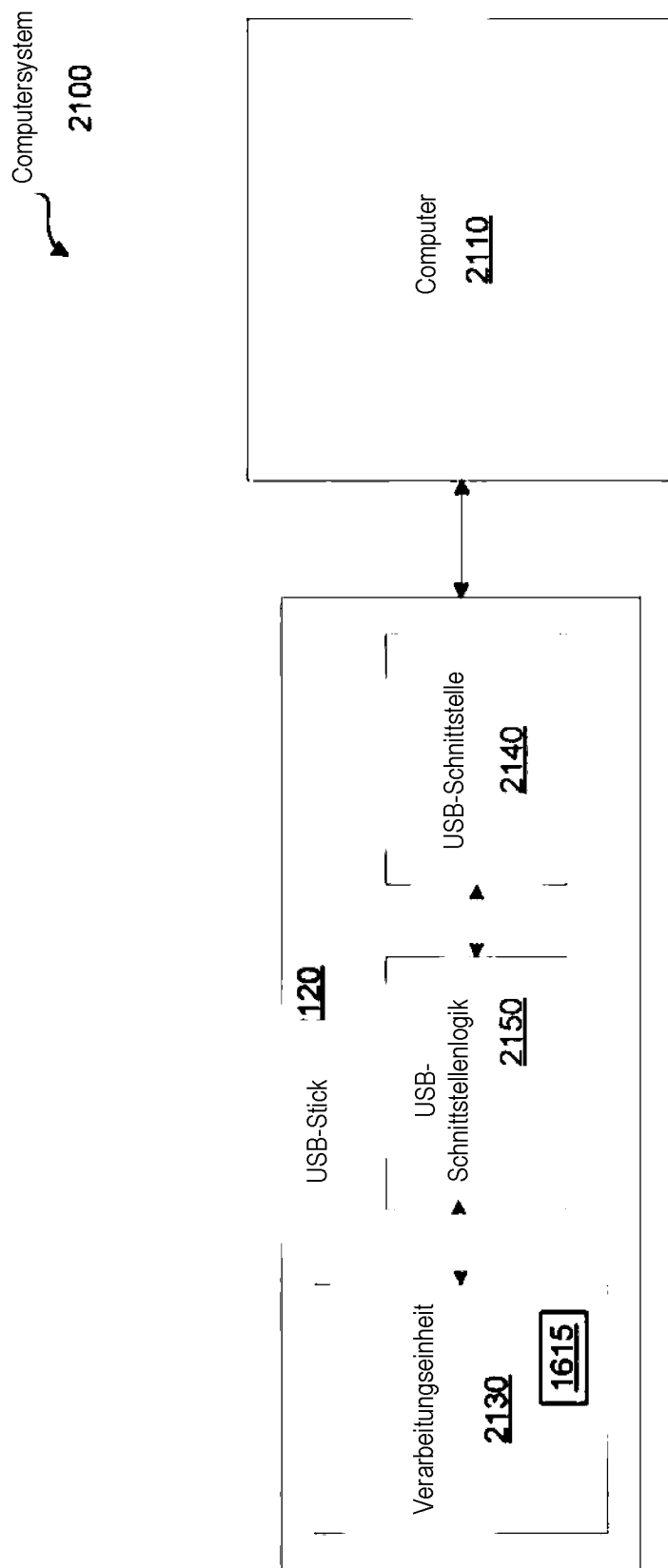


FIG. 21

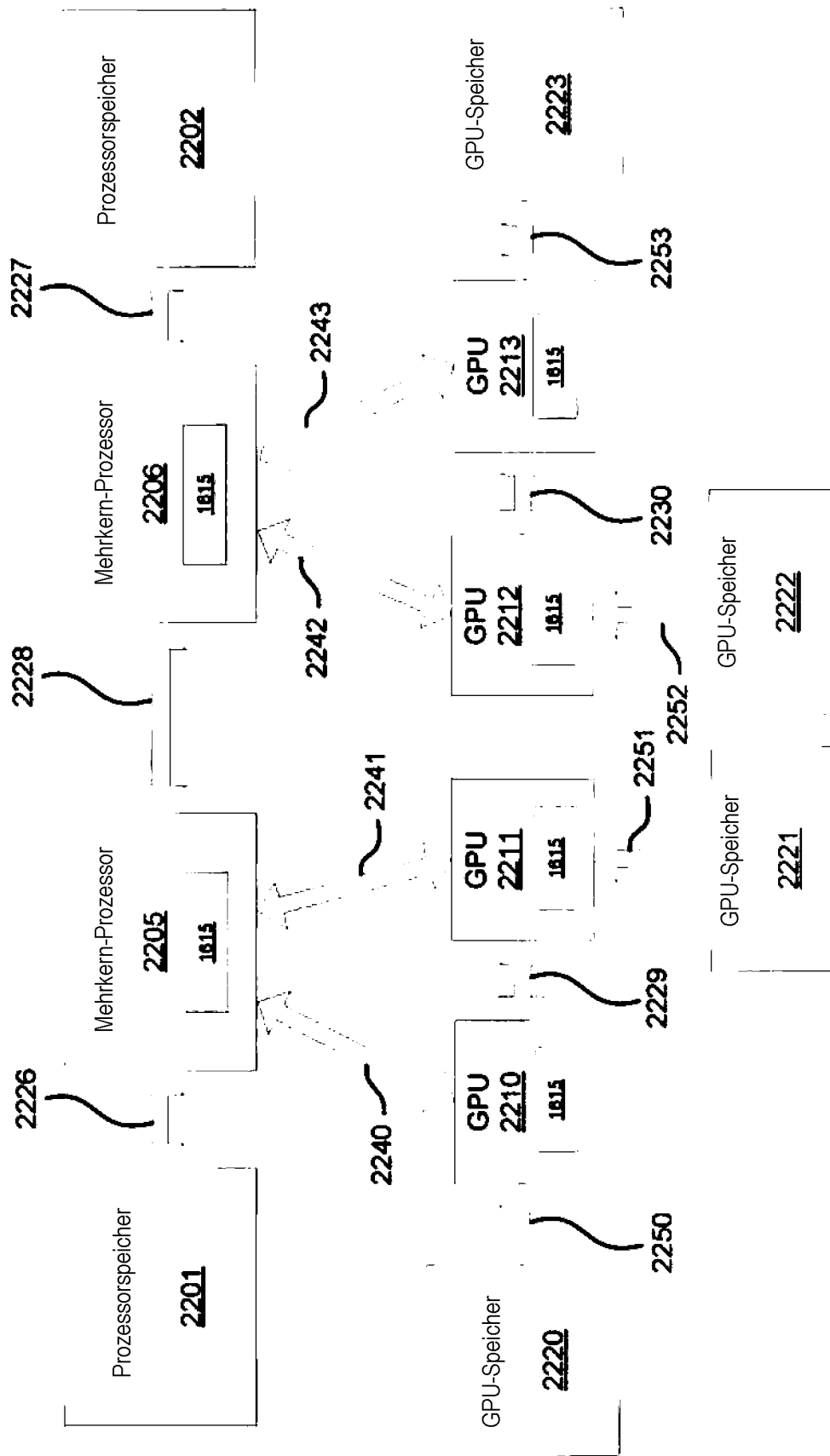


FIG. 22A

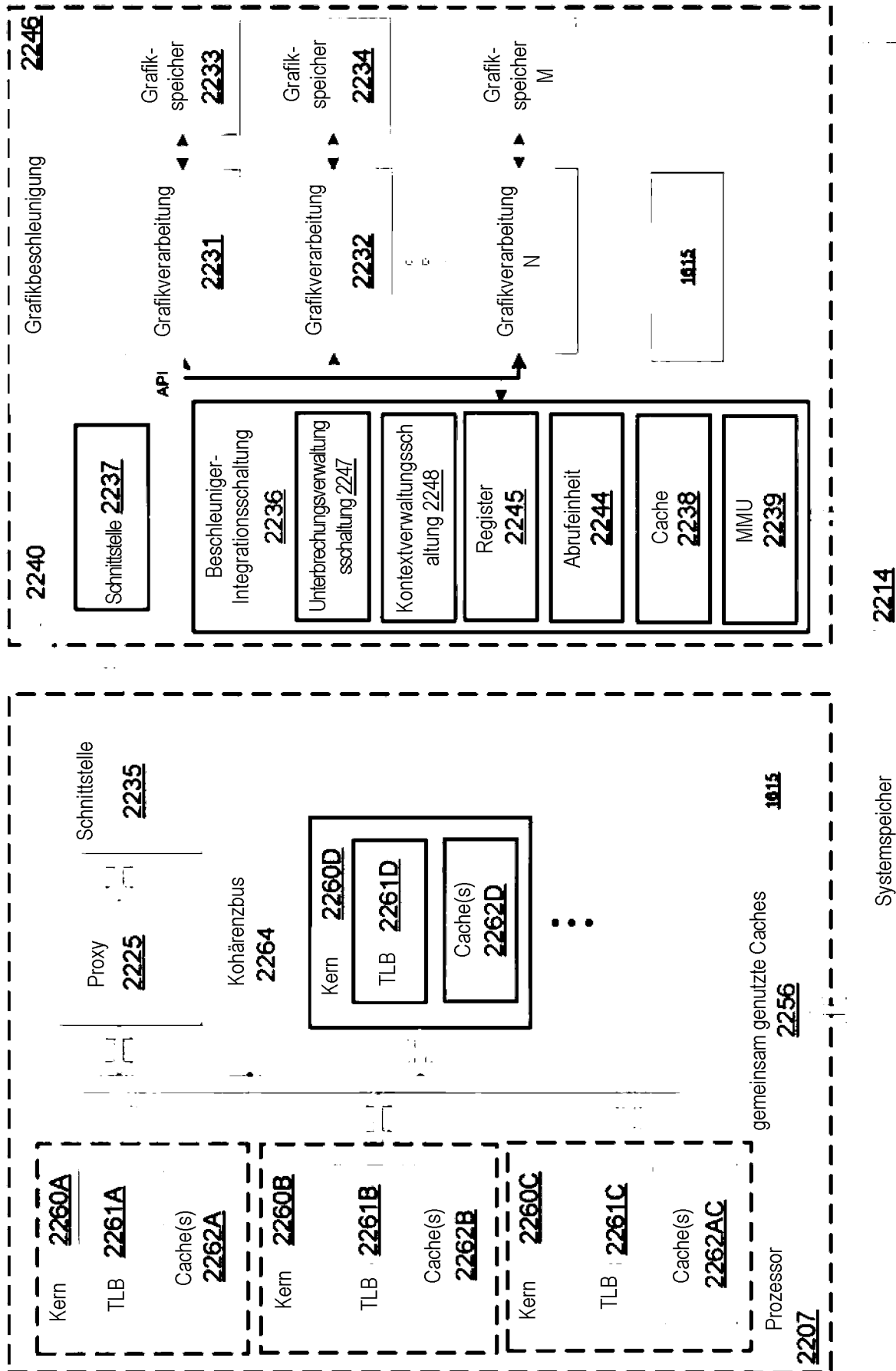


FIG. 22B

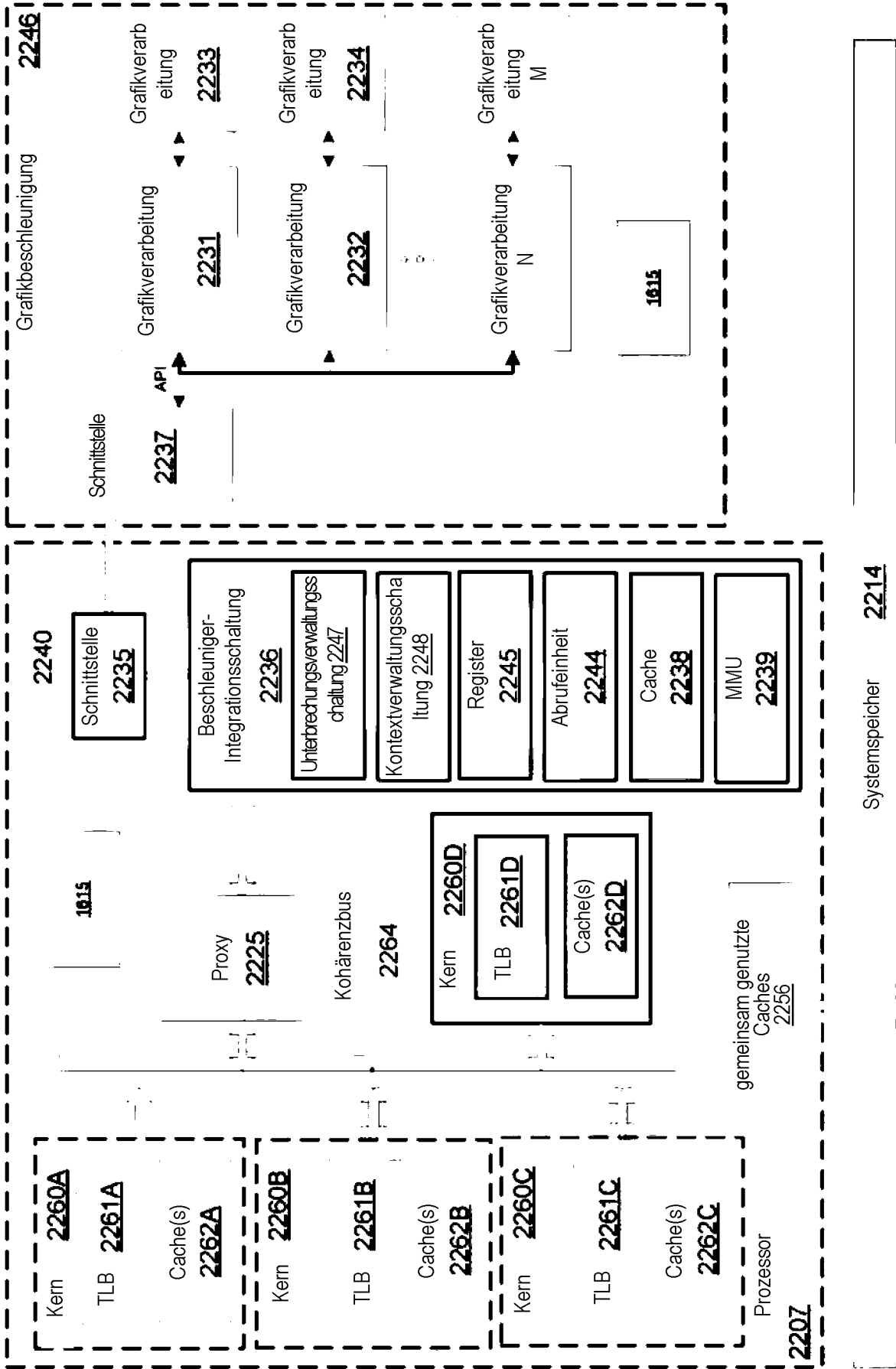


FIG. 22C

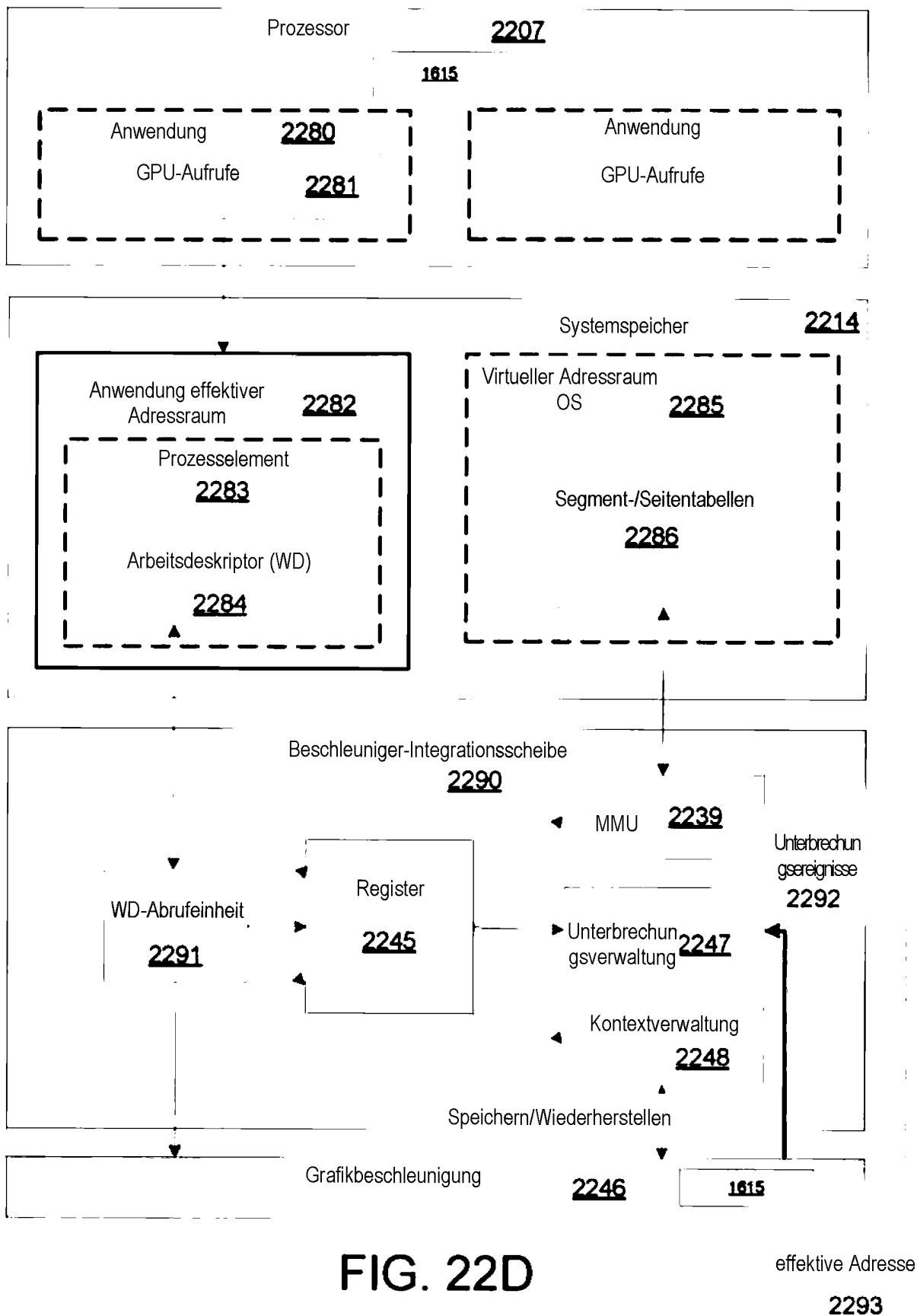
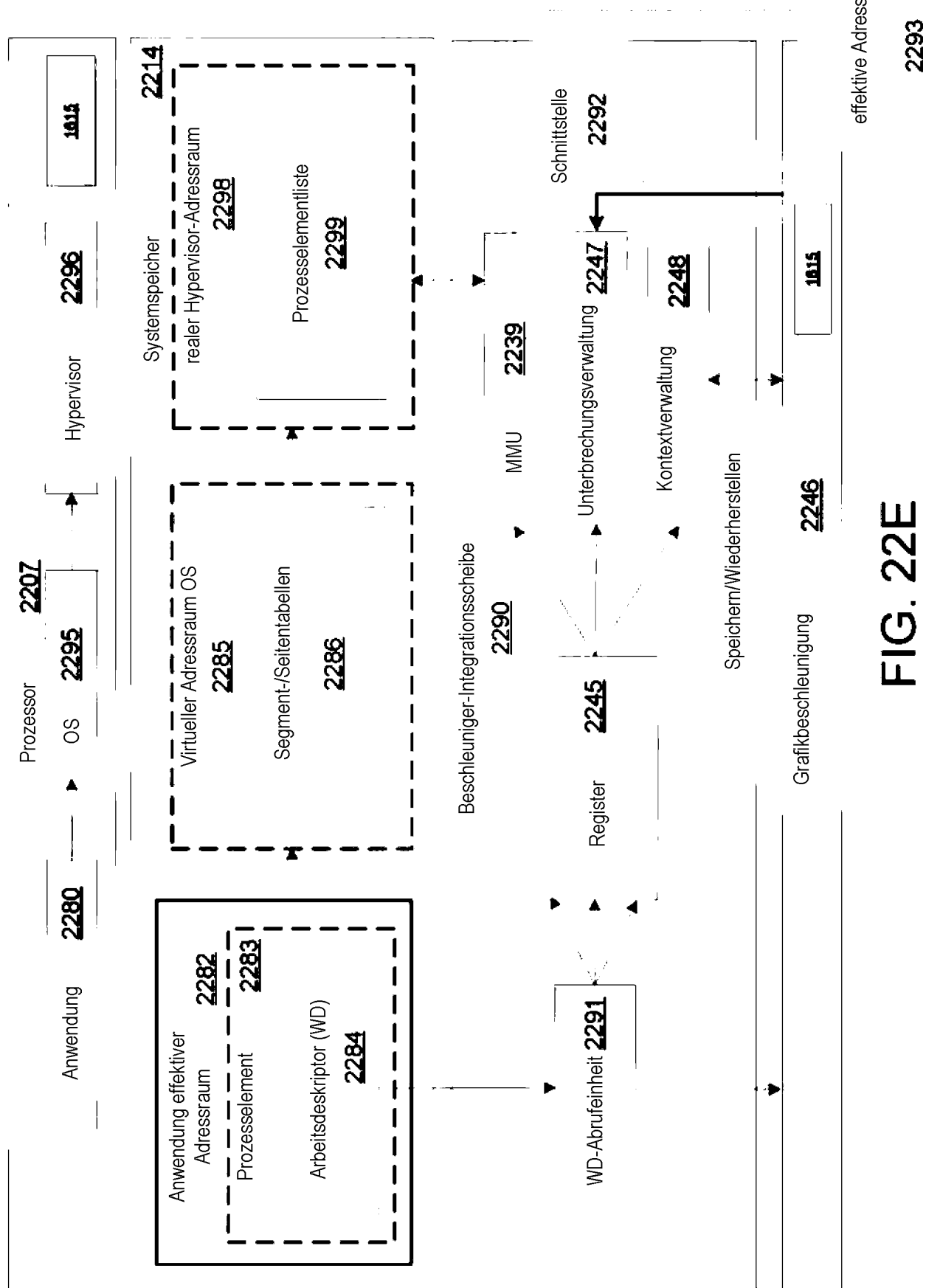


FIG. 22D



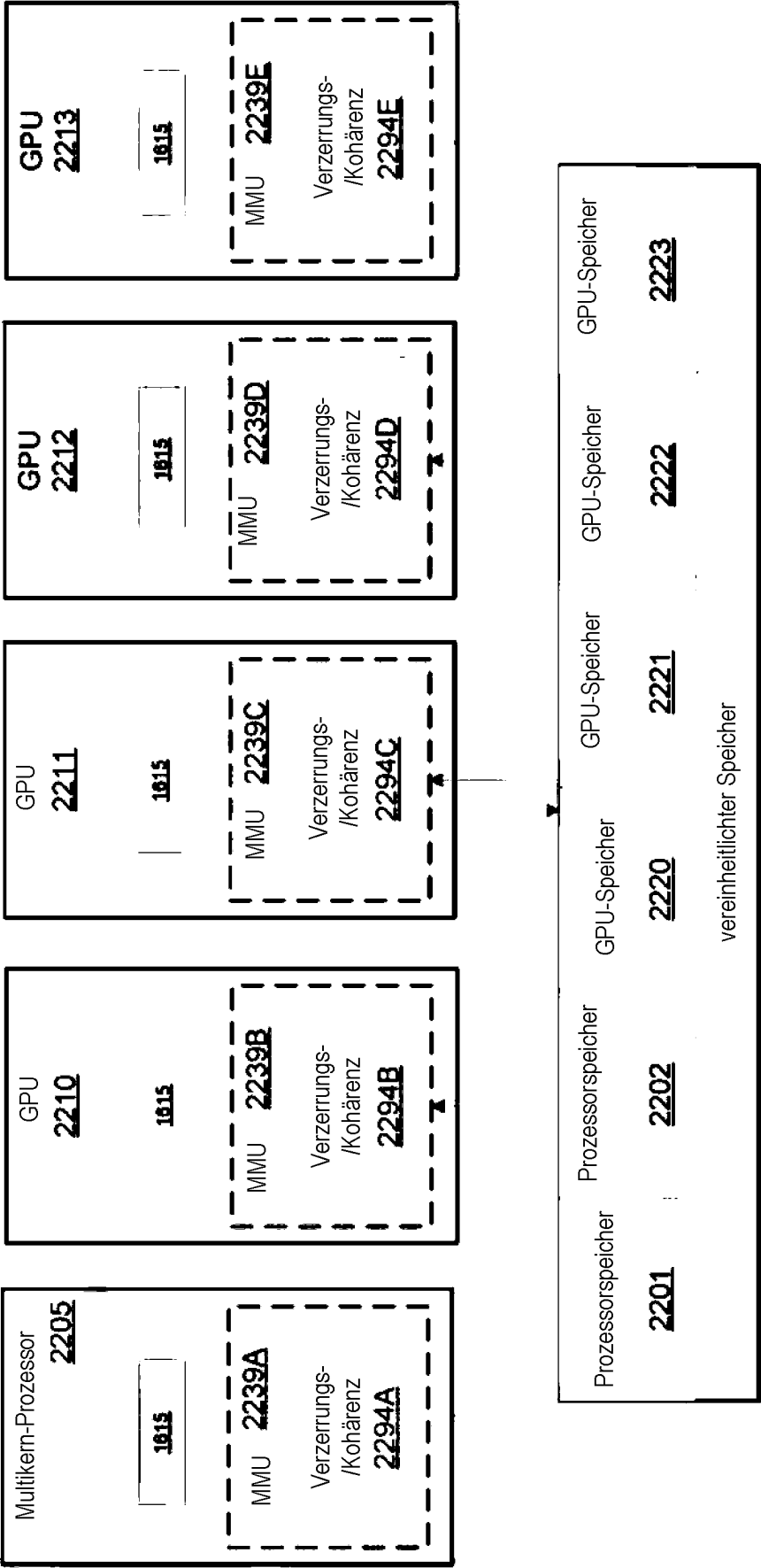


FIG. 22F

SoC integrierte Schaltung

2300

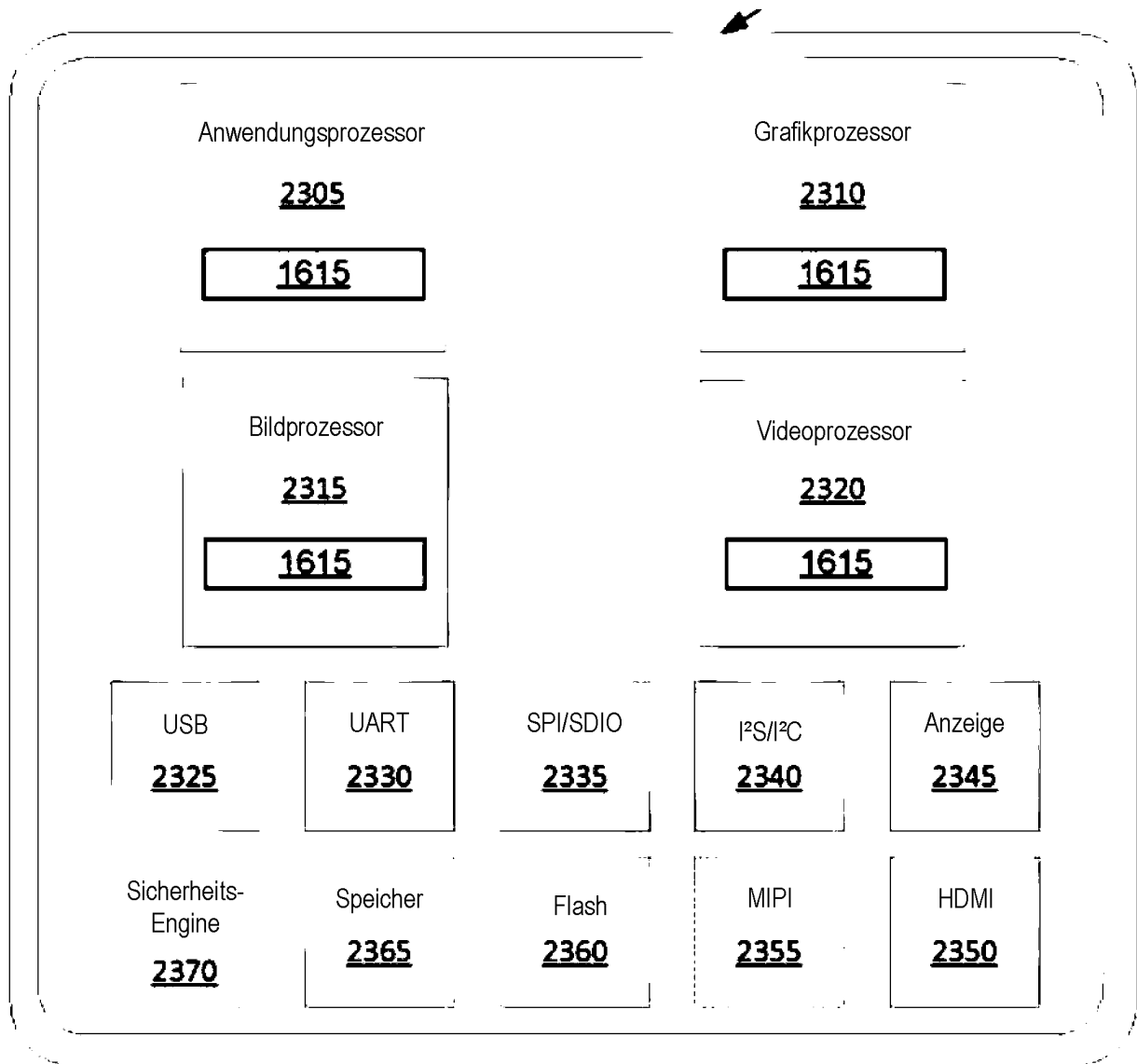


FIG. 23

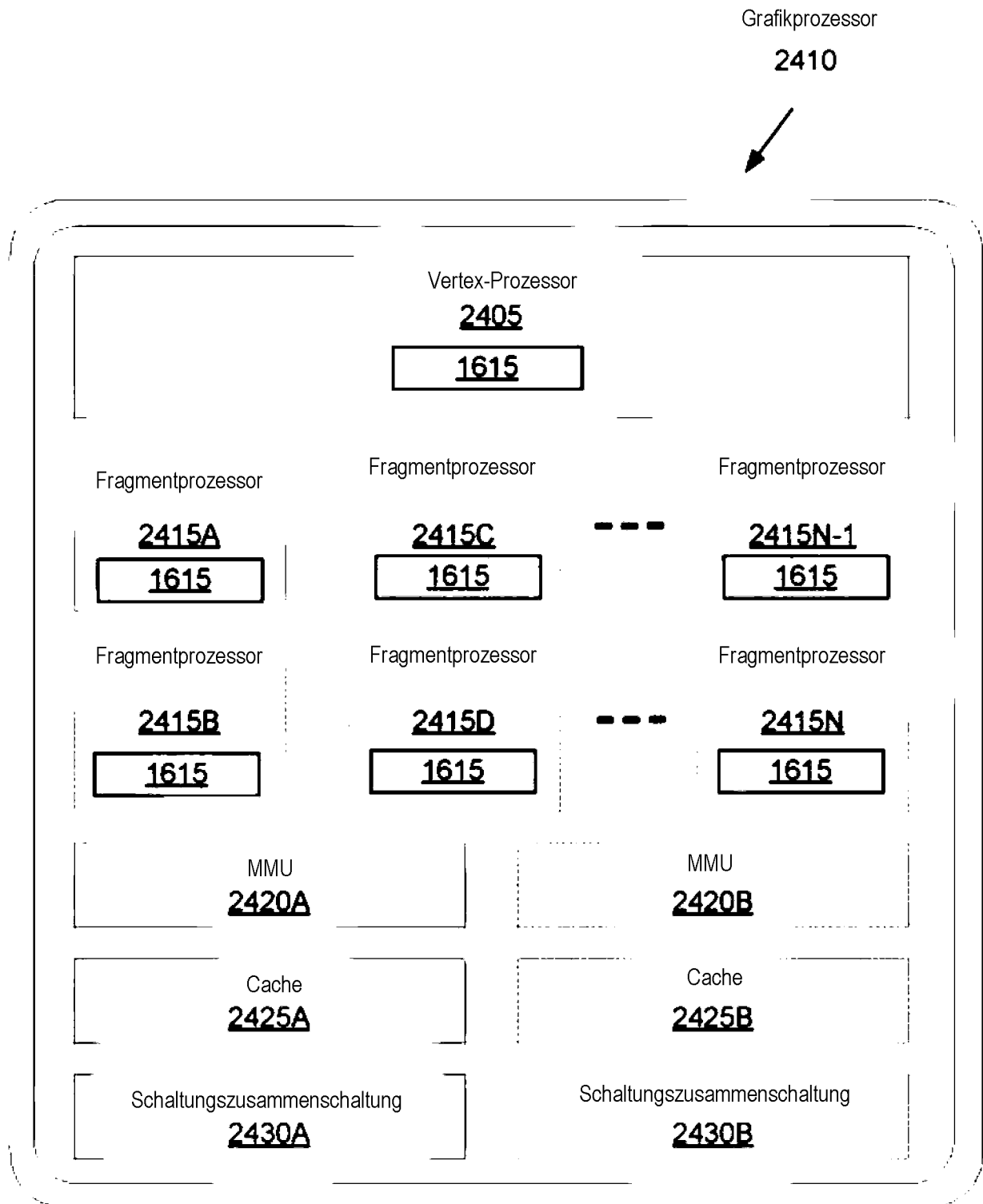


FIG. 24A

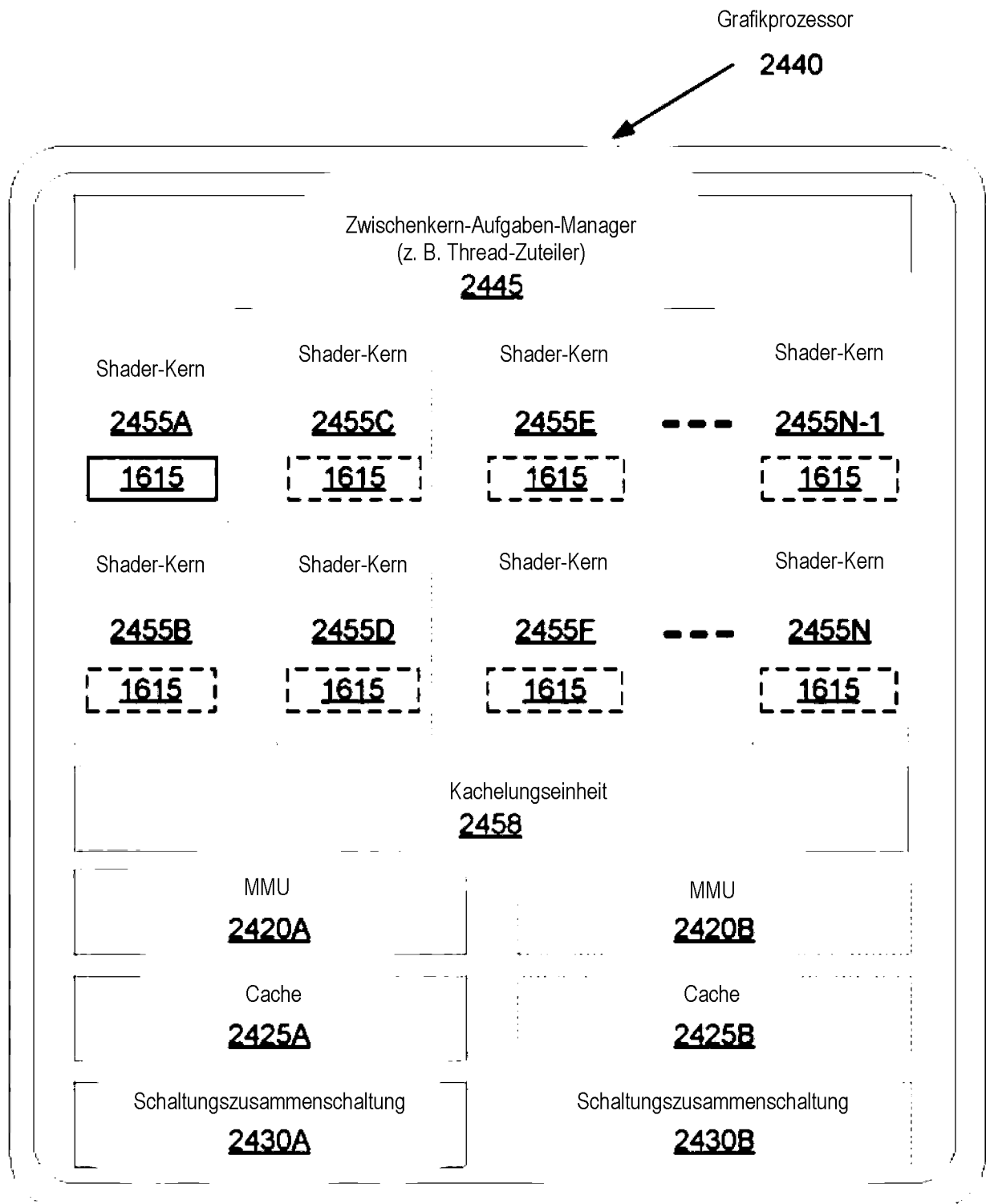


FIG. 24B

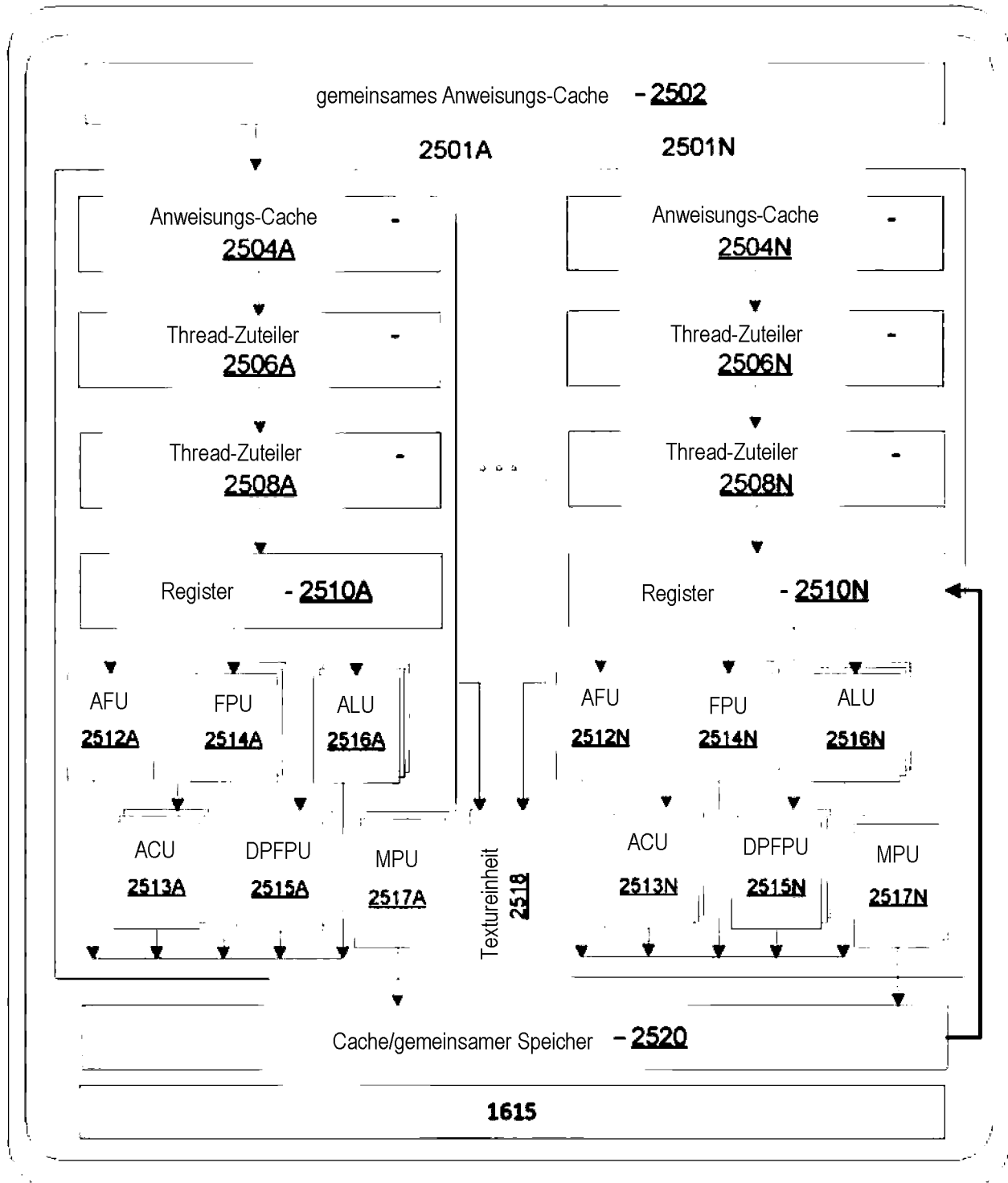
Grafikprozessor
2500

FIG. 25A

2530

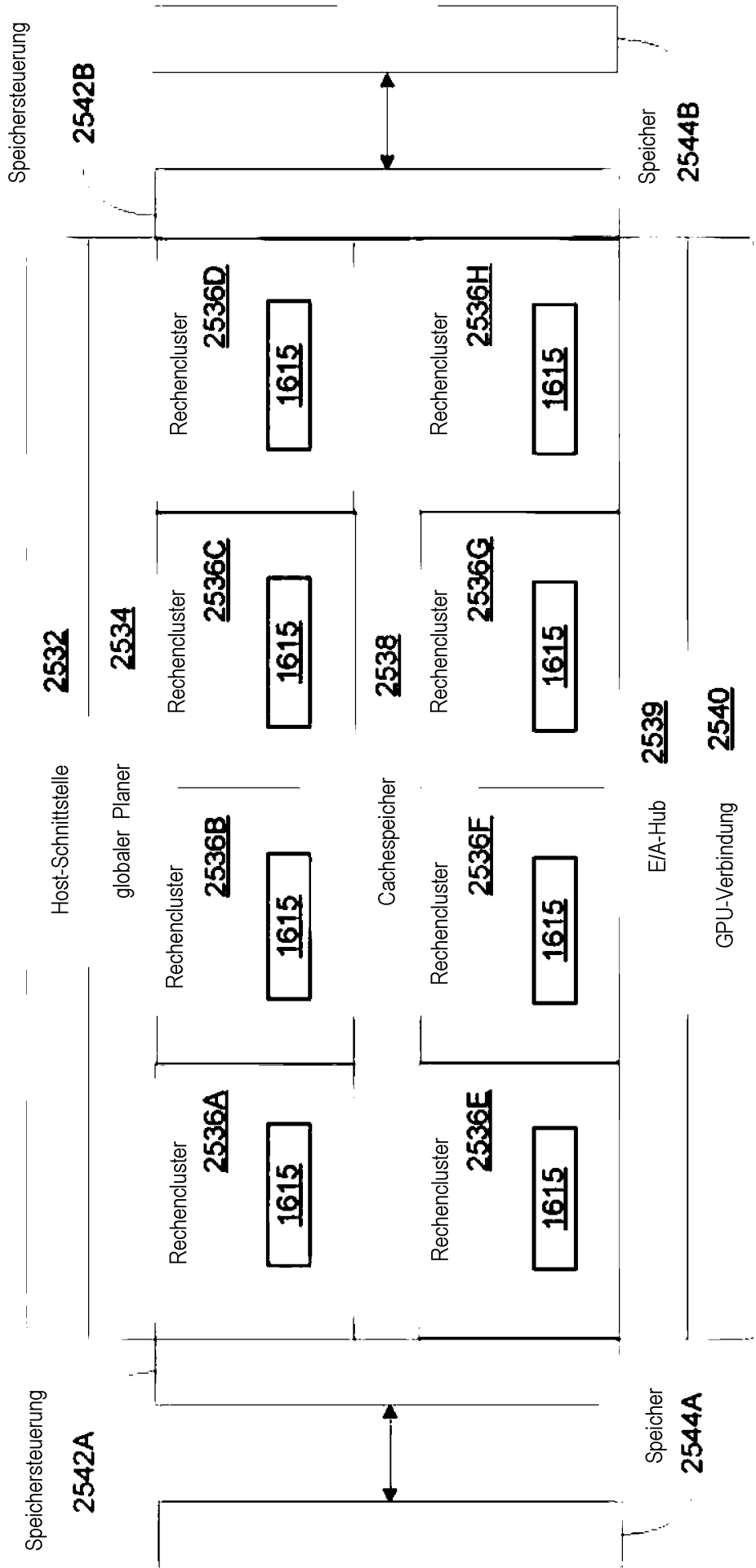


FIG. 25B

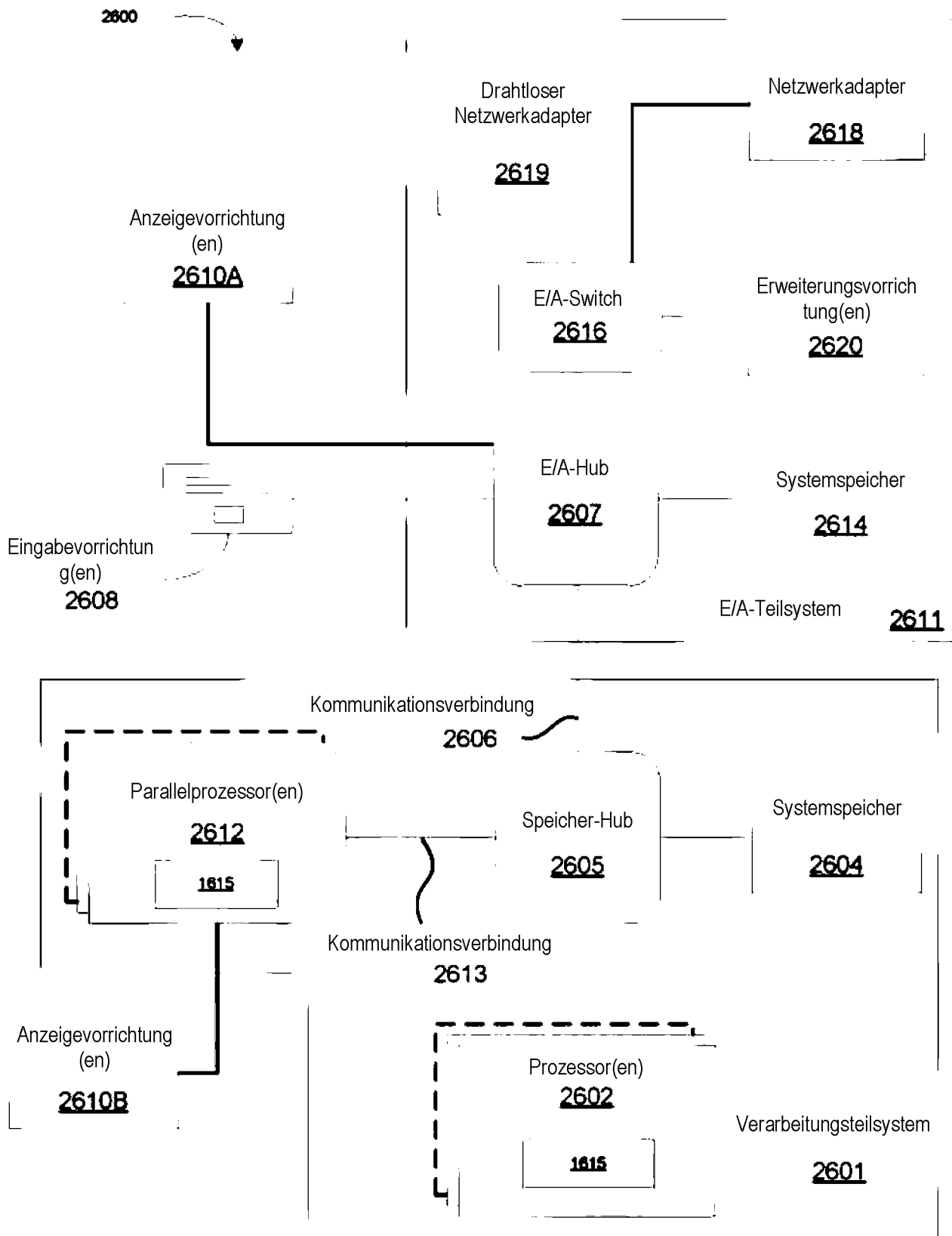


FIG. 26

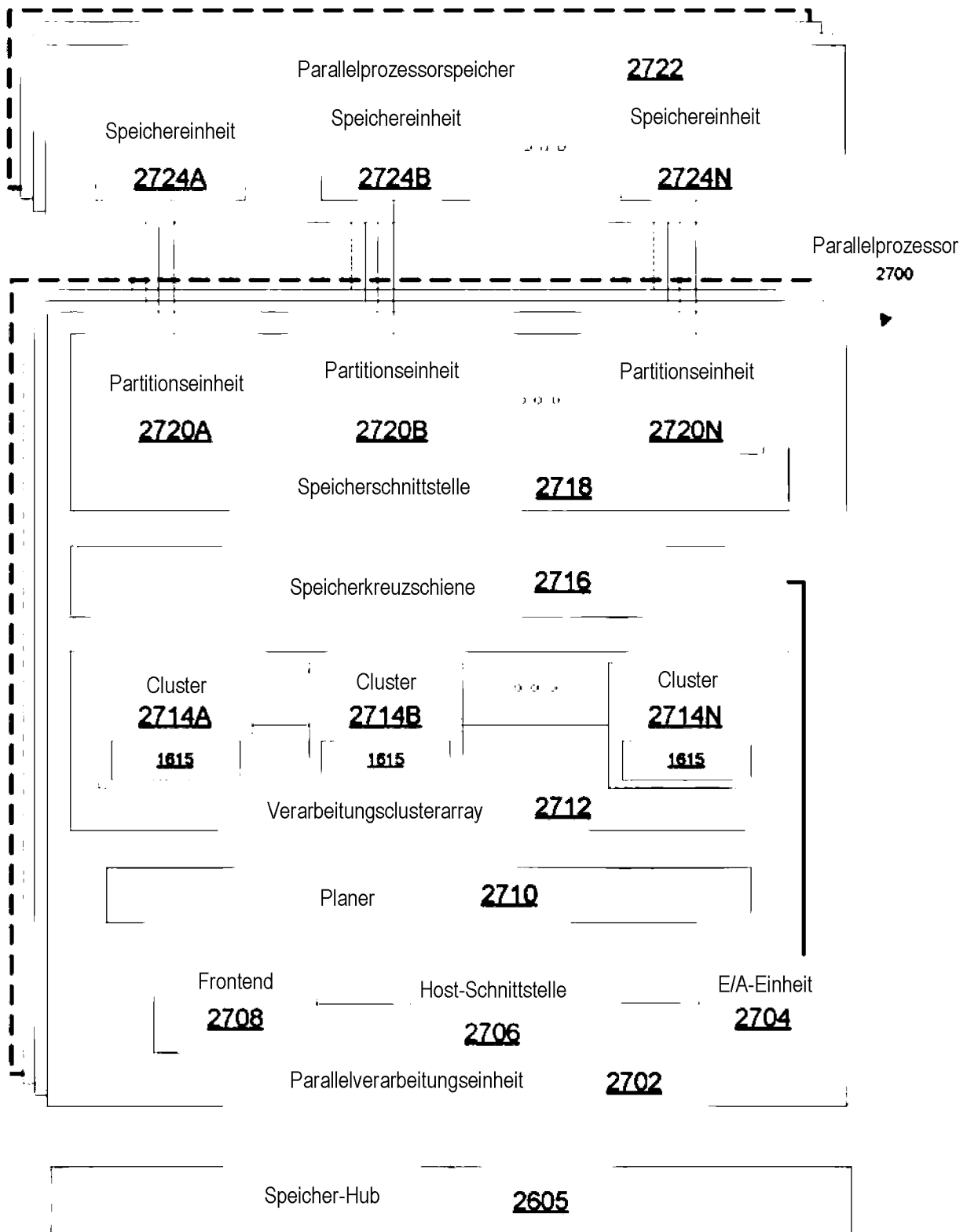


FIG. 27A

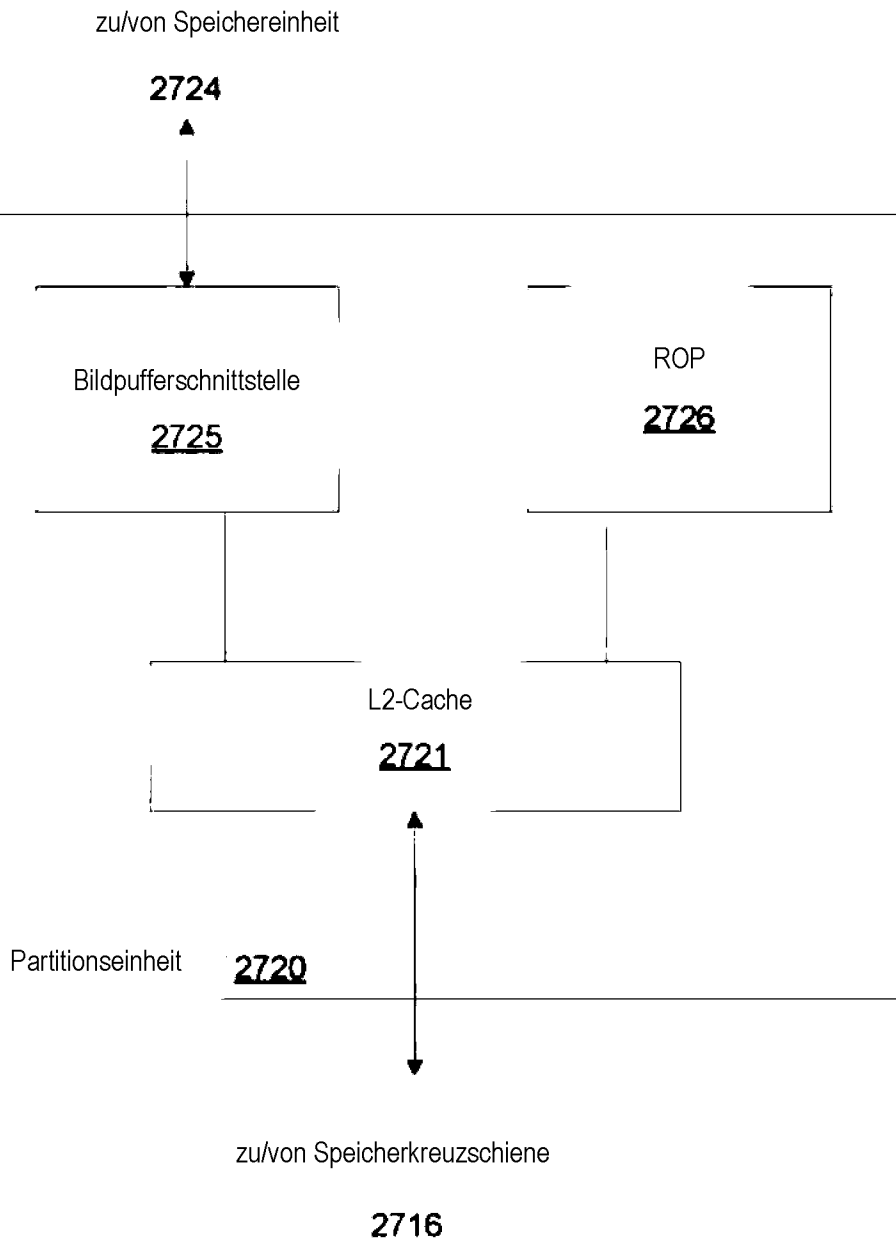


FIG. 27B

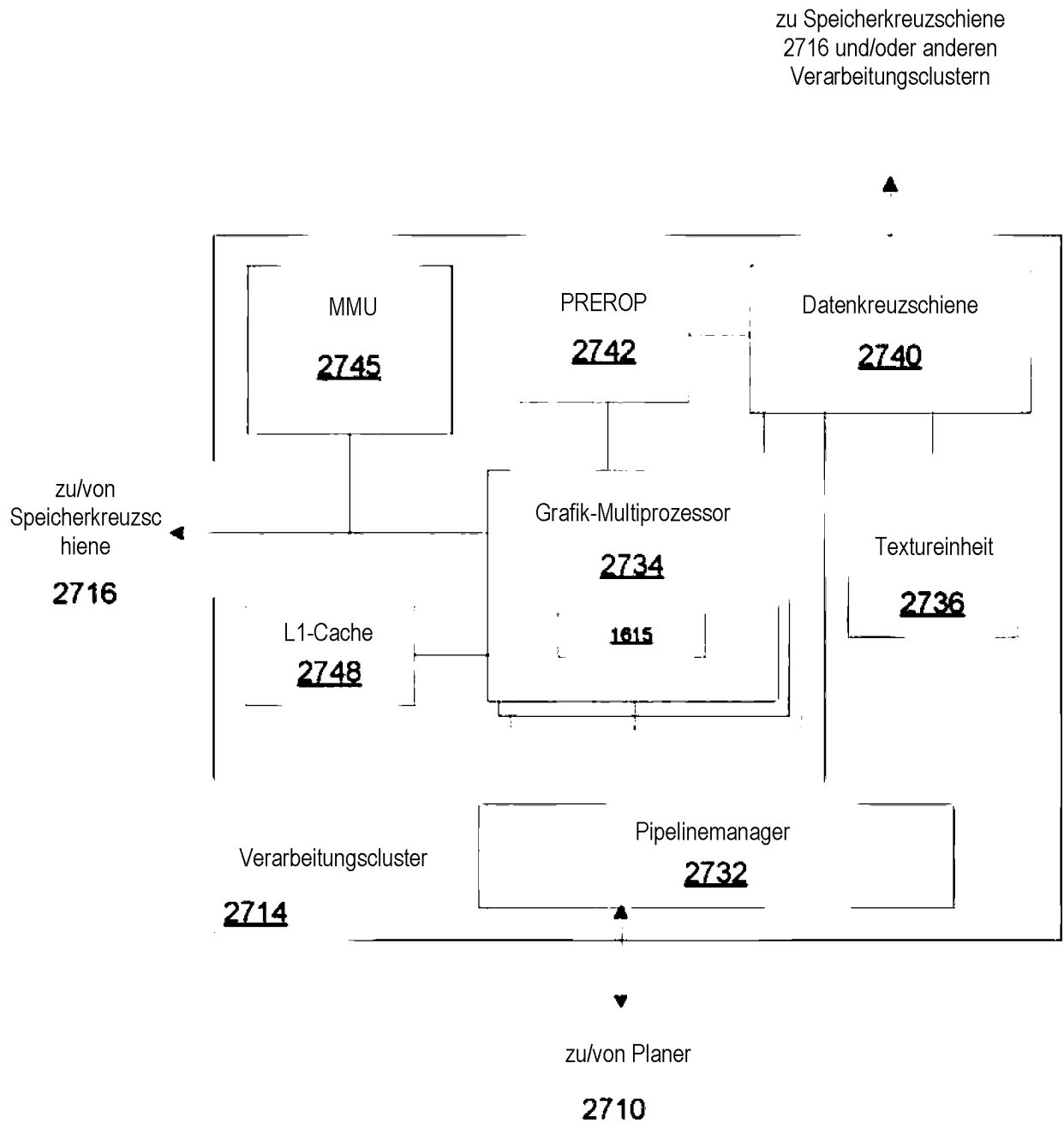


FIG. 27C

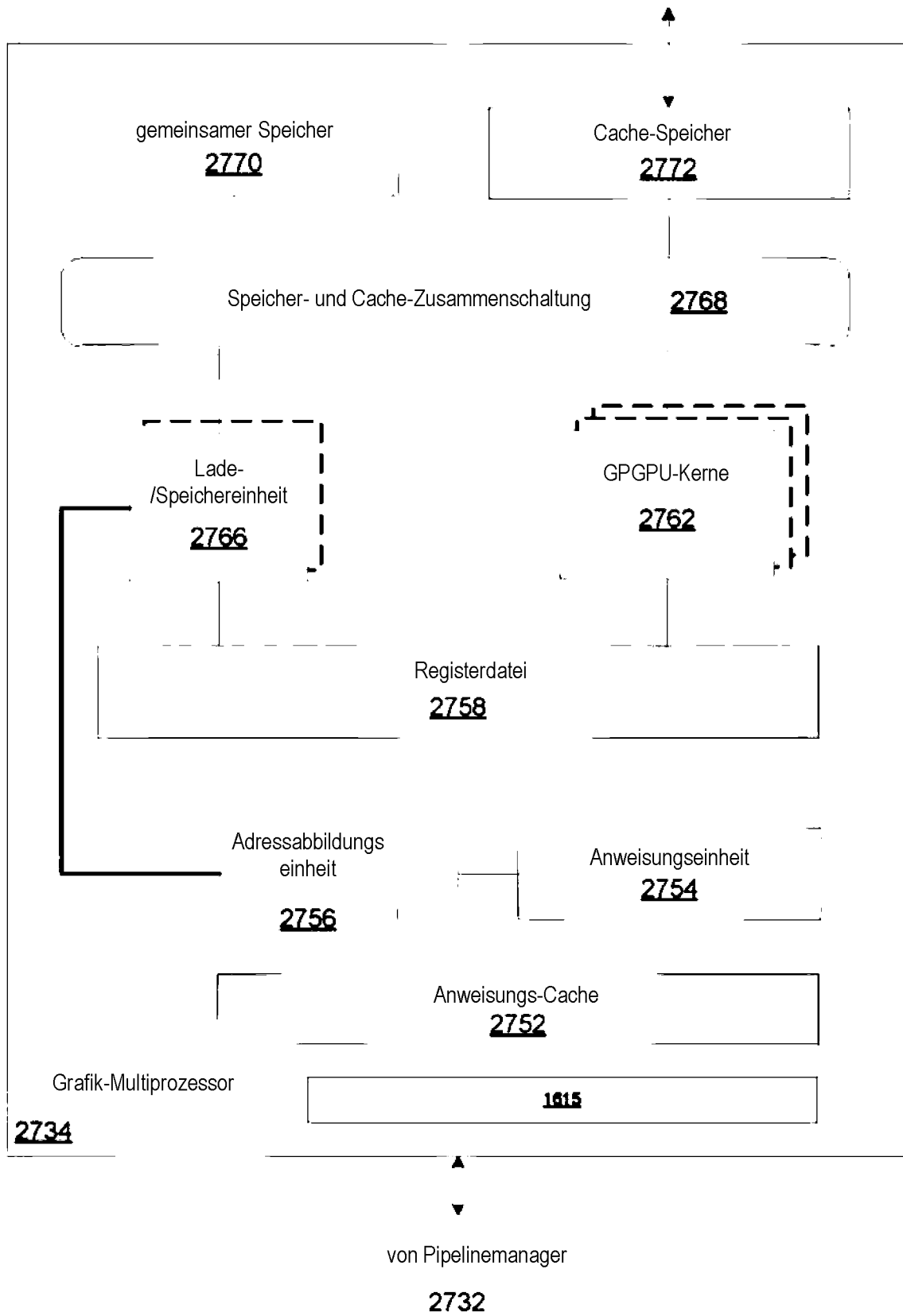


FIG. 27D

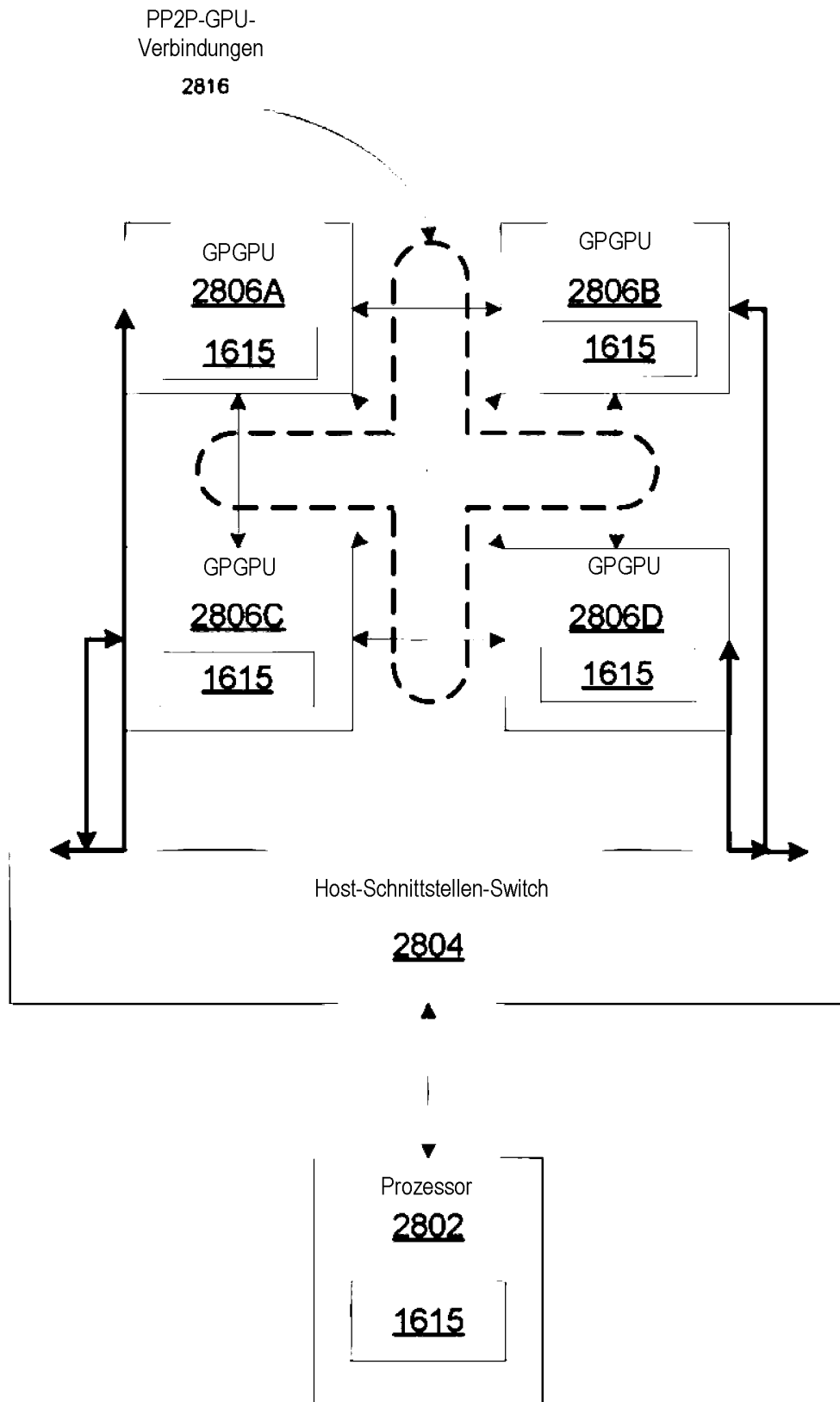


FIG. 28

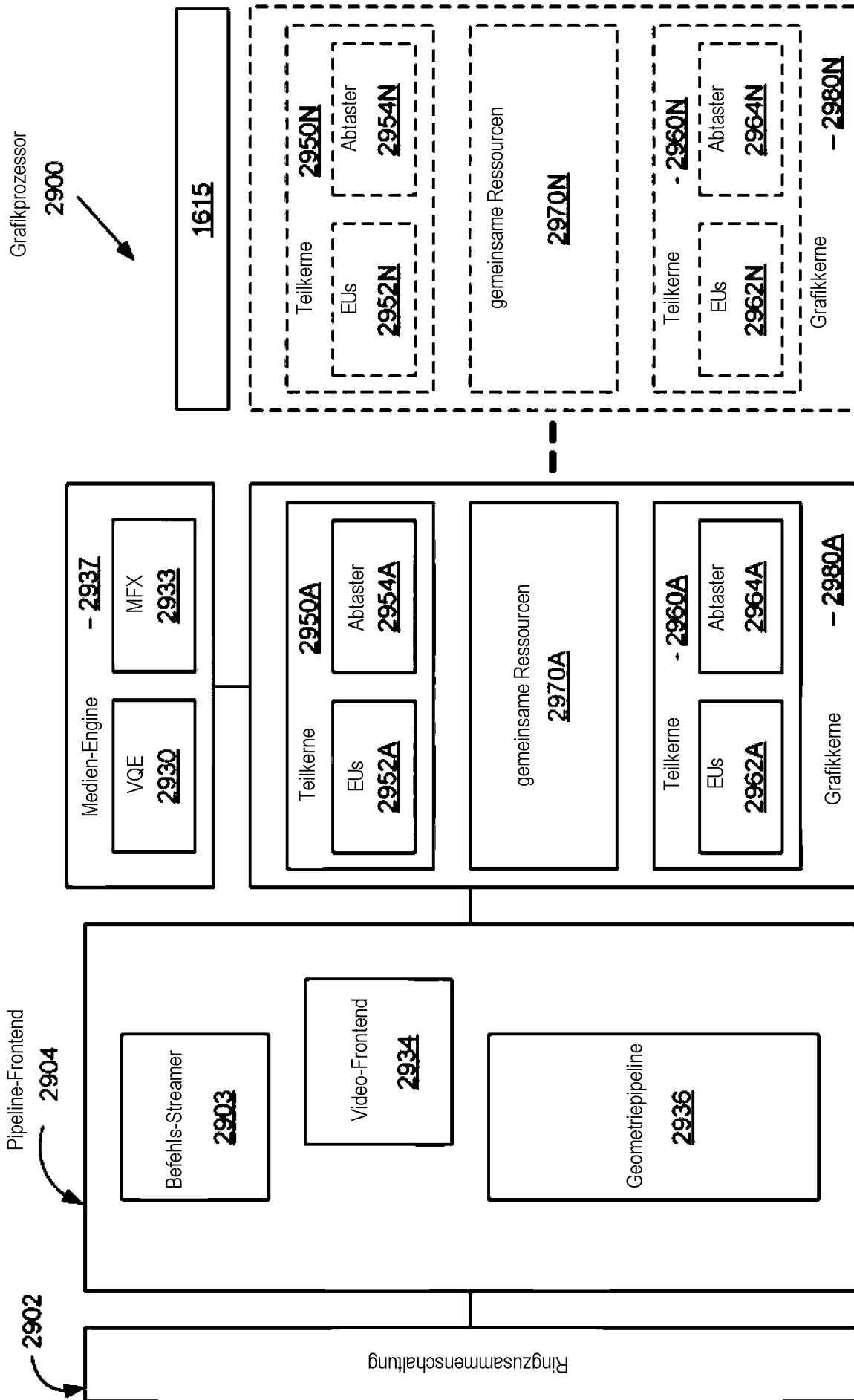


FIG. 29

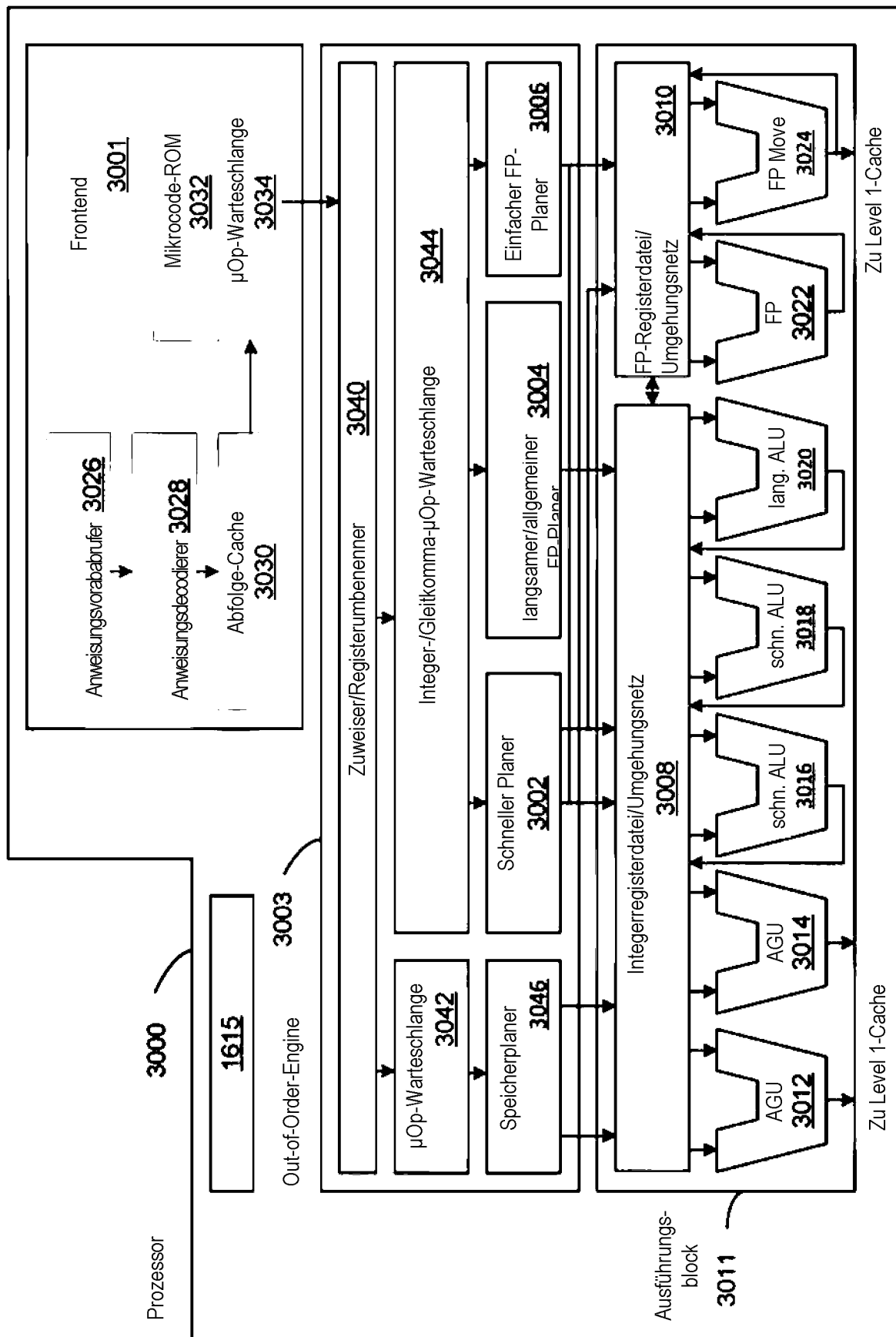


FIG. 30

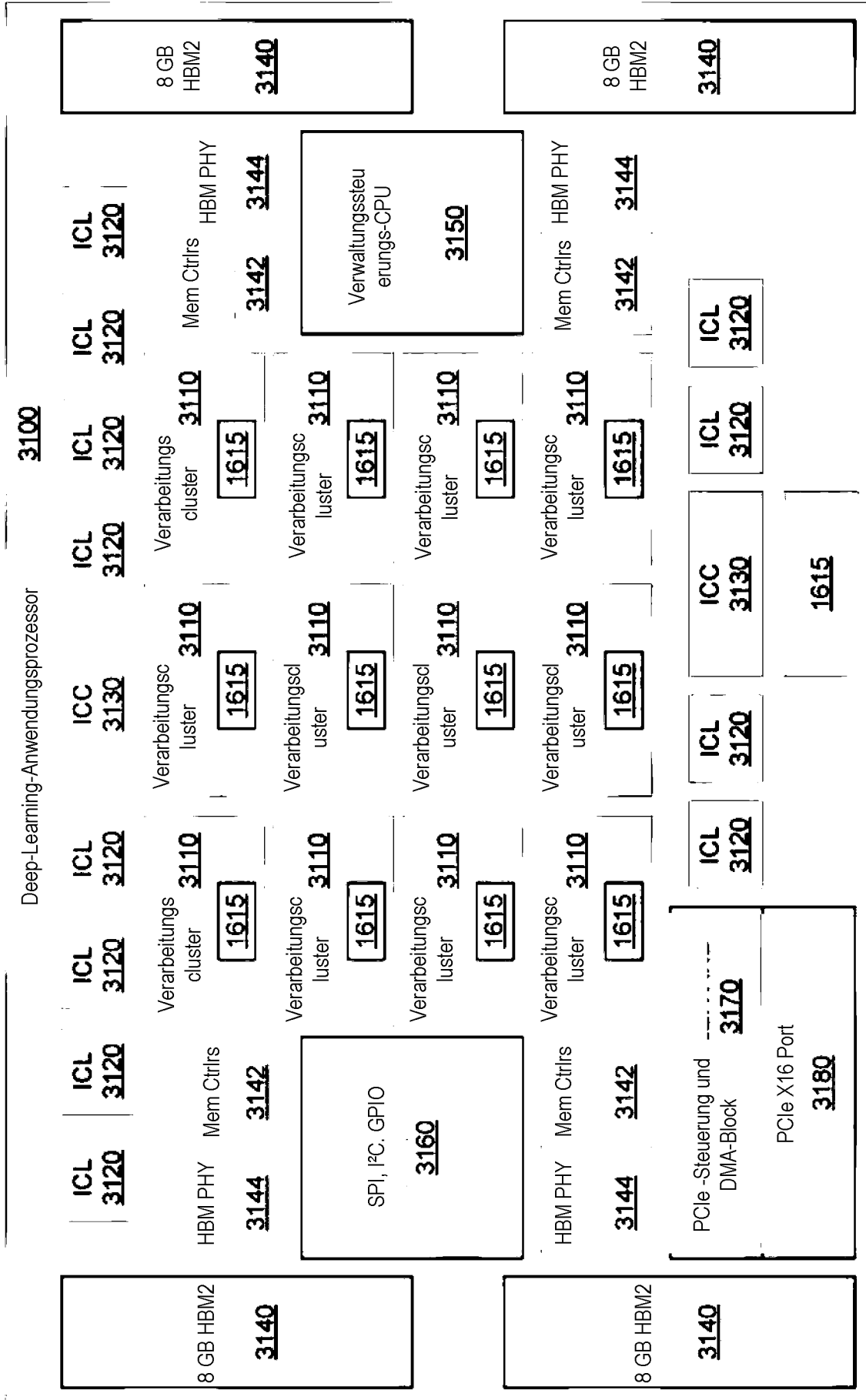
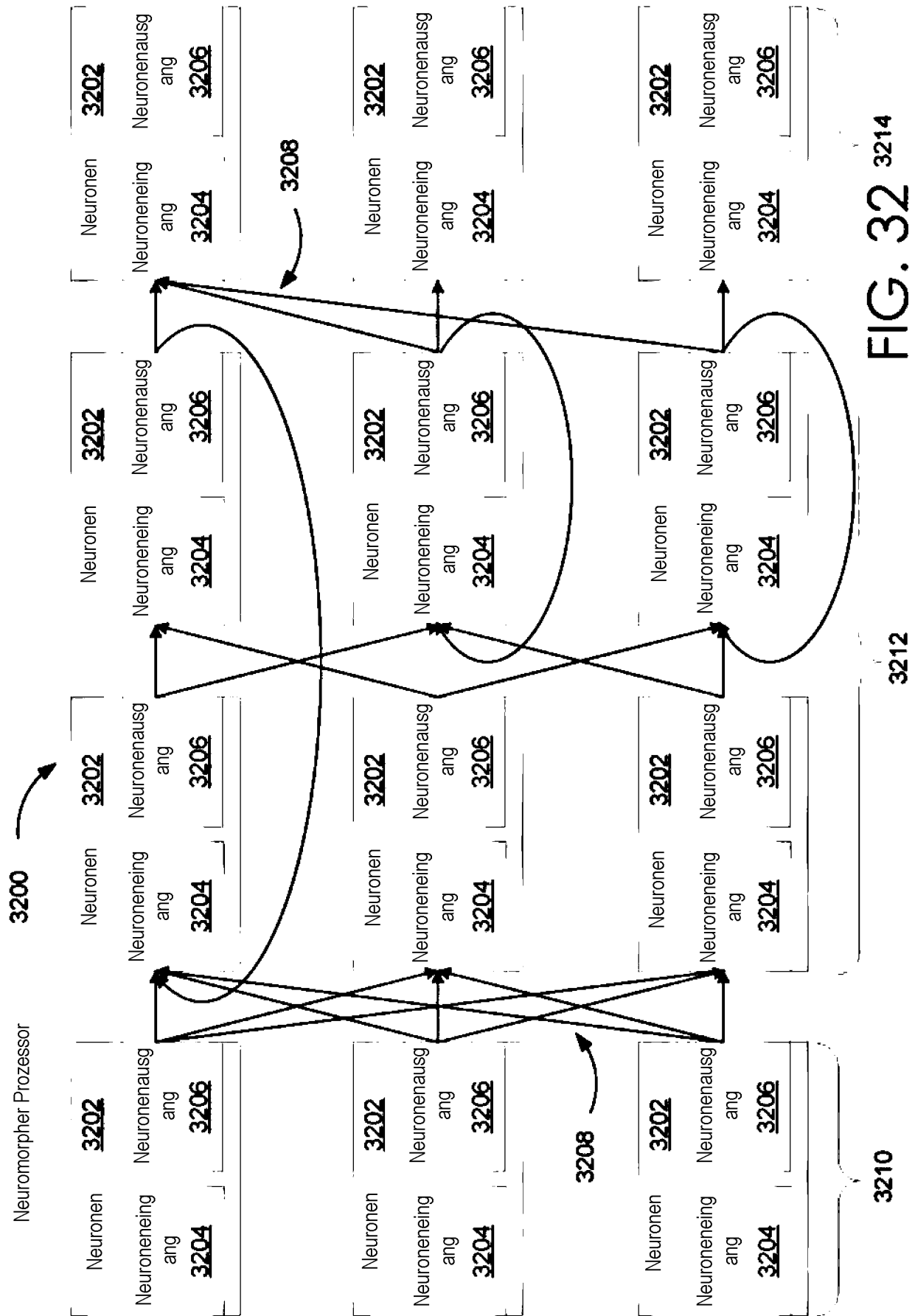


FIG. 31



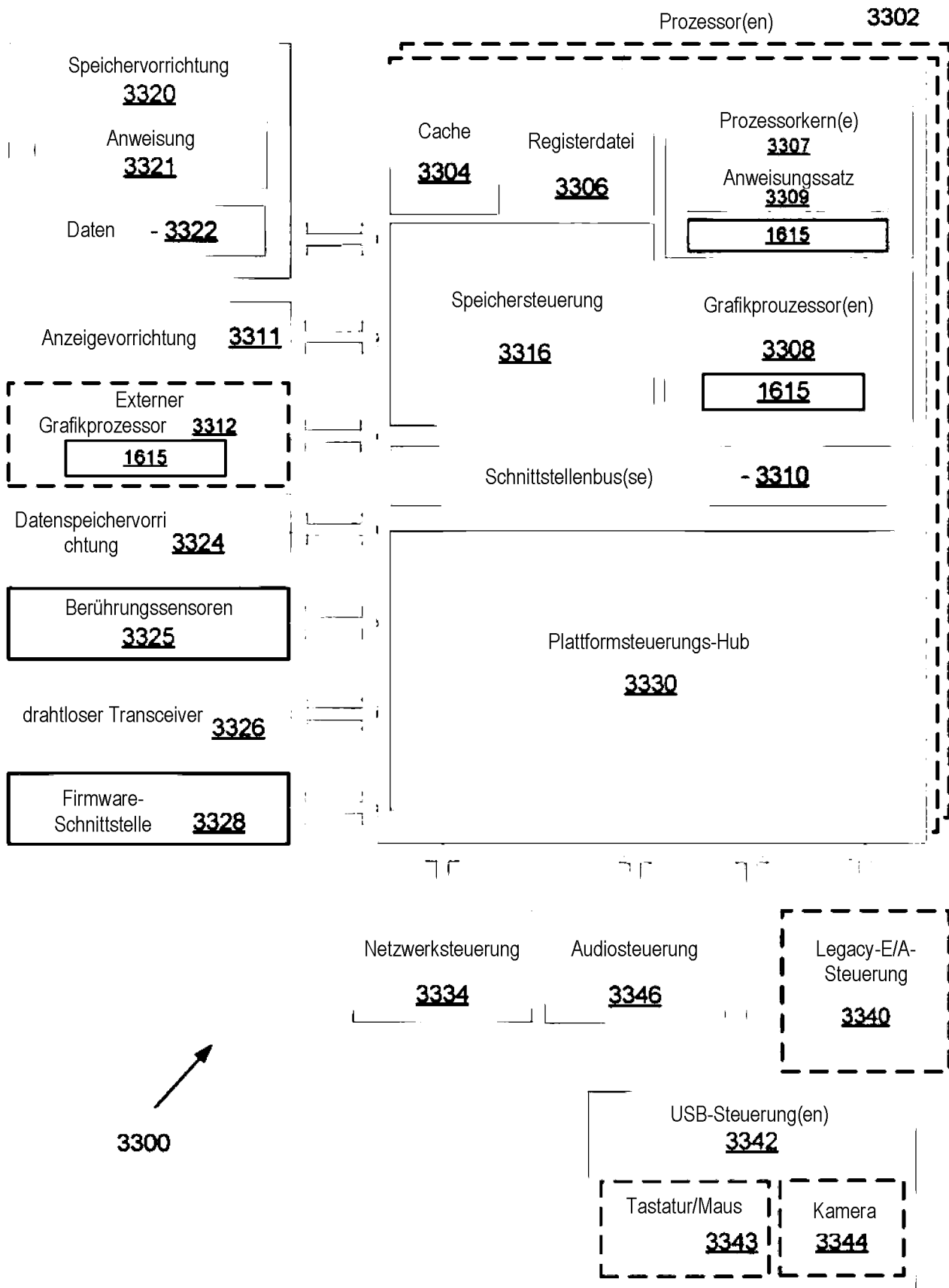


FIG. 33

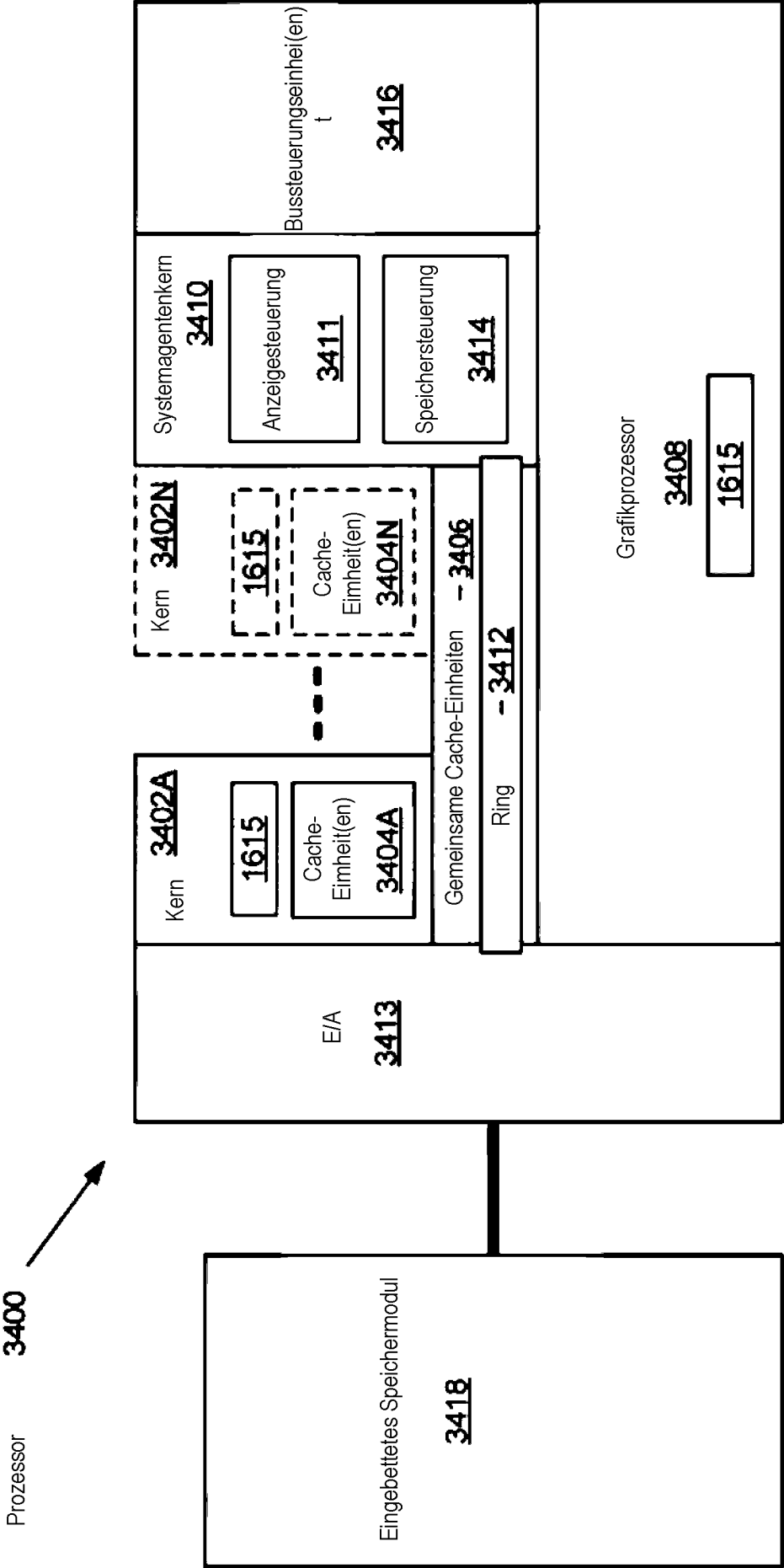


FIG. 34

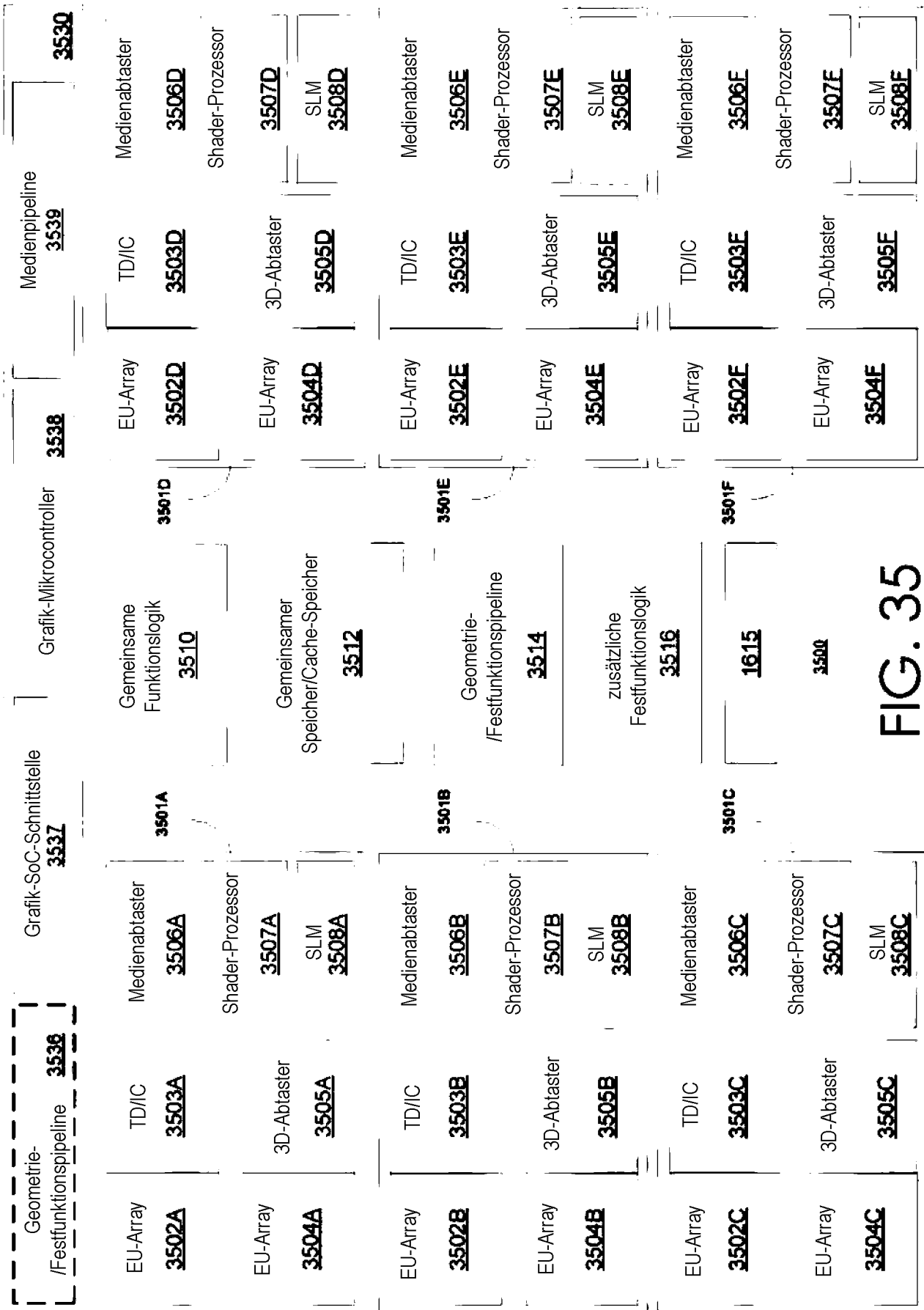


FIG. 35

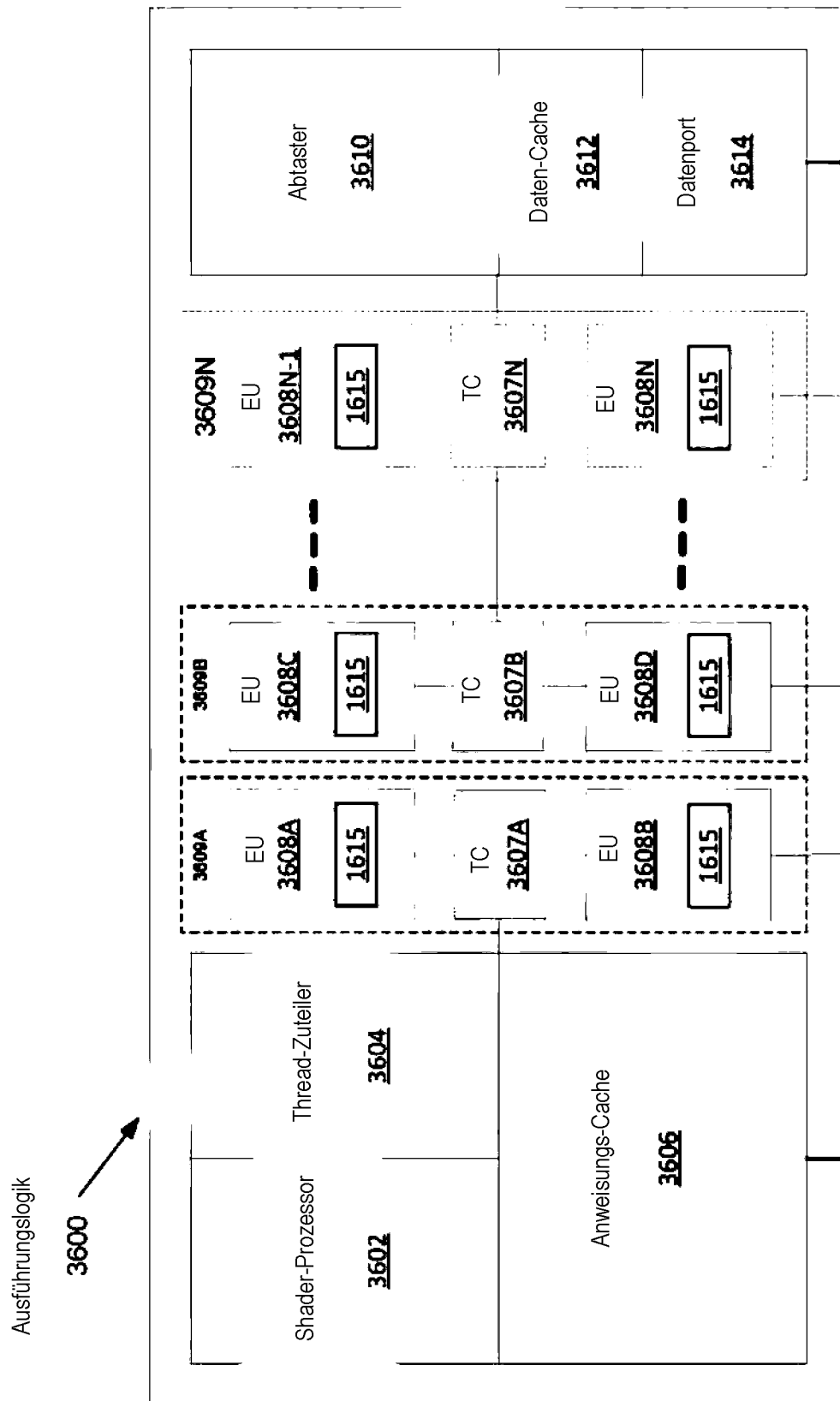


FIG. 36A

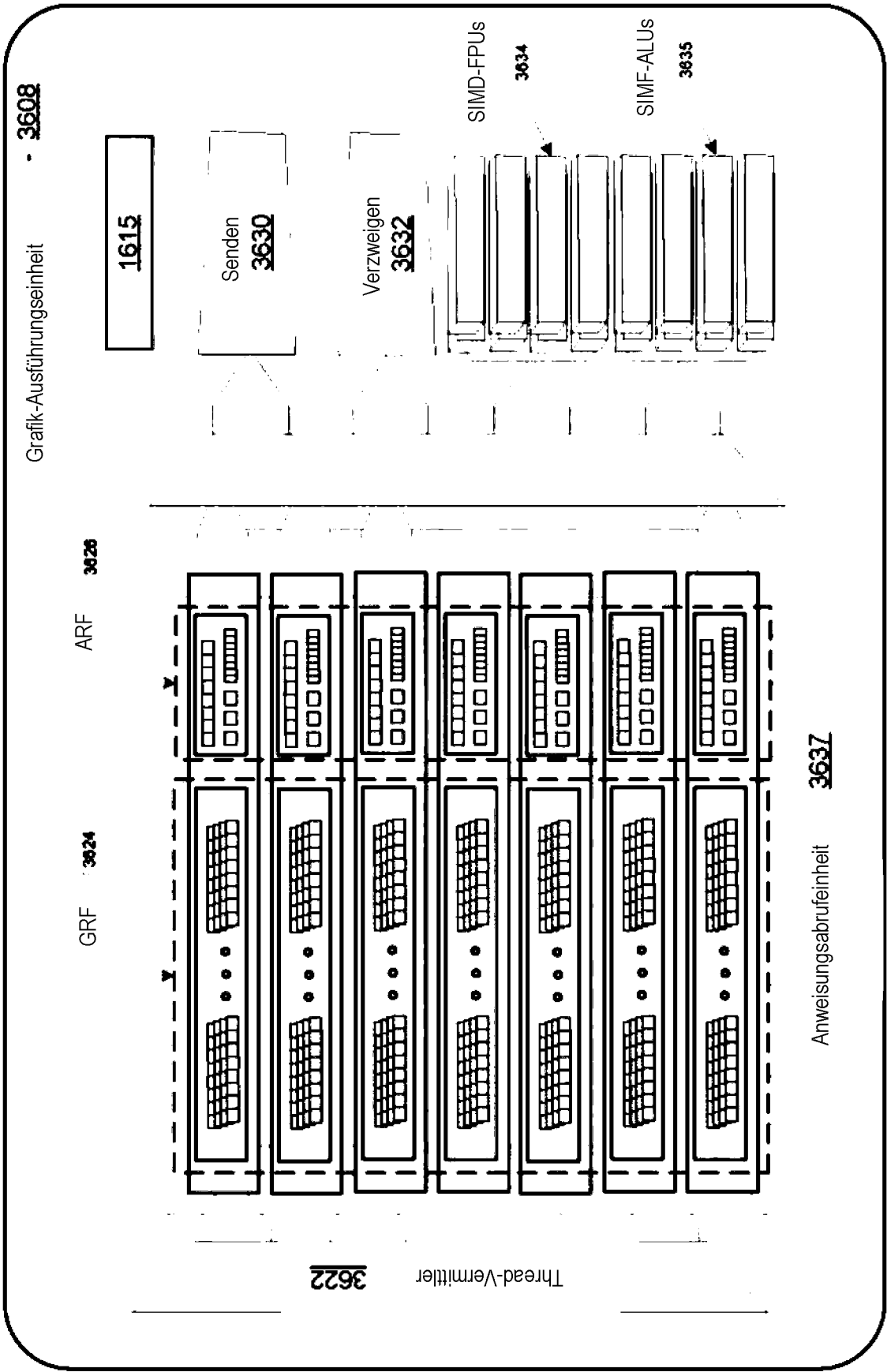


FIG. 36B

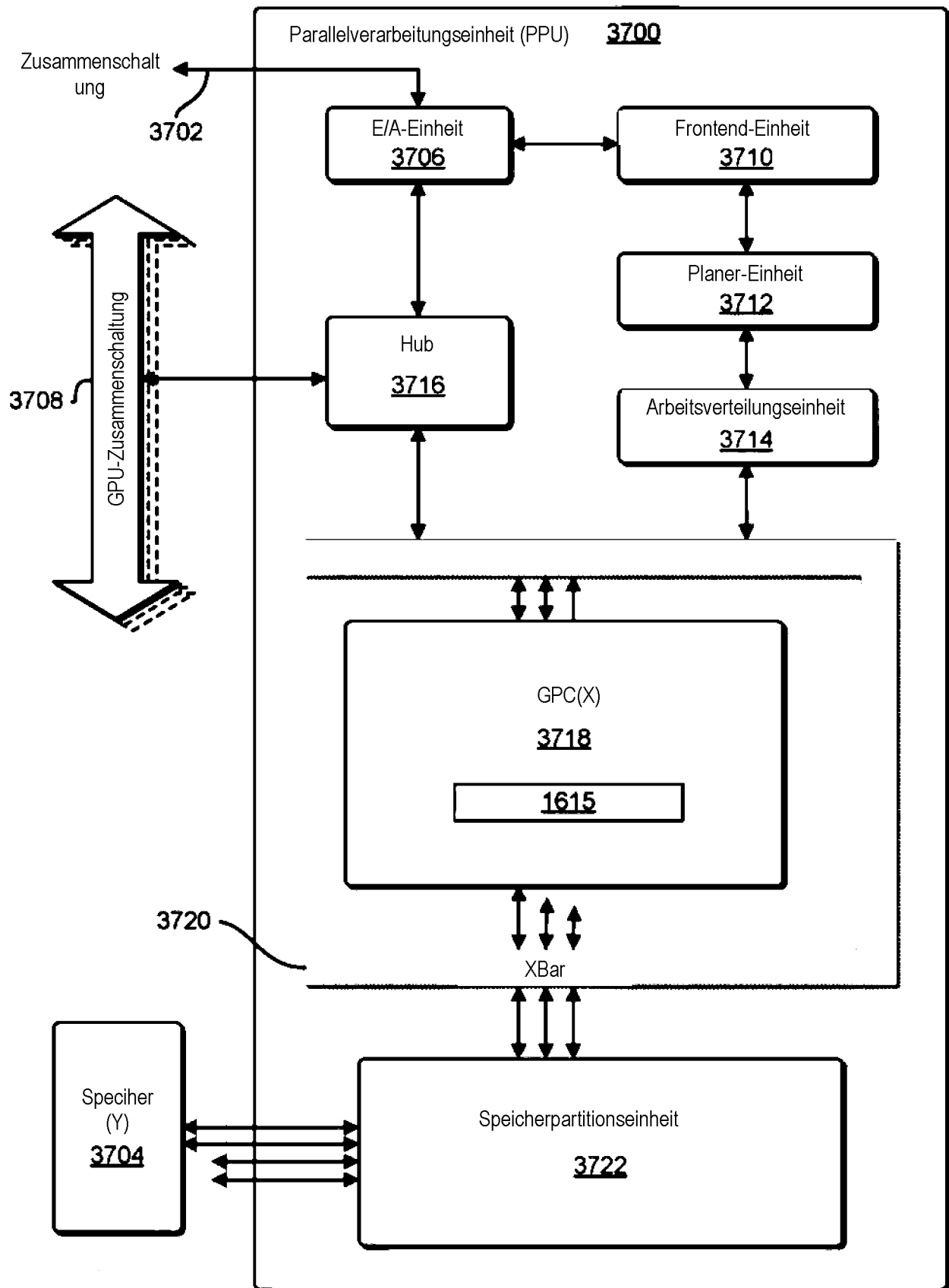


FIG. 37

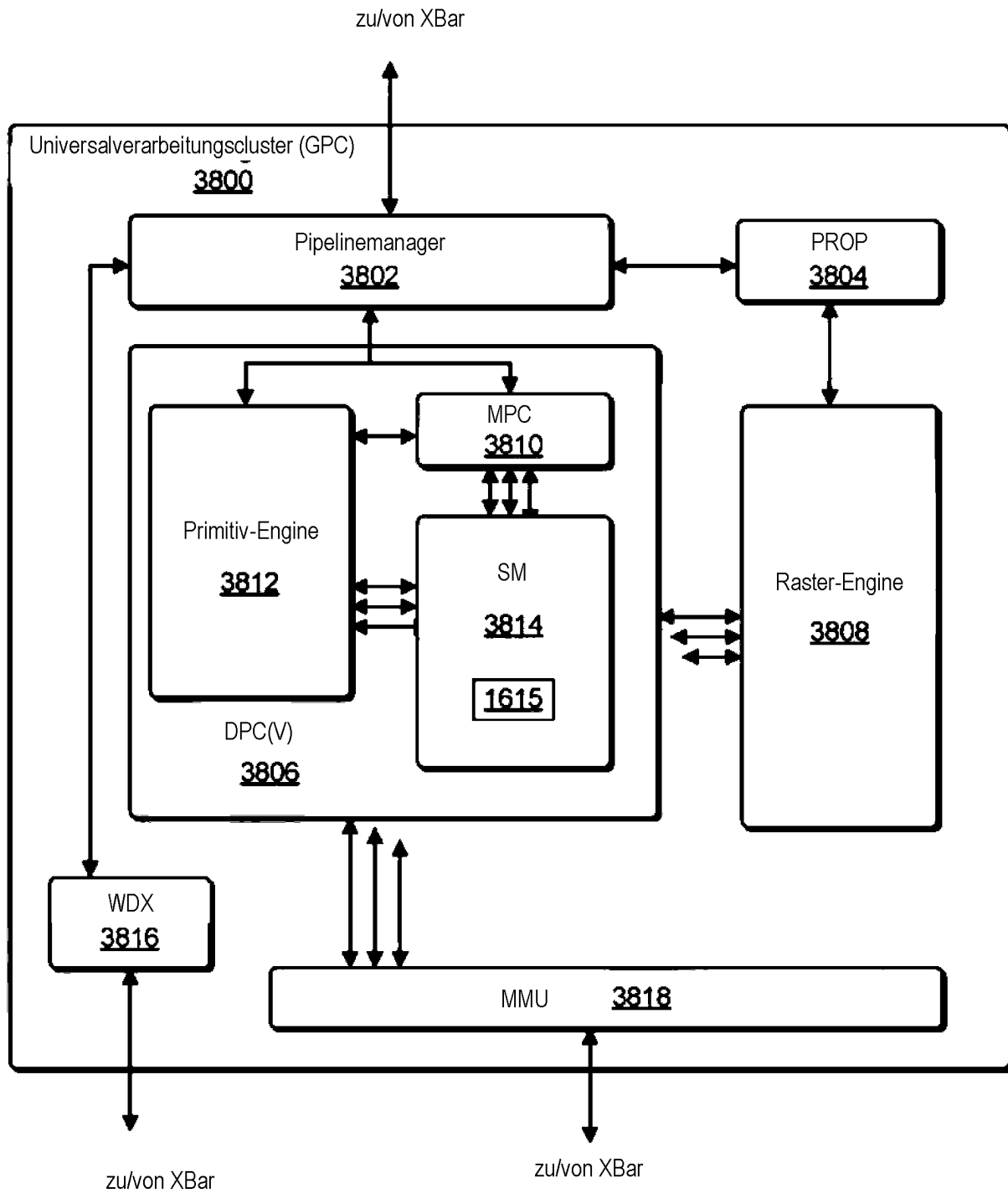


FIG. 38

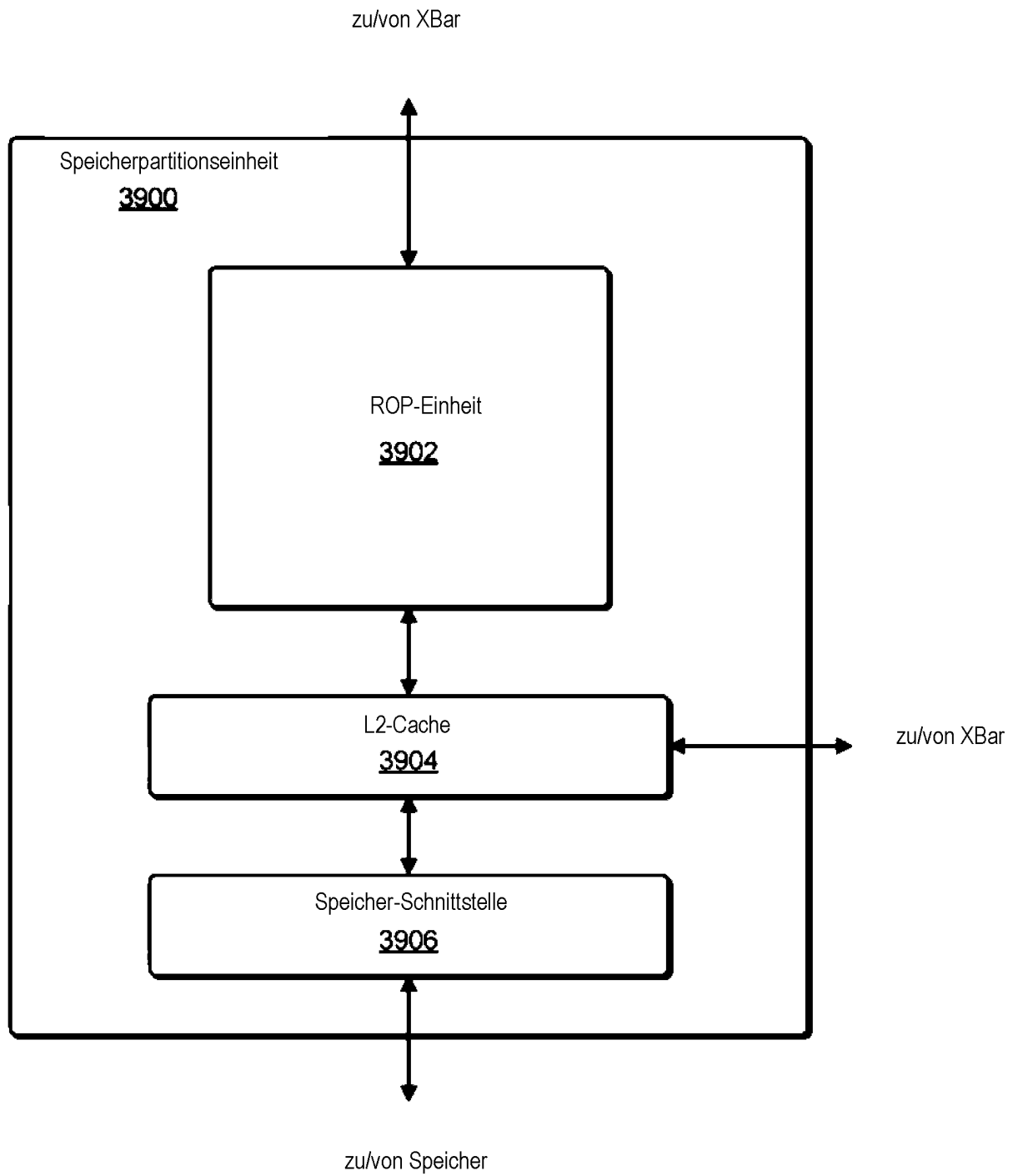


FIG. 39

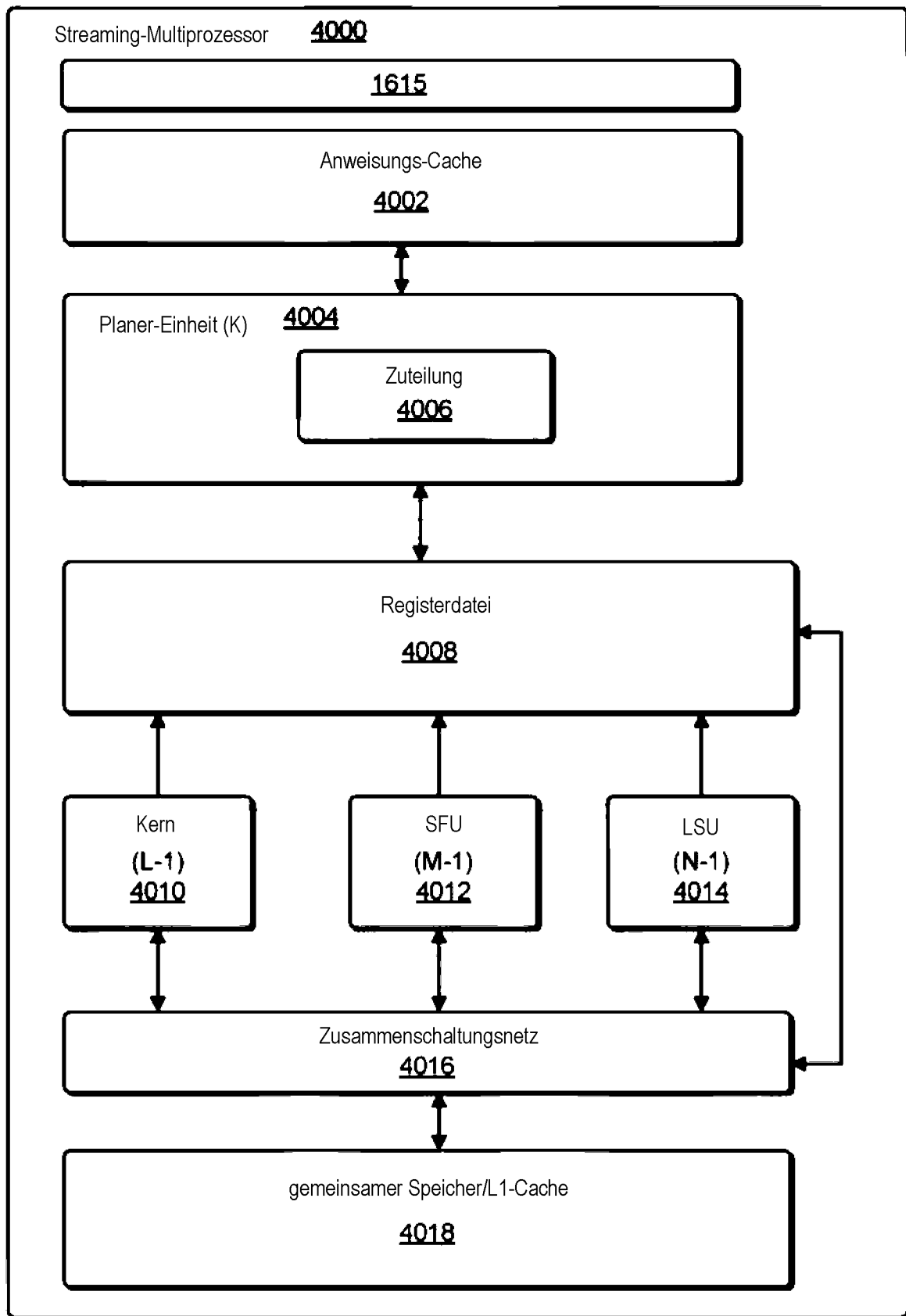


FIG. 40