



(10) **DE 10 2021 132 069 A1** 2022.06.09

(12)

Offenlegungsschrift

(21) Aktenzeichen: **10 2021 132 069.0**

(22) Anmeldetag: **06.12.2021**

(43) Offenlegungstag: **09.06.2022**

(51) Int Cl.: **G06V 10/82 (2022.01)**

(30) Unionspriorität:

17/114,144 07.12.2020 US

(71) Anmelder:

NVIDIA Corporation, Santa Clara, CA, US

(74) Vertreter:

Kraus & Weisert Patentanwälte PartGmbB, 80539 München, DE

(72) Erfinder:

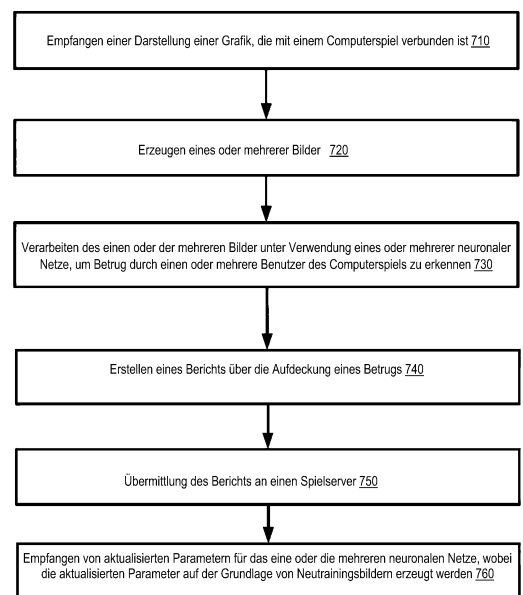
**Jonnalagadda, Aditya, Santa Clara, CA, US;
Frosio, Iuri, Santa Clara, CA, US; Kim, Joohwan,
Santa Clara, CA, US; Schneider, Seth, Santa Clara,
CA, US**

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **GRAFIKVERARBEITUNGSEINHEITEN ZUR ERKENNUNG VON BETRUG MITTELS NEURONALER NETZE**

700 



(57) Zusammenfassung: Vorrichtungen, Systeme und Techniken zur Erkennung von Betrug in einem Computerspiel. In mindestens einem Anspruch verwenden ein oder mehrere Schaltkreise einen oder mehrere neuronale Netze, um Betrug durch einen oder mehrere Benutzer eines Computerspiels zumindest teilweise auf der Grundlage eines oder mehrerer durch das Computerspiel erzeugter Bilder zu erkennen.

Beschreibung

TECHNISCHES GEBIET

[0001] Mindestens eine Ausführungsform betrifft Grafikverarbeitungseinheiten, die so konfiguriert sind, dass sie mit Hilfe neuronaler Netze vor der Verwendung unerlaubter Informationen in Bildern schützen. Zum Beispiel bezieht sich mindestens eine Ausführungsform auf Vorgänge, die beim Training und bei der Verwendung neuronaler Netze, die auf Grafikverarbeitungseinheiten ausgeführt werden, auftreten, um sich gegen die Verwendung von betrügerischen oder anderen unerlaubten Informationen in Bildern zu schützen, die den Benutzern von Rechengaräten angezeigt werden.

ALLGEMEINER STAND DER TECHNIK

[0002] In verschiedenen Rechenanwendungen können mehrere Teilnehmer an einer kollektiven Aktivität beteiligt sein, bei der einige Informationen mit anderen Teilnehmern geteilt werden, während andere Informationen privat gehalten werden und nur für einige der Teilnehmer zugänglich sind. In Online-Computerspielen kann beispielsweise der Besitz von Informationen über den Aufenthaltsort gegnerischer Spieler, ihren Gesundheitszustand, ihre Rüstung, ihren Munitionsstatus und ähnliches untersagt sein. Allerdings können Spieler verschiedene Betrugssoftware erwerben, um Zugang zu solchen unerlaubten (betrügerischen) Informationen zu erhalten. Umfragen zufolge geben mehr als 10% der Spieler zu, dass sie schummeln, und mehr als drei Viertel der Spieler sehen das Schummeln als ernsthaftes Problem an und hören wahrscheinlich auf zu spielen, wenn sie vermuten, dass andere Spieler schummeln. Betrug basierend auf Berichten von Spielern aufdecken, kann ein langsamer und unzuverlässiger Prozess sein. Umgekehrt kann die Verwendung eines schnelleren Verfahrens zur Untersuchung des Speichers auf den Computern der Betrüger und/oder der Protokolle der Aktivitäten der Teilnehmer Probleme mit dem Datenschutz mit sich bringen.

Figurenliste

Fig. 1A veranschaulicht Ableitungs- und/oder Trainingslogik gemäß mindestens einer Ausführungsform;

Fig. 1B veranschaulicht Ableitungs- und/oder Trainingslogik gemäß mindestens einer Ausführungsform;

Fig. 2 veranschaulicht das Training und den Einsatz eines neuronalen Netzes gemäß mindestens einer Ausführungsform;

Fig. 3 ist ein Blockdiagramm eines beispielhaften Computersystems, in dem die Erkennung unerlaubter Informationen in Bildern, die zur Anzeige für einen oder mehrere Benutzer wiedergegeben werden, in Übereinstimmung mit mindestens einer Ausführungsform durchgeführt werden kann;

Fig. 4 ist eine schematische Darstellung eines Bildes, das unerlaubte Informationen enthält, die von einer Grafikverarbeitungseinheit auf einem Display eines Benutzergeräts gemäß mindestens einer Ausführungsform dargestellt werden;

Fig. 5A ist eine schematische Darstellung eines erweiterten unzulässigen Trainingsbildes mit einer reduzierten Menge an unzulässigen Informationen für ein effizientes Training neuronaler Netze zur Erkennung unzulässiger Informationen in Bildern, die zur Anzeige für einen oder mehrere Benutzer wiedergegeben werden, in Übereinstimmung mit mindestens einer Ausführungsform;

Fig. 5B ist eine schematische Darstellung eines erweiterten echten Trainingsbildes, das Einfügeartefakte enthält, für ein effizientes Training neuronaler Netze zur Erkennung unzulässiger Informationen in Bildern, die zur Anzeige für einen oder mehrere Benutzer wiedergegeben werden, in Übereinstimmung mit mindestens einer Ausführungsform

Fig. 6 ist eine schematische Darstellung eines Systems mit einem oder mehreren neuronalen Netzmodellen zur Erkennung unerlaubter Informationen in Bildern, die von einer Grafikverarbeitungseinheit auf einem Display eines Benutzergeräts gemäß mindestens einer Ausführungsform dargestellt werden;

Fig. 7 ist ein Flussdiagramm eines Beispielverfahrens zur Verwendung eines oder mehrerer neuronaler Netze zur Erkennung unerlaubter Informationen in Bildern, die von einer Grafikverarbeitungseinheit auf einem Display eines Benutzergeräts gemäß mindestens einer Ausführungsform dargestellt werden;

Fig. 8 ist ein Flussdiagramm eines Beispielverfahrens zur Verwendung eines oder mehrerer neuronaler Netze zur Erkennung unerlaubter Informationen in Bildern, die von einer Grafikverarbeitungseinheit auf einem Display eines Benutzergeräts gemäß mindestens einer Ausführungsform dargestellt werden;

Fig. 9 veranschaulicht ein beispielhaftes Datencentersystem gemäß mindestens einer Ausführungsform;

Fig. 10A veranschaulicht ein Beispiel für ein autonomes Fahrzeug gemäß mindestens einer Ausführungsform;

Fig. 10B veranschaulicht ein Beispiel für Kamerapositionen und Sichtfelder für das autonome Fahrzeug von **Fig. 10A** gemäß mindestens einer Ausführungsform;

Fig. 10C ist ein Blockdiagramm, das eine Beispielsystemarchitektur für das autonome Fahrzeug von **Fig. 10A** gemäß mindestens einer Ausführungsform veranschaulicht;

Fig. 10D ist ein Diagramm, das ein System für die Kommunikation zwischen dem/den cloudbasierten Server(n) und dem autonomen Fahrzeug von **Fig. 10A** gemäß mindestens einer Ausführungsform veranschaulicht;

Fig. 11 ist ein Blockdiagramm, das ein Computersystem veranschaulicht, gemäß mindestens einer Ausführungsform;

Fig. 12 ist ein Blockdiagramm, das ein Computersystem veranschaulicht, gemäß mindestens einer Ausführungsform;

Fig. 13 veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;

Fig. 14 veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;

Fig. 15A veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;

Fig. 15B veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;

Fig. 15C veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;

Fig. 15D veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;

Fig. 15E und **Fig. 15F** veranschaulichen ein gemeinsam genutztes Programmiermodell gemäß mindestens einer Ausführungsform;

Fig. 16 veranschaulicht beispielhafte integrierte Schaltungen und damit assoziierte Grafikprozessoren gemäß mindestens einer Ausführungsform;

Fig. 17A-17B veranschaulichen beispielhafte integrierte Schaltungen und damit assoziierte Grafikprozessoren gemäß mindestens einer Ausführungsform;

Fig. 18A-18B veranschaulichen zusätzliche beispielhafte Grafikprozessorlogik gemäß mindestens einer Ausführungsform;

Fig. 19 veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;

Fig. 20A veranschaulicht einen Parallelprozessor gemäß mindestens einer Ausführungsform;

Fig. 20B veranschaulicht eine Partitionseinheit gemäß mindestens einer Ausführungsform;

Fig. 20C veranschaulicht ein Verarbeitungscluster gemäß mindestens einer Ausführungsform;

Fig. 20D veranschaulicht einen Grafikmultiprozessor gemäß mindestens einer Ausführungsform;

Fig. 21 veranschaulicht ein System mit Multigrafikverarbeitungseinheit (graphics processing unit - GPU) gemäß mindestens einer Ausführungsform;

Fig. 22 veranschaulicht einen Grafikprozessor gemäß mindestens einer Ausführungsform;

Fig. 23 ist ein Blockdiagramm, das eine Prozessor-Mikroarchitektur für einen Prozessor veranschaulicht, gemäß mindestens einer Ausführungsform;

Fig. 24 veranschaulicht einen Deep-Learning-Anwendungsprozessor gemäß mindestens einer Ausführungsform;

Fig. 25 ist ein Blockdiagramm, das einen beispielhaften neuromorphen Prozessor veranschaulicht, gemäß mindestens einer Ausführungsform;

Fig. 26 veranschaulicht mindestens Abschnitte eines Grafikprozessors gemäß einer oder mehreren Ausführungsformen;

Fig. 27 veranschaulicht mindestens Abschnitte eines Grafikprozessors gemäß einer oder mehreren Ausführungsformen;

Fig. 28 veranschaulicht mindestens Abschnitte eines Grafikprozessors gemäß einer oder mehreren Ausführungsformen;

Fig. 29 ist ein Blockdiagramm einer Grafikverarbeitungsengine eines Grafikprozessors gemäß mindestens einer Ausführungsform;

Fig. 30 ist ein Blockdiagramm von mindestens Abschnitten eines Grafikprozessorkerns gemäß mindestens einer Ausführungsform;

Fig. 31A-31B veranschaulichen Thread-Ausführungslogik, die ein Array von Verarbeitungselementen eines Grafikprozessorkerns beinhaltet, gemäß mindestens einer Ausführungsform;

Fig. 32 veranschaulicht eine Parallelverarbeitungseinheit (parallel processing unit - „PPU“) gemäß mindestens einer Ausführungsform;

Fig. 33 veranschaulicht ein allgemeines Verarbeitungscluster (general processing cluster - „GPC“) gemäß mindestens einer Ausführungsform;

Fig. 34 veranschaulicht eine Speicherpartitionseinheit einer Parallelverarbeitungseinheit („PPU“) gemäß mindestens einer Ausführungsform;

Fig. 35 veranschaulicht einen Streaming-Multiprozessor gemäß mindestens einer Ausführungsform.

Fig. 36 ist ein beispielhaftes Datenflussdiagramm für eine weiterentwickelte Rechenpipeline gemäß mindestens einer Ausführungsform;

Fig. 37 ist eine Systemdarstellung für ein beispielhaftes System zum Trainieren, Anpassen, Instanzieren und Einsetzen von Modellen des maschinellen Lernens in einer weiterentwickelten Rechenpipeline gemäß mindestens einer Ausführungsform;

DETAILLIERTE BESCHREIBUNG

INFERENZ- UND TRAININGSLOGIK

[0003] **Fig. 1A** stellt Inferenz- und/oder Trainingslogik 115 dar, die verwendet wird, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugehörig sind.

[0004] In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 ohne Einschränkung Code- und/oder Datenspeicher 101 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 115 Code- und/oder Datenspeicher 101 beinhalten oder an diesen gekoppelt sein, um Kurvendiagrammencode 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) oder einfach Schaltungen bezeichnet). In mindestens einer Ausführungsform lädt Code, wie etwa Kurvendiagrammencode, 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 101 Gewichtungsparmeter und/oder Eingabe-/Ausgabedaten jeder Schicht eines neuronalen Netzwerks, die mit einer oder mehreren Ausführungsformen trainiert oder in Verbindung damit verwendet wird, während der Vorwärtspropagation von Eingabe-/Ausgabedaten und/oder Gewichtungsparmetern während des Trainings und/oder der Ableitung unter Verwendung von Aspekten einer oder mehrerer Ausführungsformen. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Code- und/oder Datenspeichers 101 in einem anderen chipinternen oder chip externen Datenspeicher, einschließlich eines L1-, L2- oder L3-Caches oder Systemspeichers eines Prozessors, enthalten sein.

[0005] In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Code- und/oder Datenspeichers 101 innerhalb oder außerhalb eines oder mehrerer Prozessoren oder anderer Hardware-Logikvorrichtungen oder -Schaltungen vorliegen. In mindestens einer Ausführungsform kann der Code und/oder Code- und/oder Datenspeicher 101 Cache-Speicher, dynamischer zufällig adressierbarer Speicher (dynamic randomly addressable memory - „DRAM“), statischer zufällig adressierbarer Speicher (static randomly addressable memory - „SRAM“), nichtflüchtiger Speicher (z.B. Flash-Speicher) oder anderer Speicher sein. In mindestens einer Ausführungsform kann eine Wahl, ob der Code und/oder Code- und/oder Datenspeicher 101

zum Beispiel innerhalb oder außerhalb eines Prozessors vorliegt oder DRAM, SRAM, Flash oder einen anderen Speichertyp umfasst, von dem verfügbaren chipinternen oder chipexternen Speicher, den Latenzanforderungen der Trainings- und/oder Inferenzfunktionen, die durchgeführt werden, der Batch-Größe der Daten, die beim Inferenzieren und/oder Trainieren eines neuronalen Netzes verwendet werden, oder einer Kombination dieser Faktoren abhängen.

[0006] In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 ohne Einschränkung einen Code- und/oder Datenspeicher 105 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 105 Gewichtsparameter und/oder Eingabe-/Ausgabedaten jeder Schicht eines neuronalen Netzes, die in Verbindung mit einer oder mehreren Ausführungsformen trainiert oder verwendet wird, während der Rückwärtspropagation von Eingabe-/Ausgabedaten und/oder Gewichtsparametern während des Trainierens und/oder Inferenzierens unter Verwendung von Aspekten einer oder mehrerer Ausführungsformen. In mindestens einer Ausführungsform kann die Trainingslogik 115 einen Code- und/oder Datenspeicher 105 beinhalten oder mit diesem gekoppelt sein, um Grafikcode oder andere Software zur Steuerung der zeitlichen Abfolge und/oder der Reihenfolge zu speichern, in der Gewichts- und/oder andere Parameterinformationen geladen werden sollen, um die Logik zu konfigurieren, einschließlich Integer- und/oder Fließkommazahlen (zusammen als arithmetische Logikeinheiten (ALUs) bezeichnet).

[0007] In mindestens einer Ausführungsform bewirkt Code, wie etwa Kurvendiagrammencode, das Laden von den Gewichts- oder anderen 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 105 in einem anderen chipinternen oder chipexternen Datenspeicher, einschließlich eines L1-, L2- oder L3-Caches oder Systemspeichers eines Prozessors, enthalten sein. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Code- und/oder Datenspeichers 105 innerhalb oder außerhalb eines oder mehrerer Prozessoren oder anderer Hardware-Logikvorrichtungen oder -Schaltungen vorliegen. In mindestens einer Ausführungsform kann der Code- und/oder Datenspeicher 105 Cachespeicher, DRAM, SRAM, nichtflüchtiger Speicher (z. B. Flash-Speicher) oder anderer Speicher sein. In mindestens einer Ausführungsform kann eine Wahl, ob der Code- und/oder Datenspeicher 105 zum Beispiel innerhalb oder außerhalb eines Prozessors vorliegt oder DRAM, SRAM, Flash-Speicher oder einen anderen Speichertyp umfasst, von dem verfügbaren chipinternen oder chipexternen Speicher, den Latenzanforderungen der Trainings- und/oder Inferenzfunktionen, die durchgeführt werden, der Batch-Größe der Daten, die beim Inferenzieren und/oder Trainieren eines neuronalen Netzes verwendet werden, oder einer Kombination dieser Faktoren abhängen.

[0008] In mindestens einer Ausführungsform können der Code- und/oder Datenspeicher 101 und der Code- und/oder Datenspeicher 105 separate Speicherstrukturen sein. In mindestens einer Ausführungsform können der Code- und/oder Datenspeicher 101 und der Code- und/oder Datenspeicher 105 eine kombinierte Speicherstruktur sein. In mindestens einer Ausführungsform können der Code- und/oder Datenspeicher 101 und der Code- und/oder Datenspeicher 105 teilweise kombiniert und teilweise separat sein. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Code- und/oder Datenspeichers 101 und des Code- und/oder Datenspeichers 105 in einem anderen chipinternen oder chipexternen Datenspeicher, einschließlich des L1-, L2- oder L3-Caches oder Systemspeichers eines Prozessors, enthalten sein.

[0009] In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 ohne Einschränkung eine oder mehrere arithmetisch-logische Einheiten („ALU(s)“) 110 beinhalten, einschließlich Ganzzahl- und/oder Gleitkommazahlen, um logische und/oder mathematische Operationen durchzuführen, die mindestens zum Teil auf Trainings- und/oder Inferenzcode (z. B. Kurvendiagrammencode) 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 120 gespeichert werden und abhängig von Eingabe/Ausgabe- und/oder Gewichtsparameterdaten sind, die in dem Code- und/oder Datenspeicher 101 und/oder dem Code- und/oder Datenspeicher 105 gespeichert sind. In mindestens einer Ausführungsform werden in dem Aktivierungsspeicher 120 gespeicherte Aktivierungen gemäß linearer algebraischer und/oder matrixbasierter Mathematik erzeugt, die durch die ALU(s) 110 als Reaktion auf das Durchführen von Anweisungen oder anderem Code durchgeführt wird, wobei in dem Code- und/oder Datenspeicher 105 und/oder dem Datenspeicher 101 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 105 oder dem Code- und/oder Datenspeicher 101 oder einem anderen chipinternen oder -externen Speicher gespeichert sein können.

[0010] In mindestens einer Ausführungsform sind die ALU(s) 110 in einem oder mehreren Prozessoren oder anderen Hardware-Logikvorrichtungen oder -Schaltungen enthalten, während in einer anderen Ausführungsform die ALU(s) 110 außerhalb eines Prozessors oder einer anderen Hardware-Logikvorrichtung oder -Schaltung vorliegen können, der/die sie verwendet (z. B. ein Koprozessor). In mindestens einer Ausführungsform können die ALUs 110 in Ausführungseinheiten eines Prozessors oder anderweitig in einer Bank von ALUs enthalten sein, auf die 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 101, der Code- und/oder Datenspeicher 105 und der Aktivierungsspeicher 120 einen Prozessor oder eine andere Hardware-Logikvorrichtung oder -schaltung teilen, während sie sich in einer anderen Ausführungsform in unterschiedlichen Prozessoren oder anderen Hardware-Logikvorrichtungen oder -Schaltungen oder in einer Kombination aus gleichen und unterschiedlichen Prozessoren oder anderen Hardware-Logikvorrichtungen oder -Schaltungen befinden können. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Aktivierungsspeichers 120 in einem anderen chipinternen oder chipexternen Datenspeicher, einschließlich des L1-, L2- oder L3-Caches oder Systemspeichers eines Prozessors, integriert sein. Darüber hinaus kann der Inferenzierungs- 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 logischen Schaltungen eines Prozessors abgerufen und/oder verarbeitet wird.

[0011] In mindestens einer Ausführungsform kann der Aktivierungsspeicher 120 Cachespeicher, DRAM, SRAM, nichtflüchtiger Speicher (z. B. Flash-Speicher) oder anderer Speicher sein. In mindestens einer Ausführungsform kann sich der Aktivierungsspeicher 120 ganz oder teilweise innerhalb oder außerhalb eines oder mehrerer Prozessoren oder anderer logischer Schaltungen befinden. In mindestens einer Ausführungsform kann eine Wahl, ob der Aktivierungsspeicher 120 zum Beispiel zu einem Prozessor intern oder extern ist oder DRAM, SRAM, Flash-Speicher oder einen anderen Speichertyp umfasst, von dem verfügbaren chipinternen oder chip externen Speicher, den Latenzanforderungen der Trainings- und/oder Inferenzfunktionen, die durchgeführt werden, der Batch-Größe der Daten, die beim Inferenzieren und/oder Trainieren eines neuronalen Netzes verwendet werden, oder einer Kombination dieser Faktoren abhängen.

[0012] In mindestens einer Ausführungsform kann die in **Fig. 1A** dargestellte Inferenz- und/oder Trainingslogik 115 in Verbindung mit einer anwendungsspezifischen integrierten Schaltung (application-specific integrated circuit - „ASIC“) verwendet werden, wie etwa einer 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. 1A** dargestellte Inferenz- und/oder Trainingslogik 115 in Verbindung mit Hardware der Zentraleinheit (central processing unit - „CPU“), Hardware der Grafikverarbeitungseinheit („GPU“) oder anderer Hardware, wie etwa feldprogrammierbaren Gate-Arrays (field programmable gate arrays - „FPGAs“), verwendet werden.

[0013] **Fig. 1B** veranschaulicht Ableitungs- und/oder Trainingslogik 115 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 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. 1B** dargestellte Inferenz- und/oder Trainingslogik 115 in Verbindung mit einer anwendungsspezifischen integrierten Schaltung (application-specific integrated circuit - „ASIC“) verwendet werden, wie etwa einer 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. 1B** dargestellte Inferenz- und/oder Trainingslogik 115 in Verbindung mit Hardware der Zentraleinheit (central processing unit - „CPU“), Hardware der Grafikverarbeitungseinheit („GPU“) oder anderer Hardware, wie etwa feldprogrammierbaren Gate-Arrays (field programmable gate arrays - „FPGAs“), verwendet werden. In mindestens einer Ausführungsform beinhaltet die Inferenz- und/oder Trainingslogik 115 ohne Einschränkung den Code- und/oder Datenspeicher 101 und den Code- und/oder Datenspeicher 105, die zum Speichern von Code (z. B. Kurvendigrammencode), Gewichtungswerten und/oder anderen Informationen, einschließlich Verzerrungswerten, Gradienteninformationen, Momentwerten und/oder anderer Parameter- oder Hyperparameterinformationen, verwendet werden

können. In mindestens einer Ausführungsform, die in **Fig. 1B** veranschaulicht ist, ist jeder des Code- und/oder Datenspeichers 101 und des Code- und/oder Datenspeichers 105 mit einer dedizierten Rechenressource, wie etwa der Rechen-Hardware 102 bzw. der Rechen-Hardware 106, assoziiert. In mindestens einer Ausführungsform umfasst jede der Rechen-Hardware 102 und der Rechen-Hardware 106 eine oder mehrere ALUs, die mathematische Funktionen, wie etwa lineare algebraische Funktionen, nur an Informationen durchführen, die in dem Code- und/oder Datenspeicher 101 bzw. dem Code- und/oder Datenspeicher 105 gespeichert sind, wobei das Ergebnis davon in dem Aktivierungsspeicher 120 gespeichert wird.

[0014] In mindestens einer Ausführungsform entspricht jedes der Code- und/oder Datenspeicher 101 und 105 und der entsprechenden Rechen-Hardware 102 bzw. 106 unterschiedlichen Schichten eines neuronalen Netzes, sodass die resultierende Aktivierung von einem Speicher-/Rechenpaar 101/102 des Code- und/oder Datenspeichers 101 und der Rechen-Hardware 102 als Eingabe einem nächsten Speicher-/Rechenpaar 105/106 des Code- und/oder Datenspeichers 105 und der Rechen-Hardware 106 bereitgestellt wird, um eine konzeptionelle Organisation eines neuronalen Netzes widerzuspiegeln. In mindestens einer Ausführungsform kann jedes der Speicher-/Rechenpaare 101/102 und 105/106 mehr als einer Schicht des neuronalen Netzes entsprechen. In mindestens einer Ausführungsform können zusätzliche Speicher-/Rechenpaare (nicht gezeigt) im Anschluss an oder parallel zu den Speicher-/Rechenpaaren 101/102 und 105/106 in der Inferenz- und/oder Trainingslogik 115 enthalten sein.

TRAINING UND EINSATZ VON NEURONALEN NETZEN

[0015] **Fig. 2** veranschaulicht das Training und den Einsatz eines tiefen neuronalen Netzes gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform wird das untrainierte neuronale Netz 206 unter Verwendung eines Trainingsdatensatzes 202 trainiert. In mindestens einer Ausführungsform ist der Trainingsrahmen 204 ein PyTorch-Rahmen, wohingegen der Trainingsrahmen 204 in anderen Ausführungsformen ein TensorFlow-, Boost-, Caffe-, Microsoft-Cognitive-Toolkit/CNTK-, MXNet-, Chainer-, Keras-, DeepLearning4j- oder ein anderer Trainingsrahmen ist. In mindestens einer Ausführungsform trainiert der Trainingsrahmen 204 ein untrainiertes neuronales Netz 206 und ermöglicht, dass es unter Verwendung der hierin beschriebenen Verarbeitungsressourcen trainiert wird, um ein trainiertes neuronales Netz 208 zu erzeugen. In mindestens einer Ausführungsform können Gewichtungen zufällig oder durch Vorabtraining 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.

[0016] In mindestens einer Ausführungsform wird das untrainierte neuronale Netz 206 unter Verwendung von überwachtem Lernen trainiert, wobei der Trainingsdatensatz 202 eine Eingabe beinhaltet, die mit einer gewünschten Ausgabe für eine Eingabe gepaart ist, oder wobei der Trainingsdatensatz 202 eine Eingabe beinhaltet, die eine bekannte Ausgabe aufweist, und eine Ausgabe des neuronalen Netzes 206 manuell bewertet wird. In mindestens einer Ausführungsform wird das untrainierte neuronale Netz 206 auf überwachter Weise trainiert und verarbeitet Eingaben aus dem Trainingsdatensatz 202 und vergleicht die resultierenden Ausgaben mit einem Satz von erwarteten oder gewünschten Ausgaben. In mindestens einer Ausführungsform werden Fehler dann durch das untrainierte neuronale Netz 206 rückpropagiert. In mindestens einer Ausführungsform stellt der Trainingsrahmen 204 Gewichtungen ein, die das untrainierte neuronale Netz 206 steuern. In mindestens einer Ausführungsform beinhaltet der Trainingsrahmen 204 Werkzeuge, um zu überwachen, wie gut das untrainierte neuronale Netz 206 zu einem Modell konvergiert, wie etwa dem trainierten neuronalen Netz 208, das dazu geeignet ist, korrekte Antworten zu erzeugen, wie etwa in Ergebnis 214, die auf Eingabedaten wie etwa einem neuen Datensatz 212 basieren. In mindestens einer Ausführungsform trainiert der Trainingsrahmen 204 das untrainierte neuronale Netz 206 wiederholt, während Gewichtungen eingestellt werden, um eine Ausgabe des untrainierten neuronalen Netzes 206 unter Verwendung einer Verlustfunktion und eines Einstellungsalgorithmus, wie etwa des stochastischen Gradientenabstiegs, zu verfeinern. In mindestens einer Ausführungsform trainiert der Trainingsrahmen 204 das untrainierte neuronale Netz 206, bis das untrainierte neuronale Netz 206 eine gewünschte Genauigkeit erreicht. In mindestens einer Ausführungsform kann das trainierte neuronale Netz 208 dann zum Implementieren einer beliebigen Anzahl von Operationen des maschinellen Lernens eingesetzt werden.

[0017] In mindestens einer Ausführungsform wird das untrainierte neuronale Netz 206 unter Verwendung von nicht überwachtem Lernen trainiert, wobei das untrainierte neuronale Netz 206 versucht, sich selbst unter Verwendung von unbeschrifteten Daten zu trainieren. In mindestens einer Ausführungsform beinhaltet der Trainingsdatensatz 202 für nicht überwachtes Lernen Eingabedaten ohne zugehörige Ausgabedaten oder „Ground-Truth“-Daten. In mindestens einer Ausführungsform kann das untrainierte neuronale Netz 206

Gruppierungen innerhalb des Trainingsdatensatzes 202 erlernen und bestimmen, wie einzelne Eingaben mit dem untrainierten Datensatz 202 in Beziehung stehen. In mindestens einer Ausführungsform kann nicht überwachtes Training verwendet werden, um eine selbstorganisierende Karte in dem trainierten neuronalen Netz 208 zu erzeugen, die dazu in der Lage ist, Operationen durchzuführen, die beim Reduzieren der Dimensionalität des neuen Datensatzes 212 nützlich sind. In mindestens einer Ausführungsform kann nicht überwachtes Training auch verwendet werden, um Anomaliedetektion durchzuführen, was die Identifizierung von Datenpunkten in dem neuen Datensatz 212 ermöglicht, die von normalen Mustern des neuen Datensatzes 212 abweichen.

[0018] In mindestens einer Ausführungsform kann halbüberwachtes Lernen verwendet werden, wobei es sich um eine Technik handelt, bei der der Trainingsdatensatz 202 eine Mischung aus beschrifteten und unbeschrifteten Daten beinhaltet. In mindestens einer Ausführungsform kann der Trainingsrahmen 204 verwendet werden, um inkrementelles Lernen durchzuführen, wie etwa durch Transferlertechniken. In mindestens einer Ausführungsform ermöglicht das inkrementelle Lernen es dem trainierten neuronalen Netz 208, sich an den neuen Datensatz 212 anzupassen, ohne das Wissen zu vergessen, das dem trainierten neuronalen Netz 208 während des anfänglichen Trainings beigebracht wurde.

ERKENNUNG VON BETRUG MIT NEURONALEN NETZEN

[0019] In mindestens einer Ausführungsform können die Herausgeber von Anwendungen (z. B. Spielehersteller oder -entwickler) das Eigentum (z. B. das Urheberrecht und die Kontrolle) an Bildern (z. B. Spielbildern) behalten, die von Anwendungen (z. B. Spielsoftware) erzeugt werden, die auf den Computern der Benutzer installiert sind. In mindestens einer Ausführungsform können die Bilder vor, parallel zu und/oder nach der Anzeige der Bilder für einen Benutzer (z. B. einen Spieler) einer Analyse durch eine Erkennungssoftware und/oder andere Software unterzogen werden. In mindestens einer Ausführungsform kann das Vorhandensein von Betrugsbildern, betrügerischen Einschüben oder anderen gefälschten Informationen in echten Bildern, die einem Benutzer angezeigt werden, mit einem oder mehreren trainierten neuronalen Netzen auf lokaler (Pixelebene) und/oder globaler (Bildebene) Ebene erkannt werden. In mindestens einer Ausführungsform können die Bilder von einer Grafikverarbeitungseinheit (GPU) erzeugt werden, bei der es sich um einen spezialisierten Prozessor für die effiziente Darstellung von grafischen Ausgaben (z. B. dreidimensionale Bilder von Spielumgebungen und Landschaften) auf den Bildschirmen der Benutzer handeln kann. In mindestens einer Ausführungsform kann die GPU in Verbindung mit einer Zentraleinheit (CPU) des Benutzercomputers arbeiten; die CPU kann verschiedene Aufgaben ausführen, die eine schnelle serielle Verarbeitung erfordern. So kann die CPU beispielsweise die Wege der Spieler, die Schusslinien, das Treffen oder Verfehlen von Zielen, die Verfolgung des Inventars, die Kommunikation mit Computergeräten anderer Spieler (z. B. über das Internet oder ein anderes Netzwerk) usw. bestimmen. In mindestens einer Ausführungsform kann die GPU Aufgaben ausführen, die sich für eine parallele Verarbeitung eignen, wie z. B. die Durchführung von Matrixoperationen (z. B. Drehungen und Übersetzungen auf Pixelebene), die bestimmen, wie die Spielumgebung dem Benutzer erscheinen soll, und die Erzeugung eines endgültigen (kombinierten) Bildes, das dem Benutzer bereitgestellt wird. In mindestens einer Ausführungsform kann die GPU auf der Grundlage der von der CPU empfangenen Befehle den endgültigen Bildern auch verschiedene Hilfsdaten hinzufügen, wie Informationen über verfügbare Waffen, Munition, den aktuellen Gesundheitszustand und ähnliches. In mindestens einer Ausführungsform kann die CPU Anweisungen ausführen, die von einer Spielesoftware erzeugt wurden. In mindestens einer Ausführungsform kann ein Benutzer auch eine Betrugssoftware installieren, die, wenn sie zusammen mit einer Spielesoftware ausgeführt wird, die CPU veranlassen kann, modifizierte Befehle an die GPU auszugeben und die GPU zu veranlassen, Bilder wiederzugeben (oder direkt in den GPU-Speicher zu schreiben), die unzulässige oder gefälschte Informationen in Form von Fenstern, Einblendungen, halbtransparenten Überlagerungen und dergleichen enthalten.

[0020] In mindestens einer Ausführungsform können GPUs oder andere Verarbeitungseinheiten, die Bilder auf dem Bildschirm des Benutzers darstellen, ein oder mehrere trainierte neuronale Netze mit den dargestellten Bildern versorgen. Beispielsweise können Bilder aus einem Puffer abgerufen werden, in dem Bilder vor der Darstellung auf einem Display zwischengespeichert werden. In mindestens einer Ausführungsform können trainierte neuronale Netze verwendet werden, um festzustellen, ob ein Bild unerlaubte (z. B. betrügerische oder gefälschte) Informationen enthält oder nicht. In mindestens einer Ausführungsform kann die GPU (oder eine andere Verarbeitungslogik), wenn unter Verwendung trainierter neuronaler Netze festgestellt wird, dass das Bild Betrugsinformationen enthält, einen Bericht erstellen und den erstellten Bericht (z. B. über ein Netzwerk) an einen Server (z. B. einen Spielserver oder einen Veröffentlichungsserver eines Spieleanbieters) übermitteln, um den Server über einen erkannten Betrugsfall zu informieren. Trainierte neuronale Netze können auch einen Vertrauenswert liefern, der den Grad des Vertrauens in die Feststellung angibt, ob ein Bild

Betrugsinformationen enthält oder nicht. In mindestens einer Ausführungsform können die bereitgestellten Vertrauenswerte verwendet werden, um die Anzahl der falsch-positiven Fälle von Betrugserkennung zu reduzieren. Wenn das Vertrauen gering ist (z. B. unterhalb eines ersten Schwellenwerts), wird ein mutmaßlicher Betrugsfall möglicherweise nicht gemeldet. In mindestens einer Ausführungsform kann der Server, wenn das Vertrauensniveau (z. B. über einen bestimmten Satz von Bildern) unter einen bestimmten Wert (z. B. einen zweiten Schwellenwert) fällt, dies als Anzeichen dafür werten, dass eine neue Betrugssoftware verfügbar geworden ist oder eine neue oder geänderte Anwendung bereitgestellt wurde (z. B. ein neues Spielszenario oder eine neue Episode zum Spiel hinzugefügt wurde) und dass neuronale Netze nicht mehr verwendet werden können, um genaue Bestimmungen zu liefern, und dass sie unter Verwendung neuerer Bilder, z. B. von neuer Betrugssoftware erzeugter Bilder, neu trainiert werden sollten.

[0021] Obwohl es in den hier verwendeten Beispielen um die Erkennung von betrügerischen Informationen in Spielbildern geht, können ähnliche Geräte und Techniken auch zur Erkennung unzulässiger (oder gefälschter) Informationen in Bildern in anderen Zusammenhängen verwendet werden, in denen ein Benutzer in der Lage sein könnte, den Bereich des autorisierten Zugriffs zu überschreiten. In mindestens einer Ausführungsform können solche Zusammenhänge beispielsweise den Zugang zu Finanzinformationen, medizinischen Informationen, Versicherungsinformationen, Geheiminformationen oder anderen sicheren und/oder privaten Informationen umfassen. In mindestens einer Ausführungsform können solche Kontexte auch kostenpflichtige Abonnements, digitale Informationen, die durch Zugriffsschutzmaßnahmen (Passwörter, Signaturen, Mehrfaktor-Authentifizierungsmaßnahmen und ähnliches) geschützt sind, und Online-Prüfungen umfassen, bei denen ein Prüfer vom Besitz bestimmter Informationen oder vom Zugriff auf bestimmte Ressourcen (z. B. den Inhalt des Computers des Benutzers) ausgeschlossen ist. In mindestens einer Ausführungsform können unzulässige Informationen alle Arten von Informationen umfassen, zu denen ein Benutzer keinen rechtmäßigen Zugang haben darf oder soll. In mindestens einer Ausführungsform können unzulässige Informationen alle Informationen sein, zu denen der Zugang aufgrund von Gesetzen, Verträgen, arbeitsrechtlichen Vorschriften oder anderen Regeln oder Vereinbarungen eingeschränkt ist.

[0022] Fig. 3 ist ein Blockdiagramm eines beispielhaften Computersystems 100, in dem die Erkennung unerlaubter Informationen in Bildern, die zur Anzeige für einen oder mehrere Benutzer wiedergegeben werden, in Übereinstimmung mit mindestens einer Ausführungsform durchgeführt werden kann; In mindestens einer Ausführungsform, die in Fig. 3 dargestellt ist, kann das beispielhafte Computersystem 100 einen Veröffentlichungsserver 302 (z. B. einen Spielveröffentlichungsserver), einen Spielserver 304, einen oder mehrere Benutzerrechner 310, 312, 313 ..., einen Trainingsserver 340 und ein Trainingsbildarchiv 350 umfassen, von denen einige oder alle mit einem Netzwerk 308 verbunden sein können. In mindestens einer Ausführungsform kann das Netzwerk 308 ein öffentliches Netzwerk (z. B. das Internet), ein privates Netzwerk (z. B. ein lokales Netzwerk (LAN) oder ein Weitverkehrsnetzwerk (WAN)) oder eine Kombination davon sein. In mindestens einer Ausführungsform kann ein Benutzerrechner, z. B. der Benutzerrechner 310, über einen Netzwerkadapter 312 mit dem Netzwerk 308 verbunden sein. In mindestens einer Ausführungsform kann die Benutzermaschine 310 einen Speicher 314, eine Verarbeitungsvorrichtung, z. B. eine CPU 316, und ein Eingabe-/Ausgabemodul 318 umfassen. Bei der CPU 316 kann es sich um ein beliebiges Gerät handeln, das Befehle ausführen kann, die arithmetische, logische oder E/A-Operationen kodieren. In mindestens einer Ausführungsform kann die Speichervorrichtung 314 ein flüchtiger oder nichtflüchtiger Speicher sein, z. B. ein Direktzugriffsspeicher (RAM), ein Festwertspeicher (ROM), ein elektrisch löschbarer programmierbarer Festwertspeicher (EEPROM) oder eine andere Vorrichtung, die Daten speichern kann. In mindestens einer Ausführungsform kann die CPU 316 dem Von-Neumann-Architekturmodell folgen und eine oder mehrere arithmetische Logikeinheiten (ALU), Steuereinheiten und eine Vielzahl von Registern umfassen. In mindestens einer Ausführungsform kann die CPU 316 ein Einkernprozessor sein, der jeweils einen Befehl ausführen kann, oder ein Mehrkernprozessor, der gleichzeitig mehrere Befehle ausführen kann. In mindestens einer Ausführungsform kann die CPU 316 als einzelner integrierter Schaltkreis, als zwei oder mehr integrierte Schaltkreise oder als Bestandteil eines Multi-Chip-Moduls ausgeführt sein. In mindestens einer Ausführungsform kann die Benutzermaschine 310 auch ein oder mehrere E/A-Geräte 319 enthalten, die vom E/A-Modul 318 unterstützt werden. In mindestens einer Ausführungsform können die E/A-Geräte 319 Tastaturen, Mäuse, Zeiger, Touchscreens, Touchpads, Kameras, Mikrofone oder andere Arten von Sensoren umfassen, die Benutzereingaben erkennen können.

[0023] In mindestens einer Ausführungsform kann das Benutzergerät 310 eine GPU 320 beinhalten. Die GPU 320 kann mehrere Kerne 322 haben, wobei jeder Kern in der Lage ist, mehrere Themen gleichzeitig und parallel auszuführen. Die GPU-Kerne 322 können Zugriff auf den GPU-Speicher 324 haben, der private (themen-spezifische) Register, gemeinsam genutzte Register, Cache (z. B. L1, L2, L3 usw.) und/oder andere Speichergeräte umfassen kann. In mindestens einer Ausführungsform kann der GPU-Speicher 324 einen

nichtflüchtigen Speicher enthalten, in dem Modelle 325 eines neuronalen Netzes (NN) zur Erkennung unzulässiger Bilder gespeichert sind, z. B. Bilder, die zumindest einige unerlaubte Informationen enthalten, zu deren Besitz ein Benutzer nicht berechtigt ist. Die NN-Modelle 325 können von einem oder mehreren Kernen 322 ausgeführt werden. In mindestens einer Ausführungsform können die NN-Modelle 325 von einer in **Fig. 3** nicht dargestellten Verarbeitungsvorrichtung separat oder in Verbindung mit den Kernen 322 ausgeführt werden. In mindestens einer Ausführungsform kann die GPU 320 einen Puffer 326 zum Speichern von Spielbildern (oder anderen Bildern) enthalten, die von der GPU 320 für die Anzeige auf dem Display 329 vorbereitet wurden. In mindestens einer Ausführungsform kann der Puffer 229 ein FIFO-Puffer (first-in-first-out) sein. In mindestens einer Ausführungsform kann es sich bei den Bildern um beliebige Rasterbilder handeln, einschließlich Bildern in beliebigen pixelbasierten digitalen Formaten, wie z. B. BMP, TIFF, CIB, JPEG, DIMAP, GIF, NITF, PNG und so weiter. In mindestens einer Ausführungsform kann es sich bei den Bildern um vektorisierte Bilder handeln, bei denen die Objekte durch mathematische Beziehungen dargestellt werden, z. B. durch die Definition von Linien, Kurven, vordefinierten Formen oder einer beliebigen Kombination davon. In mindestens einer Ausführungsform kann der GPU-Speicher 324 anstelle oder zusätzlich zu den NN-Modellen 325 ein oder mehrere Modelle des maschinellen Lernens enthalten, die sich von trainierten neuronalen Netzen unterscheiden, z. B. einen Baumklassifikator, einen Bayes-Klassifikator, einen Regressionsklassifikator, einen stochastischen Gradientenabstiegsklassifikator, einen K-Nächste-Nachbarn-Klassifikator oder andere Algorithmen zur Erkennung unzulässiger Informationen.

[0024] In mindestens einer Ausführungsform können die NN-Modelle 325 vom Trainingsserver 340 trainiert werden. In mindestens einer Ausführungsform kann der Trainingsserver 340 ein Rackmount-Server, ein Router-Computer, ein Personal Computer, ein Laptop-Computer, ein Tablet-Computer, ein Desktop-Computer, ein Media Center oder eine beliebige Kombination davon sein (und/oder umfassen). In mindestens einer Ausführungsform kann der Trainingsserver 340 eine Trainingsmaschine 342 beinhalten. In mindestens einer Ausführungsform kann die Trainingsmaschine 342 maschinelle Lernmodelle, z. B. NN-Modelle 325, erstellen. In mindestens einer Ausführungsform können NN-Modelle 325 von der Trainingsmaschine 342 unter Verwendung von Trainingsdaten trainiert werden, die Trainingseingaben 347 und entsprechende Zielausgaben 348 enthalten. In mindestens einer Ausführungsform können die Zielausgaben 348 korrekte Assoziationen (Zuordnungsdaten 349) von Trainingsbildern mit dem Vorhandensein oder Nichtvorhandensein unerlaubter Informationen in Trainingsbildern enthalten. In mindestens einer Ausführungsform kann die Trainingsmaschine 342 Muster in den Trainingseingaben 347 ableiten, die die Trainingseingaben 347 auf die Zielausgaben 348 abbilden (die korrekte Assoziationen enthalten können, die während des Trainings vorhergesagt werden sollen) und NN-Modelle 325 trainieren, um solche Muster zu erfassen. In mindestens einer Ausführungsform können die Muster anschließend von trainierten NN-Modellen 325 für künftige Inferenzen verwendet werden, die für neue Bilder durchgeführt werden. Nachdem beispielsweise auf ein oder mehrere neue Bilder im Puffer 326 zugegriffen wurde, können trainierte NN-Modelle 325 das Vorhandensein oder Nichtvorhandensein unzulässiger Informationen in neuen Bildern bestimmen. In mindestens einer Ausführungsform kann der Trainingsserver 340 in der Lage sein, die NN-Modelle 325 neu zu trainieren, wenn neue Informationen (z. B. über die Verfügbarkeit neuer Betrugssoftware, Änderungen/Aktualisierungen bestehender Spiele oder das Aufkommen neuer Spiele oder Spielszenarien) verfügbar werden. In mindestens einer Ausführungsform kann der Trainingsserver 340 in der Lage sein, nach dem anfänglichen Training oder einem späteren Neutaining geänderte Parameter der NN-Modelle 325 an verschiedene GPUs (z. B. GPU 320) von Benutzermaschinen (z. B. Benutzermaschine 310) zu senden, um die darauf installierten NN-Modelle zu aktualisieren.

[0025] In mindestens einer Ausführungsform können die Trainingsbilder 352 in einem Trainingsbildspeicher 350 gespeichert werden, auf den der Trainingsserver 340 direkt oder über das Netzwerk 308 zugreifen kann. In mindestens einer Ausführungsform kann der Trainingsbildspeicher 350 ein dauerhafter Speicher sein, der sowohl Trainingsbilder als auch Metadaten für gespeicherte Trainingsbilder speichern kann. In mindestens einer Ausführungsform kann das Verzeichnis der Trainingsbilder 350 von einem oder mehreren Speichergeräten gehostet werden, z. B. von einem Hauptspeicher, magnetischen oder optischen Speicherplatten, Bändern oder Festplatten, NAS, SAN usw. Obwohl in der Abbildung getrennt vom Trainingsserver 340 dargestellt, kann in mindestens einer Ausführungsform der Bildspeicher 350 ein Teil des Trainingsservers 340 sein. In mindestens einer Ausführungsform kann der Bildspeicher 350 ein netzgebundener Dateiserver sein, während in anderen Ausführungsformen der Bildspeicher 350 eine andere Art von dauerhaftem Speicher sein kann, wie z. B. eine objektorientierte Datenbank, eine relationale Datenbank usw., die von einem oder mehreren verschiedenen Rechnern gehostet werden kann, die mit dem Trainingsserver 340 über das Netzwerk 308 kommunizieren.

[0026] In mindestens einer Ausführungsform können Trainingsbilder 352 durch Wiederholung von Spielen unter Verwendung der auf dem Spieleserver 304 gespeicherten Spielprotokolle 306 erzeugt werden. Insbesondere können die Spielprotokolle 306 verwendet werden, um Spielkontexte erneut abzuspielen, Bilder mit Hilfe von Spielsoftware neu wiederzugeben und unzulässige Informationen hinzuzufügen, um einen Satz von Trainingsbildern zu erstellen. In mindestens einer Ausführungsform können unzulässige Informationen zu echten (nicht unzulässigen) Spielbildern (oder Bildern aus anderen Quellen) hinzugefügt werden. In mindestens einer Ausführungsform können verschiedene Überlagerungen (z. B. aus öffentlich zugänglichen Video- und Grafikressourcen) zu Trainingsbildern hinzugefügt werden, um NN-Modelle 325 so zu trainieren, dass sie Überlagerungen ignorieren, die keine unzulässigen Informationen enthalten.

[0027] In mindestens einer Ausführungsform können die Trainingsbilder 352 mit Hilfe von Betrugssoftware 346 erzeugt werden, z. B. mit einer Betrugssoftware, die von betrügenden Nutzern (z. B. Spielern) oder anderen nicht autorisierten Nutzern unerlaubter Informationen verwendet werden kann. In mindestens einer Ausführungsform, die im Folgenden näher beschrieben wird, können einige der Trainingsbilder 352 echte (Nichtbetrugs-)Bilder sein, die von einer legitimen Software, z. B. der Spielesoftware 345, erzeugt wurden. In mindestens einer Ausführungsform kann es sich bei echten Bildern um Bilder handeln, die keine unzulässigen Informationen enthalten. In mindestens einer Ausführungsform können echte Bilder von jeder legitimen Software erzeugt werden, die sich von Spielsoftware unterscheidet, z. B. Finanz-, Versicherungs-, Buchhaltungs-, Gesundheitssoftware usw. In mindestens einer Ausführungsform können die Trainingsbilder 352 Bilder enthalten, die unzulässige Informationen enthalten (Trainingsbetrugsbilder), die von einer Betrugssoftware 346 erzeugt und von einem Trainingsbildgenerator 344 zu echten Bildern hinzugefügt werden. In mindestens einer Ausführungsform kann der Trainingsbildgenerator 344 die Menge der betrügerischen Informationen in den Trainingsbetrugsbildern variieren, indem er Teile der unzulässigen Informationen, die von der Betrugssoftware 346 generiert wurden, mit Teilen der Nichtbetrugsbilder, die von der Spielesoftware 345 generiert wurden, kombiniert (überlagert, überblendet, verdeckt, ersetzt und dergleichen). In mindestens einer Ausführungsform kann dies dadurch erreicht werden, dass NN-Modelle 325 so trainiert werden, dass sie Betrugsbilder erkennen, die auch nur einen geringen Anteil an unzulässigen Informationen enthalten, um zu verhindern, dass NN-Modelle 325 ihre Aufmerksamkeit auf die offensichtlichsten betrügerischen Informationen richten, die von ausgeklügelten Betrügern, die nur einen kleinen Wettbewerbsvorteil suchen und ihre Betrugssoftware so konfigurieren, dass sie nur kleine Bruchteile der verfügbaren betrügerischen Informationen liefern, ausgeschaltet werden könnten. In mindestens einer Ausführungsform können die NN-Modelle 325 so trainiert werden, dass sie die (durch Einfügen verursachten) Artefakte in den Trainingsbildern ignorieren. Genauer gesagt, können nicht unzulässige Teile (z. B. Teile ohne unzulässige Informationen) von Betrugsbildern auf ähnliche Teile von nicht betrügerischen Trainingsbildern geklebt werden. Folglich können sich Trainingsbetrugsbilder und nicht betrügerische Trainingsbilder durch das Vorhandensein unzulässiger Informationen unterscheiden (für deren Erkennung NN-Modelle 325 trainiert werden), während sie die gleichen oder ähnliche Einfügeartefakte aufweisen (für deren Ignorieren NN-Modelle 325 trainiert werden).

[0028] In mindestens einer Ausführungsform kann die GPU 320 ein Zertifizierungsmodul 328 enthalten, um zu zertifizieren, dass ein oder mehrere Benutzer des Benutzersystems 310 eine Spielehardware (z. B. GPU 320) verwenden, die in der Lage ist, Betrug zu erkennen. In mindestens einer Ausführungsform kann das Zertifizierungsmodul 328 dem Veröffentlichungsserver 302 und/oder dem Spieleserver 304 eine Nachricht übermitteln, die anzeigt, dass die GPU 320 eine zertifizierte Anti-Betrugs-GPU ist. In mindestens einer Ausführungsform kann die übermittelte Nachricht die GPU 320 anhand ihrer eindeutigen Kennung (ID) identifizieren, bei der es sich um eine MAC-Adresse (Media Access Control) der GPU 320 oder eine andere Form der Identifizierung handeln kann (z. B. eine eindeutige ID, die einer bestimmten GPU 320 zugewiesen ist, die auf dem Benutzerrechner 310 installiert ist). In mindestens einer Ausführungsform kann die Nachricht zu Beginn der Spielsitzung des Benutzers übermittelt werden. In mindestens einer Ausführungsform können zusätzliche Nachrichten während der laufenden Spielsitzung des Benutzers (z. B. in regelmäßigen Abständen) und/oder nach Abschluss der Spielsitzung des Benutzers übermittelt werden. Solche zusätzlichen Meldungen können Berichte darüber enthalten, ob Betrugsfälle aufgedeckt worden sind.

[0029] In mindestens einer Ausführungsform können die in Verbindung mit den **Fig. 3-8** beschriebenen Systeme und Verfahren in Geräten implementiert werden, die sich von der GPU 320 unterscheiden, z. B. in einer Display-Treiberkarte (DDB), die Bilder auf dem Display 329 wiedergibt. In mindestens einer Ausführungsform können sich die NN-Modelle 325 auf der DDB befinden und auf Bilder zugreifen, die von der DDB auf der Anzeige 329 wiedergegeben werden. In mindestens einer Ausführungsform können die in Verbindung mit den **Fig. 3-8** beschriebenen Systeme und Verfahren dazu verwendet werden, Betrug in Bildern zu erkennen, die von Geräten wiedergegeben werden, die sich von den Erkennungsgeräten unterscheiden (z. B. können die Erkennungsgeräte von der GPU 320 getrennt sein). In mindestens einer Ausführungsform kann ein

Erkennungsgerät (z. B. GPU 320) auf Bilder zugreifen, die von anderen Geräten und/oder Anwendungen, wie z. B. einer DDB, auf dem Display 329 dargestellt werden. In mindestens einer Ausführungsform kann die Betrugssoftware unzulässige Informationen auf dem Display 329 getrennt von der Anzeige von Bildern anzeigen, z. B. in einem separaten Fenster, unter Verwendung einer Überlagerung, die von anderen Display-Controllern als der GPU 320 erzeugt wird, usw. In mindestens einer Ausführungsform kann der Grafikprozessor 320 (oder ein anderes Erkennungsgerät, z. B. eine auf die Anzeige 329 gerichtete Kamera) Bilder abrufen, die auf der Anzeige 329 wiedergegeben werden (oder für eine spätere Wiedergabe auf der Anzeige 329 gespeichert werden), z. B. durch eine Anzeigetreiberplatine, und die Erkennung von Betrug (oder anderen unzulässigen Informationen) in ähnlicher Weise durchführen, wie unzulässige Informationen erkannt werden (wie unten beschrieben), wenn die endgültigen Bilder vom Grafikprozessor 320 wiedergegeben werden.

[0030] Fig. 4 ist eine schematische Darstellung eines Bildes, das unerlaubte Informationen enthält, die von einer Grafikverarbeitungseinheit auf einem Display eines Benutzergeräts gemäß mindestens einer Ausführungsform dargestellt werden. Fig. 4 ist ein unzulässiges Bild 400 einer beispielhaften Spielumgebung, die Darstellungen von gegnerischen Spielern, Strukturen, Gebäuden, Hindernissen, Pfaden, natürlichen Gegenständen und anderen Objekten, die in einem bestimmten Spiel vorhanden sein können, enthalten kann. Obwohl Fig. 4 eine Umgebung eines Computerspiels zeigt, kann in mindestens einer Ausführungsform auch ein Bild einer beliebigen anderen Art von Information angezeigt werden - Finanz-, Bildungs-, medizinische Informationen usw. In mindestens einer Ausführungsform kann eine angezeigte Umgebung (z. B. eine Spielumgebung) unzulässige Informationen enthalten, die dem Benutzer einen unfairen Vorteil verschaffen. Die in Fig. 4 gezeigte Spielumgebung kann beispielsweise globale unzulässige Informationen 402 enthalten, wie z. B. Informationen, die sich auf ein Spiel als Ganzes beziehen, z. B. die Anzahl und die Eigenschaften von gegnerischen Spielern, eine Karte der Spielumgebung, die für den Benutzer nicht zugänglich sein sollte, Hinweise auf Munition und medizinische Versorgung und so weiter. In mindestens einer Ausführungsform kann die Spielumgebung darüber hinaus eine Vielzahl lokaler unzulässiger Informationen enthalten, wie z. B. Informationen über einen bestimmten Spieler oder ein bestimmtes Objekt auf einem Bildschirm, z. B. die Informationen 404-410. Lokale unzulässige Informationen 404 können z.B. Informationen über Gesundheit, Munition, Status usw. enthalten, die mit einer Darstellung (z.B. Abbildung) eines gegnerischen Spielers 420 verbunden sind, lokale unzulässige Informationen 406 können dem Benutzer die Anwesenheit von Gegnern anzeigen, die sich hinter einer Wand 430 verstecken, lokale unzulässige Informationen 408 können dem Benutzer die Kommunikation unterhalb einer Struktur 440 anzeigen, lokale Betrugsinformationen 410 können dem Benutzer die Anwesenheit von Gegnern innerhalb der Struktur 450 anzeigen, usw. In mindestens einer Ausführungsform werden unzulässige Informationen von der CPU 316 hinzugefügt (und möglicherweise von der GPU 320 auf Anweisung der CPU 316 wiedergegeben), die Software ausführt, die auf dem Benutzerrechner 310 installiert und im Speicher 314 gespeichert ist. Illegale Informationen können über echte Bilder, die von der Spielesoftware 345 erzeugt wurden, eingefügt, hinzugefügt, überlagert usw. werden. In mindestens einer Ausführungsform können die unzulässigen Informationen in einem beliebigen Format dargestellt werden, z. B. als Text, gerasterte Bilder, vektorisierte Objekte, Videos, Animationen usw. oder in einer beliebigen Kombination davon. In mindestens einer Ausführungsform kann die GPU 320 ein letztes Gerät sein, das die Bilder auf dem Display 329 wiedergibt. Dementsprechend kann die GPU 320 im Besitz von Grafikinformatoren sein, die dem Benutzer tatsächlich angezeigt werden, da keine zusätzliche Komponente die angezeigten Bilder bearbeitet, z. B. keine Komponente unzulässige Informationen zu ansonsten nicht unzulässigen Bildern hinzufügt oder unzulässige Informationen abfängt (und möglicherweise unzulässige Informationen entfernt, nachdem sie einem Benutzer angezeigt wurden).

[0031] In mindestens einer Ausführungsform kann die auf dem Trainingsserver 340 installierte Betrugssoftware 346 Bilder erzeugen, die denen ähneln, die von betrügenden Spielern (oder anderen Nutznießern oder Benutzern unzulässiger Informationen) gesehen werden. Die Betrugssoftware 346 kann beispielsweise dieselbe oder eine ähnliche Software sein, die von betrügenden Benutzern verwendet wird. In mindestens einer Ausführungsform werden unmodifizierte unzulässige Trainingsbilder (wie die in Fig. 4) von der Betrugssoftware 346 zum Trainieren der NN-Modelle 325 verwendet. In mindestens einer Ausführungsform können zumindest einige der Trainingsbilder auf Bildern basieren, die von der Betrugssoftware 346 erzeugt, aber zusätzlich vom Trainingsbildgenerator 344 modifiziert wurden. Genauer gesagt kann die Menge an unzulässigen Informationen in den Trainingsbildern variiert werden, z. B. erhöht oder verringert im Vergleich zur Menge an unzulässigen Informationen in den Originalbildern, die von der Betrugssoftware 346 erzeugt wurden. In mindestens einer Ausführungsform können erhöhte Mengen verwendet werden, um das anfängliche Training der NN-Modelle 325 zu erleichtern (z. B. um das anfängliche Training zu beschleunigen), während geringere Mengen verwendet werden können, um die NN-Modelle 325 so zu trainieren, dass sie selbst ein geringes Vorhandensein unzulässiger Informationen erkennen. In mindestens einer Ausführungsform können NN-Modelle 325, die mit unterschiedlichen Mengen an unzulässigen Informationen trainiert wurden, selbst

vorsichtige Spieler effizient identifizieren, die (in dem Bestreben, unentdeckt zu bleiben) einen relativ bescheidenen Vorteil gegenüber der Konkurrenz anstreben, indem sie die Menge der angezeigten unzulässigen Informationen begrenzen. In mindestens einer Ausführungsform können die NN-Modelle 325 anhand von Bildern trainiert werden, die schrittweise weniger unzulässige Informationen enthalten.

[0032] In mindestens einer Ausführungsform können unterschiedliche Mengen unzulässiger Informationen in Trainingsbildern durch die unten beschriebenen Augmentierungstechniken und -verfahren erreicht werden. **Fig. 5A** ist eine schematische Darstellung eines erweiterten unzulässigen Trainingsbildes 500 mit einer reduzierten Menge an unzulässigen Informationen für ein effizientes Training neuronaler Netze zur Erkennung unzulässiger Informationen in Bildern, die zur Anzeige für einen oder mehrere Benutzer wiedergegeben werden, in Übereinstimmung mit mindestens einer Ausführungsform. **Fig. 5A** ist ein Bild einer Spielumgebung, die aus einem unzulässigen Bild 400 gewonnen werden kann, das dem in **Fig. 4** dargestellten Bild ähnelt. In mindestens einer Ausführungsform wird ein Teil der unzulässigen Informationen im Bild 400 zurückgehalten. So werden beispielsweise lokale unzulässige Informationen 406 (die auf das Vorhandensein von Gegnern hinweisen), lokale unzulässige Informationen 408 (die auf verdeckte Kommunikation hinweisen) und lokale unzulässige Informationen 410 (die ebenfalls auf das Vorhandensein von Gegnern hinweisen) angezeigt. In mindestens einer Ausführungsform kann die Menge der unzulässigen Informationen in den Trainingsbildern begrenzt werden, z. B. durch Änderung der Einstellungen der Betrugssoftware. In mindestens einer Ausführungsform kann ein Teil der unzulässigen Informationen entfernt werden. Beispielsweise können sowohl globale unzulässige Informationen 402 als auch lokale unzulässige Informationen 404 (die den Gesundheitszustand, die Munition, den Status usw. des Gegners anzeigen) entfernt werden. In mindestens einer Ausführungsform kann ein Teil der unzulässigen Informationen entfernt werden, wenn Paare von echten/unzulässigen (Trainings-)Bildern verfügbar sind. In mindestens einer Ausführungsform kann ein echtes Trainingsbild von einer Spielsoftware und ein unzulässiges Trainingsbild von einer Betrugssoftware erzeugt werden. In mindestens einer Ausführungsform kann die Entfernung unzulässiger Informationen dadurch erfolgen, dass ein Bereich des unzulässigen Trainingsbildes durch einen entsprechenden Bereich eines echten Bildes ersetzt wird (z. B. ein echtes Bild der gleichen oder einer ähnlichen Umgebung, das von der Spielsoftware 345 erzeugt wurde). In mindestens einer Ausführungsform kann eine Ersatzregion 502 globale unzulässige Informationen 402 abdecken (ersetzen, überkleben, überlagern, überdecken usw.), während eine Ersatzregion 504 lokale unzulässige Informationen 404 abdecken kann.

[0033] In mindestens einer Ausführungsform kann das erhaltene erweiterte unzulässige Trainingsbild 500 an den Stellen, an denen das Einfügen oder Ersetzen stattgefunden hat, Artefakte durch Einfügen/Überlagern/-Überdecken usw. aufweisen. In mindestens einer Ausführungsform kann ein echtes Bild, das von der Spielsoftware 345 erzeugt wird, keine exakte Pixel-zu-Pixel-Synchronisation mit einem unzulässigen Bild aufweisen, das von einer Betrugssoftware (allein oder in Kombination mit der Spielsoftware 345) erzeugt wird. In mindestens einer Ausführungsform können unzulässige Bilder und/oder echte Bilder einer Komprimierung unterzogen werden, wodurch die Pixel-zu-Pixel-Synchronisation ebenfalls gestört werden kann. Dies kann zu Einfügeartefakten in den ergänzten unzulässigen Bildern führen. Selbst wenn sie für das menschliche Auge kaum sichtbar sind, können NN-Modelle 325 bei einer Pixel-für-Pixel-Analyse durch das Einfügen von Artefakten in unzulässige Trainingsbilder dazu führen, dass NN-Modelle 325 unzulässige Bilder identifizieren, indem sie sich auf Artefakte (statt auf unzulässige Informationen) konzentrieren. In mindestens einer Ausführungsform können ähnliche Artefakte in echte Trainingsbilder eingefügt werden, um die Entwicklung einer solchen verzerrten Wahrnehmung durch NN-Modelle 325 zu verhindern.

[0034] **Fig. 5B** ist eine schematische Darstellung eines erweiterten echten Trainingsbildes 550, das Einfügeartefakte enthält, für ein effizientes Training neuronaler Netze zur Erkennung unzulässiger Informationen in Bildern, die zur Anzeige für einen oder mehrere Benutzer wiedergegeben werden, in Übereinstimmung mit mindestens einer Ausführungsform. **Fig. 5B** ist ein Bild 550 einer Spielumgebung, das aus einem echten Bild nach einem oder mehreren Einfügungen/Ersetzungen gewonnen werden kann. Zum Beispiel sind die Ersatzregionen 554, 556 und 558 als hinzugefügt dargestellt. Das Hinzufügen von Ersatzregionen führt zu Einfügeartefakten in echten Bildern und - aufgrund des Vorhandenseins von Einfügeartefakten sowohl in echten als auch in unzulässigen Trainingsbildern - dazu, dass NN-Modelle 325 Einfügeartefakte ignorieren und echte Trainingsbilder von unzulässigen Trainingsbildern auf der Grundlage des Vorhandenseins von unzulässigen Informationen unterscheiden. In mindestens einer Ausführungsform werden die Ersatzregionen 554, 556 und 558 aus unzulässigen Bildern entnommen, z. B. aus unzulässigen Bildern, die demselben Computerspiel, demselben Aufbau, denselben Spielern, derselben Szenerie usw. entsprechen. In mindestens einer Ausführungsform werden die Ersatzregionen 554, 556 und 558 aus solchen Regionen (von unzulässigen Bildern) entnommen, die keine unzulässigen Informationen enthalten, um eine Kontamination der erweiterten echten Trainingsbilder 550 durch Pixel zu verhindern, die unzulässigen Informationen entsprechen.

[0035] In mindestens einer Ausführungsform kann die Trainingssoftware 342 (automatisch oder auf Befehl eines Entwicklers) die Spielesoftware 345 starten, um echte Bilder zu erzeugen, die mit dem Spiel verbunden sind, um erweiterte echte Trainingsbilder und erweiterte unzulässige Trainingsbilder zu erzeugen. Zusätzlich kann (separat oder parallel) in mindestens einer Ausführungsform das Trainingsbild (und/oder der Entwickler) die Betrugssoftware 346 starten, um (allein oder in Zusammenarbeit mit der Spielesoftware 345) ein oder mehrere unzulässige Basisbilder zu erzeugen, bei denen es sich um unmodifizierte unzulässige Bilder handeln kann, die von der Spielesoftware erzeugt wurden. In mindestens einer Ausführungsform können die Spielesoftware 345 und die Betrugssoftware 346 Bilder erzeugen, die denselben oder ähnlichen Spielkontexten entsprechen. In mindestens einer Ausführungsform kann eine Instanz (erster Kontext) der Spielesoftware 345 mit einer bestimmten anfänglichen Position der Spieler, dem Zustand von Waffen, Gesundheit, Munition usw. beginnen. Gleichzeitig kann in mindestens einer Ausführungsform eine andere Instanz (zweiter Kontext, der eine Kopie des ersten Kontexts sein kann) der Spielesoftware 345 in Verbindung mit der Betrugssoftware 346 gestartet werden und die Ausführung desselben (oder eines ähnlichen) Spielkontextes zur selben (oder einer ähnlichen) Zeit mit denselben (oder ähnlichen) Spielern, Waffen, Gesundheit, Munition usw. beginnen. In mindestens einer Ausführungsform kann der Trainingsbildgenerator 344 dazu verwendet werden, beide Kontexte etwa zur gleichen (oder annähernd gleichen) Zeit abzutasten, um Bilder (z. B. Schnappschüsse) zu extrahieren, die von zwei Instanzen des Spiels stammen. In mindestens einer Ausführungsform kann der Trainingsserver 340 über separate GPUs verfügen, die Bilder für unterschiedliche Spielkontexte erzeugen.

[0036] In mindestens einer Ausführungsform kann der Trainingsbildgenerator 344 verwendet werden, um (z. B. auf Anweisung eines Entwicklers oder automatisch) einige unzulässige Informationen aus unzulässigen Basisbildern (die durch den zweiten Kontext erzeugt wurden) in unterschiedlichem Ausmaß zu entfernen. In mindestens einer Ausführungsform kann es sich bei den unzulässigen Basisbildern um unmodifizierte unzulässige Bilder handeln, die von der Spielsoftware erzeugt wurden. In mindestens einer Ausführungsform kann ein unzulässiges Basisbild verwendet werden, um mehrere unzulässige Trainingsbilder zu erzeugen, z. B. mit 10%, 20%, 50%, 70% usw. entfernter unzulässiger Information. In mindestens einer Ausführungsform können den Basisbildern zusätzliche unzulässige Informationen hinzugefügt werden, um unzulässige Trainingsbilder mit 110%, 120% usw. an unzulässigen Informationen zu erzeugen. In mindestens einer Ausführungsform können einige der unzulässigen Informationen aus demselben Basisbild an zusätzlichen Stellen im Basisbild eingefügt werden. In mindestens einer Ausführungsform können unzulässige Informationen aus anderen ähnlichen Bildern entnommen werden, z. B. aus früheren oder späteren Bildern innerhalb desselben Spielkontexts oder aus Bildern, die aus anderen Spielkontexten stammen.

[0037] In mindestens einer Ausführungsform kann der Trainingsbildgenerator 344 außerdem dazu verwendet werden, nicht unzulässige Teile unzulässiger Bilder zu echten Bildern (die z. B. durch den ersten Kontext erzeugt wurden) hinzuzufügen (z. B. durch Einfügen, Überlagern, Übereinanderlegen usw.). In mindestens einer Ausführungsform können Menge, Größe und Position der Einfügungen in echte Bilder mit der Menge der Entfernungen (oder Einfügungen) in unzulässige Bilder vergleichbar sein, so dass erweiterte unzulässige Bilder ungefähr die gleiche (oder vergleichbare) Menge und Art von Artefakten aufweisen, die durch Einfügungen verursacht werden, wie erweiterte echte Bilder.

[0038] In mindestens einer Ausführungsform kann der Trainingsserver 340 als Ergebnis der in den vorangehenden Abschnitten beschriebenen Bildmanipulationen ein oder mehrere Trainingsbilder 352 erzeugen, einschließlich eines oder mehrerer erweiterter unzulässiger Trainingsbilder (wie z. B. Bild 500) und eines oder mehrerer erweiterter echter Trainingsbilder (wie z. B. Bild 550), die von der Trainingsmaschine 342 zum Trainieren eines oder mehrerer NN-Modelle 325 verwendet werden können. In mindestens einer Ausführungsform kann die Anreicherung von echten und unzulässigen Trainingsbildern automatisch und ohne detaillierte Eingaben eines Entwicklers erfolgen. Zum Beispiel kann ein Entwickler Regionen markieren, die eine unzulässige Information in einem unzulässigen Basisbild enthalten, während der Trainingsbildgenerator 344 nach dem Zufallsprinzip Teile der unzulässigen Information auswählen kann, die aus den unzulässigen Basisbildern entfernt werden, oder Teile der echten Basisbilder, die durch nicht unzulässige Regionen der unzulässigen Bilder ersetzt werden.

[0039] **Fig. 6** ist eine schematische Darstellung eines Systems 600 mit einem oder mehreren neuronalen Netzmodellen zur Erkennung unerlaubter Informationen in Bildern, die von einer Grafikverarbeitungseinheit auf einem Display eines Benutzergeräts gemäß mindestens einer Ausführungsform dargestellt werden. In mindestens einer Ausführungsform kann das Training eines oder mehrerer in **Fig. 6** dargestellter NN-Modelle (oder im Folgenden einfach neuronaler Netze) von der Trainingsmaschine 342 des Trainingsservers 340 durchgeführt werden. In mindestens einer Ausführungsform können die in **Fig. 6** dargestellten neuronalen Netze die NN-Modelle 325 aus **Fig. 3** sein. 3. In mindestens einer Ausführungsform können die trainierten

neuronalen Netze nach dem Training in die GPU 210 integriert werden, die wiederum in die Benutzermaschine 310 integriert werden kann. In mindestens einer Ausführungsform kann die GPU 210 eine Doppelfunktion erfüllen: Bilder auf dem Display 329 darstellen und gleichzeitig (vorher oder nachher) neuronale Netze auf die dargestellten Bilder anwenden, um das Vorhandensein unzulässiger (z. B. betrügerischer) Informationen darin zu erkennen. In mindestens einer Ausführungsform kann ein Bild 602 in das System 600 eingegeben werden. In mindestens einer Ausführungsform kann das Bild 602 eines der Trainingsbilder 352 (während der Trainingsphase verwendet) oder ein neues Bild (während der Inferenzphase) sein. In mindestens einer Ausführungsform kann das Bild 602 in ein erstes Netzwerk 610 eingegeben werden. In mindestens einer Ausführungsform kann das erste Netzwerk 610 so trainiert werden, dass es das Bild 602 auf einer lokalen Ebene klassifiziert. Die lokale Ebene kann sich auf einzelne Pixel des Bildes 602 beziehen. Alternativ kann sich die lokale Ebene auch auf Superpixel (Ansammlungen von Pixeln) beziehen, die mehrere Pixel umfassen können, z. B. Blöcke von 2×2 , 4×4 , 16×16 , 8×16 oder jede andere Superpixelgröße. Wenn im Folgenden von einem Pixel oder Pixeln die Rede ist, so gilt dies auch für ein oder mehrere Superpixel in Ausführungsformen, die Superpixel-Darstellungen von Bildern verwenden.

[0040] In mindestens einer Ausführungsform kann eine Ausgabe des ersten Netzwerks 610 eine Merkmalskarte 620 mit lokalen Scores 622 sein, die die Wahrscheinlichkeit angibt, dass ein bestimmtes Pixel (oder Superpixel) unzulässige Informationen enthält (oder zu ihnen gehört). In mindestens einer Ausführungsform kann eine lokale Punktzahl 622 eine Zahl in einem Bereich von 0 bis 1 (z. B. Wahrscheinlichkeit) oder in einem anderen vordefinierten Bereich (z. B. 0 bis 10, der mit der Wahrscheinlichkeit zusammenhängt, ihr aber nicht unbedingt entspricht) sein, wobei ein Wert nahe einem Ende des Bereichs (z. B. 0) anzeigt, dass es sehr unwahrscheinlich ist, dass ein Pixel zu einer Gruppe von Pixeln gehört, die unzulässige Informationen darstellen, und ein Wert nahe dem entgegengesetzten Ende des Bereichs (z.B. 1) anzeigt, dass ein Pixel sehr wahrscheinlich zu einer Gruppe von Pixeln gehört, die unzulässige Informationen darstellen. In mindestens einer Ausführungsform kann das Eingangsbild 602 zunächst in eine digitale Darstellung einer Pixelkarte des Bildes 602 umgewandelt werden. In mindestens einer Ausführungsform kann die Anzahl der Pixel von der Auflösung des Bildes 602 abhängen, z. B. kann ein Bild 2048×1024 Pixel, 1680×1050 Pixel oder eine andere Auflösung haben. In mindestens einer Ausführungsform kann jedes Pixel durch einen oder mehrere Intensitätswerte charakterisiert werden. In mindestens einer Ausführungsform kann ein Schwarz-Weiß-Pixel durch einen Intensitätswert gekennzeichnet sein, der die Dunkelheit des Pixels darstellt, z. B. kann der Wert 0 (oder 1) einem weißen Pixel und der Wert 1 (oder 0) einem schwarzen Pixel entsprechen. In mindestens einer Ausführungsform kann der Intensitätswert kontinuierliche (oder quasikontinuierliche) Werte zwischen 0 und 1 (oder zwischen anderen gewählten Grenzen) annehmen. Ebenso kann ein Farbpixel durch mehrere Intensitätswerte dargestellt werden. Im RGB-Schema kann es beispielsweise drei verschiedene Intensitätswerte für die Farben Rot, Grün und Blau geben. Im CMYK-Schema kann es vier Intensitätswerte geben.

[0041] In mindestens einer Ausführungsform kann die Anzahl der Eingangsknoten 612 des ersten Netzes 610 gleich der Anzahl der Pixel (oder der Gesamtzahl der Parameter, die alle Pixel beschreiben) sein. In mindestens einer Ausführungsform kann die Anzahl der Ausgabeknoten 614 auch gleich der Anzahl der Pixel sein, wobei jedem Pixel eine Punktzahl 622 zugewiesen wird. In mindestens einer Ausführungsform kann die Anzahl der Ausgabeknoten 614 geringer sein als die Anzahl der Pixel, z. B. wenn lokale Punktzahlen 622 Superpixeln zugewiesen werden. In mindestens einer Ausführungsform kann das erste Netzwerk 610 ein Faltungsnetzwerk (CNN) sein. In mindestens einer Ausführungsform können die Parameter des ersten Netzwerks 610 die Tiefe der Faltungen, die Filtergröße, die Filterwerte (Gewichte der Faltungen), die Länge der Schritte, die Verzerrungen usw. umfassen. In mindestens einer Ausführungsform kann das erste Netzwerk 610 eine Vielzahl von Neuronenschichten verwenden, wie z. B. eine Eingabeschicht, eine Ausgabeschicht und eine oder mehrere versteckte Schichten. In mindestens einer Ausführungsform kann das erste Netzwerk 610 3, 4, 5, 6, 7 usw. Schichten von Neuronen oder eine andere (niedrigere oder höhere) Anzahl von Schichten aufweisen. In mindestens einer Ausführungsform können einige der Schichten maximale Pooling-Schichten und/oder durchschnittliche Pooling-Schichten sein. In mindestens einer Ausführungsform können die Eingaben verschiedener Schichten den nachfolgenden Schichten unter Verwendung von Aktivierungsfunktionen, wie einer Softplus-Funktion, Sigmoid-Funktionen (z. B. logistische Sigmoid-Funktion), hyperbolischen Tangens-Funktionen oder verschiedenen linearen Gleichrichtereinheiten (ReLU), wie Standard-ReLU, verrauschte ReLU, exponentielle ReLU usw., bereitgestellt werden. In mindestens einer Ausführungsform kann die undichte ReLU eine parametrische ReLU mit einem oder mehreren Parametern sein, die während des Trainings zu bestimmen sind.

[0042] In mindestens einer Ausführungsform können passende Paare von unzulässigen/echten Trainingsbildern verwendet werden, so dass eine Grundwahrheit extrahiert werden kann, die verschiedene Pixel als zu

nicht unzulässigen Regionen (sowohl in unzulässigen als auch in echten Bildern) und/oder unzulässigen Regionen (in unzulässigen Bildern) gehörend identifiziert. In mindestens einer Ausführungsform können sich die übereinstimmenden Paare von Trainingsbildern auf dieselben (oder ähnliche) Spielszenarien, Episoden, Szenen usw. beziehen. In mindestens einer Ausführungsform können echte Bilder in übereinstimmenden Paaren von Trainingsbildern von einer Spielsoftware erzeugt werden, während unzulässige Bilder in übereinstimmenden Paaren von einer Betrugssoftware (oder von einer Betrugssoftware, die in Kombination mit einer Spielsoftware ausgeführt wird) erzeugt werden können. In mindestens einer Ausführungsform können während des Trainings des ersten Netzwerks 610 die lokalen Ausgangswerte 622 mit den lokalen Zielwerten (nicht dargestellt) unter Verwendung einer oder mehrerer Verlustfunktionen verglichen werden. In mindestens einer Ausführungsform kann eine Verlustfunktion eine binäre Kreuzentropie-Verlustfunktion sein. Genauer gesagt, wenn c_j die vom ersten Netzwerk 610 berechnete Wahrscheinlichkeit (z. B. die Wahrscheinlichkeit), dass ein j -ter Bildpunkt zu einem Bereich (des Bildes 602) gehört, der unzulässige Informationen enthält ($1 - c_j$ ist jeweils die berechnete Wahrscheinlichkeit, dass der Bildpunkt zu einem Bereich gehört, der keine unzulässigen Informationen enthält), und t_j die Zielwahrscheinlichkeit des gleichen Ergebnisses (Grundwahrheit) ist, kann die Verlustfunktion eine Summe sein,

$$L = -\sum_j (t_j \ln c_j + (1 - t_j) \ln (1 - c_j)),$$

über eine Vielzahl von Pixeln des Bildes 602. In mindestens einer Ausführungsform kann die Vielzahl der Pixel alle Pixel des Bildes 602 umfassen. In mindestens einer Ausführungsform können auch andere Verlustfunktionen als Kreuzentropieverlustfunktionen verwendet werden. Zum Beispiel kann das erste Netzwerk 602 (wie auch das zweite Netzwerk 630) Verlustfunktionen für den mittleren quadratischen Fehler, gewichtete Verlustfunktionen für den mittleren quadratischen Fehler, Verlustfunktionen für den mittleren absoluten Fehler, Huber-Verlustfunktionen, Hinge-Verlustfunktionen, Multi-Klassen-Kreuzentropie-Verlustfunktionen, Kullback-Liebler-Verlustfunktionen und dergleichen verwenden.

[0043] In mindestens einer Ausführungsform kann die Grundwahrheit anstelle einer Pixelklassifizierung auf lokaler Ebene alle Pixel eines unzulässigen Trainingsbildes mit einem ersten globalen Wert (z. B. Wert 1) und alle Pixel eines nicht unzulässigen Trainingsbildes mit einem zweiten globalen Wert (z. B. Wert 0) identifizieren. In mindestens einer Ausführungsform kann das erste Netzwerk 610 während des Trainings lernen, Pixel mit hoher Wahrscheinlichkeit als zu unzulässigen oder echten Bildern gehörend zu identifizieren. Zum Beispiel können Pixel, die den Pixeln ähnlich sind, die das erste Netzwerk 610 (während des Trainings) sowohl in unzulässigen Bildern als auch in echten Bildern mit gleicher Wahrscheinlichkeit (z. B. 50%) gefunden hat, als Pixel klassifiziert werden, die zu nicht unzulässigen Bildern gehören. In mindestens einer Ausführungsform können Pixel, die den Pixeln ähnlich sind, auf die das erste Netzwerk 610 mit einer bestimmten Wahrscheinlichkeit $w > 50\%$ in unzulässigen Bildern gestoßen ist, als Pixel klassifiziert werden, die zu unzulässigen Regionen gehören. In mindestens einer Ausführungsform muss zur Vermeidung falsch positiver Ergebnisse die Wahrscheinlichkeit, dass Pixel als unzulässige Regionen klassifiziert werden, einen bestimmten vorgegebenen Schwellenwert w_T überschreiten, der 60%, 70%, 75% usw. oder einen anderen Wert betragen kann, der heuristisch als optimales Gleichgewicht zwischen zu vielen falsch positiven Ergebnissen und dem Übersehen zu vieler Pixel unzulässiger Regionen ermittelt wurde. In mindestens einer Ausführungsform kann der beschriebene Ansatz ohne Extraktion der lokalen Klassifizierung von Pixeln (unter Verwendung übereinstimmender unechter/echter Paare von Trainingsbildern) verwendet werden, um das Training/Umlernen des ersten neuronalen Netzes 610 zu beschleunigen. In mindestens einer Ausführungsform kann der beschriebene Ansatz auch dann verwendet werden, wenn das Training mit nicht übereinstimmenden echten und unzulässigen Bildern durchgeführt wird. Insbesondere können die echten und die unzulässigen Trainingsbilder aus verschiedenen Spielszenen, Episoden, Spielversionen usw. oder sogar aus verschiedenen Spielen stammen.

[0044] In mindestens einer Ausführungsform kann die Trainingsmaschine 342 während des Trainings des ersten Netzwerks 610 die Backpropagation 625 verwenden, um verschiedene Parameter des ersten Netzwerks (Verzerrungen, Gewichte, Abstände, Filterverzerrungen, Wahrnehmungstiefen usw.) anzupassen, bis die beobachteten Unterschiede (Verlustfunktionen) zwischen den berechneten Ausgaben und den Zielausgaben (Grundwahrheit) minimiert sind. Das erste Netzwerk 610 kann zum Beispiel einen oder mehrere Matrixfilter verwenden, deren Parameter (Matrixelemente, Tiefen, Abstände usw.) während des Trainings angepasst werden können.

[0045] In mindestens einer Ausführungsform können die lokalen Bewertungen 622 der Merkmalskarte 620 in das zweite Netzwerk 630 eingegeben werden, das die lokale Merkmalskarte 620 verwenden kann, um eine

endgültige Entscheidung über das Bild 602 zu treffen. In mindestens einer Ausführungsform kann das zweite Netzwerk 620 so trainiert werden, dass es das Bild 602 auf einer globalen Ebene klassifiziert. Die globale Ebene kann sich auf die Feststellung auf Bildebene beziehen, ob das Bild 602 unzulässige Informationen enthält. Dementsprechend kann in mindestens einer Ausführungsform eine Ausgabe des zweiten Netzwerks 630 eine binäre Ausgabe sein, z. B. 1 oder JA, wenn Bild 602 unzulässige Informationen enthält, und 0 oder NEIN, wenn Bild 602 keine unzulässigen Informationen enthält. Zumindest in einigen Ausführungsformen können auch andere Ausgänge verwendet werden. Beispielsweise kann die globale Punktzahl 640 ein gleitender Skalenwert sein, z. B. können Ausgabewerte von 0 bis 0,5 anzeigen, dass das Bild 602 keine unzulässigen Informationen enthält, während Werte von 0,5 bis 1,0 das Vorhandensein von unzulässigen Informationen anzeigen können.

[0046] In mindestens einer Ausführungsform kann die Anzahl der Eingangsknoten 632 des zweiten Netzes 630 gleich der Anzahl der Pixel (oder der Gesamtzahl der Parameter, die alle Pixel beschreiben) sein. In mindestens einer Ausführungsform kann das zweite Netzwerk 630 eine Vielzahl von Neuronenschichten verwenden, wie z. B. eine Eingabeschicht, eine Abschlusschicht und eine oder mehrere versteckte Schichten (nicht gezeigt). In mindestens einer Ausführungsform können die Ausgangsknoten des zweiten Netzes 630 den Ausgangsknoten 636 zur Ausgabe der globalen Punktzahl 640 umfassen. In mindestens einer Ausführungsform kann das zweite Netzwerk 630 eine oder mehrere vollständig verbundene Schichten von Neuronen umfassen. Beispielsweise kann eine vollständig verbundene Eingabeschicht (Knoten 632) lokale Ergebnisse 622 in einen einzigen Vektor homogenisieren, der dann von einer nächsten Stufe (z. B. einer versteckten, vollständig verbundenen Schicht oder einer endgültigen, vollständig verbundenen Schicht) verarbeitet werden kann, wo die Anwendung von Gewichten und Verzerrungen auf eine Ausgabe der Eingabeschicht zur Vorhersage des globalen Ergebnisses 640 verwendet werden kann. In mindestens einer Ausführungsform können zusätzliche versteckte Schichten verwendet werden, um die Wahrnehmung der Merkmalskarte 620 durch das zweite Netz 630 zu verbessern. In mindestens einer Ausführungsform können das erste neuronale Netz 610 und/oder das zweite neuronale Netz 630 überspringende Verbindungen aufweisen, die nicht aufeinanderfolgende Schichten von Neuronen verbinden. Der Kürze halber wird eine einzelne überspringende Verbindung eines Knotens in einer ersten verborgenen Schicht des ersten neuronalen Netzes 610 und eines Knotens in einer zweiten Schicht des zweiten neuronalen Netzes 630 gezeigt, aber es versteht sich, dass eine beliebige Anzahl von überspringenden Verbindungen im System 600 vorhanden sein kann, einschließlich einer beliebigen Anzahl von überspringenden Verbindungen zwischen nicht aufeinanderfolgenden Schichten des ersten Netzes 610, des zweiten Netzes 630 oder überspringenden Verbindungen, die Schichten des ersten Netzes 610 und des zweiten Netzes 630 verbinden.

[0047] In mindestens einer Ausführungsform können während des Trainings des zweiten Netzes 630 verschiedene Verlustfunktionen, z. B. wie in Bezug auf das erste Netz 610 beschrieben, verwendet werden. In mindestens einer Ausführungsform kann eine Verlustfunktion eine binäre Kreuzentropie-Verlustfunktion sein. In mindestens einer Ausführungsform kann die Trainingsmaschine 342 während des Trainings des zweiten Netzwerks 630 die Backpropagation 645 verwenden, um verschiedene Parameter des zweiten Netzwerks (Verzerrungen, Gewichte usw.) anzupassen, bis die beobachteten Unterschiede (Verlustfunktion) zwischen den berechneten Ausgaben und den Zielausgaben (Grundwahrheit) minimiert sind.

[0048] In mindestens einer Ausführungsform kann das Training des ersten Netzes 610 und des zweiten Netzes 630 getrennt durchgeführt werden. Zum Beispiel kann das erste Netzwerk 610 mit Hilfe von Kartendaten 349 trainiert werden, die bekannte Positionen (z. B. Bounding Boxes) von Regionen enthalten, in denen sich Pixel mit unzulässigen Informationen befinden. In mindestens einer Ausführungsform können bekannte Orte Regionen mit komplexen Formen sein, wie z. B. Polygone (einschließlich unregelmäßiger Polygone, konkaver Polygone usw.), nicht-polygonale Formen, die mit einer einzigen geschlossenen Linie gezeichnet sind, Formen, die mit mehreren geschlossenen Linien gezeichnet sind, und/oder beliebige Formen, die mit einer oder mehreren offenen Linien gezeichnet sind. In mindestens einer Ausführungsform können bekannte Stellen als unzulässige Informationen enthaltend identifiziert (oder anderweitig zugeordnet, unter Verwendung eines beliebigen Zuordnungsschemas oder einer Ressource, wie z. B. einer Zuordnungstabelle) werden, z. B. unter Verwendung einer geeigneten Identifizierung auf Pixelebene (wie z. B. Wert 1 für ein Pixel, das zur Anzeige einer unzulässigen Information verwendet wird, und Wert 0 für ein Pixel, das keine unzulässige Information anzeigt). In mindestens einer Ausführungsform kann sich der Begriff „Regionen“ auf ganze Bilder beziehen, wobei der Wert 1 ein Bild kennzeichnet, das zumindest einige unzulässige Informationen enthält, und der Wert 0 ein echtes Bild (ohne unzulässige Informationen) kennzeichnet. In mindestens einer Ausführungsform kann das zweite Netzwerk 630 auf der Grundlage von binären Zuordnungsdaten 349 trainiert werden, die die Identifizierung der Trainingsbilder als echt oder unzulässig umfassen. In mindestens einer Ausführungsform kann das Training des ersten Netzwerks 610 oder des zweiten Netzwerks 630 (oder beider

Netzwerke) unter Verwendung von Trainingsbildern mit unterschiedlichen Mengen an unzulässigen Informationen durchgeführt werden. So können beispielsweise unzulässige Trainingsbilder in Stapeln gruppiert werden, die durch die Menge der in den Trainingsbildern enthaltenen unzulässigen Informationen definiert sind. In mindestens einer Ausführungsform können die Chargen eine abnehmende Menge an unzulässigem Inhalt aufweisen. In mindestens einer Ausführungsform können nach dem Training des ersten Netzwerks 610 und/oder des zweiten Netzwerks 630 mit unzulässigen Trainingsbildern, in denen 100% (oder mehr) unzulässige Informationen enthalten sind, die nächsten Trainingsbilder (z. B. Bilder, in denen 80% unzulässige Informationen enthalten sind) verwendet werden, und so weiter. In mindestens einer Ausführungsform kann der Übergang von einer Charge zur nächsten erfolgen, nachdem eine bestimmte Ziel-Erfolgsrate der Bestimmung erreicht wurde. In mindestens einer Ausführungsform kann dieser Prozess fortgesetzt werden, bis eine letzte Charge (mit X% zurückbehaltener unzulässiger Informationen) angewendet wird. In mindestens einer Ausführungsform kann ein Mindestprozentsatz an unzulässigen Informationen, X, auf der Grundlage einer Reihe von Gesichtspunkten festgelegt werden. In mindestens einer Ausführungsform kann X auf der Grundlage dessen festgelegt werden, was vorsichtige Betrüger verwenden könnten, um nicht entdeckt zu werden. In einer anderen Ausführungsform kann X auf der Grundlage eines ausreichenden Trainings festgelegt werden. Wenn beispielsweise festgestellt wird, dass die zuverlässige Erkennung unzulässiger Bilder problematisch wird, wenn der Prozentsatz unter X fällt, kann dieser Wert X als Mindestprozentsatz für die letzte Trainingsserie verwendet werden. In mindestens einer Ausführungsform können einige unzulässige Informationen teilweise unkenntlich gemacht werden, anstatt sie vollständig zu entfernen. So können beispielsweise Regionen, die Pixel mit unzulässigen Informationen enthalten, mit einem verminderten Kontrast versehen werden, damit sich diese Pixel weniger deutlich vom Hintergrund abheben. In mindestens einer Ausführungsform kann die Farbdarstellung von Pixeln mit unzulässigen Informationen geändert werden, z. B. können Pixel mit weißen, gelben, orangefarbenen, roten usw. Farben so verändert werden, dass sie mehr Farben im blauen, grünen usw. Teil des Spektrums enthalten.

[0049] In mindestens einer Ausführungsform kann das System 600 auch einen Zuverlässigkeitsgrad 645 vorhersagen, der die Wahrscheinlichkeit angibt, dass das ausgegebene globale Ergebnis 640 das Vorhandensein oder Nichtvorhandensein von unzulässigen Informationen im Bild 602 korrekt vorhersagt. In mindestens einer Ausführungsform kann das Zuverlässigkeitsniveau 645 die Wahrscheinlichkeit charakterisieren, dass die ausgegebene globale Punktzahl 640 für ein Ensemble von Bildern, die dem Bild 602 ähnlich sind, korrekt ist. In mindestens einer Ausführungsform kann die Zuverlässigkeitsstufe 645 von einem zusätzlichen Ausgangsknoten 638 (dargestellt durch einen schattierten Kreis) ausgegeben werden. In mindestens einer Ausführungsform können während des Trainings einige der Trainingsbilder verwendet werden, um Ensembles für die Bestimmung von Zuverlässigkeitsstufen zu erzeugen. In mindestens einer Ausführungsform können verschiedene Methoden zur Ermittlung von Zuverlässigkeitsstufen verwendet werden.

[0050] In mindestens einer Ausführungsform kann einem Trainingsbild I_0 ein Zufallsrauschen hinzugefügt werden, um ein Trainingsbild I_1 zu erzeugen. In ähnlicher Weise kann ein weiteres Trainingsbild I_2 aus einem Bild I_0 mit einer anderen Art von hinzugefügtem Rauschen erzeugt werden, und so weiter, bis eine Anordnung $\{I_k\}$ von Trainingsbildern erhalten wird. In mindestens einer Ausführungsform kann jedes Bild der Anordnung $\{I_k\}$ von einem ersten Netzwerk 610 und einem zweiten Netzwerk 630 analysiert werden, und für jedes Bild kann eine globale Bewertung 640 ausgegeben werden. Anschließend kann in mindestens einer Ausführungsform die Statistik der Gesamtpunktzahl $\{GS_k\}$ für die Anordnung $\{I_k\}$ bestimmt und mit der Zuverlässigkeitsstufe der Ausgabe verglichen werden. In mindestens einer Ausführungsform kann die zu vergleichende Zuverlässigkeitsstufe die Zuverlässigkeitsstufe I_0 für das Ausgangsbild sein. Alternativ dazu kann in mindestens einer Ausführungsform die zu vergleichende Zuverlässigkeitsstufe eine durchschnittliche Zuverlässigkeitsstufe für die gesamte Anordnung $\{I_k\}$ sein. In mindestens einer Ausführungsform kann dann eine zusätzliche Verlustfunktion verwendet werden, um die Differenz zwischen der Zuverlässigkeitsstufe und der Statistik der ausgegebenen globalen Punktzahlen $\{GS_k\}$ zu bewerten. Wenn z. B. eine Anordnung von 10 globalen Punktzahlen $\{GS_k\}$ für eine Anordnung von 10 Bildern $\{I_k\}$ in 7 von 10 Fällen die Bilder korrekt als unzulässig identifiziert, kann die angestrebte Zuverlässigkeitsstufe 0,7 betragen. In mindestens einer Ausführungsform kann die Differenz $0,7 - 0,4 = 0,3$ durch das zweite Netz 630 und/oder das erste Netz 610 zurückverfolgt werden, bis die Differenz minimiert ist, wenn die vom Knoten 638 ausgegebene Zuverlässigkeitsstufe (für ein Bild I_0 oder gemittelt für den gesamten Satz $\{I_k\}$) 0,4 beträgt. In mindestens einer Ausführungsform kann jede geeignete Verlustfunktion zur Bewertung der Differenz verwendet werden (z. B. Kreuzentropie, quadratischer Fehler usw.). In mindestens einer Ausführungsform können sich bei der Änderung von Parametern des ersten Netzwerks 610 und/oder des zweiten Netzwerks 630 die ausgegebenen globalen Punktzahlen $\{GS_k\}$ zusammen mit den ausgegebenen Zuverlässigkeitsstufen ändern. In mindestens einer Ausführungsform kann die Änderung der Parameter der Netze so lange fortgesetzt werden, bis beide Differenzen - sowohl die Gesamtausgangsbewertung als auch die Zuverlässigkeitsstufen - minimiert sind. Anschließend

können in mindestens einer Ausführungsform auch ein oder mehrere zusätzliche Ausgangsbilder ausgewählt, entsprechende Anordnungen von Trainingsbildern erstellt und das Training der neuronalen Netze wiederholt werden. In mindestens einer Ausführungsform erwirbt das System 600 als Ergebnis des Anordnungstrainings die Fähigkeit, Zuverlässigkeitsstufen 645 auszugeben, die die Wahrscheinlichkeit vorhersagen, dass die globalen Punktzahlen 640 die Typen (unzulässig vs. echt) der Bilder 602 genau bestimmen.

[0051] In mindestens einer Ausführungsform können die Zuverlässigkeitsstufen auf der Grundlage der Einführung von Zufälligkeiten in die Architektur des neuronalen Netzes bestimmt werden. Genauer gesagt kann ein Trainingsbild von einem oder mehreren neuronalen Netzen mehrfach verarbeitet werden, wobei eine Anordnung von neuronalen Netzen verwendet wird, bei der zumindest einige (oder alle) Knoten (mit einer gewissen Wahrscheinlichkeit) entfernt (herausgenommen) werden. Das Entfernen eines Knotens bedeutet, dass alle eingehenden und ausgehenden Netzwerkverbindungen für den Knoten eliminiert werden. In mindestens einer Ausführungsform können die Entfernungswahrscheinlichkeiten von der Schicht abhängen, in der sich der entfernte Knoten befindet, und können für versteckte Schichten größer sein als für Eingabe-/Ausgabeschichten. In mindestens einer Ausführungsform kann eine Zuverlässigkeitsstufe bestimmt werden, die darauf basiert, wie erfolgreich verschiedene neuronale Netze des erzeugten Anordnungen Bilder (oder Pixel von Bildern) als unzulässig oder echt identifizieren. In mindestens einer Ausführungsform können zur Bestimmung der Zuverlässigkeitsstufen verschiedene Verfahren zur Klassifizierung der Unsicherheit verwendet werden, einschließlich des Verfahrens der Variationsverhältnisse, des Verfahrens der gegenseitigen Information oder anderer ähnlicher Verfahren (z. B. des Verfahrens der prädiktiven Entropie). Beispielsweise kann das Verfahren des Variationsverhältnisses eine Anzahl erfolgreicher Identifizierungen M von Zielergebnissen (die ein Modus einer Gesamtverteilung von Ergebnissen sein kann) und zugrundeliegende Zuverlässigkeitsstufe unter Verwendung des Variationsverhältnisses $R = 1 - M/N$ zur Gesamtzahl aller Ergebnisse N ermitteln. In mindestens einer Ausführungsform kann die Zuverlässigkeitsstufe durch das ermittelte Variationsverhältnis angegeben werden. In mindestens einer Ausführungsform kann die Zuverlässigkeitsstufe eine Funktion (z. B. eine nichtlineare Funktion) des ermittelten Variationsverhältnisses sein. Bei dem Verfahren der gegenseitigen Information kann der Grad der Korrelation zwischen der Verteilung der Ergebnisse in der Anordnung mit einer Verteilung der richtigen Ergebnisse maximiert werden. In mindestens einer Ausführungsform kann der Grad der Korrelation durch die Kullback-Leibler-Divergenz zweier Verteilungen dargestellt werden.

[0052] In mindestens einer Ausführungsform kann die Klassifizierung eines Pixels eines Bildes (oder einer Region eines Bildes oder eines ganzen Bildes) über die Klassen „unzulässig“, „nicht unzulässig“ und „unsicher“ erfolgen. In mindestens einer Ausführungsform kann die Klassifizierung eines Pixels eines Bildes (oder einer Region eines Bildes oder eines ganzen Bildes) über die Klassen „nicht unzulässig“ mit einer Wahrscheinlichkeit p_0 , „unzulässig“ mit einer Wahrscheinlichkeit p_1 und „unsicher“ mit einer Wahrscheinlichkeit u verfolgen, so dass $p_0 + p_1 + u = 1$. die Unsicherheit $p_0 + p_1 + u = 1$. unter Verwendung multinomialer Wahrscheinlichkeiten, die auf Dirichlet-Verteilungen beruhen, bestimmt werden kann. In mindestens einer Ausführungsform kann dies durch die Definition einer Verlustfunktion und die Berechnung des Bayes-Risikos in Bezug auf den Klassenprädiktor (z. B. „unzulässig“, „nicht unzulässig“ und „unsicher“) erfolgen. In mindestens einer Ausführungsform kann die Verlustfunktion eine mittlere quadratische Verlustfunktion sein. In mindestens einer Ausführungsform kann die Verlustfunktion eine Kullback-Leibler-Divergenz sein. In mindestens einer Ausführungsform kann die Verlustfunktion eine Maximum-Likelihood-Funktion vom Typ II sein.

[0053] In mindestens einer Ausführungsform kann während der Inferenzphase der Vorgänge des/der neuronalen Netzes/Netze die Wahrscheinlichkeit der Ungewissheit u in eine Zuverlässigkeitsstufe umgewandelt werden, z.B. kann die Zuverlässigkeitsstufe eine Funktion der Ungewissheit $CL = f(u)$, sein, die eine abnehmende Funktion der Ungewissheit sein kann, so dass höhere Ungewissheiten niedrigeren Zuverlässigkeitsstufen CL entsprechen. In mindestens einer Ausführungsform kann jede geeignete Funktion $f(u)$ zur Ausgabe der Zuverlässigkeitsstufe verwendet werden (z. B. innerhalb eines Intervalls von 0 bis 1, 0% bis 100% oder eines anderen Intervalls).

[0054] In mindestens einer Ausführungsform können während des Trainings die Parameter des/der neuronalen Netzes/Netze geändert werden, um die Zuverlässigkeitsstufe zu maximieren CL . In mindestens einer Ausführungsform können die Parameter des/der neuronalen Netzes/Netze geändert werden, um das gewünschte Gleichgewicht zwischen der Erhöhung der Zuverlässigkeitsstufe CL und der Erhöhung der Wahrscheinlichkeit der Erkennung von Betrug zu erreichen. So kann beispielsweise eine höhere Zuverlässigkeitsstufe bedeuten, dass ein Teil der Betrüger unentdeckt bleibt. Dementsprechend kann in mindestens einer Ausführungsform eine gewisse Verringerung der Zuverlässigkeitsstufe absichtlich in Kauf genommen werden, um eine umfassendere Erkennung von Betrug zu erreichen. In mindestens einer Ausführungsform

kann das Training des neuronalen Netzes/der neuronalen Netze die Optimierung der Zuverlässigkeitsstufe CL gegenüber der Wahrscheinlichkeit P einer korrekten Klassifizierung von Bildern über einen Satz (oder einen Teilsatz) von Trainingsbildern umfassen. In mindestens einer Ausführungsform kann die Optimierung darin bestehen, einen Maximalwert einer Funktion zu finden, $F(CL, P)$ z.B. $F(CL, P) = a(CL)^2 + b(P)^2$, mit Koeffizienten, a die b in Abhängigkeit von den Erfassungszielen gewählt werden. Genauer gesagt können höhere Werte von a die Erkennung von weniger Betrugsfällen mit höherer Sicherheit begünstigen, während höhere Werte von b die Erkennung von mehr potenziellen Betrugsfällen begünstigen, auch wenn einige dieser Fälle falsch positiv (oder falsch negativ) sein können. In mindestens einer Ausführungsform $F(CL, P)$ können auch andere Funktionen verwendet werden. In mindestens einer Ausführungsform kann die Optimierung so lange durchgeführt werden, bis eine höchstmögliche Rate korrekter Erkennungen unter der Bedingung erreicht ist, dass die Zuverlässigkeitsstufe einen Mindestschwellenwert erreicht. In mindestens einer Ausführungsform kann die Optimierung so lange durchgeführt werden, bis eine höchstmögliche Zuverlässigkeitsstufe unter der Bedingung erreicht ist, dass die Rate der korrekten Erkennung einen Mindestschwellenwert erreicht.

[0055] In mindestens einer Ausführungsform können Verfahren der Intervallgrenzenausbreitung (IBP) während des Trainings verwendet werden, um Zuverlässigkeitsstufen zu schätzen CL. Zum Beispiel können Trainingsbilder durch Hinzufügen von Rauschen (wie oben beschrieben) verändert werden, und die Ausgaben des/der neuronalen Netzes/Netze - Klassifizierungen von Bildern oder Teilen von Bildern als unzulässig oder echt - können generiert werden. In mindestens einer Ausführungsform kann dann für eine vorgegebene Unsicherheit $\pm \epsilon$ der Eingaben (z. B. aufgrund von Rauschen) eine Obergrenze für die Abweichung $\pm \Delta$ der Ausgaben von Klassifizierungen geschätzt werden. In mindestens einer Ausführungsform kann die Obergrenze Δ mathematisch geschätzt werden, indem ein oder mehrere IBP-Modelle verwendet werden. In mindestens einer Ausführungsform kann die Obergrenze Δ empirisch bestimmt werden, und zwar als maximale Abweichung, die für einen Satz von mit Rauschen modifizierten Trainingsbildern beobachtet wurde. In mindestens einer Ausführungsform kann die geschätzte oder ermittelte Obergrenze Δ zur Bestimmung oder Anpassung der Zuverlässigkeitsstufen CL verwendet werden. In mindestens einer Ausführungsform können die Zuverlässigkeitsstufen für alle Bilder eine Obergrenze bilden. So kann beispielsweise eine Zuverlässigkeitsstufe von CL = 80% auf 70% gesenkt werden, wenn die geschätzte obere Grenze $\Delta = 10\%$ ist. In mindestens einer Ausführungsform können die Parameter des/der neuronalen Netze(s), die Zuverlässigkeitsstufen ausgeben (die auch Anpassungen der Obergrenze umfassen können), während des Trainings bestimmt werden (wenn die Grundwahrheit bekannt ist) und auch während der Inferenzphase verwendet werden (wenn die Grundwahrheit nicht bekannt ist).

[0056] In mindestens einer Ausführungsform können das erste neuronale Netz 610 und das zweite neuronale Netz 620 so trainiert werden, dass sie Zuverlässigkeitsstufen durch Hinzufügen von Rauschen zu den Parametern der neuronalen Netze (und nicht zu den Trainingsbildern) ausgeben, z. B. durch Hinzufügen von Rauschen zu den Gewichten, Verzerrungs- und/oder Aktivierungsparametern in verschiedenen Schichten (z. B. versteckte Schichten) des ersten Netzes 610 und/oder des zweiten Netzes 620. In mindestens einer Ausführungsform kann das Hinzufügen von Rauschen zu den Parametern zusätzlich zum Hinzufügen von Rauschen zu den Trainingsbildern durchgeführt werden.

[0057] In mindestens einer Ausführungsform können während der Inferenzphase die für ein Bild oder eine Reihe von Bildern auf dem Computer des Benutzers (Spielers) ermittelten Zuverlässigkeitsstufen für verschiedene Zwecke verwendet werden, einschließlich, aber nicht beschränkt auf das Melden von Fällen von erkanntem Betrug, das Unterlassen des Meldens einiger Fälle von vermutetem Betrug, das Signalisieren, wann das/die neuronale(n) Netz(e) neu trainiert werden sollte(n), und so weiter. Wenn beispielsweise die Zuverlässigkeitsstufe hoch ist, d. h. über einem bestimmten vorgegebenen Schwellenwert CL_1 liegt, können erkannte Betrugsfälle an den Veröffentlichungsserver 302, den Spielservers 304 oder andere Ressourcen gemeldet werden. In mindestens einer Ausführungsform können erkannte Betrugsfälle nicht gemeldet werden, wenn die Zuverlässigkeitsstufen geringer als CL_1 sind, um falsch positive Feststellungen zu vermeiden. In mindestens einer Ausführungsform kann eine Meldung erfolgen, sobald dem Benutzer eine bestimmte, vorher festgelegte Anzahl von Bildern angezeigt wird (oder eine bestimmte, vorher festgelegte Zeit, in der dem Benutzer Bilder angezeigt werden) und diese mit mindestens einer Zuverlässigkeitsstufe als unzulässig eingestuft werden CL_1 . In mindestens einer Ausführungsform kann ein Bericht an den Veröffentlichungsserver 302 oder den Spielservers 304 gesendet werden, wenn eine Reihe von (auslösenden) Bildern gefunden wird, bei denen die Bestimmung mit einer Zuverlässigkeitsstufe unterhalb eines zweiten Schwellenwerts CL_2 (der mit dem Schwellenwert CL_1 identisch oder von diesem verschieden sein kann) durchgeführt wird. In mindestens einer Ausführungsform kann der Empfang eines solchen Berichts dem Veröffentlichungsserver 302 oder dem Spielservers 304 signalisieren, dass eine neue Betrugssoftware (oder eine neue Version einer

bestehenden Software) verfügbar geworden sein könnte und dass ein erneutes Training des neuronalen Netzes/der neuronalen Netze mit Bildern, die mit der neuen Betrugssoftware erzeugt wurden, durchgeführt werden sollte. In mindestens einer Ausführungsform sollte eine Mindestanzahl verschiedener Bilder (oder eine Mindestanzahl verschiedener Spielszenen, die von verschiedenen Nutzern empfangen wurden) vorliegen, bevor ein Bericht gesendet wird, um Falschmeldungen zu vermeiden, die durch verschiedene Artefakte auf dem Computer eines Nutzers (wie Softwarefehler, Stromstöße usw.) verursacht werden können.

[0058] In mindestens einer Ausführungsform kann sich während des Trainings des Systems 600 die Backpropagation 646 (mit gestrichelten Linien dargestellt) sowohl über das zweite Netzwerk 630 als auch über das erste Netzwerk 610 erstrecken. Dementsprechend können in mindestens einer Ausführungsform die Zielausgaben 348 für das Training des Systems 650 einen einzigen Wert enthalten, der beschreibt, ob es sich bei einer Trainingseingabe um ein echtes Bild oder um ein unzulässiges Bild handelt. In solchen Ausführungsformen enthalten die Zielausgaben keine Grundwahrheit für die Merkmalskarte 620, die den Charakter einzelner Pixel beschreibt, da es sich bei den lokalen Punktwerten 622 um interne Werte handeln kann, die nicht explizit zugänglich sind.

[0059] In mindestens einer Ausführungsform wird das System 600 als eine Kombination aus dem ersten Netzwerk 610 und dem zweiten Netzwerk 210 dargestellt, um dem Leser verschiedene Funktionalitäten zu veranschaulichen, über die das System 600 verfügen kann. In mindestens einer Ausführungsform stellen das erste Netzwerk 610 und das zweite Netzwerk 620 ein einziges Netzwerk dar. In mindestens einer Ausführungsform kann die Merkmalskarte 620 aus verschiedenen Zwischenausgangsknoten (z. B. Knoten 614) extrahiert und für das Training verwendet werden, wenn die lokale Backpropagation 625 angewendet wird. In mindestens einer Ausführungsform kann die Merkmalskarte 620 nicht explizit ausgegeben werden (und kann ein Zwischenergebnis darstellen, auf das nicht direkt zugegriffen wird), z. B. in Fällen, in denen eine einzige globale Backpropagation 646 verwendet wird.

[0060] In mindestens einer Ausführungsform kann das System 600 gegen gegnerische Angriffe trainiert werden. Ein gegnerischer Angriff (z. B. von Herstellern oder Vertreibern von Betrugssoftware) kann auf kleinen Änderungen von Betrugsbildern beruhen, die für einen Menschen nicht wahrnehmbar sind, aber dennoch ausreichen, um ein neuronales Netz so zu verwirren, dass es ein unzulässiges Bild fälschlicherweise als nicht unzulässiges Bild einstuft. In mindestens einer Ausführungsform bietet das System 600 intrinsische Sicherheitsvorkehrungen gegen gegnerische Angriffe. Da der Grafikprozessor 320 die endgültigen Bilder auf dem Display 329 mit einer Geschwindigkeit von 30-60 Bildern pro Sekunde oder mehr wiedergibt, hat ein potenzieller Angriff (z. B. durch eine Betrugssoftware auf dem Benutzerrechner 310) nur eine sehr begrenzte Zeit, was einen erfolgreichen Angriff verhindert. Daher ist es in mindestens einer Ausführungsform wahrscheinlich, dass ein Angreifer, der versucht, die Reaktion eines oder mehrerer neuronaler Netze (z. B. NN-Modelle 325) mit Hilfe gegnerischer Testbilder zu testen, entdeckt wird, bevor ein Bild entwickelt werden kann, das in der Lage ist, ein oder mehrere neuronale Netze erfolgreich zu verwirren. Dementsprechend kann die GPU 320 in mindestens einer Ausführungsform ein Muster erkennen, bei dem die anfängliche Erkennung unzulässiger Bilder nach einer gewissen Zeit aufhört. In mindestens einer Ausführungsform kann die GPU 320 ein solches Muster als Hinweis darauf melden, dass der Benutzer wahrscheinlich eine Betrugssoftware verwendet, die in der Lage ist, erfolgreiche gegnerische Angriffe zu entwickeln.

[0061] In mindestens einer Ausführungsform können Parameter (Gewichte, Verzerrungen, Parameter von Aktivierungsfunktionen usw.) von neuronalen Netzen (z. B. NN-Modellen 325) geheim gehalten werden, z. B. kryptografisch mit Schlüsseln, Hashes und anderen Schutzmechanismen geschützt werden, die potenzielle Angreifer daran hindern, erfolgreiche Offline-Angriffe (z. B. universelle gegnerische Bildstörungen) an anderer Stelle zu entwickeln, die anschließend auf den Benutzerrechner 310 übertragen werden könnten. Außerdem hat ein Angreifer möglicherweise keinen Zugriff auf die Architektur des/der neuronalen Netzes/Netze. In mindestens einer Ausführungsform, da die Zeit, die für gegnerische Angriffe zur Verfügung steht, kurz ist, ist ein gegnerischer Angriff on-the-fly wahrscheinlich von einem relativ einfachen Typ, wie z.B. Madry-Angriffe, Angriffe nach der Fast-Gradient-Sign-Methode (FGSM), und so weiter. In mindestens einer Ausführungsform können zum Schutz vor solchen Angriffen während des Trainings des Systems 600 gegnerische Bilder vorbereitet werden - unter Verwendung von Algorithmen, die wahrscheinlich bei solchen Angriffen on-the-fly eingesetzt werden -, um das System 600 so zu trainieren, dass es gegnerische Störungen ignoriert, die ein Angreifer wahrscheinlich verwenden würde. In mindestens einer Ausführungsform können IBP-Verfahren verwendet werden, um das System 600 gegen gegnerische Angriffe zu trainieren. Während des Trainings kann der IBP beispielsweise für eine ausgewählte Unsicherheit $\pm \epsilon$ der gegnerischen Eingaben (z. B. Störungen) bestimmt werden, die wahrscheinlich für Angriffe verwendet werden. In mindestens einer Ausführungsform können solche vorbereiteten gegnerischen Eingaben durch ein oder mehrere neuronale Netze weiterge-

geben werden, und es kann eine Obergrenze Δ für die ausgewählte Unsicherheit ϵ bestimmt werden. In mindestens einer Ausführungsform kann das Training dann auf der Grundlage einer Zuverlässigkeitsstufe durchgeführt werden, die (auf der Grundlage des Worst-Case-Szenarios) unter Verwendung einer bestimmten Obergrenze angepasst wurde.

[0062] Fig. 7 ist ein Flussdiagramm eines Beispielverfahrens 700 zur Verwendung eines oder mehrerer neuronaler Netze zur Erkennung unerlaubter Informationen in Bildern, die von einer Grafikverarbeitungseinheit auf einem Display eines Benutzergeräts gemäß mindestens einer Ausführungsform dargestellt werden. In mindestens einer Ausführungsform kann das Verfahren 700 von einem oder mehreren Schaltkreisen durchgeführt werden, die ein oder mehrere neuronale Netze verwenden. In mindestens einer Ausführungsform kann das Verfahren 700 von einer Verarbeitungslogik der Benutzermaschine 310 durchgeführt werden. Genauer gesagt kann das Verfahren 700 von der GPU 320 durchgeführt werden, die einen oder mehrere Schaltkreise (z. B. GPU-Kerne 322) und eine oder mehrere Speichervorrichtungen, wie die GPU 324, umfasst. In mindestens einer Ausführungsform kann das Verfahren 700 von einem einzigen Verarbeitungsprozess durchgeführt werden. Alternativ kann in mindestens einer Ausführungsform das Verfahren 700 von zwei oder mehreren Verarbeitungs-Threads durchgeführt werden, wobei jeder Thread eine oder mehrere individuelle Funktionen, Routinen, Unterprogramme oder Vorgänge des Verfahrens 700 ausführt. In mindestens einer Ausführungsform können Verarbeitungs-Threads, die das Verfahren 700 implementieren, synchronisiert werden (z. B. unter Verwendung von Semaphoren, kritischen Abschnitten und/oder anderen Thread-Synchronisierungsmechanismen). Alternativ können in mindestens einer Ausführungsform die Verarbeitungs-Threads, die das Verfahren 700 implementieren, asynchron zueinander ausgeführt werden. In mindestens einer Ausführungsform können verschiedene Vorgänge des Verfahrens 700 in einer anderen Reihenfolge durchgeführt werden als in Fig. 7. In mindestens einer Ausführungsform können einige Vorgänge des Verfahrens 700 gleichzeitig mit anderen Vorgängen durchgeführt werden. In mindestens einer Ausführungsform können ein oder mehrere der in Fig. 7 gezeigten Vorgänge nicht durchgeführt werden. Obwohl verschiedene Vorgänge des Verfahrens 700 in Bezug auf Spielanwendungen und Betrugsoftware beschrieben werden, versteht es sich von selbst, dass in mindestens einer Ausführungsform ähnliche Vorgänge verwendet werden können, um unzulässige Informationen in Bildern zu erkennen, die mit beliebigen anderen Arten von Anwendungen verbunden sind.

[0063] In mindestens einer Ausführungsform kann die Verarbeitungslogik, die das Verfahren 700 durchführt, ein oder mehrere neuronale Netze (z. B. NN-Modelle 325) ausführen, die in einem Inferenzmodus arbeiten, um zu bestimmen, ob Bilder, die unzulässige (z. B. betrügerische) Informationen enthalten, wiedergegeben und einem Benutzer (z. B. einem Spieler) angezeigt werden. In mindestens einer Ausführungsform werden die angezeigten Bilder von derselben Verarbeitungsvorrichtung wiedergegeben, die das Verfahren 700 durchführt. In mindestens einer Ausführungsform können die angezeigten Bilder von einer anderen Verarbeitungsvorrichtung wiedergegeben werden. In mindestens einer Ausführungsform kann die Verarbeitungslogik, die das Verfahren 700 durchführt, in Block 710 von einer Verarbeitungsvorrichtung (z. B. CPU 316) eine Darstellung einer mit einem Computerspiel verbundenen Grafik empfangen. Zum Beispiel kann die CPU 316 Anweisungen an die GPU ausgeben, um Grafiken entsprechend einer Spielsituation auf dem Bildschirm des Benutzers wiederzugeben. In mindestens einer Ausführungsform können die Grafiken ein oder mehrere Bilder umfassen, die auf dem Bildschirm des Benutzers gleichzeitig oder nacheinander (z. B. zu aufeinanderfolgenden Zeitpunkten) angezeigt werden sollen. In mindestens einer Ausführungsform kann die Darstellung von Grafiken in einem beliebigen elektronischen (z. B. digitalen) Format erfolgen, auf das die Verarbeitungslogik, die das Verfahren 700 durchführt (z. B. GPU 324), zugreifen kann.

[0064] In mindestens einer Ausführungsform kann die Verarbeitungslogik, die das Verfahren 700 durchführt, in Block 720 auf der Grundlage der empfangenen Darstellung ein oder mehrere Bilder erzeugen. In mindestens einer Ausführungsform können die erzeugten Bilder in einem Puffer (z. B. Puffer 326) gespeichert werden, bevor sie an das Display des Benutzers (z. B. Display 329) gesendet werden. In mindestens einer Ausführungsform kann die Verarbeitungsvorrichtung in Block 730 ein oder mehrere trainierte neuronale Netze verwenden, um Betrug durch einen oder mehrere Benutzer des Computerspiels zu erkennen. In mindestens einer Ausführungsform können ein oder mehrere Bilder aus dem Puffer entnommen werden, bevor, während oder nachdem die Bilder zur Anzeige gesendet werden. In mindestens einer Ausführungsform kann es sich bei den neuronalen Netzen um solche handeln, die in Verbindung mit Fig. 6. In mindestens einer Ausführungsform können neuronale Netze eine Bestimmung ausgeben (z. B. die globale Punktzahl 640), die anzeigt, dass ein oder mehrere Bilder Pixel enthalten, die zur Anzeige von Betrugsinformationen verwendet werden.

[0065] In mindestens einer Ausführungsform kann die Verarbeitungslogik unter Verwendung eines oder mehrerer neuronaler Netze eine Zuverlässigkeitsstufe generieren, die die Wahrscheinlichkeit charakterisiert, dass ein oder mehrere Bilder eine Betrugsinformation enthalten. In mindestens einer Ausführungsform kann die Verarbeitungslogik in Block 750 unter Verwendung eines oder mehrerer neuronaler Netze einen Bericht erstellen, der die Erkennung von Betrug anzeigt. In mindestens einer Ausführungsform kann der Bericht einen Hinweis auf die Entdeckung eines Betrugs enthalten. In mindestens einer Ausführungsform kann der Bericht auch die erstellte Zuverlässigkeitsstufe enthalten. In mindestens einer Ausführungsform kann der Bericht in Block 760 an einen Spieleserver übermittelt werden, bei dem es sich um einen Verlagsserver, einen Server eines Spieleentwicklers oder einen anderen Server handeln kann, der mit Herstellern oder Vertriebern von Spielsoftware verbunden ist.

[0066] In mindestens einer Ausführungsform kann der Spieleserver nach Erhalt des Berichts (automatisch oder auf Befehl eines menschlichen Entwicklers/Ingenieurs) ein Neutraining eines oder mehrerer neuronaler Netze einleiten. Wenn beispielsweise eine bestimmte (z. B. vorgegebene) Anzahl von Bildern mit einem Konfidenzniveau unterhalb eines Schwellenwerts (z. B. 20%, 50% oder ein anderer vorgegebener Wert) erkannt wurde, kann der Spieleserver (und/oder der Entwickler/Ingenieur) zu dem Schluss kommen, dass eine neue Betrugssoftware (oder eine neue Version einer bestehenden Betrugssoftware) verfügbar geworden ist. In mindestens einer Ausführungsform kann eine neue Betrugssoftware die Platzierung der Betrugsinformationen, das Format der Anzeige der Betrugsinformationen (z. B. Schriftart, Farbe, Transparenz der Einblendungen usw.), den Umfang und die Art der Betrugsinformationen usw. geändert haben. Infolgedessen können in mindestens einer Ausführungsform ein oder mehrere neuronale Netze Veränderungen im Erscheinungsbild von Betrugsbildern erkennen und über eine verringerte Zuverlässigkeitsstufe signalisieren, dass ein Neutraining eines oder mehrerer neuronaler Netze durchgeführt werden sollte. In mindestens einer Ausführungsform kann der Spieleserver auf Spielprotokolle (z. B. Spielprotokolle 306) für den jeweiligen Spieler zugreifen, die auf dem Spieleserver gespeichert sein können, um dabei zu helfen, einen Kontext zu identifizieren, in dem die neue Betrugssoftware arbeitet, eine Art von Vorteil, den die neue Betrugssoftware bietet, und so weiter. In mindestens einer Ausführungsform können der Spieleserver 304, der Veröffentlichungsserver 302 und/oder der Trainingsserver 340 die Spielprotokolle 306 verwenden, um den Spielkontext erneut abzuspielen, Bilder mit Hilfe von Spielsoftware neu wiederzugeben und unzulässige Informationen hinzuzufügen, um einen neuen Satz von Trainingsbildern zu erstellen. In mindestens einer Ausführungsform kann der Entwickler/Ingenieur nach Erhalt des Berichts mit der verringerten Zuverlässigkeitsstufe und nach Zugriff auf die Spielprotokolle für einen oder mehrere Benutzer eine neue Betrugssoftware beschaffen und auf dem Spieleserver und/oder dem Trainingsserver 304 installieren. In mindestens einer Ausführungsform kann der Spieleserver und/oder der Trainingsserver eine Reihe von neuronalen Netzen enthalten, die mit einem oder mehreren auf dem Gerät des Benutzers installierten neuronalen Netzen identisch oder ihnen ähnlich sind. In mindestens einer Ausführungsform können neuronale Netze mit einem neuen Satz von Neutrainingbildern neu trainiert werden, die in einer Ausführungsform mit dem unten beschriebenen Verfahren 800 erhalten werden können. In mindestens einer Ausführungsform können nach dem Neutraining unter Verwendung von Neutrainingbildern aktualisierte Parameter (Verzerrungen, Gewichte, Filterparameter, Aktivierungsparameter usw.) für ein oder mehrere neuronale Netze erhalten und an das Gerät des Benutzers übermittelt werden (z. B. während der Ausfallzeit des Geräts des Benutzers). In mindestens einer Ausführungsform kann die Verarbeitungslogik auf dem Gerät des Benutzers in Block 770 aktualisierte Parameter empfangen und ein oder mehrere neuronale Netze anhand der empfangenen Parameter aktualisieren.

[0067] **Fig. 8** ist ein Flussdiagramm eines Beispielverfahrens 800 zur Verwendung eines oder mehrerer neuronaler Netze zur Erkennung unerlaubter Informationen in Bildern, die von einer Grafikverarbeitungseinheit auf einem Display eines Benutzergeräts gemäß mindestens einer Ausführungsform dargestellt werden. In mindestens einer Ausführungsform kann das Verfahren 800 von einer Verarbeitungslogik des Trainingsservers 340 durchgeführt werden. In mindestens einer Ausführungsform kann das Verfahren 800 von einem einzigen Verarbeitungsprozess durchgeführt werden. Alternativ kann in mindestens einer Ausführungsform das Verfahren 800 von zwei oder mehreren Verarbeitungs-Threads durchgeführt werden, wobei jeder Thread eine oder mehrere individuelle Funktionen, Routinen, Unterprogramme oder Vorgänge des Verfahrens 800 ausführt. In mindestens einer Ausführungsform können Verarbeitungs-Threads, die das Verfahren 800 implementieren, synchronisiert werden (z. B. unter Verwendung von Semaphoren, kritischen Abschnitten und/oder anderen Thread-Synchronisierungsmechanismen). Alternativ können in mindestens einer Ausführungsform die Verarbeitungs-Threads, die das Verfahren 800 implementieren, asynchron zueinander ausgeführt werden. In mindestens einer Ausführungsform können verschiedene Vorgänge des Verfahrens 800 in einer anderen Reihenfolge durchgeführt werden als in **Fig. 8**. In mindestens einer Ausführungsform können einige Vorgänge des Verfahrens 800 gleichzeitig mit anderen Vorgängen durchgeführt werden. In mindestens einer Ausführungsform können ein oder mehrere der in **Fig. 8** gezeigten Vorgänge nicht durchgeführt werden.

Obwohl verschiedene Vorgänge des Verfahrens 800 in Bezug auf Spielanwendungen und Betrugssoftware beschrieben werden, versteht es sich von selbst, dass ähnliche Vorgänge verwendet werden können, um unzulässige Informationen in Bildern zu erkennen, die mit anderen Arten von Anwendungen verbunden sind. In mindestens einer Ausführungsform kann das Verfahren 800 sowohl für das Training neuronaler Netze als auch für das erneute Training bereits trainierter neuronaler Netze verwendet werden, wenn eine neue Betrugssoftware verfügbar wird.

[0068] In mindestens einer Ausführungsform kann die Verarbeitungslogik, die das Verfahren 800 durchführt, ein oder mehrere neuronale Netze (z. B. NN-Modelle 325) ausführen, die in einem Trainingsmodus arbeiten, um Parameter eines oder mehrerer neuronaler Netze zu bestimmen, um Betrugs- (oder andere unerlaubte) Bilder zu erkennen, die wiedergegeben und einem Benutzer (z. B. einem Spieler) angezeigt werden. In mindestens einer Ausführungsform kann das Verfahren 800 von einer Verarbeitungslogik durchgeführt werden, die Zugriff auf eine Spielesoftware (z. B. eine legale Software) hat, die in der Lage ist, echte (Nichtbetrugs-) Bilder zu erzeugen. In mindestens einer Ausführungsform kann das Verfahren 800 von einer Verarbeitungslogik durchgeführt werden, die Zugriff auf eine Betrugssoftware hat, die allein oder in Kombination mit einer Spielesoftware verwendet werden kann, um unzulässige Bilder (Betrugsbilder) zu erzeugen, bei denen es sich um Bilder handeln kann, die Informationen anzeigen, die einen unlauteren Vorteil verschaffen oder nicht autorisierte Informationen enthalten. Die Blöcke 810, 820 und 830 beziehen sich auf die Erzeugung von erweiterten betrügerischen Trainingsbildern, während die Blöcke 812, 822 und 832 sich auf die Erzeugung von erweiterten nicht betrügerischen Trainingsbildern beziehen. In mindestens einer Ausführungsform können die Blöcke 810, 820 und 830 unabhängig (z. B. parallel oder nacheinander) von den Blöcken 812, 822 und 832 durchgeführt werden.

[0069] In mindestens einer Ausführungsform kann die Verarbeitungslogik, die das Verfahren 800 durchführt, in Block 810 mit Hilfe von Spielsoftware ein Nichtbetrugsbild erzeugen. In mindestens einer Ausführungsform kann die Verarbeitungslogik, die das Verfahren 800 durchführt, in Block 820 ein grundlegendes Betrugsbild unter Verwendung von Betrugssoftware erzeugen. In mindestens einer Ausführungsform kann die Verarbeitungslogik, die das Verfahren 800 durchführt, in Block 830 einen Teil des grundlegenden Betrugsbildes durch einen Teil des Nichtbetrugsbildes ersetzen. Als Ergebnis kann in mindestens einer Ausführungsform ein erweitertes Betrugstrainingsbild erzeugt werden, das im Vergleich zum grundlegenden Betrugsbild eine geringere Menge an Betrugsinformationen enthält. In mindestens einer Ausführungsform können die Blöcke 810, 820 und 830 mehrfach wiederholt werden, um einen Stapel von mehreren erweiterten Betrugstrainingsbildern (basierend auf demselben grundlegenden Betrugsbild) mit unterschiedlichen Mengen an Betrugsinformationen zu erzeugen. Zusätzlich können in mindestens einer Ausführungsform die Blöcke 810, 820 und 830 unter Verwendung verschiedener grundlegender Betrugsbilder wiederholt werden, um mehrere Stapel (basierend auf verschiedenen grundlegenden Betrugsbildern) mit unterschiedlichen Mengen an Betrugsinformationen zu erhalten.

[0070] In mindestens einer Ausführungsform kann eine ähnliche Kreuzaugmentierung durchgeführt werden, um Nichtbetrugsbilder mit Einfügeartefakten zu versehen, die den Einfügeartefakten ähneln, die Betrugsbildern verliehen werden. In mindestens einer Ausführungsform kann die Verarbeitungslogik, die das Verfahren 800 durchführt, in Block 812 mit Hilfe von Spielsoftware ein grundlegendes Nichtbetrugsbild erzeugen. In mindestens einer Ausführungsform kann die Verarbeitungslogik, die das Verfahren 800 durchführt, in Block 822 ein Betrugsbild unter Verwendung von Betrugssoftware erzeugen. In mindestens einer Ausführungsform kann die Verarbeitungslogik, die das Verfahren 800 durchführt, in Block 832 einen Teil des grundlegenden Betrugsbildes durch einen Teil des Betrugsbildes ersetzen. In mindestens einer Ausführungsform kann als Ergebnis ein erweitertes, nicht betrügerisches Trainingsbild erzeugt werden, das Einfügeartefakte aufweist, die den Artefakten in erweiterten vergrößerten Betrugsbildern ähnlich sind. In mindestens einer Ausführungsform enthalten Teile des Betrugsbildes, die in die grundlegenden Nichtbetrugsbilder eingefügt werden, nur Nichtbetrugsinformationen (Nichtbetrugsbereiche des Betrugsbildes).

[0071] In mindestens einer Ausführungsform können die Blöcke 812, 822 und 832 mehrfach wiederholt werden, um einen Stapel von mehreren erweiterten Betrugstrainingsbildern (basierend auf demselben grundlegenden Nichtbetrugsbild) mit unterschiedlichen Mengen an Einfügeartefakten zu erzeugen. Zusätzlich können in mindestens einer Ausführungsform die Blöcke 812, 822 und 832 unter Verwendung verschiedener grundlegender Nichtbetrugsbilder wiederholt werden, um mehrere Stapel (basierend auf verschiedenen grundlegenden Nichtbetrugsbildern) mit unterschiedlichen Mengen an Einfügeartefakten zu erhalten.

[0072] In mindestens einer Ausführungsform kann die Verarbeitungslogik, die das Verfahren 800 durchführt, in Block 840 die erhaltenen (über die Blöcke 810, 820 und 830) erweiterten Betrugsbilder verwenden, um ein

oder mehrere neuronale Netzwerke zu trainieren, um Betrug durch einen oder mehrere Benutzer eines Computerspiels zu erkennen. In mindestens einer Ausführungsform kann die Verarbeitungslogik, die das Verfahren 800 durchführt, auch die (über die Blöcke 812, 822 und 832) erhaltenen erweiterten Nichtbetrugsbilder verwenden, um ein oder mehrere neuronale Netze zu trainieren.

RECHENZENTRUM

[0073] Fig. 9 stellt ein beispielhaftes Rechenzentrum 900 dar, in dem mindestens eine Ausführungsform verwendet werden kann. In mindestens einer Ausführungsform beinhaltet das Rechenzentrum 900 eine Rechenzentrumsinfrastrukturschicht 910, eine Rahmenschicht 920, eine Softwareschicht 930 und eine Anwendungsschicht 940.

[0074] In mindestens einer Ausführungsform, wie in Fig. 9 gezeigt, kann die Rechenzentrumsinfrastrukturschicht 910 einen Ressourcenorchestrierer 912, gruppierte Rechenressourcen 914 und Knoten-Rechenressourcen (node computing resources - „Knoten-CRs“) 916(1)-916(N) beinhalten, wobei „N“ eine positive ganze Zahl darstellt (die eine andere ganze Zahl „N“ sein kann, als in anderen Figuren verwendet). In mindestens einer Ausführungsform können die Knoten-CRs 916(1)-916(N) eine beliebige Anzahl von Zentraleinheiten („CPUs“) oder anderen Prozessoren (einschließlich Beschleunigern, feldprogrammierbarer Gate-Arrays (FPGAs), Grafikprozessoren usw.), Speichervorrichtungen 918(1)-918(N) (z. B. dynamischen Festwertspeicher, Festkörperspeicher oder Festplattenlaufwerke), Vorrichtungen zur Netz-Eingabe/Ausgabe (network input/output - „NW-I/O“), Netz-Switches, virtuellen Maschinen (virtual machines - „VMs“), Leistungsmodulen und Kühlmodulen usw. beinhalten, ohne darauf beschränkt zu sein. In mindestens einer Ausführungsform kann es sich bei einer oder mehreren Knoten-CRs aus den Knoten-CRs 916(1)-916(N) um einen Server handeln, der eine oder mehrere der vorstehend erwähnten Rechenressourcen aufweist.

[0075] In mindestens einer Ausführungsform können die gruppierten Rechenressourcen 914 separate Gruppierungen von Knoten-CRs 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-CRs innerhalb der gruppierten Rechenressourcen 914 können in mindestens einer Ausführungsform gruppierte Rechen-, Netz-, Arbeitsspeicher- oder Datenspeicherressourcen beinhalten, die dazu konfiguriert oder zugewiesen sein können, einen oder mehrere Workloads zu unterstützen. In mindestens einer Ausführungsform können mehrere Knoten-CRs, die CPUs oder Prozessoren beinhalten, in einem oder mehreren Racks gruppiert sein, um Rechenressourcen zur Unterstützung eines oder mehrerer Workloads bereitzustellen. In mindestens einer Ausführungsform können ein oder mehrere Racks auch eine beliebige Anzahl von Leistungsmodulen, Kühlmodulen und Netz-Switches in beliebiger Kombination beinhalten.

[0076] In mindestens einer Ausführungsform kann der Ressourcenorchestrierer 912 einen oder mehrere Knoten-CRs 916(1)-916(N) und/oder gruppierte Berechnungsressourcen 914 konfigurieren oder anderweitig steuern. In mindestens einer Ausführungsform kann der Ressourcenorchestrierer 912 eine Softwaredesigninfrastruktur („SDI“-)Verwaltungsinstanz für das Rechenzentrum 900 beinhalten. In mindestens einer Ausführungsform kann der Ressourcenorchestrierer 112 Hardware, Software oder eine Kombination davon beinhalten.

[0077] In mindestens einer Ausführungsform, wie in Fig. 9 gezeigt, die Frameworkschicht 920 einen Aufgabenplaner 922, einen Konfigurationsverwalter 924, einen Ressourcenverwalter 926 und ein verteiltes Dateisystem 928. In mindestens einer Ausführungsform kann die Frameworkschicht 920 ein Framework beinhalten, um Software 932 der Softwareschicht 930 und/oder eine oder mehrere Anwendungen 942 der Anwendungsschicht 940 zu unterstützen. In mindestens einer Ausführungsform kann/können die Software 932 bzw. die Anwendung(en) 942 webbasierte Dienst-Software oder -anwendungen beinhalten, wie sie beispielsweise von Amazon Web Services, Google Cloud und Microsoft Azure bereitgestellt werden. In mindestens einer Ausführungsform kann es sich bei der Framework-Schicht 920 um eine Art freies und Open-Source-Software-Webanwendungs-Framework wie etwa Apache Spark™ (im Folgenden „Spark“) handeln, das das verteilte Dateisystem 928 für die Verarbeitung großer Datenmengen (z. B. „Big Data“) verwenden kann, ohne darauf beschränkt zu sein. In mindestens einer Ausführungsform kann der Taskplaner 922 einen Spark-Treiber beinhalten, um die zeitliche Planung von Workloads zu erleichtern, die von verschiedenen Schichten des Rechenzentrums 900 unterstützt werden. In mindestens einer Ausführungsform kann der Konfigurationsverwalter 924 dazu in der Lage sein, unterschiedliche Schichten, wie etwa die Software-Schicht 930 und die Framework-Schicht 920 einschließlich Spark und des verteilten Dateisystems 928 zu konfigurieren, um die Verarbeitung großer Datenmengen zu unterstützen. In mindestens einer Ausführungsform kann

der Ressourcenverwalter 926 in der Lage sein, geclusterte oder gruppierte Berechnungsressourcen zu verwalten, die dem verteilten Dateisystem 928 und dem Taskplaner 922 zur Unterstützung zugeordnet oder zugewiesen sind. In mindestens einer Ausführungsform können geclusterte oder gruppierte Rechenressourcen gruppierte Rechenressourcen 914 in der Rechenzentrumsinfrastrukturschicht 910 beinhalten. In mindestens einer Ausführungsform kann sich der Ressourcenverwalter 926 mit dem Ressourcenorchestrator 912 koordinieren, um diese abgebildeten oder zugewiesenen Rechenressourcen zu verwalten.

[0078] In mindestens einer Ausführungsform kann die in der Software-Schicht 930 enthaltene Software 932 Software beinhalten, die mindestens durch Abschnitte der Knoten-CRs 916(1)-916(N), der gruppierten Rechenressourcen 914 und/oder des verteilten Dateisystems 928 der Rahmenschicht 920 verwendet wird. Zu einem oder mehreren Typen von Software können in mindestens einer Ausführungsform Software zum Durchsuchen von Internet-Webseiten, Software zum Scannen von E-Mails auf Viren, Datenbank-Software und Software für Streaming-Videoinhalte gehören, ohne darauf beschränkt zu sein.

[0079] In mindestens einer Ausführungsform können die in der Anwendungsschicht 940 enthaltenen Anwendung(en) 942 einen oder mehrere Typen von Anwendungen beinhalten, die mindestens durch Abschnitte der Knoten-CRs 916(1)-916(N), der gruppierten Rechenressourcen 914 und/oder des verteilten Dateisystems 928 der Rahmenschicht 920 verwendet werden. Zu einem oder mehreren Typen von Anwendungen können in mindestens einer Ausführungsform eine beliebige Anzahl von einer Genomikanwendung, einer Anwendung zur kognitiven Berechnung und einer Anwendung für maschinelles Lernen gehören, einschließlich Trainings- oder Inferenz-Software, Rahmen-Software für maschinelles Lernen (z. B. PyTorch, TensorFlow, Caffe usw.) oder anderer Anwendungen für maschinelles Lernen, die in Verbindung mit einer oder mehreren Ausführungsformen verwendet werden, ohne darauf beschränkt zu sein.

[0080] In mindestens einer Ausführungsform können beliebige des Konfigurationsverwalters 924, des Ressourcenverwalters 926 und des Ressourcenorchestrators 912 eine beliebige Anzahl und einen beliebigen Typ von selbstmodifizierenden Handlungen implementieren, die auf einer beliebigen Menge und einem beliebigen Typ von Daten basieren, die auf eine beliebige technisch machbare Weise erfasst wurden. In mindestens einer Ausführungsform können selbstmodifizierende Handlungen einen Rechenzentrumsbetreiber des Rechenzentrums 900 dahingehend entlasten, möglicherweise schlechte Konfigurationsentscheidungen zu treffen und möglicherweise nicht ausgelastete und/oder schlecht funktionierende Abschnitte eines Rechenzentrums zu vermeiden.

[0081] In mindestens einer Ausführungsform kann das Rechenzentrum 900 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 hierin beschriebener Ausführungsformen vorherzusagen oder zu inferenzieren. 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 900 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 900 beschriebenen Ressourcen zu inferenzieren oder vorherzusagen, indem Gewichtungsparmeter verwendet werden, die durch eine oder mehrere hierin beschriebene Trainingstechniken berechnet werden.

[0082] 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.

[0083] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 in dem System **Fig. 9** für Inferenz- oder Vorhersageoperationen mindestens zum Teil auf Grundlage von Gewichtungsparmetern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze,

Funktionen und/oder Architekturen von neuronalen Netzen oder hierin beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

AUTONOMES FAHRZEUG

[0084] Fig. 10A veranschaulicht ein Beispiel für ein autonomes Fahrzeug 1000 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann das autonome Fahrzeug 1000 (hier alternativ als „Fahrzeug 1000“ bezeichnet) ohne Einschränkung ein Personenfahrzeug sein, wie z. B. ein Pkw, ein Lkw, ein Bus und/oder eine andere Art von Fahrzeug, das einen oder mehrere Fahrgäste aufnimmt. In mindestens einer Ausführungsform kann das Fahrzeug 1000 ein Sattelschlepper sein, der zum Transport von Gütern verwendet wird. In mindestens einer Ausführungsform kann das Fahrzeug 1000 ein Flugzeug, ein Roboterfahrzeug oder eine andere Art von Fahrzeug sein.

[0085] Autonome Fahrzeuge können in Form von Automatisierungsgraden beschrieben werden, die von der National Highway Traffic Safety Administration („NHTSA“), einer Abteilung des US-Verkehrsministeriums, und der Society of Automotive Engineers („SAE“) „Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles“ (z. B. Standard Nr. J3016-201806, veröffentlicht am 15. Juni 2018, Standard Nr. J3016-201609, veröffentlicht am 30. September 2016, sowie frühere und künftige Versionen dieses Standards) definiert werden. In mindestens einer Ausführungsform kann das Fahrzeug 1000 in der Lage sein, Funktionen gemäß einer oder mehrerer der Stufen 1 bis 5 der autonomen Fahrstufen zu erfüllen. Zum Beispiel kann das Fahrzeug 1000 in mindestens einer Ausführungsform je nach Ausführungsform bedingt automatisiert (Stufe 3), hochautomatisiert (Stufe 4) und/oder vollständig automatisiert (Stufe 5) sein.

[0086] In mindestens einer Ausführungsform kann das Fahrzeug 1000 ohne Einschränkung Komponenten wie ein Fahrgestell, eine Fahrzeugkarosserie, Räder (z. B. 2, 4, 6, 8, 18 usw.), Reifen, Achsen und andere Komponenten eines Fahrzeugs umfassen. In mindestens einer Ausführungsform kann das Fahrzeug 1000 ohne Einschränkung ein Antriebssystem 1050 umfassen, wie z. B. einen Verbrennungsmotor, ein Hybrid-Elektrokraftwerk, einen reinen Elektromotor und/oder einen anderen Antriebssystemtyp. In mindestens einer Ausführungsform kann das Antriebssystem 1050 mit einem Antriebsstrang des Fahrzeugs 1000 verbunden sein, der unter anderem ein Getriebe umfassen kann, um den Antrieb des Fahrzeugs 1000 zu ermöglichen. In mindestens einer Ausführungsform kann das Antriebssystem 1050 als Reaktion auf den Empfang von Signalen von einer Drosselklappe/einem Gaspedal (1052) gesteuert werden.

[0087] In mindestens einer Ausführungsform wird ein Lenksystem 1054, das unter anderem ein Lenkrad umfassen kann, verwendet, um das Fahrzeug 1000 zu lenken (z. B. entlang eines gewünschten Weges oder einer Route), wenn das Antriebssystem 1050 in Betrieb ist (z. B. wenn das Fahrzeug 1000 in Bewegung ist). In mindestens einer Ausführungsform kann das Lenksystem 1054 Signale von dem/den Lenkaktuator(en) 1056 empfangen. In mindestens einer Ausführungsform kann ein Lenkrad für die vollständige Automatisierung (Stufe 5) optional sein. In mindestens einer Ausführungsform kann ein Bremssensorsystem 1046 verwendet werden, um die Fahrzeugbremsen in Reaktion auf den Empfang von Signalen von Bremsbetätigungselementen 1048 und/oder Bremssensoren zu betätigen.

[0088] In mindestens einer Ausführungsform kann (können) das (die) Steuergerät(e) 1036, das (die) ohne Einschränkung ein oder mehrere System-on-Chips („SoCs“) umfassen (nicht dargestellt in **Fig. 10A**) und/oder Grafikverarbeitungseinheit(en) („GPU(s)“) Signale (z. B. in Form von Befehlen) an eine oder mehrere Komponenten und/oder Systeme des Fahrzeugs 1000 liefern. In mindestens einer Ausführungsform kann (können) das (die) Steuergerät(e) 1036 beispielsweise Signale zur Betätigung der Fahrzeugbremsen über den (die) Bremsaktuator(en) 1048, zur Betätigung des Lenksystems 1054 über den (die) Lenkaktuator(en) 1056 und zur Betätigung des Antriebssystems 1050 über die (das) Drosselklappe/Gaspedal 1052 senden. In mindestens einer Ausführungsform kann (können) das (die) Steuergerät(e) 1036 ein oder mehrere eingebaute (z. B. integrierte) Rechengерäte umfassen, die Sensorsignale verarbeiten und Betriebsbefehle (z. B. Signale, die Befehle darstellen) ausgeben, um autonomes Fahren zu ermöglichen und/oder einen menschlichen Fahrer beim Führen des Fahrzeugs 1000 zu unterstützen. In mindestens einer Ausführungsform kann (können) das (die) Steuergerät(e) 1036 ein erstes Steuergerät für autonome Fahrfunktionen, ein zweites Steuergerät für funktionale Sicherheitsfunktionen, ein drittes Steuergerät für Funktionen der künstlichen Intelligenz (z. B. Computer Vision), ein viertes Steuergerät für Infotainment-Funktionen, ein fünftes Steuergerät für Redundanz in Notfällen und/oder andere Steuergeräte umfassen. In mindestens einer Ausführungsform kann ein einziges Steuergerät zwei oder mehr der oben genannten Funktionen übernehmen, zwei oder mehr Steuergeräte können eine einzige Funktion übernehmen und/oder eine beliebige Kombination davon.

[0089] In mindestens einer Ausführungsform liefern das/die Steuergerät(e) 1036 Signale zur Steuerung einer oder mehrerer Komponenten und/oder Systeme des Fahrzeugs 1000 in Reaktion auf Sensordaten, die von einem oder mehreren Sensoren (z. B. Sensoreingaben) empfangen werden. In mindestens einer Ausführungsform können Sensordaten beispielsweise und ohne Einschränkung von folgenden Sensoren empfangen werden: GNSS-Sensor(en) 1058 (z. B. Global Positioning System-Sensor(en)), RADAR-Sensor(en) 1060, Ultraschallsensor(en) 1062, LIDAR-Sensor(en) 1064, IMU-Sensor(en) 1066 (z. B., Beschleunigungsmesser, Gyroskop(e), Magnetkompass oder Magnetkompass, Magnetometer usw.), Mikrofon(e) 1096, Stereokamera(s) 1068, Weitwinkelkamera(s) 1070 (z. B. Fischaugenkameras), Infrarotkamera(s) 1072, Umgebungskamera(s) 1074 (z. B. 360-Grad-Kameras), Fernkameras (nicht dargestellt in **Fig. 10A**), Mittelbereichskamera(s) (nicht dargestellt in **Fig. 10A**), Geschwindigkeitssensor(en) 1044 (z. B. zur Messung der Geschwindigkeit des Fahrzeugs 1000), Schwingungssensor(en) 1042, Lenksensor(en) 1040, Bremssensor(en) (z. B. als Teil des Bremssensorsystems 1046) und/oder andere Sensortypen.

[0090] In mindestens einer Ausführungsform können ein oder mehrere Steuergerät(e) 1036 Eingaben (z. B. in Form von Eingabedaten) von einem Kombiinstrument 1032 des Fahrzeugs 1000 empfangen und Ausgaben (z. B. in Form von Ausgabedaten, Anzeigedaten usw.) über eine Mensch-Maschine-Schnittstelle („HMI“-Anzeige 1034, einen akustischen Melder, einen Lautsprecher und/oder über andere Komponenten des Fahrzeugs 1000 liefern. In mindestens einer Ausführungsform können die Ausgaben Informationen wie Fahrzeuggeschwindigkeit, Geschwindigkeit, Zeit, Kartendaten (z. B. eine High-Definition-Karte (nicht dargestellt in **Fig. 10A**), Standortdaten (z. B. der Standort des Fahrzeugs 1000, wie auf einer Karte), Richtung, Standort anderer Fahrzeuge (z. B. ein Belegungsrastrer), Informationen über Objekte und den Status von Objekten, wie von dem/den Steuergerät(en) 1036 wahrgenommen, usw. Zum Beispiel kann in mindestens einer Ausführungsform die HMI-Anzeige 1034 Informationen über das Vorhandensein eines oder mehrerer Objekte (z. B. ein Straßenschild, ein Warnschild, eine sich ändernde Ampel usw.) und/oder Informationen über Fahrmanöver anzeigen, die das Fahrzeug durchgeführt hat, durchführt oder durchführen wird (z. B. Spurwechsel jetzt, Ausfahrt 34B in zwei Meilen usw.).

[0091] In mindestens einer Ausführungsform umfasst das Fahrzeug 1000 außerdem eine Netzwerkschnittstelle 1024, die drahtlose Antenne(n) 1026 und/oder Modem(e) zur Kommunikation über ein oder mehrere Netzwerke verwenden kann. In mindestens einer Ausführungsform kann die Netzwerkschnittstelle 1024 beispielsweise zur Kommunikation über Long-Term Evolution („LTE“), Wideband Code Division Multiple Access („WCDMA“), Universal Mobile Telecommunications System („UMTS“), Global System for Mobile communication („GSM“), IMT-CDMA Multi-Carrier („CDMA2000“) Netze usw. geeignet sein. In mindestens einer Ausführungsform können die drahtlose(n) Antenne(n) 1026 auch die Kommunikation zwischen Objekten in der Umgebung (z. B. Fahrzeuge, mobile Geräte usw.) ermöglichen, wobei lokale Netzwerke wie Bluetooth, Bluetooth Low Energy („LE“), Z-Wave, ZigBee usw. und/oder Weitverkehrsnetze mit geringer Leistung („LPWANs“) wie LoRaWAN, SigFox usw. verwendet werden.

[0092] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 in dem System **Fig. 10A** für Inferenz- oder Vorhersageoperationen mindestens zum Teil auf Grundlage von Gewichtsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder hierin beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0093] **Fig. 10B** veranschaulicht ein Beispiel für Kamerapositionen und Sichtfelder für das autonome Fahrzeug 1000 von **Fig. 10A** gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform sind die Kameras und die jeweiligen Sichtfelder ein Beispiel für eine Ausführungsform und nicht als Einschränkung zu verstehen. Zum Beispiel können in mindestens einer Ausführungsform zusätzliche und/oder alternative Kameras enthalten sein und/oder die Kameras können sich an verschiedenen Stellen des Fahrzeugs 1000 befinden.

[0094] In mindestens einer Ausführungsform können Kameratypen für Kameras Digitalkameras umfassen, die für die Verwendung mit Komponenten und/oder Systemen des Fahrzeugs 1000 angepasst werden können, sind aber nicht darauf beschränkt. In mindestens einer Ausführungsform kann (können) die Kamera(s) auf der Fahrzeugsicherheitsstufe („ASIL“) B und/oder einer anderen ASIL betrieben werden. In mindestens einer Ausführungsform können die Kameratypen jede beliebige Bildaufnahmegeschwindigkeit erreichen, z. B. 60 Bilder pro Sekunde (fps), 1220 fps, 240 fps usw., je nach Ausführungsform. In mindestens einer Aus-

führungsform können die Kameras Rollverschlüsse, Global Shutter, einen anderen Verschlusstyp oder eine Kombination davon verwenden. In mindestens einer Ausführungsform kann das Farbfilter-Array ein Rot-Klar-Klar-Farbfilter-Array („RCCC“), ein Rot-Klar-Klar-Blau-Farbfilter-Array („RCCB“), ein Rot-Blau-Grün-Klar-Farbfilter-Array („RBGC“), ein Foveon X3-Farbfilter-Array, ein Bayer-Sensor-Farbfilter-Array („RGGB“), ein Monochromsensor-Farbfilter-Array und/oder eine andere Art von Farbfilter-Array umfassen. In mindestens einer Ausführungsform können zur Erhöhung der Lichtempfindlichkeit Clear-Pixel-Kameras, wie z. B. Kameras mit einer RCCC-, einer RCCB- und/oder einer RBGC-Farbfilteranordnung, verwendet werden.

[0095] In mindestens einer Ausführungsform können eine oder mehrere der Kameras verwendet werden, um fortschrittliche Fahrerassistenzsysteme („ADAS“) auszuführen (z. B. als Teil einer redundanten oder ausfallsicheren Konstruktion). In mindestens einer Ausführungsform kann beispielsweise eine Multifunktions-Monokamera installiert werden, die Funktionen wie Spurhalteassistent, Verkehrszeichenassistent und intelligente Scheinwerfersteuerung bietet. In mindestens einer Ausführungsform können eine oder mehrere der Kameras (z. B. alle Kameras) gleichzeitig Bilddaten (z. B. Video) aufzeichnen und bereitstellen.

[0096] In mindestens einer Ausführungsform können eine oder mehrere Kameras in einer Montagevorrichtung, z. B. einer kundenspezifisch entworfenen (dreidimensionalen („3D“) gedruckten) Vorrichtung, montiert werden, um Streulicht und Reflexionen aus dem Inneren des Fahrzeugs 1000 (z. B. Reflexionen vom Armaturenbrett, die in den Spiegeln der Windschutzscheibe reflektiert werden) auszuschalten, die die Fähigkeit der Kamera zur Bilddatenerfassung beeinträchtigen können. In Bezug auf die Montage von Außenspiegeln kann in mindestens einer Ausführungsform der Außenspiegel kundenspezifisch 3D-gedrukt werden, so dass eine Kameramontageplatte der Form eines Außenspiegels entspricht. In mindestens einer Ausführungsform können die Kamera(s) in die Außenspiegel integriert werden. Bei mindestens einer Ausführungsform können die Kameras für Seitenkameras auch in vier Säulen an jeder Ecke der Kabine integriert werden.

[0097] In mindestens einer Ausführungsform können Kameras mit einem Sichtfeld, das Teile einer Umgebung vor dem Fahrzeug 1000 einschließt (z. B. nach vorne gerichtete Kameras), für die Umgebungsansicht verwendet werden, um dabei zu helfen, nach vorne gerichtete Pfade und Hindernisse zu identifizieren, sowie mit Hilfe eines oder mehrerer Steuergeräte 1036 und/oder Steuer-SoCs bei der Bereitstellung von Informationen zu helfen, die für die Erstellung eines Belegungsgitters und/oder die Bestimmung bevorzugter Fahrzeugpfade entscheidend sind. In mindestens einer Ausführungsform können nach vorne gerichtete Kameras verwendet werden, um viele ähnliche ADAS-Funktionen wie LIDAR auszuführen, einschließlich, ohne Einschränkung, Notbremsung, Fußgängererkennung und Kollisionsvermeidung. In mindestens einer Ausführungsform können nach vorne gerichtete Kameras auch für ADAS-Funktionen und -Systeme verwendet werden, einschließlich, aber nicht beschränkt auf Spurhaltewarnungen (Lane Departure Warnings - „LDW“), autonome Geschwindigkeitssteuerung (Autonomous Cruise Control - „ACC“) und/oder andere Funktionen wie die Verkehrszeichenerkennung.

[0098] In mindestens einer Ausführungsform können verschiedene Kameras in einer nach vorne gerichteten Konfiguration verwendet werden, z. B. eine monokulare Kameraplattform mit einem CMOS-Farbbildsensor („complementary metal oxide semiconductor“). In mindestens einer Ausführungsform kann eine Weitwinkelkamera 1070 verwendet werden, um Objekte zu erkennen, die aus dem Umkreis ins Blickfeld kommen (z. B. Fußgänger, kreuzende Fahrzeuge oder Fahrräder). Obwohl in **Fig. 10B** nur eine Weitwinkelkamera 1070 dargestellt ist, können in anderen Ausführungsformen beliebig viele (einschließlich null) Weitwinkelkameras am Fahrzeug 1000 vorhanden sein. In mindestens einer Ausführungsform kann eine beliebige Anzahl von Weitbereichskameras 1098 (z. B. ein Langstrecken-Stereokamerapaar) zur tiefenbasierten Objekterkennung verwendet werden, insbesondere für Objekte, für die ein neuronales Netz noch nicht trainiert wurde. In mindestens einer Ausführungsform kann (können) die Weitbereichskamera(s) 1098 auch zur Objekterkennung und -klassifizierung sowie zur grundlegenden Objektverfolgung verwendet werden.

[0099] In mindestens einer Ausführungsform kann auch eine beliebige Anzahl von Stereokameras 1068 in einer nach vorne gerichteten Konfiguration enthalten sein. In mindestens einer Ausführungsform können eine oder mehrere Stereokameras 1068 eine integrierte Steuereinheit mit einer skalierbaren Verarbeitungseinheit enthalten, die eine programmierbare Logik („FPGA“) und einen Mehrkern-Mikroprozessor mit einer integrierten Controller Area Network („CAN“)- oder Ethernet-Schnittstelle auf einem einzigen Chip bereitstellen kann. In mindestens einer Ausführungsform kann eine solche Einheit verwendet werden, um eine 3D-Karte der Umgebung des Fahrzeugs 1000 zu erstellen, einschließlich einer Abstandsschätzung für alle Punkte in einem Bild. In mindestens einer Ausführungsform können eine oder mehrere der Stereokamera(s) 1068 ohne Einschränkung kompakte Stereosicht-Sensoren umfassen, die ohne Einschränkung zwei Kameralinsen (je eine links und rechts) und einen Bildverarbeitungschip enthalten können, der die Entfernung zwi-

schen Fahrzeug 1000 und Zielobjekt messen und die erzeugten Informationen (z. B. Metadaten) verwenden kann, um autonome Notbrems- und Spurhaltewarnfunktionen zu aktivieren. In mindestens einer Ausführungsform können zusätzlich oder alternativ zu den hier beschriebenen Stereokameras auch andere Typen von Stereokameras (1068) verwendet werden.

[0100] In mindestens einer Ausführungsform können Kameras mit einem Sichtfeld, das Teile der Umgebung an den Seiten des Fahrzeugs 1000 einschließt (z. B. Seitenkameras), für die Umgebungsansicht verwendet werden und Informationen liefern, die zur Erstellung und Aktualisierung eines Belegungsrasters sowie zur Erzeugung von SeitenaufprallWarnungen verwendet werden. In mindestens einer Ausführungsform könnte (n) beispielsweise die Umgebungskamera(s) 1074 (z. B. vier Umgebungskameras, wie in **Fig. 10B** dargestellt) am Fahrzeug 1000 positioniert werden. In mindestens einer Ausführungsform kann die Umgebungskamera(n) 1074 ohne Einschränkung eine beliebige Anzahl und Kombination von Weitwinkelkameras, Fischaugenkameras, 360-Grad-Kameras und/oder ähnlichen Kameras umfassen. Zum Beispiel können in mindestens einer Ausführungsform vier Fischaugenkameras an der Vorderseite, am Heck und an den Seiten des Fahrzeugs 1000 angebracht werden. In mindestens einer Ausführungsform kann das Fahrzeug 1000 drei Umgebungskameras 1074 (z. B. links, rechts und hinten) verwenden und eine oder mehrere andere Kamera(s) (z. B. eine nach vorne gerichtete Kamera) als vierte Umgebungskamera nutzen.

[0101] In mindestens einer Ausführungsform können Kameras mit einem Sichtfeld, das Teile der Umgebung hinter dem Fahrzeug 1000 einschließt (z. B. Rückfahrkameras), für die Einparkhilfe, die Umgebungsansicht, Heckkollisionswarnungen und das Erstellen und Aktualisieren eines Belegungsrasters verwendet werden. In mindestens einer Ausführungsform kann eine Vielzahl von Kameras verwendet werden, einschließlich, aber nicht beschränkt auf Kameras, die auch als nach vorne gerichtete Kamera(s) geeignet sind (z. B. Weitbereichskameras 1098 und/oder Mittelbereichskamera(s) 1076, Stereokamera(s) 1068), Infrarotkamera(s) 1072 usw.), wie hier beschrieben.

[0102] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 in dem System **Fig. 10B** für Inferenz- oder Vorhersageoperationen mindestens zum Teil auf Grundlage von Gewichtsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder hierin beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0103] **Fig. 10C** ist ein Blockdiagramm, das eine Beispielsystemarchitektur für das autonome Fahrzeug 1000 von **Fig. 10A** gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist jedes der Komponenten, Merkmale und Systeme des Fahrzeugs 1000 in **Fig. 10C** dargestellt als über einen Bus 1002 verbunden. In mindestens einer Ausführungsform kann der Bus 1002 ohne Einschränkung eine CAN-Datenschnittstelle umfassen (hier auch als „CAN-Bus“ bezeichnet). In mindestens einer Ausführungsform kann ein CAN ein Netzwerk innerhalb des Fahrzeugs 1000 sein, das zur Unterstützung der Steuerung verschiedener Merkmale und Funktionen des Fahrzeugs 1000 verwendet wird, wie z. B. Betätigung der Bremsen, Beschleunigung, Bremsen, Lenkung, Scheibenwischer usw. In mindestens einer Ausführungsform kann der Bus 1002 so konfiguriert sein, dass er Dutzende oder sogar Hunderte von Knoten aufweist, von denen jeder seine eigene eindeutige Kennung hat (z. B. eine CAN-ID). In mindestens einer Ausführungsform kann der Bus 1002 ausgelesen werden, um den Lenkradwinkel, die Fahrgeschwindigkeit, die Motordrehzahl pro Minute („RPM“), die Tastenpositionen und/oder andere Fahrzeugstatusanzeigen zu ermitteln. In mindestens einer Ausführungsform kann der Bus 1002 ein CAN-Bus sein, der ASIL B-konform ist.

[0104] In mindestens einer Ausführungsform können zusätzlich zu oder alternativ zu CAN auch FlexRay- und/oder Ethernet-Protokolle verwendet werden. In mindestens einer Ausführungsform kann es eine beliebige Anzahl von Bussen geben, die den Bus 1002 bilden, darunter, ohne Einschränkung, null oder mehr CAN-Busse, null oder mehr FlexRay-Busse, null oder mehr Ethernet-Busse und/oder null oder mehr andere Arten von Bussen, die unterschiedliche Protokolle verwenden. In mindestens einer Ausführungsform können zwei oder mehr Busse verwendet werden, um verschiedene Funktionen auszuführen, und/oder sie können zur Redundanz verwendet werden. So kann beispielsweise ein erster Bus für die Kollisionsvermeidung und ein zweiter Bus für die Betätigungssteuerung verwendet werden. In mindestens einer Ausführungsform kann jeder Bus des Busses 1002 mit beliebigen Komponenten des Fahrzeugs 1000 kommunizieren, und zwei oder mehr Busse des Busses 1002 können mit entsprechenden Komponenten kommunizieren. In mindestens einer Ausführungsform kann jede beliebige Anzahl von System(en) auf Chip(s) („SoC(s)“) 1004 (wie SoC

1004(A) und SoC 1004(B)), jedes Steuergerät 1036 und/oder jeder Computer im Fahrzeug Zugriff auf dieselben Eingangsdaten (z. B. Eingaben von Sensoren des Fahrzeugs 1000) haben und mit einem gemeinsamen Bus, wie dem CAN-Bus, verbunden sein.

[0105] In mindestens einer Ausführungsform kann das Fahrzeug 1000 ein oder mehrere Steuergerät(e) 1036 enthalten, wie sie hier in Bezug auf **Fig. 10 A**. In mindestens einer Ausführungsform kann das/die Steuergerät(e) 1036 für eine Vielzahl von Funktionen verwendet werden. In mindestens einer Ausführungsform kann (können) das (die) Steuergerät(e) 1036 mit verschiedenen anderen Komponenten und Systemen des Fahrzeugs 1000 gekoppelt sein und für die Steuerung des Fahrzeugs 1000, die künstliche Intelligenz des Fahrzeugs 1000, das Infotainment des Fahrzeugs 1000 und/oder andere Funktionen verwendet werden.

[0106] In mindestens einer Ausführungsform kann das Fahrzeug 1000 eine beliebige Anzahl von SoCs 1004 enthalten. In mindestens einer Ausführungsform kann jeder der SoCs 1004 ohne Einschränkung zentrale Verarbeitungseinheiten („CPU(s)“) 1006, Grafikverarbeitungseinheiten („GPU(s)“) 1008, Prozessor(en) 1010, Cache(s) 1012, Beschleuniger 1014, Datenspeicher 1016 und/oder andere nicht dargestellte Komponenten und Merkmale umfassen. In mindestens einer Ausführungsform können die SoC(s) 1004 zur Steuerung des Fahrzeugs 1000 in einer Vielzahl von Plattformen und Systemen verwendet werden. Zum Beispiel können in mindestens einer Ausführungsform SoC(s) 1004 in einem System (z.B. dem System des Fahrzeugs 1000) mit einer High-Definition („HD“)-Karte 1022 kombiniert werden, die über die Netzwerkschnittstelle 1024 von einem oder mehreren Servern Auffrischungen und/oder Aktualisierungen der Karte erhalten kann (nicht dargestellt in **Fig. 10C**).

[0107] In mindestens einer Ausführungsform kann (können) die CPU(s) 1006 einen CPU-Cluster oder CPU-Komplex (hier auch als „CCPLEX“ bezeichnet) umfassen. In mindestens einer Ausführungsform kann (können) die CPU(s) 1006 mehrere Kerne und/oder Level Two („L2“) Cache enthalten. In mindestens einer Ausführungsform können die CPU(s) 1006 beispielsweise acht Kerne in einer kohärenten Multiprozessoranordnung umfassen. In mindestens einer Ausführungsform kann (können) die CPU(s) 1006 vier Dual-Core-Cluster umfassen, wobei jeder Cluster über einen dedizierten L2-Cache verfügt (z. B. einen 2-Megabyte (MB) L2-Cache). In mindestens einer Ausführungsform kann (können) die CPU(s) 1006 (z. B. CPLEX) so konfiguriert werden, dass sie gleichzeitige Cluster-Operationen unterstützen, so dass eine beliebige Kombination von Clustern von CPU(s) 1006 zu einem bestimmten Zeitpunkt aktiv sein kann.

[0108] In mindestens einer Ausführungsform können eine oder mehrere der CPU(s) 1006 Energieverwaltungsfunktionen implementieren, die ohne Einschränkung eines oder mehrere der folgenden Merkmale umfassen: einzelne Hardwareblöcke können im Leerlauf automatisch getaktet werden, um dynamische Energie zu sparen; jeder Kerntakt kann getaktet werden, wenn ein solcher Kern aufgrund der Ausführung von Wait for Interrupt („WFI“)/Wait for Event („WFE“)-Befehlen nicht aktiv Befehle ausführt; jeder Kern kann unabhängig stromgesteuert sein; jeder Kerncluster kann unabhängig taktgesteuert sein, wenn alle Kerne taktgesteuert oder stromgesteuert sind; und/oder jeder Kerncluster kann unabhängig stromgesteuert sein, wenn alle Kerne stromgesteuert sind. In mindestens einer Ausführungsform kann (können) die CPU(s) 1006 außerdem einen erweiterten Algorithmus für die Verwaltung von Energiezuständen implementieren, bei dem zulässige Energiezustände und erwartete Aufwachzeiten festgelegt werden und die Hardware/der Mikrocode bestimmt, welcher Energiezustand für Kern, Cluster und CPLEX am besten geeignet ist. In mindestens einer Ausführungsform können die Verarbeitungskerne vereinfachte Sequenzen für die Eingabe des Energiezustands in der Software unterstützen, wobei die Arbeit auf den Mikrocode ausgelagert wird.

[0109] In mindestens einer Ausführungsform kann (können) die GPU(s) 1008 eine integrierte GPU (hier alternativ als „iGPU“ bezeichnet) umfassen. In mindestens einer Ausführungsform können die GPU(s) 1008 programmierbar und für parallele Arbeitslasten effizient sein. In mindestens einer Ausführungsform können die GPU(s) 1008 einen erweiterten Tensor-Befehlssatz verwenden. In mindestens einer Ausführungsform kann (können) die GPU(s) 1008 einen oder mehrere Streaming-Mikroprozessoren enthalten, wobei jeder Streaming-Mikroprozessor einen Level-1-Cache („L1“) enthalten kann (z. B. einen L1-Cache mit mindestens 96 KB Speicherkapazität), und zwei oder mehr Streaming-Mikroprozessoren können sich einen L2-Cache teilen (z. B. einen L2-Cache mit einer Speicherkapazität von 512 KB). In mindestens einer Ausführungsform kann (können) die GPU(s) 1008 mindestens acht Streaming-Mikroprozessoren umfassen. In mindestens einer Ausführungsform können die GPU(s) 1008 Anwendungsprogrammierschnittstelle(n) (API(s)) für Berechnungen verwenden. In mindestens einer Ausführungsform können die GPU(s) 1008 eine oder mehrere parallele Rechenplattformen und/oder Programmiermodelle (z. B. das CUDA-Modell von NVIDIA) verwenden.

[0110] In mindestens einer Ausführungsform können eine oder mehrere der GPU(s) 1008 für die beste Leistung in Automobil- und eingebetteten Anwendungsfällen optimiert werden. In mindestens einer Ausführungsform könnte(n) die GPU(s) 1008 beispielsweise auf Fin-Feldeffekttransistor-Schaltungen („FinFET“) hergestellt werden. In mindestens einer Ausführungsform kann jeder Streaming-Mikroprozessor eine Reihe von in mehrere Blöcke unterteilten gemischt-präzisen Verarbeitungskernen enthalten. So könnten beispielsweise 64 PF32-Kerne und 32 PF64-Kerne in vier Verarbeitungsblöcke unterteilt werden, ohne darauf beschränkt zu sein. In mindestens einer Ausführungsform könnten jedem Verarbeitungsblock 16 FP32-Kerne, 8 FP64-Kerne, 16 INT32-Kerne, zwei NVIDIA Tensor-Kerne mit gemischter Genauigkeit für Deep-Learning-Matrixarithmetik, ein Befehls-cache der Ebene Null („L0“), ein Warp-Scheduler, eine Dispatch-Einheit und/oder eine 64-KB-Registerdatei zugewiesen werden. In mindestens einer Ausführungsform können Streaming-Mikroprozessoren unabhängige parallele Ganzzahl- und Gleitkomma-Datenpfade enthalten, um eine effiziente Ausführung von Arbeitslasten mit einer Mischung aus Berechnungen und Adressierungsberechnungen zu ermöglichen. In mindestens einer Ausführungsform können Streaming-Mikroprozessoren eine unabhängige Thread-Planungsfunktion enthalten, um eine feinkörnigere Synchronisierung und Zusammenarbeit zwischen parallelen Threads zu ermöglichen. In mindestens einer Ausführungsform können Streaming-Mikroprozessoren einen kombinierten L1-Datencache und eine gemeinsam genutzte Speichereinheit enthalten, um die Leistung zu verbessern und gleichzeitig die Programmierung zu vereinfachen.

[0111] In mindestens einer Ausführungsform können eine oder mehrere der GPU(s) 1008 einen Speicher mit hoher Bandbreite („HBM“) und/oder ein 16-GB-Speicher-Subsystem mit hoher Bandbreite der zweiten Generation („HBM2“) enthalten, um in einigen Beispielen eine Spitzen-Speicherbandbreite von etwa 900 GB/Sekunde bereitzustellen. In mindestens einer Ausführungsform kann zusätzlich oder alternativ zum HBM-Speicher ein synchroner Grafik-Direktzugriffsspeicher („SGRAM“) verwendet werden, z. B. ein synchroner Grafik-Doppeldatenraten-Direktzugriffsspeicher vom Typ 5 („GDDR5“).

[0112] In mindestens einer Ausführungsform können die GPU(s) 1008 eine einheitliche Speichertechnologie enthalten. In mindestens einer Ausführungsform kann die Unterstützung von Adressübersetzungsdiensten („ATS“) verwendet werden, um der/den GPU(s) 1008 den direkten Zugriff auf Seitentabellen der CPU(s) 1006 zu ermöglichen. In mindestens einer Ausführungsform kann eine Adressübersetzungsanforderung an die CPU(s) 1006 übermittelt werden, wenn eine GPU der GPU(s) 1008 Speicherverwaltungseinheit („MMU“) einen Fehler feststellt. Als Reaktion darauf kann 2 CPU der CPU(s) 1006 in ihren Seitentabellen nach einer virtuell-physikalischen Zuordnung für eine Adresse suchen und in mindestens einer Ausführungsform die Übersetzung zurück an die GPU(s) 1008 übertragen. In mindestens einer Ausführungsform kann die Unified-Memory-Technologie einen einzigen, einheitlichen virtuellen Adressraum für den Speicher sowohl der CPU(s) 1006 als auch der GPU(s) 1008 ermöglichen, wodurch die Programmierung der GPU(s) 1008 und die Portierung von Anwendungen auf die GPU(s) 1008 vereinfacht wird.

[0113] In mindestens einer Ausführungsform kann (können) die GPU(s) 1008 eine beliebige Anzahl von Zugriffszählern enthalten, die die Häufigkeit des Zugriffs der GPU(s) 1008 auf den Speicher anderer Prozessoren verfolgen können. In mindestens einer Ausführungsform können Zugriffszähler dazu beitragen, dass Speicherseiten in den physischen Speicher eines Prozessors verschoben werden, der am häufigsten auf Seiten zugreift, wodurch die Effizienz von Speicherbereichen, die von mehreren Prozessoren gemeinsam genutzt werden, verbessert wird.

[0114] In mindestens einer Ausführungsform können ein oder mehrere SoC(s) 1004 eine beliebige Anzahl von Cache(s) 1012 enthalten, einschließlich der hier beschriebenen. In mindestens einer Ausführungsform könnte(n) der/die Cache(s) 1012 beispielsweise einen Level-3-Cache („L3“) enthalten, der sowohl für die CPU(s) 1006 als auch für die GPU(s) 1008 verfügbar ist (z. B. mit der/den CPU(s) 1006 und der/den GPU(s) 1008 verbunden ist). In mindestens einer Ausführungsform kann (können) der (die) Cache(s) 1012 einen Write-Back-Cache umfassen, der den Zustand von Zeilen verfolgen kann, z. B. durch Verwendung eines Cache-Kohärenzprotokolls (z. B. MEI, MESI, MSI usw.). In mindestens einer Ausführungsform kann ein L3-Cache je nach Ausführungsform 4 MB oder mehr Speicher umfassen, obwohl auch kleinere Cache-Größen verwendet werden können.

[0115] In mindestens einer Ausführungsform können ein oder mehrere SoC(s) 1004 einen oder mehrere Beschleuniger 1014 enthalten (z. B. Hardware-Beschleuniger, Software-Beschleuniger oder eine Kombination davon). In mindestens einer Ausführungsform können die SoC(s) 1004 einen Hardware-Beschleunigungs-Cluster enthalten, der optimierte Hardware-Beschleuniger und/oder einen großen On-Chip-Speicher umfassen kann. In mindestens einer Ausführungsform kann ein großer On-Chip-Speicher (z. B. 4 MB SRAM) einen Hardware-Beschleunigungscluster zur Beschleunigung neuronaler Netze und anderer Berech-

nungen ermöglichen. In mindestens einer Ausführungsform kann ein Hardware-Beschleunigungscluster zur Ergänzung der GPU(s) 1008 und zur Entlastung einiger Aufgaben der GPU(s) 1008 verwendet werden (z. B. um mehr Zyklen der GPU(s) 1008 für die Durchführung anderer Aufgaben freizugeben). In mindestens einer Ausführungsform könnte(n) der/die Beschleuniger 1014 für gezielte Arbeitslasten (z. B. Wahrnehmung, Faltungsneuronale Netze („CNNs“), rekurrente Neuronale Netze („RNNs“) usw.) verwendet werden, die stabil genug sind, um sich für eine Beschleunigung zu eignen. In mindestens einer Ausführungsform kann ein CNN ein regionenbasiertes oder regionales neuronales Faltungsnetzwerk („RCNN“) und ein schnelles RCNN (z. B. für die Objekterkennung) oder eine andere Art von CNN umfassen.

[0116] In mindestens einer Ausführungsform kann (können) der (die) Beschleuniger 1014 (z. B. ein Hardware-Beschleunigungscluster) einen oder mehrere Deep-Learning-Beschleuniger („DLA“) umfassen. In mindestens einer Ausführungsform können die DLA(s) ohne Einschränkung eine oder mehrere Tensor Processing Units („TPUs“) enthalten, die so konfiguriert sein können, dass sie zusätzliche zehn Billionen Operationen pro Sekunde für Deep-Learning-Anwendungen und Inferenzierung bereitstellen. In mindestens einer Ausführungsform können TPUs Beschleuniger sein, die für die Ausführung von Bildverarbeitungsfunktionen (z. B. für CNNs, RCNNs usw.) konfiguriert und optimiert sind. In mindestens einer Ausführungsform können die DLA(s) darüber hinaus für einen bestimmten Satz neuronaler Netzwerktypen und Gleitkommaoperationen sowie für die Inferenzierung optimiert werden. In mindestens einer Ausführungsform kann das Design von DLA(s) mehr Leistung pro Millimeter bieten als ein typischer Allzweck-GPU und übertrifft in der Regel die Leistung einer CPU bei weitem. In mindestens einer Ausführungsform kann (können) die TPU(s) mehrere Funktionen ausführen, einschließlich einer Einzelinstanz-Faltungsfunktion, die z. B. INT8-, INT16- und FP16-Datentypen sowohl für Merkmale als auch für Gewichte unterstützt, sowie Postprozessorfunktionen. In mindestens einer Ausführungsform können DLA(s) schnell und effizient neuronale Netze, insbesondere CNNs, auf verarbeiteten oder unverarbeiteten Daten für eine Vielzahl von Funktionen ausführen, darunter beispielsweise und ohne Einschränkung: ein CNN für die Identifizierung und Erkennung von Objekten unter Verwendung von Daten von Kamerasensoren; ein CNN für die Abstandsschätzung unter Verwendung von Daten von Kamerasensoren; ein CNN für die Erkennung und Identifizierung von Einsatzfahrzeugen und die Erkennung unter Verwendung von Daten von Mikrofonen; ein CNN für die Gesichtserkennung und die Identifizierung von Fahrzeugbesitzern unter Verwendung von Daten von Kamerasensoren; und/oder ein CNN für sicherheitsrelevante Ereignisse.

[0117] In mindestens einer Ausführungsform können DLA(s) jede Funktion von GPU(s) 1008 ausführen, und durch die Verwendung eines Inferenzbeschleunigers kann ein Entwickler beispielsweise entweder DLA(s) oder GPU(s) 1008 für jede Funktion vorsehen. In mindestens einer Ausführungsform kann ein Entwickler beispielsweise die Verarbeitung von CNNs und Gleitkommaoperationen auf DLA(s) konzentrieren und andere Funktionen den GPU(s) 1008 und/oder Beschleunigern 1014 überlassen.

[0118] In mindestens einer Ausführungsform kann der/die Beschleuniger 1014 einen programmierbaren Bildverarbeitungsbeschleuniger („PVA“) umfassen, der hier auch als Computer-Vision-Beschleuniger bezeichnet werden kann. In mindestens einer Ausführungsform kann der PVA so konzipiert und konfiguriert sein, dass sie Computer-Vision-Algorithmen für fortschrittliche Fahrerassistenzsysteme („ADAS“) 1038, autonomes Fahren, Augmented-Reality-Anwendungen („AR“) und/oder Virtual-Reality-Anwendungen („VR“) beschleunigt. In mindestens einer Ausführungsform kann der PVA ein Gleichgewicht zwischen Leistung und Flexibilität bieten. In mindestens einer Ausführungsform kann jeder PVA beispielsweise und ohne Einschränkung eine beliebige Anzahl von Rechenkernen mit reduziertem Befehlssatz („RISC“), direkten Speicherzugriff („DMA“) und/oder eine beliebige Anzahl von Vektorprozessoren umfassen.

[0119] In mindestens einer Ausführungsform können die RISC-Kerne mit Bildsensoren (z. B. den Bildsensoren der hier beschriebenen Kameras), Bildsignalprozessoren usw. zusammenwirken. In mindestens einer Ausführungsform kann jeder RISC-Kern eine beliebige Menge an Speicher enthalten. In mindestens einer Ausführungsform können die RISC-Kerne je nach Ausführungsform eines von mehreren Protokollen verwenden. In mindestens einer Ausführungsform können RISC-Kerne ein Echtzeitbetriebssystem („RTOS“) ausführen. In mindestens einer Ausführungsform können RISC-Kerne mit einem oder mehreren integrierten Schaltkreisen, anwendungsspezifischen integrierten Schaltkreisen („ASICs“) und/oder Speicherbausteinen implementiert werden. In mindestens einer Ausführungsform könnten RISC-Kerne beispielsweise einen Befehls-cache und/oder einen eng gekoppelten RAM-Speicher enthalten.

[0120] In mindestens einer Ausführungsform kann DMA den Komponenten der PVA ermöglichen, unabhängig von der/den CPU(s) 1006 auf den Systemspeicher zuzugreifen. In mindestens einer Ausführungsform kann DMA eine beliebige Anzahl von Merkmalen unterstützen, die zur Optimierung einer PVA verwendet

werden, einschließlich, aber nicht beschränkt auf die Unterstützung mehrdimensionaler Adressierung und/oder zirkulärer Adressierung. In mindestens einer Ausführungsform kann DMA bis zu sechs oder mehr Dimensionen der Adressierung unterstützen, die ohne Einschränkung Blockbreite, Blockhöhe, Blocktiefe, horizontale Blockabstufung, vertikale Blockabstufung und/oder Tiefenabstufung umfassen können.

[0121] In mindestens einer Ausführungsform können Vektorprozessoren programmierbare Prozessoren sein, die so konzipiert sein können, dass sie die Programmierung von Computer-Vision-Algorithmen effizient und flexibel ausführen und Signalverarbeitungsfunktionen bereitstellen. In mindestens einer Ausführungsform kann eine PVA einen PVA-Kern und zwei Vektorverarbeitungs-Subsystem-Partitionen umfassen. In mindestens einer Ausführungsform kann ein PVA-Kern ein Prozessor-Subsystem, DMA-Engine(s) (z. B. zwei DMA-Engines) und/oder andere Peripheriegeräte umfassen. In mindestens einer Ausführungsform kann ein Vektorverarbeitungs-Subsystem als primäre Verarbeitungsmaschine einer PVA arbeiten und eine Vektorverarbeitungseinheit („VPU“), einen Befehls-cache und/oder einen Vektorspeicher (z. B. „VMEM“) umfassen. In mindestens einer Ausführungsform kann der VPU-Kern einen digitalen Signalprozessor enthalten, wie z. B. einen digitalen Signalprozessor mit einem einzigen Befehl und mehreren Daten („SIMD“) und einem sehr langen Befehlswort („VLIW“). In mindestens einer Ausführungsform kann eine Kombination aus SIMD und VLIW den Durchsatz und die Geschwindigkeit erhöhen.

[0122] In mindestens einer Ausführungsform kann jeder der Vektorprozessoren einen BefehlsCache enthalten und mit einem speziellen Speicher verbunden sein. Daher kann in mindestens einer Ausführungsform jeder der Vektorprozessoren so konfiguriert werden, dass er unabhängig von anderen Vektorprozessoren arbeitet. In mindestens einer Ausführungsform können Vektorprozessoren, die in einer bestimmten PVA enthalten sind, so konfiguriert sein, dass sie Datenparallelität verwenden. Zum Beispiel können in mindestens einer Ausführungsform mehrere Vektorprozessoren in einer einzigen PVA einen gemeinsamen Computer-Vision-Algorithmus ausführen, jedoch für unterschiedliche Bereiche eines Bildes. In mindestens einer Ausführungsform können die in einer bestimmten PVA enthaltenen Vektorprozessoren gleichzeitig verschiedene Bildverarbeitungsalgorithmen für ein Bild oder sogar verschiedene Algorithmen für aufeinander folgende Bilder oder Teile eines Bildes ausführen. In mindestens einer Ausführungsform kann unter anderem eine beliebige Anzahl von PVAs in einem Hardware-Beschleunigungscluster enthalten sein und jede PVA kann eine beliebige Anzahl von Vektorprozessoren enthalten. In mindestens einer Ausführungsform kann die PVA einen zusätzlichen Speicher mit Fehlerkorrekturcode („ECC“) enthalten, um die allgemeine Systemsicherheit zu erhöhen.

[0123] In mindestens einer Ausführungsform kann (können) der (die) Beschleuniger 1014 ein Computer-Vision-Netzwerk auf dem Chip und einen statischen Direktzugriffsspeicher („SRAM“) umfassen, um einen SRAM mit hoher Bandbreite und geringer Latenz für den (die) Beschleuniger 1014 bereitzustellen. In mindestens einer Ausführungsform kann der On-Chip-Speicher mindestens 4 MB SRAM umfassen, z. B. und ohne Einschränkung acht feldkonfigurierbare Speicherblöcke, auf die sowohl eine PVA als auch eine DLA zugreifen können. In mindestens einer Ausführungsform kann jedes Paar von Speicherblöcken eine erweiterte Peripheriebus-Schnittstelle (APB), Konfigurationsschaltungen, einen Controller und einen Multiplexer umfassen. In mindestens einer Ausführungsform kann jede Art von Speicher verwendet werden. In mindestens einer Ausführungsform können eine PVA und eine DLA über einen Backbone auf den Speicher zugreifen, der einer PVA und einer DLA einen Hochgeschwindigkeitszugriff auf den Speicher ermöglicht. In mindestens einer Ausführungsform kann ein Backbone ein Computer-Vision-Netzwerk auf dem Chip umfassen, das eine PVA und eine DLA mit dem Speicher verbindet (z. B. unter Verwendung von APB).

[0124] In mindestens einer Ausführungsform kann ein Computer-Vision-Netz auf dem Chip eine Schnittstelle enthalten, die vor der Übertragung von Steuersignalen/Adressen/Daten feststellt, dass sowohl eine PVA als auch eine DLA einsatzbereite und gültige Signale liefern. In mindestens einer Ausführungsform kann eine Schnittstelle getrennte Phasen und getrennte Kanäle für die Übertragung von Steuersignalen/Adressen/Daten sowie eine Burst-Kommunikation für die kontinuierliche Datenübertragung vorsehen. In mindestens einer Ausführungsform kann eine Schnittstelle den Normen der Internationalen Organisation für Normung („ISO“) 26262 oder der Internationalen Elektrotechnischen Kommission („IEC“) 61508 entsprechen, obwohl auch andere Normen und Protokolle verwendet werden können.

[0125] In mindestens einer Ausführungsform können ein oder mehrere SoC(s) 1004 einen Echtzeit-Raytracing-Hardwarebeschleuniger enthalten. In mindestens einer Ausführungsform kann ein Echtzeit-Strahlenverfolgungs-Hardwarebeschleuniger zur schnellen und effizienten Bestimmung von Positionen und Ausdehnungen von Objekten (z. B. innerhalb eines Weltmodells), zur Erzeugung von Echtzeit-Visualisierungssimulationen, zur RADAR-Signalinterpretation, zur Schallausbreitungssynthese und/oder

-analyse, zur Simulation von SONAR-Systemen, zur allgemeinen Wellenausbreitungssimulation, zum Vergleich mit LIDAR-Daten zum Zwecke der Lokalisierung und/oder anderer Funktionen und/oder für andere Zwecke verwendet werden.

[0126] In mindestens einer Ausführungsform können der oder die Beschleuniger 1014 für das autonome Fahren auf vielfältige Weise eingesetzt werden. In mindestens einer Ausführungsform kann eine PVA für wichtige Verarbeitungsschritte in ADAS und autonomen Fahrzeugen verwendet werden. In mindestens einer Ausführungsform eignen sich die Fähigkeiten einer PVA gut für algorithmische Bereiche, in denen eine vorhersehbare Verarbeitung mit geringem Stromverbrauch und niedriger Latenz erforderlich ist. Mit anderen Worten, eine PVA eignet sich gut für halbdichte oder dichte regelmäßige Berechnungen, selbst bei kleinen Datensätzen, die vorhersehbare Laufzeiten mit geringer Latenz und niedrigem Stromverbrauch erfordern können. In mindestens einer Ausführungsform, wie z. B. im Fahrzeug 1000, könnten PVAs so konzipiert sein, dass sie klassische Computer-Vision-Algorithmen ausführen, da sie bei der Objekterkennung und der Verarbeitung ganzzahliger mathematischer Daten effizient sind.

[0127] Zum Beispiel wird gemäß mindestens einer Ausführungsform der Technologie eine PVA verwendet, um Computer-Stereovision durchzuführen. In mindestens einer Ausführungsform kann in einigen Beispielen ein Algorithmus auf der Grundlage eines semiglobalen Abgleichs verwendet werden, wobei dies nicht als Einschränkung zu verstehen ist. In mindestens einer Ausführungsform verwenden Anwendungen für das autonome Fahren der Stufen 3 bis 5 Bewegungsschätzungen/Stereoabgleich während der Fahrt (z. B. Struktur aus Bewegung, Fußgängererkennung, Fahrspurerkennung usw.). In mindestens einer Ausführungsform kann eine PVA Computer-Stereo-Vision-Funktionen auf Eingaben von zwei monokularen Kameras ausführen.

[0128] In mindestens einer Ausführungsform kann eine PVA zur Durchführung eines dichten optischen Flusses verwendet werden. Zum Beispiel könnte eine PVA in mindestens einer Ausführungsform RADAR-Rohdaten verarbeiten (z. B. unter Verwendung einer 4D-Fast-FourierTransformation), um verarbeitete RADAR-Daten zu liefern. In mindestens einer Ausführungsform wird eine PVA für die Verarbeitung der Flugzeittiefe verwendet, z. B. durch Verarbeitung von Flugzeit-Rohdaten, um verarbeitete Flugzeitdaten zu erhalten.

[0129] In mindestens einer Ausführungsform kann ein DLA verwendet werden, um jede Art von Netzwerk zu betreiben, um die Kontrolle und die Fahrsicherheit zu verbessern, einschließlich beispielsweise und ohne Einschränkung ein neuronales Netzwerk, das ein Maß an Zuverlässigkeit für jede Objekterkennung ausgibt. In mindestens einer Ausführungsform kann die Zuverlässigkeit als Wahrscheinlichkeit dargestellt oder interpretiert werden, oder als relative „Gewichtung“ jeder Erkennung im Vergleich zu anderen Erkennungen. In mindestens einer Ausführungsform ermöglicht ein Zuverlässigkeitsmaß dem System, weitere Entscheidungen darüber zu treffen, welche Erkennungen als echte positive Erkennungen und welche als falsch-positive Erkennungen betrachtet werden sollten. In mindestens einer Ausführungsform kann ein System einen Schwellenwert für die Zuverlässigkeit festlegen und nur Erkennungen, die den Schwellenwert überschreiten, als echte positive Erkennungen betrachten. In einer Ausführungsform, in der ein automatisches Notbremsensystem („AEB“) verwendet wird, würden falsch positive Erkennungen dazu führen, dass das Fahrzeug automatisch eine Notbremsung durchführt, was natürlich unerwünscht ist. In mindestens einer Ausführungsform können sehr zuverlässige Erkennungen als Auslöser für AEB angesehen werden. In mindestens einer Ausführungsform kann eine DLA ein neuronales Netz zur Regression des Zuverlässigkeitswerts einsetzen. In mindestens einer Ausführungsform kann das neuronale Netz als Eingabe mindestens eine Teilmenge von Parametern verwenden, wie z. B. die Abmessungen des Begrenzungsrahmens, die (z. B. von einem anderen Teilsystem) erhaltene Bodenebenenschätzung, die Ausgabe des/der IMU-Sensors/en 1066, die mit der Ausrichtung des Fahrzeugs 1000 korreliert, die Entfernung, die 3D-Positionsschätzungen des Objekts, die vom neuronalen Netz und/oder anderen Sensoren (z. B. LIDAR-Sensor(en) 1064 oder RADAR-Sensor(en) 1060) erhalten werden, und andere.

[0130] In mindestens einer Ausführungsform können ein oder mehrere SoC(s) 1004 einen oder mehrere Datenspeicher 1016 (z. B. Speicher) enthalten. In mindestens einer Ausführungsform kann es sich bei dem/den Datenspeicher(n) 1016 um einen On-Chip-Speicher des/der SoC(s) 1004 handeln, in dem neuronale Netze gespeichert werden können, die auf der/den GPU(s) 1008 und/oder einer DLA ausgeführt werden. In mindestens einer Ausführungsform kann die Kapazität des/der Datenspeicher(s) 1016 groß genug sein, um mehrere Instanzen neuronaler Netze aus Gründen der Redundanz und Sicherheit zu speichern. In mindestens einer Ausführungsform kann (können) der (die) Datenspeicher 1016 L2 oder L3 Cache(s) umfassen.

[0131] In mindestens einer Ausführungsform können ein oder mehrere SoC(s) 1004 eine beliebige Anzahl von Prozessoren 1010 (z. B. eingebettete Prozessoren) enthalten. In mindestens einer Ausführungsform können der/die Prozessor(en) 1010 einen Boot- und Energieverwaltungsprozessor enthalten, bei dem es sich um einen speziellen Prozessor und ein Subsystem zur Verwaltung der Boot-Energie und der Verwaltungsfunktionen sowie zur Durchsetzung der damit verbundenen Sicherheit handeln kann. In mindestens einer Ausführungsform kann ein Boot- und Energieverwaltungsprozessor Teil einer Boot-Sequenz von SoC(s) 1004 sein und Laufzeit-Energieverwaltungsdienste bereitstellen. In mindestens einer Ausführungsform kann ein Prozessor für die Boot-Energieversorgung und -Verwaltung Takt- und Spannungsprogrammierung, Unterstützung bei Übergängen in einen Zustand mit geringem Stromverbrauch, Verwaltung von SoC(s) 1004-Temperaturen und Temperatursensoren und/oder Verwaltung von SoC(s) 1004-Energiezuständen bereitstellen. In mindestens einer Ausführungsform kann jeder Temperatursensor als Ringoszillator implementiert sein, dessen Ausgangsfrequenz proportional zur Temperatur ist, und SoC(s) 1004 kann/können Ringoszillatoren verwenden, um Temperaturen von CPU(s) 1006, GPU(s) 1008 und/oder Beschleuniger(n) 1014 zu erfassen. In mindestens einer Ausführungsform kann ein Boot- und Energieverwaltungsprozessor in eine Temperaturfehleroutine eintreten, wenn festgestellt wird, dass die Temperaturen einen Schwellenwert überschreiten, und die SoC(s) 1004 in einen Zustand mit geringerer Leistung versetzen und/oder das Fahrzeug 1000 in einen Modus Fahrer zum sicheren Halt bringen (z. B. das Fahrzeug 1000 zu einem sicheren Halt bringen).

[0132] In mindestens einer Ausführungsform kann (können) der (die) Prozessor(en) 1010 darüber hinaus eine Reihe eingebetteter Prozessoren enthalten, die als Audioverarbeitungsmaschine dienen können, bei der es sich um ein Audio-Subsystem handeln kann, das eine vollständige Hardware-Unterstützung für Mehrkanal-Audio über mehrere Schnittstellen und eine breite und flexible Palette von Audio-E/A-Schnittstellen ermöglicht. In mindestens einer Ausführungsform handelt es sich bei der Audioverarbeitungsmaschine um einen speziellen Prozessorkern mit einem digitalen Signalprozessor mit eigenem RAM.

[0133] In mindestens einer Ausführungsform kann (können) der (die) Prozessor(en) 1010 außerdem eine ständig eingeschaltete Prozessormaschine enthalten, die die erforderlichen Hardware-Funktionen zur Unterstützung der Sensorverwaltung mit geringem Stromverbrauch und des Aufweckens von Anwendungsfällen bereitstellen kann. In mindestens einer Ausführungsform kann eine ständig eingeschaltete Prozessormaschine unter anderem einen Prozessorkern, einen eng gekoppelten RAM-Speicher, unterstützende Peripheriegeräte (z. B. Zeitgeber und Unterbrechungssteuerungen), verschiedene E/A-Steuerungsperipheriegeräte und eine Leitweglogik umfassen.

[0134] In mindestens einer Ausführungsform kann (können) der (die) Prozessor(en) 1010 außerdem eine Sicherheits-Cluster-Maschine enthalten, die ohne Einschränkung ein spezielles Prozessor-Subsystem für das Sicherheitsmanagement von Automobilanwendungen umfasst. In mindestens einer Ausführungsform kann eine Sicherheits-Cluster-Maschine ohne Einschränkung zwei oder mehr Prozessorkerne, einen eng gekoppelten Arbeitsspeicher, unterstützende Peripheriegeräte (z. B. Zeitgeber, eine Unterbrechungssteuerung usw.) und/oder Leitweglogik umfassen. In einem Sicherheitsmodus können zwei oder mehr Kerne in mindestens einer Ausführungsform in einem Lockstep-Modus arbeiten und als ein einziger Kern mit einer Vergleichslogik zur Erkennung von Unterschieden zwischen ihren Vorgängen funktionieren. In mindestens einer Ausführungsform kann (können) der (die) Prozessor(en) 1010 außerdem eine Echtzeit-Kameramaschine enthalten, die ohne Einschränkung ein spezielles Prozessor-Subsystem für die Verwaltung von Echtzeit-Kameras umfassen kann. In mindestens einer Ausführungsform kann (können) der (die) Prozessor(en) 1010 außerdem einen Signalprozessor mit hohem Dynamikbereich umfassen, der ohne Einschränkung einen Bildsignalprozessor umfassen kann, der eine Hardwaremaschine ist, die Teil einer Kameraverarbeitungspipeline ist.

[0135] In mindestens einer Ausführungsform kann (können) der (die) Prozessor(en) 1010 einen Videobildkompositor enthalten, der ein Verarbeitungsblock sein kann (z. B. auf einem Mikroprozessor implementiert), der Videonachbearbeitungsfunktionen implementiert, die von einer Videowiedergabeanwendung benötigt werden, um ein endgültiges Bild für ein Player-Fenster zu erzeugen. In mindestens einer Ausführungsform kann ein Videobildkompositor eine Linsenverzerrungskorrektur an der (den) Weitwinkelkamera(s) 1070, der (den) Surround-Kamera(s) 1074 und/oder an den Sensoren der Überwachungskamera(s) in der Kabine vornehmen. In mindestens einer Ausführungsform werden die Kamerasensoren zur Überwachung in der Kabine vorzugsweise von einem neuronalen Netz überwacht, das auf einer anderen Instanz des SoC 1004 läuft und so konfiguriert ist, dass es Ereignisse in der Kabine erkennt und entsprechend reagiert. In mindestens einer Ausführungsform kann ein System im Fahrzeuginnenen ohne Einschränkung Lippenlesen durchführen, um den Mobilfunkdienst zu aktivieren und einen Telefonanruf zu tätigen, E-Mails zu diktieren, den Zielort eines Fahrzeugs zu ändern, das Infotainmentsystem und die Einstellungen eines Fahrzeugs zu aktivieren oder zu

ändern oder sprachgesteuertes Surfen im Internet zu ermöglichen. In mindestens einer Ausführungsform stehen dem Fahrer bestimmte Funktionen zur Verfügung, wenn das Fahrzeug in einem autonomen Modus betrieben wird, und sind ansonsten deaktiviert.

[0136] In mindestens einer Ausführungsform kann ein Videobild-Compositor eine verbesserte zeitliche Rauschunterdrückung sowohl für die räumliche als auch für die zeitliche Rauschunterdrückung enthalten. In mindestens einer Ausführungsform werden beispielsweise bei Bewegung in einem Video die räumlichen Informationen durch die Rauschunterdrückung entsprechend gewichtet, wobei die Gewichtung der Informationen benachbarter Bilder verringert wird. In mindestens einer Ausführungsform, in der ein Bild oder ein Teil eines Bildes keine Bewegung enthält, kann die vom Videobild-Compositor durchgeführte zeitliche Rauschunterdrückung Informationen aus einem früheren Bild verwenden, um das Rauschen im aktuellen Bild zu reduzieren.

[0137] In mindestens einer Ausführungsform kann ein Videobild-Compositor auch so konfiguriert sein, dass er eine Stereo-Entzerrung der eingegebenen Stereo-Linsenbilder durchführt. In mindestens einer Ausführungsform kann ein Videobild-Compositor auch für die Gestaltung der Benutzeroberfläche verwendet werden, wenn ein Betriebssystem-Desktop in Gebrauch ist und die GPU(s) 1008 nicht ständig neue Oberflächen wiedergeben müssen. In mindestens einer Ausführungsform kann, wenn die GPU(s) 1008 eingeschaltet sind und aktiv 3D-Wiedergabe betreiben, ein Videobild-Compositor verwendet werden, um die GPU(s) 1008 zu entlasten und die Leistung und Reaktionsfähigkeit zu verbessern.

[0138] In mindestens einer Ausführungsform können ein oder mehrere SoC von SoC(s) 1004 außerdem eine serielle MIPI-Kameraschnittstelle für den Empfang von Video und Eingaben von Kameras, eine Hochgeschwindigkeitsschnittstelle und/oder einen Videoeingabeblock enthalten, der für eine Kamera und damit verbundene Pixeleingabefunktionen verwendet werden kann. In mindestens einer Ausführungsform können ein oder mehrere SoC(s) 1004 außerdem einen oder mehrere Eingangs-/Ausgangs-Controller enthalten, die durch Software gesteuert werden können und für den Empfang von E/A-Signalen verwendet werden können, die keiner bestimmten Rolle zugeordnet sind.

[0139] In mindestens einer Ausführungsform können ein oder mehrere SoC(s) 1004 außerdem eine breite Palette von Peripherieschnittstellen enthalten, um die Kommunikation mit Peripheriegeräten, Audio-Encodern/Decodern („Codecs“), Energieverwaltung und/oder anderen Geräten zu ermöglichen. In mindestens einer Ausführungsform können SoC(s) 1004 verwendet werden, um Daten von Kameras (z. B. über Gigabit Multimedia Serial Link und Ethernet-Kanäle verbunden), Sensoren (z. B., LIDAR-Sensor(en) 1064, RADAR-Sensor(en) 1060 usw., die über Ethernet-Kanäle angeschlossen sein können), Daten von Bus 1002 (z. B. Geschwindigkeit des Fahrzeugs 1000, Lenkradposition usw.), Daten von GNSS-Sensor(en) 1058 (z. B. über einen Ethernet-Bus oder einen CAN-Bus angeschlossen) usw. In mindestens einer Ausführungsform können ein oder mehrere SoC von SoC(s) 1004 außerdem dedizierte Hochleistungs-Massenspeicher-Controller enthalten, die ihre eigenen DMA-Engines enthalten können und dazu verwendet werden können, die CPU(s) 1006 von Routineaufgaben der Datenverwaltung zu entlasten.

[0140] In mindestens einer Ausführungsform kann (können) der (die) SoC 1004 eine End-to-End-Plattform mit einer flexiblen Architektur sein, die die Automatisierungsstufen 3 bis 5 umfasst und dadurch eine umfassende funktionale Sicherheitsarchitektur bereitstellt, die Computer-Vision- und ADAS-Techniken für Diversität und Redundanz nutzt und eine Plattform für einen flexiblen, zuverlässigen Fahrsoftware-Stack zusammen mit Deep-Learning-Tools bereitstellt. In mindestens einer Ausführungsform können die SoC(s) 1004 schneller, zuverlässiger und sogar energie- und platzsparender sein als herkömmliche Systeme. Zum Beispiel können in mindestens einer Ausführungsform der oder die Beschleuniger 1014 in Kombination mit CPU(s) 1006, GPU(s) 1008 und Datenspeicher(n) 1016 eine schnelle, effiziente Plattform für autonome Fahrzeuge der Stufe 3-5 bilden.

[0141] In mindestens einer Ausführungsform können Computer-Vision-Algorithmen auf CPUs ausgeführt werden, die mit Hilfe einer höheren Programmiersprache, wie z. B. C, so konfiguriert werden können, dass sie eine Vielzahl von Verarbeitungsalgorithmen für eine Vielzahl visueller Daten ausführen können. In mindestens einer Ausführungsform sind CPUs jedoch oft nicht in der Lage, die Leistungsanforderungen vieler Bildverarbeitungsanwendungen zu erfüllen, z. B. in Bezug auf die Ausführungszeit und den Stromverbrauch. In mindestens einer Ausführungsform sind viele CPUs nicht in der Lage, komplexe Objekterkennungsalgorithmen in Echtzeit auszuführen, die in fahrzeuginternen ADAS-Anwendungen und in praktischen autonomen Fahrzeugen der Stufe 3-5 verwendet werden.

[0142] Die hier beschriebenen Ausführungsformen ermöglichen es, mehrere neuronale Netze gleichzeitig und/oder nacheinander auszuführen und die Ergebnisse miteinander zu kombinieren, um autonome Fahrfunktionen der Stufe 3-5 zu ermöglichen. Zum Beispiel kann in mindestens einer Ausführungsform ein CNN, das auf einem DLA oder einer diskreten GPU (z. B. GPU(s) 1020) ausgeführt wird, eine Text- und Worterkennung beinhalten, die das Lesen und Verstehen von Verkehrszeichen ermöglicht, einschließlich Zeichen, für die ein neuronales Netz nicht speziell trainiert wurde. In mindestens einer Ausführungsform kann ein DLA außerdem ein neuronales Netz enthalten, das in der Lage ist, ein Zeichen zu identifizieren, zu interpretieren und semantisch zu verstehen und dieses semantische Verständnis an Pfadplanungsmodule weiterzugeben, die auf einem CPU-Komplex laufen.

[0143] In mindestens einer Ausführungsform können mehrere neuronale Netze gleichzeitig betrieben werden, z. B. für die Fahrstufen 3, 4 oder 5. Zum Beispiel kann in mindestens einer Ausführungsform ein Warnschild mit der Aufschrift „Vorsicht: Blinklicht weist auf Vereisung hin“ zusammen mit einem elektrischen Licht unabhängig oder gemeinsam von mehreren neuronalen Netzen interpretiert werden. In mindestens einer Ausführungsform kann ein solches Warnschild selbst von einem ersten eingesetzten neuronalen Netz (z. B. einem trainierten neuronalen Netz) als Verkehrszeichen identifiziert werden, und der Text „Blinklicht weist auf Glatteis hin“ kann von einem zweiten eingesetzten neuronalen Netz interpretiert werden, das die Wegplanungssoftware eines Fahrzeugs (die vorzugsweise auf einem CPU-Komplex ausgeführt wird) darüber informiert, dass Glatteis vorliegt, wenn Blinklicht erkannt wird. In mindestens einer Ausführungsform kann ein Blinklicht identifiziert werden, indem ein drittes eingesetztes neuronales Netz über mehrere Frames hinweg betrieben wird und die Wegplanungssoftware eines Fahrzeugs über das Vorhandensein (oder Nichtvorhandensein) von Blinklichtern informiert. In mindestens einer Ausführungsform können alle drei neuronalen Netze gleichzeitig laufen, z. B. innerhalb einer DLA und/oder auf GPU(s) 1008.

[0144] In mindestens einer Ausführungsform kann ein CNN zur Gesichtserkennung und Identifizierung des Fahrzeugbesitzers Daten von Kamerasensoren verwenden, um die Anwesenheit eines autorisierten Fahrers und/oder Besitzers des Fahrzeugs 1000 zu identifizieren. In mindestens einer Ausführungsform kann eine ständig eingeschaltete Sensorverarbeitungsmaschine dazu verwendet werden, ein Fahrzeug zu entriegeln, wenn sich ein Besitzer der Fahrertür nähert und das Licht einschaltet, und in einem Sicherheitsmodus das Fahrzeug zu deaktivieren, wenn ein Besitzer das Fahrzeug verlässt. Auf diese Weise sorgen die SoC(s) 1004 für Sicherheit gegen Diebstahl und/oder Carjacking.

[0145] In mindestens einer Ausführungsform kann ein CNN zur Erkennung und Identifizierung von Einsatzfahrzeugen Daten von Mikrofonen 1096 verwenden, um Sirenen von Einsatzfahrzeugen zu erkennen und zu identifizieren. In mindestens einer Ausführungsform verwenden die SoC(s) 1004 ein CNN zur Klassifizierung von Umwelt- und Stadtgeräuschen sowie zur Klassifizierung visueller Daten. In mindestens einer Ausführungsform wird ein CNN, das auf einer DLA läuft, darauf trainiert, eine relative Annäherungsgeschwindigkeit eines Einsatzfahrzeugs zu erkennen (z. B. durch Nutzung des Dopplereffekts). In mindestens einer Ausführungsform kann ein CNN auch darauf trainiert werden, Einsatzfahrzeuge zu identifizieren, die für ein lokales Gebiet, in dem ein Fahrzeug unterwegs ist, spezifisch sind und von GNSS-Sensor(en) 1058 identifiziert werden. In mindestens einer Ausführungsform versucht ein CNN, wenn es in Europa eingesetzt wird, europäische Sirenen zu erkennen, und wenn es in Nordamerika eingesetzt wird, versucht ein CNN, nur nordamerikanische Sirenen zu erkennen. In mindestens einer Ausführungsform kann ein Steuerprogramm, sobald ein Einsatzfahrzeug erkannt wird, dazu verwendet werden, eine Sicherheitsroutine für Einsatzfahrzeuge auszuführen, ein Fahrzeug abzubremesen, an den Straßenrand zu fahren, ein Fahrzeug zu parken und/oder ein Fahrzeug im Leerlauf laufen zu lassen, bis die Einsatzfahrzeuge vorbeigefahren sind, und zwar mit Hilfe des/der Ultraschallsensoren 1062.

[0146] In mindestens einer Ausführungsform kann das Fahrzeug 1000 CPU(s) 1018 (z. B. diskrete CPU(s) oder dCPU(s)) enthalten, die über eine Hochgeschwindigkeitsverbindung (z. B. PCIe) mit SoC(s) 1004 gekoppelt sein können. In mindestens einer Ausführungsform kann (können) die CPU(s) 1018 z. B. einen X86-Prozessor umfassen. Die CPU(s) 1018 kann/können verwendet werden, um eine Vielzahl von Funktionen auszuführen, einschließlich der Schlichtung potenziell widersprüchlicher Ergebnisse zwischen den ADAS-Sensoren und dem/den SoC(s) 1004 und/oder der Überwachung des Status und des Zustands des/der Controller(s) 1036 und/oder eines Infotainment-Systems auf einem Chip („Infotainment-SoC“) 1030, zum Beispiel.

[0147] In mindestens einer Ausführungsform kann das Fahrzeug 1000 GPU(s) 1020 (z.B. diskrete GPU(s) oder dGPU(s)) enthalten, die über eine Hochgeschwindigkeitsverbindung (z.B. NVIDIAs NVLINK-Kanal) mit SoC(s) 1004 gekoppelt sein können. In mindestens einer Ausführungsform kann/können GPU(s) 1020

zusätzliche Funktionen der künstlichen Intelligenz bereitstellen, z. B. durch Ausführen redundanter und/oder unterschiedlicher neuronaler Netze, und kann/können verwendet werden, um neuronale Netze zu trainieren und/oder zu aktualisieren, die zumindest teilweise auf Eingaben (z. B. Sensordaten) von Sensoren eines Fahrzeugs 1000 basieren.

[0148] In mindestens einer Ausführungsform kann das Fahrzeug 1000 außerdem eine Netzwerkschnittstelle 1024 enthalten, die ohne Einschränkung eine oder mehrere drahtlose Antennen 1026 (z. B. eine oder mehrere drahtlose Antennen für verschiedene Kommunikationsprotokolle, wie eine Mobilfunkantenne, eine Bluetooth-Antenne usw.) umfassen kann. In mindestens einer Ausführungsform kann die Netzwerkschnittstelle 1024 verwendet werden, um eine drahtlose Verbindung zu Internet-Cloud-Diensten (z. B. mit Servern und/oder anderen Netzwerkgeräten), mit anderen Fahrzeugen und/oder mit Computergeräten (z. B. Client-Geräten von Fahrgästen) zu ermöglichen. In mindestens einer Ausführungsform kann zur Kommunikation mit anderen Fahrzeugen eine direkte Verbindung zwischen Fahrzeug 1000 und einem anderen Fahrzeug und/oder eine indirekte Verbindung (z. B. über Netzwerke und das Internet) hergestellt werden. In mindestens einer Ausführungsform können direkte Verbindungen über eine Fahrzeug-zu-Fahrzeug-Kommunikationsverbindung hergestellt werden. In mindestens einer Ausführungsform kann eine Fahrzeug-zu-Fahrzeug-Kommunikationsverbindung dem Fahrzeug 1000 Informationen über Fahrzeuge in der Nähe des Fahrzeugs 1000 liefern (z. B. Fahrzeuge vor, auf einer Seite und/oder hinter dem Fahrzeug 1000). In mindestens einer Ausführungsform kann diese Funktionalität Teil einer kooperativen adaptiven Geschwindigkeitsregelungsfunktion des Fahrzeugs 1000 sein.

[0149] In mindestens einer Ausführungsform kann die Netzwerkschnittstelle 1024 einen SoC enthalten, der Modulations- und Demodulationsfunktionen bereitstellt und es dem/den Controller(n) 1036 ermöglicht, über drahtlose Netzwerke zu kommunizieren. In mindestens einer Ausführungsform kann die Netzwerkschnittstelle 1024 ein Hochfrequenz-Frontend für die Aufwärtskonvertierung von Basisband auf Hochfrequenz und die Abwärtskonvertierung von Hochfrequenz auf Basisband enthalten. In mindestens einer Ausführungsform kann die Frequenzumwandlung auf jede technisch mögliche Weise erfolgen. Die Frequenzumwandlung könnte zum Beispiel mit bekannten Verfahren und/oder mit Superheterodyn-Verfahren erfolgen.

[0150] In mindestens einer Ausführungsform kann die Hochfrequenz-Front-End-Funktionalität durch einen separaten Chip bereitgestellt werden. In mindestens einer Ausführungsform können die Netzwerkschnittstellen drahtlose Funktionen für die Kommunikation über LTE, WCDMA, UMTS, GSM, CDMA2000, Bluetooth, Bluetooth LE, Wi-Fi, Z-Wave, ZigBee, LoRaWAN und/oder andere drahtlose Protokolle umfassen.

[0151] In mindestens einer Ausführungsform kann das Fahrzeug 1000 außerdem Datenspeicher 1028 enthalten, die ohne Einschränkung auch außerhalb des Chips (z. B. außerhalb der SoC(s) 1004) gespeichert werden können. In mindestens einer Ausführungsform kann/können der/die Datenspeicher 1028 ohne Einschränkung ein oder mehrere Speicherelemente umfassen, darunter RAM, SRAM, dynamischer Direktzugriffsspeicher („DRAM“), Video-Direktzugriffsspeicher („VRAM“), Flash-Speicher, Festplatten und/oder andere Komponenten und/oder Geräte, die mindestens ein Datenbit speichern können.

[0152] In mindestens einer Ausführungsform kann das Fahrzeug 1000 außerdem GNSS-Sensor(en) 1058 (z. B. GPS- und/oder unterstützte GPS-Sensoren) enthalten, um bei der Kartierung, der Wahrnehmung, der Erstellung von Belegungsrastern und/oder den Pfadplanungsfunktionen zu helfen. In mindestens einer Ausführungsform kann eine beliebige Anzahl von GNSS-Sensoren 1058 verwendet werden, z. B. und ohne Einschränkung ein GPS, das einen Universal Serial Bus („USB“)-Anschluss mit einer Ethernet-to-Serial (z. B. RS-232)-Brücke verwendet.

[0153] In mindestens einer Ausführungsform kann das Fahrzeug 1000 außerdem RADAR-Sensor(en) 1060 enthalten. In mindestens einer Ausführungsform können der oder die RADAR-Sensoren 1060 vom Fahrzeug 1000 zur Fahrzeugerkennung selbst bei Dunkelheit und/oder schlechten Witterungsbedingungen über große Entfernungen verwendet werden. In mindestens einer Ausführungsform können die RADAR-Funktionssicherheitsstufen ASIL B sein. In mindestens einer Ausführungsform können die RADAR-Sensoren 1060 einen CAN-Bus und/oder Bus 1002 (z. B. zur Übertragung von Daten, die von den RADAR-Sensoren 1060 erzeugt wurden) zur Steuerung und zum Zugriff auf Objektverfolgungsdaten verwenden, wobei in einigen Beispielen der Zugriff auf Ethernet-Kanäle für den Zugriff auf Rohdaten möglich ist. In mindestens einer Ausführungsform kann eine breite Palette von RADAR-Sensortypen verwendet werden. Zum Beispiel, und ohne Einschränkung, können RADAR-Sensor(en) 1060 für die Verwendung von Front-, Heck- und Seiten-RADAR geeignet sein. In mindestens einer Ausführungsform sind ein oder mehrere Sensoren der RADAR-Sensoren 1060 ein Puls-Doppler-RADAR-Sensor.

[0154] In mindestens einer Ausführungsform kann (können) der (die) RADAR-Sensor(en) 1060 verschiedene Konfigurationen umfassen, z. B. mit großer Reichweite und engem Sichtfeld, mit geringer Reichweite und breitem Sichtfeld, mit seitlicher Abdeckung im Nahbereich usw. In mindestens einer Ausführungsform kann RADAR mit großer Reichweite für die adaptive Geschwindigkeitsregelung verwendet werden. In mindestens einer Ausführungsform können RADAR-Systeme mit großer Reichweite ein breites Sichtfeld bieten, das durch zwei oder mehr unabhängige Abtastungen, z. B. innerhalb eines Bereichs von 250 m (Meter), realisiert wird. In mindestens einer Ausführungsform kann/können der/die RADAR-Sensor(en) 1060 dabei helfen, zwischen statischen und sich bewegendenden Objekten zu unterscheiden, und kann/können vom ADAS-System 1038 zur Notbremsunterstützung und zur Vorwärtskollisionswarnung verwendet werden. In mindestens einer Ausführungsform können die Sensoren 1060, die in einem RADAR-System mit großer Reichweite enthalten sind, ohne Einschränkung ein monostatisches multimodales RADAR mit mehreren (z. B. sechs oder mehr) festen RADAR-Antennen und einer Hochgeschwindigkeits-CAN- und FlexRay-Schnittstelle umfassen. Bei mindestens einer Ausführung mit sechs Antennen können vier Antennen in der Mitte ein fokussiertes Strahlungsmuster erzeugen, das dazu dient, die Umgebung des Fahrzeugs 1000 bei höheren Geschwindigkeiten mit minimalen Störungen durch den Verkehr auf den angrenzenden Fahrspuren zu erfassen. In mindestens einer Ausführungsform kann das Sichtfeld durch zwei weitere Antennen erweitert werden, so dass Fahrzeuge, die in eine Fahrspur des Fahrzeugs 1000 einfahren oder diese verlassen, schnell erkannt werden können.

[0155] In mindestens einer Ausführungsform können RADAR-Systeme mittlerer Reichweite beispielsweise eine Reichweite von bis zu 160 m (vorne) oder 80 m (hinten) und ein Sichtfeld von bis zu 42 Grad (vorne) oder 150 Grad (hinten) aufweisen. In mindestens einer Ausführungsform können Kurzstrecken-RADAR-Systeme ohne Einschränkung eine beliebige Anzahl von RADAR-Sensoren 1060 umfassen, die an beiden Enden eines hinteren Stoßfängers installiert werden können. Wenn ein RADAR-Sensorsystem an beiden Enden eines hinteren Stoßfängers installiert ist, kann es in mindestens einer Ausführungsform zwei Strahlen erzeugen, die ständig tote Winkel in Richtung Heck und neben dem Fahrzeug überwachen. In mindestens einer Ausführungsform können RADAR-Systeme mit geringer Reichweite im ADAS-System 1038 zur Erkennung des toten Winkels und/oder zur Unterstützung beim Spurwechsel verwendet werden.

[0156] In mindestens einer Ausführungsform kann das Fahrzeug 1000 außerdem Ultraschallsensor(en) 1062 enthalten. In mindestens einer Ausführungsform kann/können der/die Ultraschallsensor(en) 1062, der/-die an einer vorderen, hinteren und/oder seitlichen Stelle des Fahrzeugs 1000 positioniert sein kann/können, zur Einparkhilfe und/oder zur Erstellung und Aktualisierung eines Belegungsrasters verwendet werden. In mindestens einer Ausführungsform kann eine Vielzahl von Ultraschallsensoren 1062 verwendet werden, und verschiedene Ultraschallsensoren 1062 können für unterschiedliche Erfassungsbereiche (z. B. 2,5 m, 4 m) eingesetzt werden. In mindestens einer Ausführungsform können der/die Ultraschallsensor(en) 1062 bei funktionalen Sicherheitsstufen von ASIL B arbeiten.

[0157] In mindestens einer Ausführungsform kann das Fahrzeug 1000 LIDAR-Sensor(en) 1064 enthalten. In mindestens einer Ausführungsform kann/können der/die LIDAR-Sensor(en) 1064 zur Objekt- und Fußgängererkennung, Notbremsung, Kollisionsvermeidung und/oder für andere Funktionen verwendet werden. In mindestens einer Ausführungsform kann/können der/die LIDAR-Sensor(en) 1064 auf der funktionalen Sicherheitsstufe ASIL B betrieben werden. In mindestens einer Ausführungsform kann das Fahrzeug 1000 mehrere LIDAR-Sensoren 1064 (z. B. zwei, vier, sechs usw.) umfassen, die einen Ethernet-Kanal verwenden können (z. B. zur Bereitstellung von Daten an einen Gigabit-Ethernet-Switch).

[0158] In mindestens einer Ausführungsform kann/können der/die LIDAR-Sensor(en) 1064 in der Lage sein, eine Liste von Objekten und deren Entfernungen für ein 360-Grad-Sichtfeld zu liefern. In mindestens einer Ausführungsform kann/können der/die handelsübliche(n) LIDAR-Sensor(en) 1064 eine angegebene Reichweite von etwa 100 m haben, mit einer Genauigkeit von 2 cm bis 3 cm und mit Unterstützung für eine Ethernet-Verbindung mit 100 Mbit/s, zum Beispiel. In mindestens einer Ausführungsform können ein oder mehrere nicht vorspringende LIDAR-Sensoren verwendet werden. In einer solchen Ausführungsform kann (können) der (die) LIDAR-Sensor(en) 1064 ein kleines Gerät umfassen, das in eine Front-, Heck-, Seiten- und/oder Eckposition des Fahrzeugs 1000 eingebettet werden kann. In mindestens einer Ausführungsform kann/können der/die LIDAR-Sensor(en) 1064 ein horizontales Sichtfeld von bis zu 120° und ein vertikales Sichtfeld von bis zu 35° mit einer Reichweite von 200 m selbst für Objekte mit geringem Reflexionsvermögen bieten. In mindestens einer Ausführungsform kann/können der/die an der Vorderseite montierte(n) LIDAR-Sensor(en) 1064 für ein horizontales Sichtfeld zwischen 45° und 135° konfiguriert werden.

[0159] In mindestens einer Ausführungsform können auch LIDAR-Technologien, wie z. B. 3D-Blitz-LIDAR, verwendet werden. In mindestens einer Ausführungsform verwendet 3D-Blitz-LIDAR einen Laserblitz als Übertragungsquelle, um die Umgebung des Fahrzeugs 1000 bis zu einer Entfernung von etwa 200 m zu beleuchten. In mindestens einer Ausführungsform umfasst eine Blitz-LIDAR-Einheit ohne Einschränkung einen Rezeptor, der die Laufzeit des Laserpulses und das reflektierte Licht an jedem Pixel aufzeichnet, was wiederum einer Entfernung von Fahrzeug 1000 zu Objekten entspricht. In mindestens einer Ausführungsform kann das Blitz-LIDAR es ermöglichen, mit jedem Laserblitz hochpräzise und verzerrungsfreie Bilder der Umgebung zu erzeugen. In mindestens einer Ausführungsform können vier Blitz-LIDAR-Sensoren eingesetzt werden, einer auf jeder Seite des Fahrzeugs 1000. In mindestens einer Ausführungsform umfassen 3D-Blitz-LIDAR-Systeme ohne Einschränkung eine 3D-Star-Array-LIDAR-Festkörperkamera, die außer einem Gebläse keine beweglichen Teile aufweist (z. B. ein nicht abtastendes LIDAR-Gerät). In mindestens einer Ausführungsform kann das Flash-LIDAR-Gerät einen 5-Nanosekunden-Laserimpuls der Klasse I (augensicher) pro Bild verwenden und das reflektierte Laserlicht als 3D-Entfernungspunktwolke und koregistrierte Intensitätsdaten erfassen.

[0160] In mindestens einer Ausführungsform kann das Fahrzeug 1000 außerdem IMU-Sensor(en) 1066 enthalten. In mindestens einer Ausführungsform kann/können der/die IMU-Sensor(en) 1066 in der Mitte der Hinterachse des Fahrzeugs 1000 angeordnet sein. In mindestens einer Ausführungsform kann/können der/die IMU-Sensor(en) 1066 z. B. und ohne Einschränkung Beschleunigungsmesser, Magnetometer, Gyroskop(e), einen Magnetkompass, Magnetkompass und/oder andere Sensortypen umfassen. In mindestens einer Ausführungsform, z. B. bei sechsachsigen Anwendungen, kann/können der/die IMU-Sensor(en) 1066 unter anderem Beschleunigungsmesser und Gyroskope umfassen. In mindestens einer Ausführungsform, z. B. bei neunsachsigen Anwendungen, kann/können der/die IMU-Sensor(en) 1066 unter anderem Beschleunigungsmesser, Gyroskope und Magnetometer umfassen.

[0161] In mindestens einer Ausführungsform kann/können der/die IMU-Sensor(en) 1066 als ein miniaturisiertes, hochleistungsfähiges GPS-gestütztes Trägheitsnavigationssystem („GPS/INS“) implementiert werden, das mikroelektromechanische Systeme („MEMS“) mit Trägheitssensoren, einem hochempfindlichen GPS-Empfänger und fortschrittlichen Kalman-Filteralgorithmen kombiniert, um Schätzungen von Position, Geschwindigkeit und Lage zu liefern. In mindestens einer Ausführungsform können der/die IMU-Sensor(en) 1066 das Fahrzeug 1000 in die Lage versetzen, seinen Kurs zu ermitteln, ohne dass eine Eingabe von einem Magnetsensor erforderlich ist, indem Änderungen der Geschwindigkeit von einem GPS direkt beobachtet und mit dem/den IMU-Sensor(en) 1066 korreliert werden. In mindestens einer Ausführungsform können IMU-Sensor(en) 1066 und GNSS-Sensor(en) 1058 in einer einzigen integrierten Einheit kombiniert werden.

[0162] In mindestens einer Ausführungsform kann das Fahrzeug 1000 ein oder mehrere Mikrofone 1096 enthalten, die im und/oder am Fahrzeug 1000 angebracht sind. In mindestens einer Ausführungsform kann/können das/die Mikrophon(e) 1096 u. a. zur Erkennung und Identifizierung von Einsatzfahrzeugen verwendet werden.

[0163] In mindestens einer Ausführungsform kann das Fahrzeug 1000 außerdem eine beliebige Anzahl von Kameratypen enthalten, einschließlich Stereokamera(s) 1068, Weitwinkelkamera(s) 1070, Infrarotkamera(s) 1072, Umgebungskamera(s) 1074, Fernkamera(s) 1098, Mittelbereichskamera(s) 1076 und/oder andere Kameratypen. In mindestens einer Ausführungsform können Kameras verwendet werden, um Bilddaten rund um den gesamten Umfang des Fahrzeugs 1000 zu erfassen. In mindestens einer Ausführungsform hängt es vom Fahrzeug 1000 ab, welche Arten von Kameras verwendet werden. In mindestens einer Ausführungsform kann eine beliebige Kombination von Kameratypen verwendet werden, um die erforderliche Abdeckung rund um das Fahrzeug 1000 zu gewährleisten. In mindestens einer Ausführungsform kann die Anzahl der eingesetzten Kameras je nach Ausführungsform unterschiedlich sein. In mindestens einer Ausführungsform könnte das Fahrzeug 1000 zum Beispiel sechs, sieben, zehn, zwölf oder eine andere Anzahl von Kameras enthalten. In mindestens einer Ausführungsform können die Kameras beispielsweise und ohne Einschränkung Gigabit Multimedia Serial Link („GMSL“) und/oder Gigabit Ethernet-Kommunikation unterstützen. In mindestens einer Ausführungsform kann jede Kamera so beschaffen sein, wie dies zuvor in Bezug auf **Fig. 10 A** und **Fig. 10B** beschrieben.

[0164] In mindestens einer Ausführungsform kann das Fahrzeug 1000 außerdem Vibrationssensor(en) 1042 enthalten. In mindestens einer Ausführungsform können der/die Vibrationssensor(en) 1042 Vibrationen von Komponenten des Fahrzeugs 1000, wie z. B. der Achse(n), messen. Zum Beispiel können in mindestens einer Ausführungsform Änderungen der Vibrationen auf eine Änderung der Straßenoberfläche hinweisen. In

mindestens einer Ausführungsform können bei Verwendung von zwei oder mehr Vibrationssensoren 1042 die Unterschiede zwischen den Vibrationen zur Bestimmung der Reibung oder des Schlupfes der Straßenoberfläche verwendet werden (z. B. wenn ein Unterschied in der Vibration zwischen einer angetriebenen Achse und einer frei drehenden Achse besteht).

[0165] In mindestens einer Ausführungsform kann das Fahrzeug 1000 ein ADAS-System 1038 enthalten. In mindestens einer Ausführungsform kann das ADAS-System 1038 in einigen Beispielen ohne Einschränkung einen SoC umfassen. In mindestens einer Ausführungsform kann das ADAS-System 1038 ohne Einschränkung eine beliebige Anzahl und Kombination aus einem autonomen/adaptiven/automatischen Geschwindigkeitsregelsystem („ACC“), einem kooperativen adaptiven Geschwindigkeitsregelsystem („CACC“), einem Vorwärts-Crash-Warnsystem („FCW“), einem automatischen Notbremsystem („AEB“) ein System zur Warnung vor dem Verlassen der Fahrspur („LDW“), ein Spurhalteassistent („LKA“), ein System zur Warnung vor dem toten Winkel („BSW“), ein System zur Warnung vor rückwärtigem Querverkehr („RCTW“), ein System zur Kollisionswarnung („CW“), ein System zur Fahrspurzentrierung („LC“) und/oder andere Systeme, Merkmale und/oder Funktionen.

[0166] In mindestens einer Ausführungsform kann das ACC-System RADAR-Sensor(en) 1060, LIDAR-Sensor(en) 1064 und/oder eine beliebige Anzahl von Kameras verwenden. In mindestens einer Ausführungsform kann das ACC-System ein ACC-System in Längsrichtung und/oder ein ACC-System in Querrichtung umfassen. In mindestens einer Ausführungsform überwacht und regelt ein ACC-System in Längsrichtung den Abstand zu einem anderen Fahrzeug unmittelbar vor dem Fahrzeug 1000 und passt die Geschwindigkeit des Fahrzeugs 1000 automatisch an, um einen sicheren Abstand zu vorausfahrenden Fahrzeugen einzuhalten. In mindestens einer Ausführungsform sorgt ein seitliches ACC-System für die Einhaltung des Abstands und empfiehlt dem Fahrzeug 1000, bei Bedarf die Spur zu wechseln. In mindestens einer Ausführungsform ist ein seitlicher ACC mit anderen ADAS-Anwendungen, wie LC und CW, verbunden.

[0167] In mindestens einer Ausführungsform verwendet ein CACC-System Informationen von anderen Fahrzeugen, die über die Netzwerkschnittstelle 1024 und/oder die Funkantenne(n) 1026 von anderen Fahrzeugen über eine drahtlose Verbindung oder indirekt über eine Netzwerkverbindung (z. B. über das Internet) empfangen werden können. In mindestens einer Ausführungsform können direkte Verbindungen durch eine Fahrzeug-zu-Fahrzeug-Kommunikationsverbindung („V2V“) hergestellt werden, während indirekte Verbindungen durch eine Infrastruktur-zu-Fahrzeug-Kommunikationsverbindung („I2V“) hergestellt werden können. Im Allgemeinen liefert die V2V-Kommunikation Informationen über unmittelbar vorausfahrende Fahrzeuge (z. B. Fahrzeuge, die sich unmittelbar vor und auf derselben Fahrspur wie Fahrzeug 1000 befinden), während die I2V-Kommunikation Informationen über den weiter vorausfahrenden Verkehr liefert. In mindestens einer Ausführungsform kann ein CACC-System eine oder beide I2V- und V2V-Informationsquellen enthalten. In mindestens einer Ausführungsform kann ein CACC-System angesichts der Informationen über Fahrzeuge vor dem Fahrzeug 1000 zuverlässiger sein und hat das Potenzial, den Verkehrsfluss zu verbessern und Staus auf der Straße zu verringern.

[0168] In mindestens einer Ausführungsform ist ein FCW-System so konzipiert, dass es den Fahrer vor einer Gefahr warnt, so dass dieser korrigierende Maßnahmen ergreifen kann. In mindestens einer Ausführungsform verwendet ein FCW-System eine nach vorne gerichtete Kamera und/oder RADAR-Sensor(en) 1060, die mit einem speziellen Prozessor, digitalen Signalprozessor („DSP“), FPGA und/oder ASIC verbunden sind, der elektrisch gekoppelt ist, um dem Fahrer ein Feedback zu geben, wie z. B. ein Display, einen Lautsprecher und/oder eine vibrierende Komponente. In mindestens einer Ausführungsform kann ein FCW-System eine Warnung ausgeben, z. B. in Form eines Tons, einer optischen Warnung, einer Vibration und/oder eines schnellen Bremsimpulses.

[0169] In mindestens einer Ausführungsform erkennt ein AEB-System einen drohenden Zusammenstoß mit einem anderen Fahrzeug oder einem anderen Objekt und kann automatisch die Bremsen betätigen, wenn der Fahrer nicht innerhalb eines bestimmten Zeit- oder Entfernungsparameters korrigierend eingreift. In mindestens einer Ausführungsform kann das AEB-System die nach vorne gerichtete(n) Kamera(s) und/oder RADAR-Sensor(en) 1060 verwenden, die mit einem speziellen Prozessor, DSP, FPGA und/oder ASIC verbunden sind. In mindestens einer Ausführungsform wird ein AEB-System, wenn es eine Gefahr erkennt, in der Regel zunächst den Fahrer warnen, damit er korrigierende Maßnahmen ergreift, um eine Kollision zu vermeiden, und wenn der Fahrer keine korrigierenden Maßnahmen ergreift, kann das AEB-System automatisch die Bremsen betätigen, um die Auswirkungen einer vorhergesagten Kollision zu verhindern oder zumindest abzumildern. In mindestens einer Ausführungsform kann ein AEB-System Techniken wie eine dynamische Bremsunterstützung und/oder eine Notbremsung vor einem Unfall umfassen.

[0170] In mindestens einer Ausführungsform bietet ein LDW-System optische, akustische und/oder taktile Warnungen, wie z. B. Lenkrad- oder Sitzvibrationen, um den Fahrer zu warnen, wenn das Fahrzeug 1000 die Fahrbahnmarkierungen überquert. In mindestens einer Ausführungsform wird ein LDW-System nicht aktiviert, wenn der Fahrer ein absichtliches Verlassen der Fahrspur anzeigt, z. B. durch Betätigen eines Blinkers. In mindestens einer Ausführungsform kann ein LDW-System nach vorne gerichtete Kameras verwenden, die mit einem speziellen Prozessor, DSP, FPGA und/oder ASIC verbunden sind, der elektrisch gekoppelt ist, um dem Fahrer eine Rückmeldung zu geben, z. B. ein Display, einen Lautsprecher und/oder eine vibrierende Komponente. In mindestens einer Ausführungsform ist ein LKA-System eine Variante eines LDW-Systems. In mindestens einer Ausführungsform sorgt ein LKA-System für einen Lenkeingriff oder eine Bremsung, um das Fahrzeug 1000 zu korrigieren, wenn das Fahrzeug 1000 beginnt, seine Fahrspur zu verlassen.

[0171] In mindestens einer Ausführungsform erkennt und warnt ein BSW-System den Fahrer vor Fahrzeugen im toten Winkel des Fahrzeugs. In mindestens einer Ausführungsform kann ein BSW-System eine optische, akustische und/oder taktile Warnung ausgeben, um darauf hinzuweisen, dass das Einfädeln oder Wechseln der Fahrspur unsicher ist. In mindestens einer Ausführungsform kann ein BSW-System eine zusätzliche Warnung ausgeben, wenn ein Fahrer einen Blinker betätigt. In mindestens einer Ausführungsform kann ein BSW-System nach hinten gerichtete Kamera(s) und/oder RADAR-Sensor(en) 1060 verwenden, die mit einem speziellen Prozessor, DSP, FPGA und/oder ASIC gekoppelt sind, der elektrisch mit dem Fahrerfeedback gekoppelt ist, wie z. B. einem Display, einem Lautsprecher und/oder einer vibrierenden Komponente.

[0172] In mindestens einer Ausführungsform kann ein RCTW-System eine optische, akustische und/oder taktile Benachrichtigung ausgeben, wenn ein Objekt außerhalb des Bereichs der Rückfahrkamera erkannt wird, während das Fahrzeug 1000 rückwärts fährt. In mindestens einer Ausführungsform umfasst ein RCTW-System ein AEB-System, um sicherzustellen, dass die Fahrzeugbremsen betätigt werden, um einen Unfall zu vermeiden. In mindestens einer Ausführungsform kann ein RCTW-System einen oder mehrere nach hinten gerichtete(n) RADAR-Sensor(en) 1060 verwenden, der/die mit einem speziellen Prozessor, DSP, FPGA und/oder ASIC gekoppelt ist/sind, der/die elektrisch gekoppelt ist/sind, um dem Fahrer eine Rückmeldung zu geben, wie z. B. eine Anzeige, einen Lautsprecher und/oder eine vibrierende Komponente.

[0173] In mindestens einer Ausführungsform kann es bei herkömmlichen ADAS-Systemen zu falsch positiven Ergebnissen kommen, die für den Fahrer zwar störend und ablenkend sein können, aber in der Regel nicht katastrophal sind, da herkömmliche ADAS-Systeme den Fahrer warnen und ihm die Möglichkeit geben, zu entscheiden, ob ein Sicherheitszustand wirklich vorliegt und entsprechend zu handeln. In mindestens einer Ausführungsform entscheidet das Fahrzeug 1000 bei widersprüchlichen Ergebnissen selbst, ob das Ergebnis eines Primärrechners oder eines Sekundärrechners (z. B. eines ersten Steuergeräts oder eines zweiten Steuergeräts der Steuergeräte 1036) zu berücksichtigen ist. In mindestens einer Ausführungsform kann das ADAS-System 1038 beispielsweise ein Backup- und/oder Sekundärcomputer sein, der Wahrnehmungsinformationen an ein Rationalitätsmodul des Backup-Computers liefert. In mindestens einer Ausführungsform kann ein Backup-Computer Rationalität Monitor redundante diverse Software auf Hardware-Komponenten laufen, um Fehler in der Wahrnehmung und dynamische Fahraufgaben zu erkennen. In mindestens einer Ausführungsform können die Ausgaben des ADAS-Systems 1038 an eine übergeordnete MCU weitergeleitet werden. In mindestens einer Ausführungsform bestimmt eine übergeordnete MCU, wie der Konflikt zu lösen ist, um einen sicheren Betrieb zu gewährleisten, wenn die Ausgänge eines primären Computers und die Ausgänge eines sekundären Computers in Konflikt geraten.

[0174] In mindestens einer Ausführungsform kann ein Primärcomputer so konfiguriert sein, dass er einer übergeordneten MCU einen Zuverlässigkeitswert liefert, der die Zuverlässigkeit des Primärcomputers in Bezug auf ein ausgewähltes Ergebnis angibt. In mindestens einer Ausführungsform kann die überwachende MCU der Anweisung des primären Computers folgen, wenn die Zuverlässigkeitsbewertung einen Schwellenwert überschreitet, unabhängig davon, ob der sekundäre Computer ein widersprüchliches Ergebnis liefert. In mindestens einer Ausführungsform kann eine übergeordnete MCU in Fällen, in denen eine Zuverlässigkeitsbewertung einen Schwellenwert nicht erreicht und in denen Primär- und Sekundärcomputer unterschiedliche Ergebnisse anzeigen (z. B. einen Konflikt), zwischen den Computern vermitteln, um ein geeignetes Ergebnis zu bestimmen.

[0175] In mindestens einer Ausführungsform kann eine Überwachungs-MCU so konfiguriert sein, dass sie ein neuronales Netz bzw. neuronale Netze ausführt, das bzw. die so trainiert und konfiguriert ist bzw. sind, dass es bzw. sie zumindest teilweise auf der Grundlage der Ausgaben eines Primärcomputers und der Ausgaben eines Sekundärcomputers die Bedingungen bestimmt, unter denen der Sekundärcomputer Fehl-

alarme auslöst. In mindestens einer Ausführungsform können neuronale Netze in einer übergeordneten MCU lernen, wann der Ausgabe eines sekundären Computers vertraut werden kann und wann nicht. Wenn es sich bei dem sekundären Computer beispielsweise um ein RADAR-basiertes FCW-System handelt, kann ein neuronales Netzwerk in der überwachenden MCU lernen, wenn ein FCW-System metallische Objekte identifiziert, die in Wirklichkeit keine Gefahr darstellen, wie z. B. ein Abflussgitter oder ein Gullydeckel, der einen Alarm auslöst. In mindestens einer Ausführungsform kann ein neuronales Netz in einer Überwachungs-MCU lernen, den Spurhalteassistenten außer Kraft zu setzen, wenn Radfahrer oder Fußgänger anwesend sind und ein Verlassen der Fahrspur tatsächlich das sicherste Manöver ist, wenn es sich bei dem sekundären Computer um ein kamerabasiertes Spurhalteassistentensystem handelt. In mindestens einer Ausführungsform kann eine Überwachungs-MCU mindestens eine DLA oder eine GPU enthalten, die für die Ausführung neuronaler Netze mit zugehörigem Speicher geeignet ist. In mindestens einer Ausführungsform kann eine Überwachungs-MCU eine Komponente des/der SoC(s) 1004 umfassen und/oder als solche enthalten sein.

[0176] In mindestens einer Ausführungsform kann das ADAS-System 1038 einen sekundären Computer umfassen, der die ADAS-Funktionen nach den herkömmlichen Regeln der Computer Vision ausführt. In mindestens einer Ausführungsform kann dieser Sekundärcomputer klassische Computer-Vision-Regeln (wenn-dann) verwenden, und das Vorhandensein eines neuronalen Netzes (von neuronalen Netzen) in einer übergeordneten MCU kann die Zuverlässigkeit, Sicherheit und Leistung verbessern. In mindestens einer Ausführungsform machen beispielsweise die unterschiedliche Implementierung und die absichtliche Nichtidentität ein Gesamtsystem fehlertoleranter, insbesondere gegenüber Fehlern, die durch Softwarefunktionen (oder Software-Hardware-Schnittstellen) verursacht werden. Wenn beispielsweise in mindestens einer Ausführungsform ein Softwarefehler in der auf einem Primärcomputer laufenden Software auftritt und ein nicht identischer Softwarecode auf einem Sekundärcomputer ein konsistentes Gesamtergebnis liefert, kann eine überwachende MCU mit größerer Sicherheit davon ausgehen, dass das Gesamtergebnis korrekt ist und ein Fehler in der Software oder Hardware auf dem Primärcomputer keinen wesentlichen Fehler verursacht.

[0177] In mindestens einer Ausführungsform kann eine Ausgabe des ADAS-Systems 1038 in den Wahrnehmungsblock eines Primärrechners und/oder in den Block für dynamische Fahraufgaben eines Primärrechners eingespeist werden. Wenn das ADAS-System 1038 beispielsweise eine Aufprallwarnung aufgrund eines unmittelbar vor ihm befindlichen Objekts anzeigt, kann ein Wahrnehmungsblock diese Information bei der Identifizierung von Objekten verwenden. In mindestens einer Ausführungsform kann ein sekundärer Computer über ein eigenes neuronales Netz verfügen, das trainiert ist und somit das Risiko von Fehllarmen reduziert, wie hier beschrieben.

[0178] In mindestens einer Ausführungsform kann das Fahrzeug 1000 außerdem ein Infotainment-SoC 1030 (z. B. ein bordeigenes Infotainment-System (IVI)) enthalten. Obwohl der Infotainment-System-SoC 1030 als SoC abgebildet und beschrieben ist, kann er in mindestens einer Ausführungsform kein SoC sein und ohne Einschränkung zwei oder mehr diskrete Komponenten umfassen. In mindestens einer Ausführungsform kann der Infotainment-SoC 1030 ohne Einschränkung eine Kombination aus Hardware und Software umfassen, die verwendet werden kann, um Audio (z. B. Musik, einen persönlichen digitalen Assistenten, Navigationsanweisungen, Nachrichten, Radio usw.), Video (z. B. TV, Filme, Streaming usw.), Telefon (z. B. Freisprecheinrichtung), Netzwerkkonnektivität (z. B. LTE, WiFi usw.) und/oder Informationsdienste (z. B. Navigationssysteme, Einparkhilfe hinten, ein Radiodatensystem, fahrzeugbezogene Informationen wie Kraftstoffstand, zurückgelegte Gesamtstrecke, Bremskraftstoffstand, Ölstand, Tür öffnen/schließen, Luftfilterinformationen usw.) an das Fahrzeug 1000 bereitzustellen. Der Infotainment-SoC 1030 könnte beispielsweise Radios, Disk-Player, Navigationssysteme, Videoplayer, USB- und Bluetooth-Konnektivität, Carputer, In-Car-Entertainment, WiFi, Audiobedienelemente am Lenkrad, Freisprecheinrichtung, ein Heads-up-Display („HUD“), ein HMI-Display 1034, ein Telematikgerät, ein Bedienfeld (z. B. zur Steuerung und/oder Interaktion mit verschiedenen Komponenten, Funktionen und/oder Systemen) und/oder andere Komponenten umfassen. In mindestens einer Ausführungsform kann der Infotainment-SoC 1030 außerdem dazu verwendet werden, dem/den Benutzer(n) des Fahrzeugs 1000 Informationen (z. B. visuell und/oder akustisch) zur Verfügung zu stellen, wie z. B. Informationen vom ADAS-System 1038, Informationen zum autonomen Fahren wie geplante Fahrzeugmanöver, Trajektorien, Umgebungsinformationen (z. B. Kreuzungsinformationen, Fahrzeuginformationen, Straßeninformationen usw.) und/oder andere Informationen.

[0179] In mindestens einer Ausführungsform kann der Infotainment-SoC 1030 eine beliebige Menge und Art von GPU-Funktionalität enthalten. In mindestens einer Ausführungsform kann der Infotainment-SoC 1030 über den Bus 1002 mit anderen Geräten, Systemen und/oder Komponenten des Fahrzeugs 1000 kommunizieren. In mindestens einer Ausführungsform kann der Infotainment-SoC 1030 mit einer Überwachungs-MCU gekoppelt sein, so dass eine GPU eines Infotainment-Systems einige Selbstfahrfunktionen ausführen

kann, falls die primäre(n) Steuereinheit(en) 1036 (z. B. Primär- und/oder Backup-Computer des Fahrzeugs 1000) ausfallen. In mindestens einer Ausführungsform kann der Infotainment-SoC 1030 das Fahrzeug 1000 in einen Modus Fahrer zum sicheren Halt bringen versetzen, wie hier beschrieben.

[0180] In mindestens einer Ausführungsform kann das Fahrzeug 1000 außerdem ein Kombiinstrument 1032 (z. B. ein digitales Armaturenbrett, ein elektronisches Kombiinstrument, eine digitale Instrumententafel usw.) enthalten. In mindestens einer Ausführungsform kann das Kombiinstrument 1032 ohne Einschränkung einen Controller und/oder Supercomputer (z. B. einen diskreten Controller oder Supercomputer) umfassen. In mindestens einer Ausführungsform kann das Kombiinstrument 1032 ohne Einschränkung eine beliebige Anzahl und Kombination von Instrumenten wie Tachometer, Kraftstoffstand, Öldruck, Drehzahlmesser, Kilometerzähler, Blinker, Schaltstellungsanzeige, Sicherheitsgurt-Warnleuchte(n), Parkbrems-Warnleuchte(n), Motor-Fehlfunktionsleuchte(n), Informationen über zusätzliche Rückhaltesysteme (z. B. Airbags), Beleuchtungssteuerungen, Sicherheitssystemsteuerungen, Navigationsinformationen usw. enthalten. In einigen Beispielen können Informationen angezeigt und/oder zwischen dem Infotainment-SoC 1030 und dem Kombiinstrument 1032 ausgetauscht werden. In mindestens einer Ausführungsform kann das Kombiinstrument 1032 Teil des Infotainment-SoC 1030 sein, oder umgekehrt.

[0181] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 in dem System **Fig. 10C** für Inferenz- oder Vorhersageoperationen mindestens zum Teil auf Grundlage von Gewichtsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder hierin beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0182] **Fig. 10D** ist ein Diagramm eines Systems 1078 für die Kommunikation zwischen dem/den Cloud-basierten Server(n) und dem autonomen Fahrzeug 1000 von **Fig. 10A** gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann das System 1078 ohne Einschränkung den/die Server 1078, das/die Netzwerk(e) 1090 und eine beliebige Anzahl und Art von Fahrzeugen, einschließlich des Fahrzeugs 1000, umfassen. In mindestens einer Ausführungsform kann/ können der/die Server 1078 ohne Einschränkung eine Vielzahl von GPUs 1084(A)-1084(H) (hierin gemeinsam als GPUs 1084 bezeichnet), PCIe-Switches 1082(A)-1082(D) (hierin gemeinsam als PCIe-Switches 1082 bezeichnet) und/oder CPUs 1080 (A)-1080(B) (hierin gemeinsam als CPUs 1080 bezeichnet) enthalten. In mindestens einer Ausführungsform können GPUs 1084, CPUs 1080 und PCIe-Switches 1082 mit Hochgeschwindigkeitsverbindungen, wie z. B. und ohne Einschränkung NVLink-Schnittstellen 1088, die von NVIDIA entwickelt wurden, und/oder PCIe-Verbindungen 1086 miteinander verbunden werden. In mindestens einer Ausführungsform sind die GPUs 1084 über ein NVLink- und/oder NVSwitch-SoC verbunden und die GPUs 1084 und die PCIe-Switches 1082 sind über PCIe-Zusammenschaltungen verbunden. Obwohl acht GPUs 1084, zwei CPUs 1080 und vier PCIe-Switches 1082 abgebildet sind, ist dies nicht als Einschränkung zu verstehen. In mindestens einer Ausführungsform kann jeder der Server 1078 ohne Einschränkung eine beliebige Anzahl von GPUs 1084, CPUs 1080 und/oder PCIe-Switches 1082 in beliebiger Kombination enthalten. Zum Beispiel könnte(n) in mindestens einer Ausführungsform der/die Server 1078 jeweils acht, sechzehn, zweiunddreißig und/oder mehr GPUs 1084 enthalten.

[0183] In mindestens einer Ausführungsform können die Server 1078 über das/die Netzwerk(e) 1090 und von den Fahrzeugen Bilddaten empfangen, die unerwartete oder veränderte Straßenzustände zeigen, z. B. kürzlich begonnene Straßenarbeiten. In mindestens einer Ausführungsform kann/können der/die Server 1078 über das/die Netzwerk(e) 1090 und an Fahrzeuge neuronale Netzwerke 1092, aktualisiert oder anderweitig, und/oder Karteninformationen 1094, einschließlich, ohne Einschränkung, Informationen über den Verkehr und die Straßenbedingungen, übertragen. In mindestens einer Ausführungsform können die Aktualisierungen der Karteninformationen 1094 ohne Einschränkung Aktualisierungen für die HD-Karte 1022 umfassen, z. B. Informationen über Baustellen, Schlaglöcher, Umleitungen, Überschwemmungen und/oder andere Hindernisse. In mindestens einer Ausführungsform können die neuronalen Netze 1092 und/oder die Karteninformationen 1094 aus einem neuen Training und/oder aus Erfahrungen resultieren, die in den von einer beliebigen Anzahl von Fahrzeugen in einer Umgebung empfangenen Daten enthalten sind, und/oder zumindest teilweise auf einem in einem Datenzentrum durchgeführten Training basieren (z. B. unter Verwendung von Server(n) 1078 und/oder anderen Servern).

[0184] In mindestens einer Ausführungsform kann/können der/die Server 1078 verwendet werden, um maschinelle Lernmodelle (z. B. neuronale Netze) zumindest teilweise auf der Grundlage von Trainingsdaten zu trainieren. In mindestens einer Ausführungsform können die Trainingsdaten von Fahrzeugen und/oder in einer Simulation (z. B. mit einer Spielmaschine) erzeugt werden. In mindestens einer Ausführungsform wird eine beliebige Menge von Trainingsdaten markiert (z. B. wenn das zugehörige neuronale Netz von überwachtem Lernen profitiert) und/oder einer anderen Vorverarbeitung unterzogen. In mindestens einer Ausführungsform wird eine beliebige Menge von Trainingsdaten nicht gekennzeichnet und/oder vorverarbeitet (z. B. wenn das zugehörige neuronale Netz kein überwachtes Lernen erfordert). In mindestens einer Ausführungsform können die maschinellen Lernmodelle, sobald sie trainiert sind, von den Fahrzeugen verwendet werden (z. B. durch Übertragung an die Fahrzeuge über das/die Netzwerk(e) 1090) und/oder die maschinellen Lernmodelle können von dem/den Server(n) 1078 verwendet werden, um die Fahrzeuge aus der Ferne zu überwachen.

[0185] In mindestens einer Ausführungsform kann/können der/die Server 1078 Daten von Fahrzeugen empfangen und auf aktuelle neuronale Netze in Echtzeit anwenden, um intelligente Inferenzen in Echtzeit durchzuführen. In mindestens einer Ausführungsform kann/können der/die Server 1078 Deep-Learning-Supercomputer und/oder dedizierte KI-Computer umfassen, die von GPU(s) 1084 angetrieben werden, wie z. B. die von NVIDIA entwickelten DGX- und DGX Station-Maschinen. In mindestens einer Ausführungsform kann der/die Server 1078 jedoch eine Deep-Learning-Infrastruktur enthalten, die CPU-gestützte Rechenzentren verwendet.

[0186] In mindestens einer Ausführungsform kann die Deep-Learning-Infrastruktur des/der Server(s) 1078 in der Lage sein, ein schnelles Inferenzverfahren in Echtzeit durchzuführen und diese Fähigkeit zu nutzen, um den Zustand von Prozessoren, Software und/oder zugehöriger Hardware im Fahrzeug 1000 zu bewerten und zu überprüfen. In mindestens einer Ausführungsform kann die Deep-Learning-Infrastruktur beispielsweise periodische Aktualisierungen vom Fahrzeug 1000 erhalten, wie etwa eine Bildsequenz und/oder Objekte, die das Fahrzeug 1000 in dieser Bildsequenz lokalisiert hat (z. B. über Computer Vision und/oder andere maschinelle Objektklassifizierungsverfahren). In mindestens einer Ausführungsform kann die Deep-Learning-Infrastruktur ihr eigenes neuronales Netzwerk laufen lassen, um Objekte zu identifizieren und sie mit den vom Fahrzeug 1000 identifizierten Objekten zu vergleichen. Wenn die Ergebnisse nicht übereinstimmen und die Deep-Learning-Infrastruktur zu dem Schluss kommt, dass die KI im Fahrzeug 1000 eine Fehlfunktion aufweist, kann/können der/die Server 1078 ein Signal an das Fahrzeug 1000 senden, das einen ausfallsicheren Computer des Fahrzeugs 1000 anweist, die Kontrolle zu übernehmen, die Fahrgäste zu benachrichtigen und ein sicheres Parkmanöver durchzuführen.

[0187] In mindestens einer Ausführungsform kann/können der/die Server 1078 GPU(s) 1084 und einen oder mehrere programmierbare Inferenzbeschleuniger (z.B. NVIDIA's TensorRT 3 Geräte) enthalten. In mindestens einer Ausführungsform kann eine Kombination aus GPU-gesteuerten Servern und Inferenzbeschleunigung eine Reaktionsfähigkeit in Echtzeit ermöglichen. In mindestens einer Ausführungsform, z. B. wenn die Leistung weniger kritisch ist, können Server mit CPUs, FPGAs und anderen Prozessoren für das Inferenzverfahren verwendet werden. In mindestens einer Ausführungsform wird/werden die Hardwarestruktur(en) 115 verwendet, um eine oder mehrere Ausführungsformen durchzuführen. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden.

COMPUTERSYSTEME

[0188] **Fig. 11** ist ein Blockdiagramm, das ein beispielhaftes Computersystem veranschaulicht, das ein System mit zusammengeschalteten Vorrichtungen und Komponenten, ein System auf einem Chip (SOC) oder eine Kombination davon sein kann, gebildet mit einem Prozessor, der Ausführungseinheiten zum Ausführen einer Anweisung beinhalten kann, gemäß mindestens einer. In mindestens einer Ausführungsform kann ein Computersystem 1100 ohne Einschränkung eine Komponente wie etwa einen Prozessor 1102 beinhalten, um Ausführungseinheiten einschließlich Logik zum Durchführen von Algorithmen für Prozessdaten gemäß der vorliegenden Offenbarung zu verwenden, wie etwa in der hierin beschriebenen Ausführungsform. In mindestens einer Ausführungsform kann das Computersystem 1100 Prozessoren beinhalten, wie etwa die PEN-TIUM®-Prozessorfamilie, die Mikroprozessoren Xeon™, Itanium®, XScale™ und/oder StrongARM™, Intel® Core™ oder Intel® Nervana™, die von der Intel Corporation in Santa Clara, Kalifornien, erhältlich sind, obwohl auch andere Systeme (einschließlich PCs, die andere Mikroprozessoren, Engineering-Arbeitsstationen, Set-Top-Boxen und dergleichen aufweisen) verwendet werden können. In mindestens einer Ausführungsform kann das Computersystem 1100 eine Version des Betriebssystems WINDOWS ausführen, das von der Microsoft Corporation in Redmond, Washington, erhältlich ist, obwohl auch andere Betriebssysteme

(zum Beispiel UNIX und Linux), eingebettete Software und/oder grafische Benutzerschnittstellen verwendet werden können.

[0189] Ausführungsformen können in anderen Vorrichtungen wie etwa tragbaren Vorrichtungen und eingebetteten Anwendungen verwendet werden. Einige Beispiele für tragbare Vorrichtungen sind Mobiltelefone, Internetprotokoll-Vorrichtungen, Digitalkameras, persönliche digitale Assistenten („PDAs“) und tragbare PCs. In mindestens einer Ausführungsform können eingebettete Anwendungen einen Mikrocontroller, einen 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.

[0190] In mindestens einer Ausführungsform kann das Computersystem 1100 ohne Einschränkung einen Prozessor 1102 beinhalten, der ohne Einschränkung eine oder mehrere Ausführungseinheiten 1108 beinhalten kann, um ein Training und/oder eine Ableitung für ein Modell maschinellen Lernens gemäß in dieser Schrift beschriebenen Methoden durchzuführen. In mindestens einer Ausführungsform ist das Computersystem 1100 ein Einzelprozessor-Desktop- oder -Serversystem, aber in einer anderen Ausführungsform kann das Computersystem 1100 ein Multiprozessorsystem sein. In mindestens einer Ausführungsform kann der Prozessor 1102 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 1102 mit einem Prozessorbus 1110 gekoppelt sein, der Datensignale zwischen dem Prozessor 1102 und anderen Komponenten im Computersystem 1100 übertragen kann.

[0191] In mindestens einer Ausführungsform kann der Prozessor 1102 ohne Einschränkung einen internen Level-1-(„L1“-)Cache-Speicher („Cache“) 1104 beinhalten. In mindestens einer Ausführungsform kann der Prozessor 1102 einen einzelnen internen Cache oder mehrere Ebenen von internem Cache aufweisen. In mindestens einer Ausführungsform kann sich der Cachespeicher außerhalb des Prozessors 1102 befinden. Andere Ausführungsformen können auch eine Kombination aus sowohl internen als auch externen Caches beinhalten, und zwar in Abhängigkeit von der jeweiligen Implementierung und den Anforderungen. In mindestens einer Ausführungsform kann eine Registerbank 1106 unterschiedliche Typen von Daten in verschiedenen Registern speichern, einschließlich ohne Einschränkung Integerregistern, Gleitkommaregistern, Statusregistern und eines Anweisungszeigerregisters.

[0192] In mindestens einer Ausführungsform befindet sich die Ausführungseinheit 1108, einschließlich ohne Einschränkung der Logik zum Durchführen von Integer- und Gleitkommaoperationen, ebenfalls in dem Prozessor 1102. In mindestens einer Ausführungsform kann der Prozessor 1102 auch Festwertspeicher (read only memory - „ROM“) für Mikrocode („µcode“) beinhalten, der Mikrocode für bestimmte Makrobefehle speichert. In mindestens einer Ausführungsform kann die Ausführungseinheit 1108 Logik beinhalten, um einen gepackten Anweisungssatz 1109 zu handhaben. In mindestens einer Ausführungsform können durch die Aufnahme des gepackten Anweisungssatzes 1109 in einen Anweisungssatz eines Universalprozessors zusammen mit der damit assoziierten Schaltung zum Ausführen von Anweisungen Operationen, die durch viele Multimedia-Anwendungen verwendet werden, unter Verwendung von gepackten Daten in dem Prozessor 1102 durchgeführt werden. In einer oder mehreren Ausführungsformen können viele Multimedia-Anwendungen beschleunigt und effizienter ausgeführt werden, indem die volle Breite des Datenbusses eines Prozessors zum Durchführen von Operationen an gepackten Daten verwendet wird, wodurch die Notwendigkeit entfallen kann, kleinere Dateneinheiten über den Datenbus dieses Prozessors zu übermitteln, um eine oder mehrere Operationen an einem Datenelement nach dem anderen durchzuführen.

[0193] In mindestens einer Ausführungsform kann die Ausführungseinheit 1108 auch in Mikrocontrollern, eingebetteten Prozessoren, Grafikkarten, DSPs und anderen Typen von Logikschaltungen verwendet werden. In mindestens einer Ausführungsform kann das Computersystem 1100 ohne Einschränkung einen Speicher 1120 beinhalten. In mindestens einer Ausführungsform kann der Speicher 1120 eine Vorrichtung mit dynamischem Direktzugriffsspeicher („DRAM“), eine Vorrichtung mit statischem Direktzugriffsspeicher („SRAM“), eine Flash-Speichervorrichtung oder eine andere Speichervorrichtung sein. In mindestens einer Ausführungsform kann der Speicher 1120 Anweisung(en) 1119 und/oder Daten 1121 speichern, die durch Datensignale dargestellt werden, die vom Prozessor 1102 ausgeführt werden können.

[0194] In mindestens einer Ausführungsform kann ein Systemlogikchip an den Prozessorbus 1110 und den Speicher 1120 gekoppelt sein. In mindestens einer Ausführungsform kann ein Systemlogikchip ohne Einschränkung einen Speichersteuerungs-Hub (memory controller hub - „MCH“) 1116 beinhalten und der Prozessor 1102 mit dem MCH 1116 über den Prozessorbus 1110 kommunizieren. In mindestens einer Ausführungsform kann der MCH 1116 einen Speicherpfad mit hoher Bandbreite 1118 zum Speicher 1120 zur Anweisungs- und Datenspeicherung und zur Speicherung von Grafikbefehlen, Daten und Texturen bereitstellen. In mindestens einer Ausführungsform kann der MCH 1116 Datensignale zwischen dem Prozessor 1102, dem Speicher 1120 und anderen Komponenten in dem Computersystem 1100 leiten und Datensignale zwischen dem Prozessorbus 1110, dem Speicher 1120 und einer System-E/A-Schnittstelle 1122 überbrücken. In mindestens einer Ausführungsform kann ein Systemlogikchip einen Grafikport zum Koppeln an eine Grafiksteuerung bereitstellen. In mindestens einer Ausführungsform kann der MCH 1116 durch einen Speicherpfad 1118 mit hoher Bandbreite an den Speicher 1120 gekoppelt sein und eine Grafik-/Videokarte 1112 durch eine Accelerated-Graphics-Port-(„AGP“-)-Zusammenschaltung 1114 an den MCH 1116 gekoppelt sein.

[0195] In mindestens einer Ausführungsform kann das Computersystem 1100 die System-E/A-Schnittstelle 1122 als proprietären Hub-Schnittstellenbus verwenden, um den MCH 1116 an einen E/A-Steuerungs-Hub (I/O controller hub - „ICH“) 1130 zu koppeln. In mindestens einer Ausführungsform kann der ICH 1130 direkte Verbindungen zu einigen E/A-Vorrichtungen über einen lokalen E/A-Bus bereitstellen. In mindestens einer Ausführungsform kann ein lokaler E/A-Bus ohne Einschränkung einen Hochgeschwindigkeits-E/A-Bus zum Verbinden von Peripheriegeräten mit dem Speicher 1120, einem Chipsatz und dem Prozessor 1102 beinhalten. Beispiele können ohne Einschränkung eine Audiosteuerung 1129, einen Firmware-Hub („Flash-BIOS“) 1128, einen drahtlosen Sendeempfänger 1126, einen Datenspeicher 1124, eine ältere E/A-Steuerung 1123, die Benutzereingabe- und Tastaturschnittstellen 1125 enthält, einen seriellen Erweiterungsport 1127 wie etwa einen Universal-Serial-Bus(„USB“-)Port und eine Netzsteuerung 1134 beinhalten. In mindestens einer Ausführungsform kann der Datenspeicher 1124 ein Festplattenlaufwerk, ein Diskettenlaufwerk, eine CD-ROM-Vorrichtung, eine Flash-Speichervorrichtung oder eine andere Massenspeichervorrichtung umfassen.

[0196] In mindestens einer Ausführungsform veranschaulicht **Fig. 11** ein System, das zusammenschaltete Hardware-Vorrichtungen oder „Chips“ beinhaltet, wohingegen **Fig. 11** ein beispielhaftes SoC darstellen kann. In mindestens einer Ausführungsform können die in **Fig. 11** dargestellten Vorrichtungen mit proprietären Verbindungen, standardisierten Verbindungen (z. B. PCIe) oder einer Kombination davon zusammenschaltet sein. In mindestens einer Ausführungsform sind eine oder mehrere Komponenten des Computersystems 1100 unter Verwendung von Compute-Express-Link(CXL)-Verbindungen miteinander verbunden.

[0197] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 in dem System **Fig. 11** für Inferenz- oder Vorhersageoperationen mindestens zum Teil auf Grundlage von Gewichtsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder hierin beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0198] **Fig. 12** ist ein Blockdiagramm, das eine elektronische Vorrichtung 1200 zum Nutzen eines Prozessors 1210 gemäß mindestens einer Ausführungsform darstellt. In mindestens einer Ausführungsform kann die elektronische Vorrichtung 1200 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.

[0199] In mindestens einer Ausführungsform kann die elektronische Vorrichtung 1200 ohne Einschränkung den Prozessor 1210 beinhalten, der kommunikativ an eine beliebige geeignete Anzahl oder einen beliebigen geeigneten Typ von Komponenten, Peripheriegeräten, Modulen oder Vorrichtungen gekoppelt ist. In mindestens einer Ausführungsform ist der Prozessor 1210 unter Verwendung eines Busses oder einer Schnittstelle gekoppelt, wie etwa eines I²C-Busses, eines Systemverwaltungsbus (System Management Bus - „SMBus“), eines Low-Pin-Count(LPC-)Busses, einer seriellen Peripherieschnittstelle (Serial Peripheral Interface - „SPI“), eines High-Definition-Audio(„HDA“-)Busses, eines Serial-Advance-Technology-Attachment („SATA“-)Busses, eines Universal Serial Bus („USB“) (Version 1, 2, 3 usw.) oder eines Universal-Asynchronous-Receiver/Transmitter(„UART“-)Busses. **Fig. 12** veranschaulicht ein System, das zusammenschaltete Hardware-Vorrichtungen oder „Chips“ beinhaltet, wohingegen **Fig. 12** in anderen Ausführungsformen ein beispielhaftes SoC darstellen kann. In mindestens einer Ausführungsform können die in **Fig. 12** dargestellten

Vorrichtungen mit proprietären Verbindungen, standardisierten Verbindungen (z. B. PCIe) oder einer Kombination davon zusammengeschaltet sein. In mindestens einer Ausführungsform sind eine oder mehrere Komponenten des Computersystems 12 unter Verwendung von Compute-Express-Link(CXL)-Verbindungen miteinander verbunden.

[0200] In mindestens einer Ausführungsform kann **Fig. 12** ein Display 1224, einen Touchscreen 1225, ein Touchpad 1230, eine Nahfeldkommunikations(near field communications - „NFC“-)Einheit 1245, einen Sensor-Hub 1240, einen Wärmesensor 1246, einen Express-Chipsatz („EC“) 1235, ein Trusted Platform Module („TPM“) 1238, BIOS/Firmware/Flash-Speicher („BIOS, FW Flash“) 1222, ein DSP 1260, ein Laufwerk 1220 wie eine Solid State Disk („SSD“) oder ein Festplattenlaufwerk („HDD“), eine drahtlose lokale Netzwerkeinheit („WLAN“) 1250, eine Bluetooth-Einheit 1252, eine Wireless Wide Area Network-Einheit („WWAN“) 1256, eine Global-Positioning-System(GPS)-Einheit 1255, eine Kamera („USB 3.0-Kamera“) 1254 wie etwa eine USB-3.0 Kamera und/oder eine Low-Power-Double-Data-Rate („LPDDR“-)Speichereinheit („LPDDR3“) 1215 beinhalten, die beispielsweise in einem LPDDR3-Standard implementiert ist. Diese Komponenten können jeweils auf eine beliebige geeignete Weise implementiert sein.

[0201] In mindestens einer Ausführungsform können andere Komponenten durch hierin beschriebene Komponenten kommunikativ an den Prozessor 1210 gekoppelt sein. In mindestens einer Ausführungsform können ein Beschleunigungsmesser 1241, ein Umgebungslichtsensor (ambient light sensor - „ALS“) 1242, ein Kompass 1243 und ein Gyroskop 1244 kommunikativ an den Sensor-Hub 1240 gekoppelt sein. In mindestens einer Ausführungsform können ein Thermosensor 1239, ein Lüfter 1237, eine Tastatur 1236 und ein Touchpad 1230 kommunikativ an den EC 1235 gekoppelt sein. In mindestens einer Ausführungsform können Lautsprecher 1263, Kopfhörer 1264 und ein Mikrofon („Mikro“) 1265 kommunikativ an eine Audioeinheit („Audiocodierer und Klasse-D-Verst.“) 1262 gekoppelt sein, die wiederum kommunikativ an den DSP 1260 gekoppelt sein kann. In mindestens einer Ausführungsform kann die Audioeinheit 1262 zum Beispiel und ohne Einschränkung einen Audiocodierer/-decodierer („Codec“) und einen Klasse-D-Verstärker beinhalten. In mindestens einer Ausführungsform kann eine SIM-Karte („SIM“) 1257 kommunikativ an die WWAN-Einheit 1256 gekoppelt sein. In mindestens einer Ausführungsform können Komponenten, wie zum Beispiel die WLAN-Einheit 1250 und die Bluetooth-Einheit 1252 sowie die WWAN-Einheit 1256, in einem Next Generation Form Factor („NGFF“) implementiert sein.

[0202] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 in dem System **Fig. 12** für Inferenz- oder Vorhersageoperationen mindestens zum Teil auf Grundlage von Gewichtsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder hierin beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0203] **Fig. 13** veranschaulicht ein Computersystem 1300 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist das Computersystem 1300 so konfiguriert, dass es verschiedene Prozesse und Verfahren implementiert, die in dieser Offenbarung beschrieben sind.

[0204] In mindestens einer Ausführungsform umfasst das Computersystem 1300 ohne Einschränkung mindestens eine zentrale Verarbeitungseinheit („CPU“) 1302, die mit einem Kommunikationsbus 1310 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 1300 ohne Einschränkung einen Hauptspeicher 1304 und Steuerlogik (z. B. implementiert als Hardware, Software oder eine Kombination davon) und Daten werden in dem Hauptspeicher 1304 gespeichert, der die Form eines Direktzugriffsspeichers („RAM“) annehmen kann. In mindestens einer Ausführungsform stellt ein Netzstellen-Teilsystem („Netzstellen“) 1322 eine Schnittstelle zu anderen Rechenvorrichtungen und Netzen bereit, um Daten von anderen Systemen mit dem Computersystem 1300 zu empfangen und an diese zu übertragen.

[0205] In mindestens einer Ausführungsform beinhaltet das Computersystem 1300 ohne Einschränkung in mindestens einer Ausführungsform Eingabevorrichtungen 1308, ein Parallelverarbeitungssystem 1312 und Anzeigevorrichtungen 1306, die unter Verwendung einer herkömmlichen Kathodenstrahlröhre (cathode ray

tube - „CRT“), einer Flüssigkristallanzeige (liquid crystal display - „LCD“), einer Anzeige mit Leuchtdioden (light emitting diode - „LED“), einer Plasmaanzeige oder anderer geeigneter Anzeigetechnologien implementiert sein können. In mindestens einer Ausführungsform werden Benutzereingaben von Eingabevorrichtungen 1308 wie etwa Tastatur, Maus, Touchpad, Mikrofon usw. empfangen. In mindestens einer Ausführungsform kann sich jedes hierin beschriebene Modul auf einer einzelnen Halbleiterplattform befinden, um ein Verarbeitungssystem zu bilden.

[0206] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 in dem System **Fig. 13** für Inferenz- oder Vorhersageoperationen mindestens zum Teil auf Grundlage von Gewichtsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder hierin beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0207] **Fig. 14** veranschaulicht ein Computersystem 1400 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform beinhaltet das Computersystem 1400 ohne Einschränkung einen Computer 1410 und einen USB-Stick 1420. In mindestens einer Ausführungsform kann der Computer 1410 ohne Einschränkung eine beliebige Anzahl und einen beliebigen Typ von Prozessor(en) (nicht gezeigt) und einen Speicher (nicht gezeigt) beinhalten. In mindestens einer Ausführungsform beinhaltet der Computer 1410 ohne Einschränkung einen Server, eine Cloud-Instanz, einen Laptop und einen Desktop-Computer.

[0208] In mindestens einer Ausführungsform beinhaltet der USB-Stick 1420 ohne Einschränkung eine Verarbeitungseinheit 1430, eine USB-Schnittstelle 1440 und eine USB-Schnittstellenlogik 1450. In mindestens einer Ausführungsform kann die Verarbeitungseinheit 1430 ein beliebige Anweisungsausführungssystem, -gerät oder eine beliebige Anweisungsausführungsvorrichtung sein, das bzw. die dazu in der Lage ist, Anweisungen auszuführen. In mindestens einer Ausführungsform kann die Verarbeitungseinheit 1430 ohne Einschränkung eine beliebige Anzahl und einen beliebigen Typ von Verarbeitungskernen (nicht gezeigt) beinhalten. In mindestens einer Ausführungsform umfasst die Verarbeitungseinheit 1430 eine anwendungsspezifische integrierte Schaltung („ASIC“), die zum Durchführen beliebiger Mengen und Typen von Operationen optimiert ist, die mit maschinellem Lernen assoziiert sind. Zum Beispiel ist in mindestens einer Ausführungsform die Verarbeitungseinheit 1430 eine Tensor-Verarbeitungseinheit („TPC“), die zum Durchführen von Inferenzoperationen des maschinellen Lernens optimiert ist. In mindestens einer Ausführungsform ist die Verarbeitungseinheit 1430 eine Sichtverarbeitungseinheit (vision processing unit - „VPU“), die für zum Durchführen von Inferenzoperationen des maschinellen Sehens und des maschinellen Lernens optimiert ist.

[0209] In mindestens einer Ausführungsform kann die USB-Schnittstelle 1440 ein beliebiger Typ von USB-Stecker oder USB-Buchse sein. Zum Beispiel ist in mindestens einer Ausführungsform die USB-Schnittstelle 1440 eine USB-3.0-Typ-C-Buchse für Daten und Leistung. In mindestens einer Ausführungsform ist die USB-Schnittstelle 1440 ein USB-3.0-Typ-A-Anschluss. In mindestens einer Ausführungsform kann die USB-Schnittstellenlogik 1450 eine beliebige Menge und einen beliebigen Typ von Logik beinhalten, die es der Verarbeitungseinheit 1430 ermöglicht, über den USB-Stecker 1440 eine Schnittstelle mit Vorrichtungen (z. B. dem Computer 1410) zu bilden.

[0210] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 in dem System **Fig. 14** für Inferenz- oder Vorhersageoperationen mindestens zum Teil auf Grundlage von Gewichtsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder hierin beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0211] **Fig. 15A** veranschaulicht eine beispielhafte Architektur, in der eine Vielzahl von GPUs 1510(1)-1510(N) über Hochgeschwindigkeitsverknüpfungen 1540(1)-1540(N) (z. B. Busse, Punkt-zu-Punkt-Verbindungen usw.) kommunikativ an eine Vielzahl von Mehrkernprozessoren 1505(1)-1505(M) gekoppelt ist. In mindestens einer Ausführungsform unterstützen die Hochgeschwindigkeitsverknüpfungen 1540(1)-1540(N) einen Kommunikationsdurchsatz von 4 GB/s, 30 GB/s, 80 GB/s oder höher. In mindestens einer Ausführungsform

können verschiedene Verbindungsprotokolle verwendet werden, einschließlich, aber nicht beschränkt auf, PCIe 4.0 oder 5.0 und NVLink 2.0. In verschiedenen Figuren stellen „N“ und „M“ positive ganze Zahlen dar, deren Werte von Figur zu Figur unterschiedlich sein können.

[0212] Zusätzlich und in mindestens einer Ausführungsform sind zwei oder mehr der GPUs 1510 über Hochgeschwindigkeitsverknüpfungen 1529(1)-1529(2) zusammengeschaltet, die unter Verwendung ähnlicher oder anderer Protokolle/Verknüpfungen implementiert sein können als derjenigen, die für die Hochgeschwindigkeitsverknüpfungen 1540(1)-1540(N) verwendet werden. Auf ähnliche Weise können zwei oder mehr der Mehrkernprozessoren 1505 über eine Hochgeschwindigkeitsverknüpfung 1528 verbunden sein, bei der es sich um Busse eines symmetrischen Multiprozessors (symmetric multi-processor - SMP) handeln kann, die mit 20 GB/s, 30 GB/s, 120 GB/s oder höher betrieben werden. Alternativ kann die gesamte Kommunikation zwischen den verschiedenen in **Fig. 15A** gezeigten Systemkomponenten über ähnliche Protokolle/Verbindungen erfolgen (z. B. über eine gemeinsame Verbindungsstruktur).

[0213] In mindestens einer Ausführungsform ist jeder Mehrkernprozessor 1505 jeweils über Speicherzusammenschaltungen 1526(1)-1526(M) kommunikativ an einen Prozessorspeicher 1501(1)-1501(M) gekoppelt und jede GPU 1510(1)-1510(N) jeweils über GPU-Speicherzusammenschaltungen 1550(1)-1550(N) kommunikativ an den GPU-Speicher 1520(1)-1520(N) gekoppelt. In mindestens einer Ausführungsform können die Speicherzusammenschaltungen 1526 und 1550 ähnliche oder unterschiedliche Speicherzugriffstechniken nutzen. Bei den Prozessorspeichern 1501(1)-1501(M) und den GPU-Speichern 1520 kann es sich beispielsweise und ohne Einschränkung 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 nichtflüchtige Speicher, wie etwa 3D XPoint oder Nano-Ram, handeln. In mindestens einer Ausführungsform kann ein Abschnitt der Prozessorspeicher 1501 flüchtiger Speicher sein und ein anderer Abschnitt nichtflüchtiger Speicher sein (z. B. unter Verwendung einer Speicherhierarchie mit zwei Ebenen (two-level memory - 2LM)).

[0214] Wie hierin beschrieben, können verschiedene Mehrkernprozessoren 1505 und GPUs 1510 zwar physisch an einen konkreten Speicher 1501 bzw. 1520 gekoppelt sein und/oder eine einheitliche Speicherarchitektur implementiert sein, bei der ein virtueller Systemadressraum (auch als „effektiver Adressraum“ bezeichnet) auf verschiedene physische Speicher verteilt ist. Zum Beispiel können die Prozessorspeicher 1501(1)-1501(M) jeweils 64 GB Systemspeicheradressraum umfassen und die GPU-Speicher 1520(1)-1520(N) jeweils 32 GB Systemspeicheradressraum umfassen, was zu einem adressierbaren Speicher von insgesamt 256 GB führt, wenn M=2 und N=4. Andere Werte für N und M sind möglich.

[0215] **Fig. 15 B** veranschaulicht zusätzliche Details für eine Verbindung zwischen einem Mehrkernprozessor 1507 und einem Grafikbeschleunigungsmodul 1546 gemäß einer beispielhaften Ausführungsform. In mindestens einer Ausführungsform kann das Grafikbeschleunigungsmodul 1546 einen oder mehrere GPU-Chips beinhalten, die auf einer Leiterkarte integriert sind, die über eine Hochgeschwindigkeitsverknüpfung 1540 (z. B. einen PCIe-Bus, NVLink usw.) an den Prozessor 1507 gekoppelt ist. In mindestens einer Ausführungsform kann das Grafikbeschleunigungsmodul 1546 alternativ auf einem Gehäuse oder Chip mit dem Prozessor 1507 integriert sein.

[0216] In mindestens einer Ausführungsform beinhaltet der Prozessor 1507 eine Vielzahl von Kernen 1560A-1560D, jeder mit einem Adressenübersetzungspuffer (translation lookaside buffer - „TLB“) 1561A-1561D und einem oder mehreren Caches 1562A-1562D. In mindestens einer Ausführungsform können die Kerne 1560A-1560D verschiedene andere Komponenten zum Ausführen von Anweisungen und Verarbeiten von Daten beinhalten, die nicht dargestellt sind. In mindestens einer Ausführungsform können die Caches 1562A-1562D Caches der Ebene 1 (Level 1 - L1) und Ebene 2 (L2) umfassen. Zusätzlich können ein oder mehrere gemeinsam genutzte Caches 1556 in den Caches 1562A-1562D beinhaltet sein und von den Sätzen von Kernen 1560A-1560D gemeinsam genutzt werden. Eine Ausführungsform des Prozessors 1507 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. In mindestens einer Ausführungsform sind der Prozessor 1507 und das Grafikbeschleunigungsmodul 1546 mit dem Systemspeicher 1514 verbunden, der die Prozessorspeicher 1501(1)-1501(M) aus **Fig. 15 A** beinhalten kann.

[0217] In mindestens einer Ausführungsform wird die Kohärenz für Daten und Anweisungen, die in verschiedenen Caches 1562A-1562D, 1556 und Systemspeicher 1514 gespeichert sind, über Zwischenkernkommunikation über einen Kohärenzbus 1564 aufrechterhalten. Zum Beispiel kann in mindestens einer Ausführungsform

rungsform jeder Cache eine Cache-Kohärenzlogik/-schaltung aufweisen, die damit assoziiert ist, um als Reaktion auf detektierte Lese- oder Schreibvorgänge in konkreten Cache-Zeilen über den Kohärenzbus 1564 zu kommunizieren. In mindestens einer Ausführungsform wird ein Cache-Snooping-Protokoll über den Kohärenzbus 1564 implementiert, um Cache-Zugriffe per Snooping zu kontrollieren.

[0218] In mindestens einer Ausführungsform koppelt eine Proxy-Schaltung 1525 das Grafikbeschleunigungsmodul 1546 kommunikativ an den Kohärenzbus 1564, was es dem Grafikbeschleunigungsmodul 1546 ermöglicht, an einem Cache-Kohärenzprotokoll als Peer der Kerne 1560A-1560D teilzunehmen. Insbesondere stellt in mindestens einer Ausführungsform eine Schnittstelle 1535 Verbindungsfähigkeit mit der Proxy-Schaltung 1525 über eine Hochgeschwindigkeitsverknüpfung 1540 bereit und eine Schnittstelle 1537 verbindet das Grafikbeschleunigungsmodul 1546 mit der Hochgeschwindigkeitsverknüpfung 1540.

[0219] In mindestens einer Ausführungsform stellt eine Beschleuniger-Integrationsschaltung 1536 Cache-Verwaltungs-, Speicherzugriffs-, Kontextverwaltungs- und Unterbrechungsverwaltungsdienste im Auftrag einer Vielzahl von Grafikverarbeitungsengines 1531(1)-1531(N) des Grafikbeschleunigungsmoduls 1546 bereit. In mindestens einer Ausführungsform können die Grafikverarbeitungsengines 1531(1)-1531(N) jeweils eine separate Grafikverarbeitungseinheit (GPU) umfassen. In mindestens einer Ausführungsform können die Grafikverarbeitungsengines 1531(1)-1531(N) alternativ unterschiedliche Typen von Grafikverarbeitungsengines innerhalb einer GPU umfassen, wie etwa Grafikausführungseinheiten, Medienverarbeitungs-Engines (z. B. Videocodierer/-decodierer), Abtaster und Blit-Engines. In mindestens einer Ausführungsform kann das Grafikbeschleunigungsmodul 1546 eine GPU mit einer Vielzahl von Grafikverarbeitungsengines 1531(1)-1531(N) sein oder die Grafikverarbeitungsengines 1531(1)-1531(N) können einzelne GPUs sein, die auf einem gemeinsamen Gehäuse, einer Leitungskarte oder einem Chip integriert sind.

[0220] In mindestens einer Ausführungsform beinhaltet die Beschleuniger-Integrationsschaltung 1536 eine Speicherverwaltungseinheit (MMU) 1539 zum Durchführen verschiedener Speicherverwaltungsfunktionen, wie etwa Übersetzungen von virtuellem in physischen Speicher (auch als Übersetzungen von effektivem in realen Speicher bezeichnet) und Speicherzugriffsprotokolle zum Zugreifen auf Systemspeicher 1514. Die MMU 1539 kann in mindestens einer Ausführungsform auch einen Adressenübersetzungspuffer (TLB) (nicht gezeigt) für das Caching von Übersetzungen von virtuellen/effektiven in physische/reale Adressen beinhalten. In mindestens einer Ausführungsform kann ein Cache 1538 Befehle und Daten für einen effizienten Zugriff durch die Grafikverarbeitungsengines 1531(1)-1531(N) speichern. In mindestens einer Ausführungsform werden die in dem Cache 1538 und in den Grafikspeichern 1533(1)-1533(M) gespeicherten Daten mit den Kern-Caches 1562A-1562D, 1556 und dem Systemspeicher 1514 kohärent gehalten, möglicherweise unter Verwendung einer Abrufeinheit 1544. Wie erwähnt, kann dies über die Proxy-Schaltung 1525 im Auftrag des Caches 1538 und der Speicher 1533(1)-1533(M) erzielt werden (z. B. Senden von Aktualisierungen an den Cache 1538 in Bezug auf Modifikationen/Zugriffe auf Cache-Zeilen in den Prozessor-Caches 1562A-1562D, 1556 und Empfangen von Aktualisierungen von dem Cache 1538).

[0221] In mindestens einer Ausführungsform speichert ein Satz von Registern 1545 Kontextdaten für Threads, die durch die Grafikverarbeitungsengines 1531(1)-1531(N) ausgeführt werden, und eine Kontextverwaltungsschaltung 1548 verwaltet Thread-Kontexte. Zum Beispiel kann die Kontextverwaltungsschaltung 1548 Sicherungs- und Wiederherstellungsoperationen 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 durch eine Grafikverarbeitungsengine ausgeführt werden kann). Zum Beispiel kann die Kontextverwaltungsschaltung 1548 bei einer Kontextumschaltung aktuelle Registerwerte in einer bezeichneten Region im Speicher speichern (z. B. identifiziert durch einen Kontextzeiger). Sie kann dann bei der Rückkehr zu einem Kontext die Registerwerte wiederherstellen. In mindestens einer Ausführungsform empfängt und verarbeitet eine Unterbrechungsverwaltungsschaltung 1547 von Systemvorrichtungen empfangene Unterbrechungen.

[0222] In mindestens einer Ausführungsform werden virtuelle/effektive Adressen von einer Grafikverarbeitungsengine 1531 durch die MMU 1539 in reale/physische Adressen im Systemspeicher 1514 übersetzt. In mindestens einer Ausführungsform unterstützt die Beschleunigerintegrationsschaltung 1536 mehrere (z. B. 4, 8, 16) Grafikbeschleunigermodule 1546 und/oder andere Beschleunigervorrichtungen. Das Grafikbeschleunigermodul 1546 kann in mindestens einer Ausführungsform für eine einzelne Anwendung dediziert sein, die auf dem Prozessor 1507 ausgeführt wird, oder von mehreren Anwendungen gemeinsam genutzt werden. In mindestens einer Ausführungsform ist eine virtualisierte Grafikausführungsumgebung dargestellt, in der die Ressourcen der Grafikverarbeitungsengines 1531(1)-1531(N) mit mehreren Anwendungen oder virtuellen Maschinen (VMs) gemeinsam genutzt werden. In mindestens einer Ausführungsform können die Res-

sourcen in „Slices“ unterteilt werden, die unterschiedlichen VMs und/oder Anwendungen auf Grundlage von Verarbeitungsanforderungen und Prioritäten, die mit VMs und/oder Anwendungen assoziiert sind, zugewiesen werden.

[0223] In mindestens einer Ausführungsform dient die Beschleunigerintegrationsschaltung 1536 als eine Brücke zu einem System für das Grafikbeschleunigungsmodul 1546 und stellt Adressenübersetzungs- und Systemspeicher-Cache bereit. Darüber hinaus kann die Beschleunigerintegrationsschaltung 1536 in mindestens einer Ausführungsform Virtualisierungseinrichtungen für einen Hostprozessor bereitstellen, um die Virtualisierung der Grafikverarbeitungsengines 1531(1)-1531(N), Unterbrechungen und Speicherverwaltung zu verwalten.

[0224] Da in mindestens einer Ausführungsform die Hardware-Ressourcen der Grafikverarbeitungsengines 1531(1)-1531(N) explizit auf einen realen Adressraum abgebildet sind, den der Host-Prozessor 1507 sieht, kann ein beliebiger Host-Prozessor diese Ressourcen direkt unter Verwendung eines effektiven Adresswerts adressieren. In mindestens einer Ausführungsform ist eine Funktion der Beschleunigerintegrationsschaltung 1536 die physische Trennung der Grafikverarbeitungsengines 1531(1)-1531(N), sodass sie einem System als unabhängige Einheiten erscheinen.

[0225] In mindestens einer Ausführungsform sind ein oder mehrere Grafikspeicher 1533(1)-1533(M) jeweils an jede der Grafikverarbeitungsengines 1531(1)-1531(N) gekoppelt und es gilt $N=M$. In mindestens einer Ausführungsform speichern die Grafikspeicher 1533(1)-1533(M) Anweisungen und Daten, die durch jede der Grafikverarbeitungsengines 1531(1)-1531(N) verarbeitet werden. In mindestens einer Ausführungsform kann es sich bei den Grafikspeichern 1533(1)-1533(M) um flüchtige Speicher, wie etwa DRAMs (einschließlich gestapelter DRAMs), GDDR-Speicher (z. B. GDDR5, GDDR6) oder HBM, und/oder um nichtflüchtige Speicher, wie etwa 3D XPoint oder Nano-Ram, handeln.

[0226] In mindestens einer Ausführungsform werden zum Reduzieren des Datenverkehrs über die Hochgeschwindigkeitsverknüpfung 1540 Verzerrungstechniken verwendet, um sicherzustellen, dass es sich bei den in den Grafikspeichern 1533(1)-1533(M) gespeicherten Daten um Daten handelt, die am häufigsten durch die Grafikverarbeitungsengines 1531(1)-1531(N) verwendet werden und vorzugsweise nicht durch die Kerne 1560A-1560D verwendet werden (zumindest nicht häufig). Auf ähnliche Weise versucht in mindestens einer Ausführungsform ein Verzerrungsmechanismus, Daten, die von den Kernen (und vorzugsweise nicht von den GrafikverarbeitungsEngines 1531(1)-1531(N)) benötigt werden, innerhalb der Caches 1562A-1562D, 1556 und des Systemspeichers 1514 zu behalten.

[0227] **Fig. 15C** veranschaulicht eine weitere beispielhafte Ausführungsform, in der die Beschleunigerintegrationsschaltung 1536 in den Prozessor 1507 integriert ist. In dieser Ausführungsform kommunizieren die Grafikverarbeitungsengines 1531(1)-1531(N) direkt über die Hochgeschwindigkeitsverknüpfung 1540 mit der Beschleunigerintegrationsschaltung 1536 über die Schnittstelle 1537 und die Schnittstelle 1535 (die wiederum eine beliebige Form von Bus- oder Schnittstellenprotokoll sein können). In mindestens einer Ausführungsform kann die Beschleunigerintegrationsschaltung 1536 ähnliche Operationen durchführen, wie sie in **Fig. 15B** beschrieben sind, jedoch möglicherweise mit einem höheren Durchsatz, da sie sich in unmittelbarer Nähe zum Kohärenzbus 1564 und den Caches 1562A-1562D, 1556 befindet. In mindestens einer Ausführungsform unterstützt eine Beschleunigerintegrationsschaltung unterschiedliche Programmiermodelle, die ein Programmiermodell für dedizierte Prozesse (keine Virtualisierung des Grafikbeschleunigungsmoduls) und gemeinsam genutzte Programmiermodelle (mit Virtualisierung) beinhalten, die Programmiermodelle, die durch die Beschleunigerintegrationsschaltung 1536 gesteuert werden, und Programmiermodelle beinhalten können, die durch das Grafikbeschleunigungsmodul 1546 gesteuert werden.

[0228] In mindestens einer Ausführungsform sind die Grafikverarbeitungsengines 1531(1)-1531(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 der GrafikverarbeitungsEngines 1531(1)-1531(N) lenken und so eine Virtualisierung innerhalb einer VM/Partition bereitstellen.

[0229] In mindestens einer Ausführungsform können die Grafikverarbeitungsengines 1531(1)-1531(N) von mehreren VM-/Anwendungspartitionen gemeinsam genutzt werden. In mindestens einer Ausführungsform können gemeinsam genutzte Modelle einen System-Hypervisor verwenden, um die Grafikverarbeitungsengines 1531(1)-1531(N) zu virtualisieren und den Zugriff durch jedes Betriebssystem zu ermöglichen. Bei Systemen mit einzelner Partition ohne Hypervisor befinden sich die Grafikverarbeitungsengines 1531(1)-1531(N) in

mindestens einer Ausführungsform im Besitz eines Betriebssystems. In mindestens einer Ausführungsform kann ein Betriebssystem die Grafikverarbeitungsengines 1531(1)-1531(N) virtualisieren, um Zugriff auf jeden Prozess oder jede Anwendung bereitzustellen.

[0230] In mindestens einer Ausführungsform wählt das Grafikbeschleunigungsmodul 1546 oder eine einzelne Grafikverarbeitungsengine 1531(1)-1531(N) ein Prozesselement unter Verwendung eines Prozessidentifikators aus. In mindestens einer Ausführungsform sind die Prozesselemente in dem Systemspeicher 1514 gespeichert und unter Verwendung der hierin beschriebenen Technik zur Übersetzung von effektiven Adressen in reale Adressen adressierbar. In mindestens einer Ausführungsform kann ein Prozessidentifikator ein implementationsspezifischer Wert sein, der einem Hostprozess bereitgestellt wird, wenn er seinen Kontext bei der Grafikverarbeitungsengine 1531(1)-1531(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.

[0231] Fig. 15D veranschaulicht einen beispielhaften Beschleunigerintegrations-Slice 1590. In mindestens einer Ausführungsform umfasst ein „Slice“ einen vorgegebenen Abschnitt der Verarbeitungsressourcen der Beschleunigerintegrationsschaltung 1536. In mindestens einer Ausführungsform ist eine Anwendung effektiver Adressraum 1582 innerhalb des Systemspeichers 1514, der Prozesselemente 1583 speichert. In mindestens einer Ausführungsform werden Prozesselemente 1583 als Reaktion auf GPU-Aufrufe 1581 von Anwendungen 1580, die auf dem Prozessor 1507 ausgeführt werden, gespeichert. In mindestens einer Ausführungsform enthält ein Prozesselement 1583 den Prozesszustand für die entsprechende Anwendung 1580. In mindestens einer Ausführungsform kann ein in dem Prozesselement 1583 enthaltener Arbeitsdeskriptor (work descriptor - WD) 1584 ein einzelner durch eine Anwendung angeforderter Task sein oder einen Zeiger auf eine Warteschlange von Tasks enthalten. In mindestens einer Ausführungsform ist der WD 1584 ein Zeiger auf eine Taskanforderungswarteschlange im effektiven Adressraum 1582 einer Anwendung.

[0232] In mindestens einer Ausführungsform können das Grafikbeschleunigungsmodul 1546 und/oder die einzelnen Grafikverarbeitungsengines 1531(1)-1531(N) von allen oder einer Teilmenge der Prozesse in einem System gemeinsam genutzt werden. In mindestens einer Ausführungsform kann eine Infrastruktur zum Einrichten der Prozesszustände und zum Senden eines WD 1584 an ein Grafikbeschleunigungsmodul 1546 zum Starten eines Tasks in einer virtualisierten Umgebung enthalten sein.

[0233] In mindestens einer Ausführungsform ist ein Programmiermodell mit dediziertem Prozess implementationsspezifisch. In mindestens einer Ausführungsform besitzt in diesem Modell ein einzelner Prozess das Grafikbeschleunigungsmodul 1546 oder eine einzelne Grafikverarbeitungsengine 1531. Wenn das Grafikbeschleunigungsmodul 1546 durch einen einzelnen Prozess in Besitz genommen ist, initialisiert ein Hypervisor in mindestens einer Ausführungsform die Beschleuniger-Integrationsschaltung 1536 für eine besitzende Partition und ein Betriebssystem initialisiert die Beschleuniger-Integrationsschaltung 1536 für einen besitzenden Prozess, wenn das Grafikbeschleunigungsmodul 1546 zugeordnet ist.

[0234] In mindestens einer Ausführungsform ruft im Betrieb eine WD-Abrufeinheit 1591 in der Beschleuniger-Integrationsscheibe 1590 den nächsten WD 1584 ab, der eine Angabe der Arbeit beinhaltet, die durch eine oder mehrere Grafikverarbeitungsengines des Grafikbeschleunigungsmoduls 1546 zu erledigen ist. In mindestens einer Ausführungsform können Daten von dem WD 1584 in den Registern 1545 gespeichert und durch die MMU 1539, die Unterbrechungsverwaltungsschaltung 1547 und/oder die Kontextverwaltungsschaltung 1548 verwendet werden, wie veranschaulicht. Zum Beispiel beinhaltet eine Ausführungsform der MMU 1539 eine Segment-/Page-Walk-Schaltung zum Zugreifen auf Segment-/Page-Tabellen 1586 innerhalb des virtuellen Adressbereichs 1585 des Betriebssystems. In mindestens einer Ausführungsform kann die Unterbrechungsverwaltungsschaltung 1547 von dem Grafikbeschleunigungsmodul 1546 empfangene Unterbrechungsereignisse 1592 verarbeiten. Beim Durchführen von Grafikoperationen wird in mindestens einer Ausführungsform eine durch eine Grafikverarbeitungsengine 1531(1)-1531(N) erzeugte effektive Adresse 1593 durch die MMU 1539 in eine reale Adresse übersetzt.

[0235] In mindestens einer Ausführungsform werden Register 1545 für jede Grafikverarbeitungsengine 1531(1)-1531(N) und/oder jedes Grafikbeschleunigungsmodul 1546 dupliziert und sie können durch einen Hypervisor oder ein Betriebssystem initialisiert werden. Jedes dieser duplizierten Register kann in mindestens einer Ausführungsform in einem Beschleunigerintegrations-Slice 1590 enthalten sein. Beispielhafte Register, die durch einen Hypervisor initialisiert werden können, sind in Tabelle 1 gezeigt.

Tabelle 1 - Durch Hypervisor initialisierte Register

Register Nr.	Beschreibung
1	Slice-Steuerregister
2	Bereichszeiger für geplante Prozesse für reale Adressen (real address - RA)
3	Autoritätsmasken-Überschreibungsregister
4	Unterbrechungsvektor-Tabelleneintragsversatz
5	Unterbrechungsvektor-Tabelleneintragsbegrenzung
6	Zustandsregister
7	ID einer logischen Partition
8	Hypervisor-Beschleunigernutzungsaufzeichnungszeiger für reale Adressen (RA)
9	Speicherbeschreibungsregister

[0236] Beispielhafte Register, die durch ein Betriebssystem initialisiert werden können, sind in Tabelle 2 gezeigt.

Tabelle 2 - Durch Betriebssystem initialisierte Register

Register Nr.	Beschreibung
1	Prozess- und Thread-Identifikation
2	Kontext-Sicherungs-/Wiederherstellungszeiger für effektive Adressen (EA)
3	Beschleunigernutzungsaufzeichnungszeiger für virtuelle Adressen (VA)
4	Speichersegmenttabellenzeiger für virtuelle Adressen (VA)
5	Autoritätsmaske
6	Arbeitsdeskriptor

[0237] In mindestens einer Ausführungsform ist jeder WD 1584 spezifisch für ein konkretes Grafikbeschleunigungsmodul 1546 und/oder die Grafikverarbeitungsengines 1531(1)-1531(N). In mindestens einer Ausführungsform enthält er alle Informationen, die für eine GrafikverarbeitungsEngine 1531(1)-1531(N) erforderlich sind, um Arbeit zu verrichten, oder er kann ein Zeiger auf einen Speicherort sein, an dem eine Anwendung eine Befehlswarteschlange von abzuschließender Arbeit eingerichtet hat.

[0238] **Fig. 15E** veranschaulicht zusätzliche Details für eine beispielhafte Ausführungsform eines gemeinsam genutzten Modells. Diese Ausführungsform beinhaltet einen realen Adressbereich 1598 des Hypervisors, in dem eine Prozesselementliste 1599 gespeichert ist. In mindestens einer Ausführungsform kann auf den realen Hypervisor-Adressraum 1598 über einen Hypervisor 1596 zugegriffen werden, der die Grafikbeschleunigungsmodul-Engines für das Betriebssystem 1595 virtualisiert.

[0239] 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 1546 verwenden. In mindestens einer Ausführungsform gibt es zwei Programmiermodelle, bei denen das Grafikbeschleunigungsmodul 1546 von mehreren Prozessen und Partitionen gemeinsam genutzt wird, nämlich über Zeitslices gemeinsam genutzt und über gerichtete Grafik gemeinsam genutzt.

[0240] In mindestens einer Ausführungsform besitzt in diesem Modell der System-Hypervisor 1596 das Grafikbeschleunigungsmodul 1546 und er stellt seine Funktion allen Betriebssystemen 1595 zur Verfügung. Damit ein Grafikbeschleunigungsmodul 1546 die Virtualisierung durch den System-Hypervisor 1596 unterstützt, muss in mindestens einer Ausführungsform das Grafikbeschleunigungsmodul 1546 bestimmte Anforderungen einhalten, wie etwa (1) die Aufgabenanforderung einer Anwendung muss autonom sein (das heißt, der Zustand muss zwischen den Aufgaben nicht beibehalten werden), oder das Grafikbeschleunigungsmodul 1546 muss einen Mechanismus zum Sichern und Wiederherstellen von Kontext bereitstellen, (2) das Grafikbeschleunigungsmodul 1546 garantiert, dass die Aufgabenanforderung einer Anwendung innerhalb einer vorgegebenen Zeitspanne abgeschlossen wird, einschließlich etwaiger Übersetzungsfehler, oder das Grafik-

beschleunigungsmodul 1546 stellt eine Fähigkeit bereit, die Verarbeitung einer Aufgabe vorwegzunehmen, und (3) dem Grafikbeschleunigungsmodul 1546 muss Fairness zwischen den Prozessen garantiert werden, wenn es in einem gerichteten gemeinsam genutzten Programmiermodell arbeitet.

[0241] In mindestens einer Ausführungsform ist es erforderlich, dass die Anwendung 1580 einen Systemaufruf des Betriebssystems 1595 mit einem Grafikbeschleunigungsmodultyp, einem Arbeitsdeskriptor (WD), einem Wert des Autoritätsmaskenregisters (authority mask register - AMR) und einem Kontext-Sicherungs-/Wiederherstellungsbereichszeiger (context save/restore area pointer - CSRP) vornimmt. In mindestens einer Ausführungsform beschreibt der Typ des Grafikbeschleunigungsmoduls eine gezielte Beschleunigungsfunktion für einen Systemaufruf. In mindestens einer Ausführungsform kann der Typ des Grafikbeschleunigungsmoduls ein systemspezifischer Wert sein. In mindestens einer Ausführungsform ist WD spezifisch für das Grafikbeschleunigungsmodul 1546 formatiert und kann in Form eines Befehls des Grafikbeschleunigungsmoduls 1546, eines effektiven Adresszeigers auf eine benutzerdefinierte Struktur, eines effektiven Adresszeigers auf eine Befehlswarteschlange oder einer beliebigen anderen Datenstruktur vorliegen, welche die vom Grafikbeschleunigungsmodul 1546 zu verrichtende Arbeit beschreibt.

[0242] In mindestens einer Ausführungsform ist ein AMR-Wert ein AMR-Zustand, der für einen derzeitigen Prozess zu verwenden ist. In mindestens einer Ausführungsform ist ein Wert, der an ein Betriebssystem übergeben wird, vergleichbar mit einer Anwendung, die einen AMR festlegt. Falls in mindestens einer Ausführungsform Implementierungen der Beschleunigerintegrationsschaltung 1536 (nicht gezeigt) und des Grafikbeschleunigungsmoduls 1546 kein Benutzer-Autoritätsmasken-Überschreibungsregister (User Authority Mask Override Register - UAMOR) unterstützen, kann ein Betriebssystem einen derzeitigen UAMOR-Wert auf einen AMR-Wert anwenden, bevor ein AMR in einem Hypervisor-Aufruf übergeben wird. In mindestens einer Ausführungsform kann der Hypervisor 1596 optional einen derzeitigen Wert für ein Autoritätsmasken-Überschreibungsregister (Authority Mask Override Register - AMOR) anwenden, bevor ein AMR in dem Prozesselement 1583 platziert wird. In mindestens einer Ausführungsform ist der CSRP eines der Register 1545, das eine effektive Adresse eines Bereichs im effektiven Adressbereich 1582 einer Anwendung enthält, damit das Grafikbeschleunigungsmodul 1546 den Kontextzustand speichert und wiederherstellt. In mindestens einer Ausführungsform ist dieser Zeiger optional, falls zwischen Aufgaben oder bei der Präemption einer Aufgabe kein Zustand gespeichert werden muss. In mindestens einer Ausführungsform kann der Kontext-Sicherungs-/Wiederherstellungsbereich ein gepinnter Systemspeicher sein.

[0243] Beim Empfang eines Systemaufrufs kann das Betriebssystem 1595 überprüfen, ob die Anwendung 1580 registriert wurde und die Berechtigung erhalten hat, das Grafikbeschleunigungsmodul 1546 zu verwenden. In mindestens einer Ausführungsform ruft das Betriebssystem 1595 dann den Hypervisor 1596 mit den in Tabelle 3 gezeigten Informationen auf.

Tabelle 3 - OS-zu-Hypervisor-Aufrufparameter

Parameter Nr.	Beschreibung
1	Ein Arbeitsdeskriptor (WD)
2	Ein Wert eines Autoritätsmaskenregisters (AMR) (möglicherweise maskiert)
3	Ein Kontext-Sicherungs-/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)

[0244] In mindestens einer Ausführungsform verifiziert der Hypervisor 1596 beim Empfangen eines Hypervisor-Aufrufs, dass das Betriebssystem 1595 registriert ist und die Autorität zur Verwendung des Grafikbeschleunigungsmoduls 1546 bekommen hat. In mindestens einer Ausführungsform setzt der Hypervisor 1596 dann das Prozesselement 1583 in eine mit dem Prozesselement verknüpfte Liste für einen entsprechenden Typ des Grafikbeschleunigungsmoduls 1546 ein. In mindestens einer Ausführungsform kann ein Prozesselement die in Tabelle 4 gezeigten Informationen beinhalten.

Tabelle 4 - Prozesselementinformationen

Element Nr.	Beschreibung
1	Ein Arbeitsdeskriptor (WD)
2	Ein Wert eines Autoritätsmaskenregisters (AMR) (möglicherweise maskiert).
3	Ein Kontext-Sicherungs-/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)

[0245] In mindestens einer Ausführungsform initialisiert der Hypervisor eine Vielzahl von Registern 1545 von Beschleunigerintegrations-Slices 1590.

[0246] Wie in **Fig. 15F** veranschaulicht, wird in mindestens einer Ausführungsform ein einheitlicher Speicher verwendet, der über einen gemeinsamen virtuellen Speicheradressraum adressierbar ist, der zum Zugreifen auf die physischen Prozessorspeicher 1501 (1)-1501(N) und die GPU-Speicher 1520(1)-1520(N) verwendet wird. In dieser Implementation nutzen Operationen, die auf den GPUs 1510(1)-1510(N) ausgeführt werden, einen gleichen virtuellen/effektiven Speicheradressraum für den Zugriff auf die Prozessorspeicher 1501 (1)-1501(M) und umgekehrt, was die Programmierbarkeit vereinfacht. In mindestens einer Ausführungsform wird ein erster Abschnitt eines virtuellen/effektiven Adressraums dem Prozessorspeicher 1501(1) zugewiesen, ein zweiter Abschnitt dem zweiten Prozessorspeicher 1501(N), ein dritter Abschnitt dem GPU-Speicher 1520(1) und so weiter. In mindestens einer Ausführungsform wird dadurch ein gesamter virtueller/effektiver Speicherraum (mitunter als effektiver Adressraum bezeichnet) über jeden der Prozessorspeicher 1501 und GPU-Speicher 1520 verteilt, was ermöglicht, dass ein beliebiger Prozessor oder eine beliebige GPU auf einen beliebigen physischen Speicher mit einer virtuellen Adresse zugreifen kann, die auf diesen Speicher abgebildet ist.

[0247] In mindestens einer Ausführungsform stellt die Verzerrungs-/Kohärenzverwaltungsschaltung 1594A-1594E in einer oder mehreren der MMUs 1539A-1539E die Cachekohärenz zwischen Cache von einem oder mehreren Hostprozessoren (z. B. 1505) und den GPUs 1510 sicher und setzt Verzerrungsmethoden um, die physische Speicher angeben, in denen bestimmte Arten von Daten gespeichert werden sollen. In mindestens einer Ausführungsform kann die Schaltung zur Verwaltung der Vorspannung/Kohärenz (1594A-1594E) in einer MMU eines oder mehrerer Host-Prozessoren (1505) und/oder in der Beschleuniger-Integrationsschaltung (1536) implementiert sein, während in **Fig. 15F** mehrere Instanzen der Schaltung zur Verwaltung der Vorspannung/Kohärenz dargestellt sind.

[0248] Eine Ausführungsform ermöglicht es, dass GPU-Speicher 1520 als Teil des Systemspeichers abgebildet werden und unter Verwendung von Technologie für gemeinsam genutzten virtuellen Speicher (shared virtual memory - SVM) darauf zugegriffen wird, ohne jedoch Rechenleistungsnachteile zu erleiden, die mit der vollständigen System-Cache-Kohärenz assoziiert sind. In mindestens einer Ausführungsform stellt eine Fähigkeit, dass auf GPU-Speicher 1520 als Systemspeicher ohne lästigen Cache-Kohärenz-Overhead zugegriffen wird, eine vorteilhafte Betriebsumgebung für GPU-Abladung bereit. In mindestens einer Ausführungsform ermöglicht diese Anordnung es der Software des Hostprozessors 1505, ohne den Overhead der traditionellen E/A-DMA-Datenkopien Operanden einzurichten und auf Berechnungsergebnisse zuzugreifen. In mindestens einer Ausführungsform sind an derartigen traditionellen Kopien Treiberaufrufe, Unterbrechungen und auf Speicher abgebildete E/A-Zugriffe (memory mapped I/O accesses - MMIO-Zugriffe) beteiligt, die alle

in Bezug auf einfache Speicherzugriffe ineffizient sind. In mindestens einer Ausführungsform kann eine Fähigkeit, ohne Cache-Kohärenz-Overheads auf GPU-Speicher 1520 zuzugreifen, für die Ausführungszeit einer abgeladenen Berechnung entscheidend sein. In Fällen mit erheblichem Streaming-Schreibspeicherkehr kann zum Beispiel der Cache-Kohärenz-Overhead eine effektive Schreibbandbreite, die durch eine GPU 1510 gesehen wird, in mindestens einer Ausführungsform erheblich reduzieren. In mindestens einer Ausführungsform können die Effizienz der Operandeneinrichtung, die Effizienz des Ergebniszugriffs und die Effizienz der GPU-Berechnung eine Rolle bei der Bestimmung der Effektivität einer GPU-Offload spielen.

[0249] In mindestens einer Ausführungsform wird die Auswahl von GPU-Verzerrung und Hostprozessorverzerrung durch eine Verzerrungs-Tracker-Datenstruktur angetrieben. In mindestens einer Ausführungsform kann zum Beispiel eine Verzerrungstabelle verwendet werden, die eine seitengranulare Struktur sein kann (z. B. 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-Speicher 1520 implementiert werden, mit oder ohne Verzerrungs-Cache in einer GPU 1510 (um z.B. häufig/kürzlich verwendete Einträge einer Verzerrungstabelle zwischenspeichern). Alternativ kann in mindestens einer Ausführungsform eine gesamte Verzerrungstabelle innerhalb einer GPU geführt werden.

[0250] In mindestens einer Ausführungsform wird vor dem tatsächlichen Zugriff auf einen GPU-Speicher auf einen Verzerrungstabelleneintrag zugegriffen, der mit jedem Zugriff auf den GPU-gebundenen Speicher 1520 assoziiert ist, wodurch die folgenden Operationen verursacht werden. In mindestens einer Ausführungsform werden lokale Anforderungen von einer GPU 1510, die ihre Seite in der GPU-Verzerrung finden, direkt an einen entsprechenden GPU-Speicher 1520 weitergeleitet. In mindestens einer Ausführungsform werden lokale Anforderungen von einer GPU, die ihre Seite in der Host-Verzerrung finden, an den Prozessor 1505 weitergeleitet (z. B. über eine Hochgeschwindigkeitsverknüpfung, wie hierin beschrieben). In mindestens einer Ausführungsform schließen Anforderungen von dem Prozessor 1505, die eine angeforderte Seite in der Hostprozessor-Verzerrung finden, eine Anforderung wie ein normales Lesen des Speichers ab. Alternativ können Anforderungen, die an eine GPU-Verzerrungsseite gerichtet sind, an eine GPU 1510 weitergeleitet werden. In mindestens einer Ausführungsform kann eine GPU dann eine Seite in eine Host-Prozessor-Neigung umwandeln, wenn er aktuell keine Seite verwendet. In mindestens einer Ausführungsform kann ein 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.

[0251] 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 Hostprozessors 1505 zur Verzerrung der GPU verwendet, jedoch nicht für einen entgegengesetzten Übergang.

[0252] In mindestens einer Ausführungsform wird die Cachekohärenz durch das vorübergehende Rendern von GPU-verzerrten Seiten aufrechterhalten, die vom Hostprozessor 1505 nicht zwischengespeichert werden können. Um auf diese Seiten zuzugreifen, kann in mindestens einer Ausführungsform der Prozessor 1505 Zugriff von der GPU 1510 anfordern, die den Zugriff sofort gewähren kann oder auch nicht. Um die Kommunikation zwischen dem Prozessor 1505 und der GPU 1510 zu reduzieren, ist es daher in mindestens einer Ausführungsform vorteilhaft, sicherzustellen, dass GPU-verzerrte Seiten solche sind, die durch eine GPU, aber nicht den Hostprozessor 1505, benötigt werden und umgekehrt.

[0253] Die Hardwarestruktur(en) 115 werden verwendet, um eine oder mehrere Ausführungsformen durchzuführen. Details bezüglich der Hardwarestruktur(en) von Inferenz- und/oder Trainingslogik 115 können hierin in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden.

[0254] **Fig. 16** veranschaulicht beispielhafte integrierte Schaltungen und damit assoziierte Grafikprozessoren, die unter Verwendung eines oder mehrerer IP-Kerne hergestellt werden können, gemäß verschiedenen hierin 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.

[0255] **Fig. 16** ist ein Blockdiagramm, das ein beispielhaftes System auf einer integrierten Chipschaltung 1600 veranschaulicht, die unter Verwendung eines oder mehrerer IP-Kerne gefertigt werden kann, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform beinhaltet die integrierte Schaltung 1600 einen oder mehrere Anwendungsprozessor(en) 1605 (z. B. CPUs), mindestens einen Grafikprozessor 1610 und kann zusätzlich einen Bildprozessor 1615 und/oder einen Videoprozessor 1620 beinhalten, die ein modularer IP-Kern sein kann. In mindestens einer Ausführungsform beinhaltet die integrierte Schaltung 1600 Peripherie- oder Buslogik, was eine USB-Steuerung 1625, eine UART-Steuerung 1630, eine SPI/SDIO-Steuerung 1635 und eine I²S/I²C-Steuerung 1640 beinhaltet. In mindestens einer Ausführungsform kann die integrierte Schaltung 1600 eine Anzeigevorrichtung 1645 beinhalten, die an einen oder mehrere von einer Steuerung einer Multimediaschnittstelle mit hoher Auflösung (high-definition multimedia interface - HDMI) 1650 und eine Anzeigeschnittstelle für eine mobile Industrieprozessorschnittstelle (mobile industry processor interface - MIPI) 1655 gekoppelt ist. In mindestens einer Ausführungsform kann die Speicherung durch ein Flash-Speicherteilsystem 1660 bereitgestellt sein, das Flash-Speicher und eine Flash-Speichersteuerung beinhaltet. In mindestens einer Ausführungsform kann eine Speicherschnittstelle über eine Speichersteuerung 1665 für den Zugriff auf SDRAM- oder SRAM-Speichervorrichtungen bereitgestellt sein. In mindestens einer Ausführungsform beinhalten einige integrierte Schaltungen zusätzlich eine eingebettete Sicherheits-Engine 1670.

[0256] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform kann die Ableitungs- und/oder Trainingslogik 115 in der integrierten Schaltung 1600 für Ableitungs- oder Vorhersagevorgänge verwendet werden, die mindestens teilweise auf Gewichtungsparametern basieren, die unter Verwendung von Trainingsvorgängen für neuronale Netzwerke, Funktionen und/oder Architekturen von neuronalen Netzwerken oder hierin beschriebenen Anwendungsfällen für neuronale Netzwerke berechnet wurden.

[0257] **[305]****Fig. 17A-17B** veranschaulichen beispielhafte integrierte Schaltungen und damit assoziierte Grafikprozessoren, die unter Verwendung eines oder mehrerer IP-Kerne hergestellt werden können, gemäß verschiedenen hierin 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.

[0258] **Fig. 17A-17B** sind Blockdiagramme, die beispielhafte Grafikprozessoren zur Verwendung innerhalb eines SoC gemäß hierin beschriebenen Ausführungsformen veranschaulichen. **Fig. 17A** veranschaulicht einen beispielhaften Grafikprozessor 1710 eines Systems auf einer integrierten Chipschaltung, die unter Verwendung eines oder mehrerer IP-Kerne gemäß mindestens einer Ausführungsform gefertigt werden kann. **Fig. 17B** veranschaulicht einen zusätzlichen beispielhaften Grafikprozessor 1740 eines Systems auf einer integrierten Chipschaltung, die unter Verwendung eines oder mehrerer IP-Kerne gemäß mindestens einer Ausführungsform gefertigt werden kann. In mindestens einer Ausführungsform ist der Grafikprozessor 1710 in **Fig. 17A** ein Grafikprozessorkern mit geringem Energieverbrauch. In mindestens einer Ausführungsform ist der Grafikprozessor 1740 in **Fig. 17B** ein Grafikprozessorkern mit höherer Leistung. In mindestens einer Ausführungsform kann jeder der Grafikprozessoren 1710, 1740 eine Variante des Grafikprozessors 1610 aus **Fig. 16**.

[0259] In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 1710 einen Scheitelpunktprozessor 1705 und einen oder mehrere Fragmentprozessoren 1715A-1715N (z. B. 1715A, 1715B, 1715C, 1715D bis 1715N-1 und 1715N). In mindestens einer Ausführungsform kann der Grafikprozessor 1710 unterschiedliche Shader-Programme über eine separate Logik ausführen, sodass der Vertex-Prozessor 1705 für die Ausführung von Vorgängen für Scheitelpunkt-Shader-Programme optimiert ist, während ein oder mehrere Fragmentprozessoren 1715A-1715N Fragment(z. B. Pixel)-Shading-Vorgänge für Fragment- oder Pixel-Shader-Programme ausführen. In mindestens einer Ausführungsform führt der Scheitelpunktprozessor 1705 eine Scheitelpunktverarbeitungsstufe einer 3D-Grafikpipeline durch und erzeugt Primitive und Scheitelpunkt-Daten. In mindestens einer Ausführungsform verwenden ein oder mehrere Fragmentprozessoren 1715A-1715N Primitiv- und Scheitelpunkt-Daten, die vom Scheitelpunktprozessor 1705 erzeugt werden, um einen Bildspeicher zu erzeugen, der auf einer Anzeigevorrichtung angezeigt wird. In mindestens einer Ausführungsform sind ein oder mehrere Fragmentprozessoren 1715A-1715N optimiert, um Fragment-Shader-Programme auszuführen, wie in einer OpenGL-API bereitgestellt, die verwendet werden können, um ähnliche Operationen wie ein Pixel-Shader-Programm durchzuführen, wie es in einer Direct 3D-API vorgesehen ist.

[0260] In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 1710 zusätzlich eine oder mehrere Speicherverwaltungseinheiten (memory management units - MMUs) 1720A-1720B, Cache 1725A-1725B und Schaltungsverbindung(en) 1730A-1730B. In mindestens einer Ausführungsform stellen eine oder mehrere MMU(s) 1720A-1720B eine virtuell-zu-physische Adresszuordnung für den Grafikprozessor 1710 bereit, was für den Scheitelpunktprozessor 1705 und/oder Fragmentprozessor(en) 1715A-1715N beinhaltet, die sich auf Scheitelpunkt- oder im Speicher gespeicherte Bild-/Texturdaten zusätzlich zu in einem oder mehreren Caches 1725A-1725B gespeicherten Scheitelpunkt- oder Bild-/Texturdaten beziehen. In mindestens einer Ausführungsform können eine oder mehrere MMU(s) 1720A-1720B mit anderen MMUs innerhalb des Systems synchronisiert werden, was eine oder mehrere MMUs beinhaltet, die derartig mit einem oder mehreren Anwendungsprozessoren 1605, Bildprozessoren 1015 und/oder Videoprocessoren 1620 aus **Fig. 16** verbunden sind, dass jeder Prozessor 1605-1620 an einem gemeinsam genutzten oder einheitlichen virtuellen Speichersystem teilnehmen kann. In mindestens einer Ausführungsform ermöglichen es eine oder mehrere Schaltungsverbindungen 1730A-1730B dem Grafikprozessor 1710, sich mit anderen IP-Kernen innerhalb des SoC zu verknüpfen, entweder über einen internen Bus des SoC oder über eine direkte Verbindung.

[0261] In mindestens einer Ausführungsform umfasst der Grafikprozessor 1740 einen oder mehrere Shader-Kern(e) 1755A-1755N (z. B., 1755A, 1755B, 1755C, 1755D, 1755E, 1755F, bis 1755N-1 und 1755N), wie in **Fig. 17B** dargestellt, was eine einheitliche Shader-Kern-Architektur ermöglicht, bei der ein einziger Kern oder Typ oder Kern alle Arten von programmierbarem Shader-Code ausführen kann, einschließlich Shader-Programmcode zur Implementierung von Vertex-Shadern, Fragment-Shadern und/oder Compute-Shadern. In mindestens einer Ausführungsform kann die Anzahl der Shader-Kerne variieren. In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 1740 eine Task-Verwaltung innerhalb des Kerns 1745, die als Thread-Verteiler fungiert, um Ausführungsthreads an einen oder mehrere Shader-Kerne 1755A-1755N zu verteilen, und eine Kachelungseinheit 1758, um Kachelungsoperationen für kachelbasiertes Rendering zu beschleunigen, bei dem Rendering-Operationen für eine Szene im Bildraum unterteilt werden, um zum Beispiel lokale räumliche Kohärenz innerhalb einer Szene auszunutzen oder um die Nutzung interner Caches zu optimieren.

[0262] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 in der integrierten Schaltung 11A und/oder 11B für Inferenz- oder Vorhersagevorgänge verwendet werden, die mindestens teilweise auf Gewichtungsparametern basieren, die unter Verwendung von Trainingsvorgängen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder hierin beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0263] **Fig. 18A-18B** veranschaulichen zusätzliche beispielhafte Grafikprozessorlogik gemäß hierin beschriebenen Ausführungsformen. **Fig. 18A** zeigt einen Grafikkern 1800, der in mindestens einer Ausführungsform im Grafikprozessor 1610 von **Fig. 16** enthalten sein kann und in mindestens einer Ausführungsform ein einheitlicher Shader-Kern 1755A-1755N wie in **Fig. 17B** sein kann. **Fig. 18B** veranschaulicht eine hochparallele Universal-Grafikverarbeitungseinheit (general-purpose graphics processing unit - „GPGPU“) 1830, die in mindestens einer Ausführungsform für den Einsatz auf einem Multi-Chip-Modul geeignet ist.

[0264] In mindestens einer Ausführungsform beinhaltet der Grafikkern 1800 einen gemeinsam genutzten Anweisungs-Cache 1802, eine Textureinheit 1818 und einen Cache/gemeinsam genutzten Speicher 1820, die den Ausführungsressourcen innerhalb des Grafikkerns 1800 gemeinsam sind. In mindestens einer Ausführungsform kann der Grafikkern 1800 mehrere Slices 1801A-1801N oder eine Partition für jeden Kern beinhalten und ein Grafikprozessor kann mehrere Instanzen des Grafikkerns 1800 beinhalten. In mindestens einer Ausführungsform können die Slices 1801A-1801N Unterstützungslogik beinhalten, die einen lokalen Anweisungs-Cache 1804A-1804N, einen Thread-Planer 1806A-1806N, einen Thread-Zuteiler 1808A-1808N und einen Satz von Registern 1810A-1810N beinhaltet. In mindestens einer Ausführungsform können die Slices 1801A - 1801N einen Satz zusätzlicher Funktionseinheiten (AFUs additional function units - AFUs) 1812A-1812N, Gleitkommaeinheiten (floating-point units - FPUs) 1814A-1814N, arithmetisch-logischer Einheiten für Ganzzahlen (arithmetic logic units - ALUs) 1816A-1816N, Adressberechnungseinheiten (address computational units - ACUs) 1813A-1813N, Gleitkommaeinheiten mit doppelter Genauigkeit (double-precision floating-point units - DPFPUs) 1815A-1815N und Matrixverarbeitungseinheiten (matrix processing units - MPUs) 1817A-1817N beinhalten.

[0265] In mindestens einer Ausführungsform können die FPU's 1814A-1814N Gleitkommaoperationen mit einfacher Genauigkeit (32 Bit) und halber Genauigkeit (16 Bit) durchführen, während die DPFPUs 1815A-1815N Gleitkommaoperationen mit doppelter Genauigkeit (64 Bit) durchführen. In mindestens einer Ausführungsform können die ALUs 1816A-1816N Integervorgänge mit variabler Genauigkeit bei 8-Bit-, 16-Bit- und 32-Bit-Genauigkeit durchführen und für Vorgänge mit gemischter Genauigkeit konfiguriert sein. In mindestens einer Ausführungsform können die MPU's 1817A-1817N auch für Matrixvorgänge mit gemischter Genauigkeit konfiguriert sein, einschließlich Fließkomma- und 8-Bit-Integervorgängen mit halber Genauigkeit. In mindestens einer Ausführungsform können die MPU's 1817-1817N eine Vielfalt von Matrixvorgängen durchführen, um Anwendungsrahmen für maschinelles Lernen zu beschleunigen, einschließlich der Ermöglichung der Unterstützung für eine beschleunigte allgemeine Matrix-zu-Matrix-Multiplikation (GEMM). In mindestens einer Ausführungsform können die AFUs 1812A-1812N zusätzliche logische Vorgänge durchführen, die von Fließkomma- oder Integereinheiten nicht unterstützt werden, einschließlich trigonometrischer Vorgänge (z. B. Sinus, Cosinus usw.).

[0266] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 in dem Grafikkern 1800 für Inferenz- oder Vorhersageoperationen mindestens zum Teil auf Grundlage von Gewichtsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder hierin beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0267] **Fig. 18B** veranschaulicht in mindestens einer Ausführungsform eine Universalverarbeitungseinheit (GPGPU) 1830, die so konfiguriert sein kann, dass hochparallele Rechenoperationen durch ein Array von Grafikverarbeitungseinheiten durchgeführt werden können. In mindestens einer Ausführungsform kann die GPGPU 1830 direkt mit anderen Instanzen der GPGPU 1830 verknüpft sein, um einen Mehr-GPU-Cluster zu erzeugen, um die Trainingsgeschwindigkeit für tiefe neuronale Netze zu verbessern. In mindestens einer Ausführungsform beinhaltet die GPGPU 1830 eine Host-Schnittstelle 1832, um eine Verbindung mit einem Hostprozessor zu ermöglichen. In mindestens einer Ausführungsform ist die Hostschnittstelle 1832 eine PCI-Express-Schnittstelle. In mindestens einer Ausführungsform kann die Hostschnittstelle 1832 eine anbieter-spezifische Kommunikationsschnittstelle oder Kommunikationsstruktur sein. In mindestens einer Ausführungsform empfängt die GPGPU 1830 Befehle von einem Host-Prozessor und verwendet einen globalen Planer 1834, um mit diesen Befehlen assoziierte Ausführungs-Threads an einen Satz von Rechenclustern 1836A-1836H zu senden. In mindestens einer Ausführungsform nutzen die Rechencluster 1836A-1836H einen Cache-Speicher 1838 gemeinsam. In mindestens einer Ausführungsform kann der Cachespeicher 1838 als übergeordneter Cache für Cachespeicher innerhalb der Rechencluster 1836A-1836H dienen.

[0268] In mindestens einer Ausführungsform beinhaltet die GPGPU 1830 Speicher 1844A-1844B, der über einen Satz von Speichersteuerungen 1842A-1842B an die Rechencluster 1836A-1836H gekoppelt ist. In mindestens einer Ausführungsform kann der Speicher 1844A-1844B verschiedene Typen von Speichervorrichtungen beinhalten, darunter dynamischer Direktzugriffsspeicher (DRAM) oder Grafik-Direktzugriffsspeicher, wie etwa synchroner Grafik-Direktzugriffsspeicher (SGRAM), darunter Grafik-Double-Data-Rate-(GDDR-) Speicher.

[0269] In mindestens einer Ausführungsform beinhalten die Rechencluster 1836A-1836H jeweils einen Satz von Grafikkernen, wie etwa den Grafikkern 1800 aus **Fig. 18A**, der mehrere Typen von Logikeinheiten für Integer und Gleitkommazahlen beinhalten kann, die Rechenoperationen mit einer Reihe von Genauigkeiten durchführen können, einschließlich solcher, die für Berechnungen des maschinellen Lernens geeignet sind. Zum Beispiel kann in mindestens einer Ausführungsform mindestens eine Teilmenge der Gleitkommaeinheiten in jedem der Rechencluster 1836A-1836H so konfiguriert sein, dass sie 16-Bit- oder 32-Bit-Gleitkommaoperationen durchführt, während eine andere Teilmenge der Gleitkommaeinheiten so konfiguriert sein kann, dass sie 64-Bit-Gleitkommaoperationen durchführt.

[0270] In mindestens einer Ausführungsform können mehrere Instanzen der GPGPU 1830 so konfiguriert sein, dass sie als Rechencluster arbeiten. In mindestens einer Ausführungsform variiert die Kommunikation, die durch die Rechencluster 1836A-1836H für die Synchronisation und den Datenaustausch verwendet wird, je nach Ausführungsform. In mindestens einer Ausführungsform kommunizieren mehrere Instanzen der GPGPU 1830 über die Hostschnittstelle 1832. In mindestens einer Ausführungsform beinhaltet die GPGPU 1830 einen E/A-Hub 1839, der die GPGPU 1830 mit einer GPU-Verknüpfung 1840 koppelt, die eine direkte

Verbindung zu anderen Instanzen der GPGPU 1830 ermöglicht. In mindestens einer Ausführungsform ist die GPU-Verknüpfung 1840 an eine dedizierte GPU-zu-GPU-Brücke gekoppelt, die Kommunikation und Synchronisation zwischen mehreren Instanzen der GPGPU 1830 ermöglicht. In mindestens einer Ausführungsform ist die GPU-Verknüpfung 1840 mit einer Hochgeschwindigkeitszusammenschaltung gekoppelt, um Daten an andere GPGPUs oder Parallelprozessoren zu übertragen und davon zu empfangen. In mindestens einer Ausführungsform befinden sich mehrere Instanzen der GPGPU 1830 in separaten Datenverarbeitungssystemen und sie kommunizieren über eine Netzvorrichtung, auf die über die Hostschnittstelle 1832 zugegriffen werden kann. In mindestens einer Ausführungsform GPU kann die Verknüpfung 1840 so konfiguriert sein dass eine Verbindung zu einem Host-Prozessor zusätzlich zu oder alternativ zu der Host-Schnittstelle 1832 ermöglicht wird.

[0271] In mindestens einer Ausführungsform kann die GPGPU 1830 dazu konfiguriert sein, neuronale Netze zu trainieren. In mindestens einer Ausführungsform kann die GPGPU 1830 innerhalb einer Inferenzplattform verwendet werden. In mindestens einer Ausführungsform, in der die GPGPU 1830 zum Inferenzieren verwendet wird, kann die GPGPU 1830 weniger Rechencluster 1836A-1836H beinhalten als in dem Fall, dass die GPGPU 1830 zum Trainieren eines neuronalen Netzes verwendet wird. In mindestens einer Ausführungsform kann sich die mit dem Speicher 1844A-1844B assoziierte Speichertechnologie zwischen Inferenz- und Trainingskonfigurationen unterscheiden, wobei den Trainingskonfigurationen Speichertechnologien mit höherer Bandbreite gewidmet sind. In mindestens einer Ausführungsform kann eine Inferenzkonfiguration der GPGPU 1830 inferenzspezifische Anweisungen unterstützen. In mindestens einer Ausführungsform kann eine Inferenzkonfiguration zum Beispiel Unterstützung für eine oder mehrere 8-Bit-Ganzzahl-Skalarprodukt-Anweisungen bereitstellen, die während der Inferenzoperationen für eingesetzte neuronale Netze verwendet werden können.

[0272] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 in der GPGPU 1830 für Inferenz- oder Vorhersageoperationen mindestens zum Teil auf Grundlage von Gewichtsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder hierin beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0273] **Fig. 19** ist ein Blockdiagramm, das ein Rechensystem 1900 gemäß mindestens einer Ausführungsform veranschaulicht. In mindestens einer Ausführungsform beinhaltet das Rechensystem 1900 ein Verarbeitungsteilsystem 1901, das einen oder mehrere Prozessor(en) 1902 und einen Systemspeicher 1904 aufweist, die über einen Zusammenschaltungspfad kommunizieren, der einen Speicher-Hub 1905 beinhalten kann. In mindestens einer Ausführungsform kann der Speicher-Hub 1905 eine separate Komponente innerhalb einer Chipsatzkomponente sein oder in einen oder mehrere Prozessoren 1902 integriert sein. In mindestens einer Ausführungsform ist der Speicher-Hub 1905 über eine Kommunikationsverknüpfung 1906 mit einem E/A-Teilsystem 1911 gekoppelt. In mindestens einer Ausführungsform beinhaltet das E/A-Teilsystem 1911 einen E/A-Hub 1907, der es dem Rechensystem 1900 ermöglichen kann, Eingaben von einer oder mehreren Eingabevorrichtung(en) 1908 zu empfangen. In mindestens einer Ausführungsform kann der E/A-Hub 1907 es einer Anzeigesteuerung, die in einem oder mehreren Prozessoren 1902 enthalten sein kann, ermöglichen, Ausgaben für eine oder mehrere Anzeigevorrichtungen 1910 A bereitzustellen. In mindestens einer Ausführungsform können eine oder mehrere mit dem E/A-Hub 1907 gekoppelte Anzeigevorrichtungen 1910A eine lokale, interne oder eingebettete Anzeigevorrichtung beinhalten.

[0274] In mindestens einer Ausführungsform beinhaltet das Verarbeitungsteilsystem 1901 einen oder mehrere Parallelprozessoren 1912, die über einen Bus oder eine andere Kommunikationsverknüpfung 1913 an den Speicher-Hub 1905 gekoppelt sind. In mindestens einer Ausführungsform kann die Kommunikationsverknüpfung 1913 eine/eines von einer beliebigen Anzahl von standardbasierten Kommunikationsverknüpfungstechnologien oder -protokollen verwenden, wie etwa, aber nicht beschränkt auf, PCI Express oder eine anbieterspezifische Kommunikationsschnittstelle oder Kommunikationsstruktur sein. In mindestens einer Ausführungsform bilden ein oder mehrere Parallelprozessor(en) 1912 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 der Parallelprozessor(en) 1912 ein Grafikverarbeitungsteilsystem, das Pixel an eine oder mehrere Anzeigevorrichtung(en) 1910A ausgeben kann, die über den E/A-Hub 1907 gekoppelt sind. In mindestens einer Ausführungsform können der oder die Parallel-

prozessoren 1912 zudem eine Anzeigesteuerung und eine Anzeigeschnittstelle (nicht gezeigt) beinhalten, um eine direkte Verbindung zu einer oder mehreren Anzeigevorrichtungen 1910B zu ermöglichen.

[0275] In mindestens einer Ausführungsform kann eine Systemspeichereinheit 1914 mit dem E/A-Hub 1907 verbunden sein, um einen Speichermechanismus für das Rechensystem 1900 bereitzustellen. In mindestens einer Ausführungsform kann ein E/A-Switch 1916 verwendet werden, um einen Schnittstellenmechanismus bereitzustellen, der Verbindungen zwischen dem E/A-Hub 1907 und anderen Komponenten ermöglicht, wie etwa einem Netzadapter 1918 und/oder einem drahtlosen Netzadapter 1919, die in eine Plattform integriert werden können, sowie verschiedenen anderen Vorrichtungen, die über eine oder mehrere Erweiterungsvorrichtung(en) 1920 hinzugefügt werden können. In mindestens einer Ausführungsform kann der Netzadapter 1918 ein Ethernet-Adapter oder ein anderer drahtgebundener Netzadapter sein. In mindestens einer Ausführungsform kann der drahtlose Netzadapter 1919 eine oder mehrere Wi-Fi-, Bluetooth-, Nahfeldkommunikations- (NFC-) oder eine andere Netzvorrichtung beinhalten, die ein oder mehrere drahtlose Funkgeräte beinhaltet.

[0276] In mindestens einer Ausführungsform kann das Rechensystem 1900 andere, nicht explizit gezeigte Komponenten beinhalten, einschließlich USB- oder anderer Portverbindungen, optischer Speicherlaufwerke, Videoaufnahmegeräten und dergleichen, die ebenfalls mit dem E/A-Hub 1907 verbunden sein können. In mindestens einer Ausführungsform können die Kommunikationspfade, die verschiedene Komponenten in **Fig. 19** miteinander verbinden, unter Verwendung beliebiger geeigneter Protokolle implementiert werden, z. B. auf PCI (Peripheral Component Interconnect) basierende Protokolle (z. B. PCI-Express) oder andere Bus- oder Punkt-zu-Punkt-Kommunikationsschnittstellen und/oder -protokoll(e), z. B. NV-Link High-Speed-Interconnect- oder -Interconnect-Protokolle.

[0277] In mindestens einer Ausführungsform beinhalten der oder die Parallelprozessoren 1912 Schaltungen, die für die Grafik- und Videoverarbeitung optimiert sind, einschließlich zum Beispiel Videoausgabeschaltungen, und sie stellen eine Grafikverarbeitungseinheit (GPU) dar. In mindestens einer Ausführungsform beinhalten der oder die Parallelprozessoren 1912 Schaltungen, die für Universalverarbeitung optimiert sind. In mindestens einer Ausführungsform können die Komponenten des Rechensystems 1900 in ein oder mehrere andere Systemelemente auf einer einzelnen integrierten Schaltung integriert sein. Zum Beispiel können in mindestens einer Ausführungsform der oder die Parallelprozessoren 1912, der Speicher-Hub 1905, der oder die Prozessoren 1902 und der E/A-Hub 1907 in eine integrierte Schaltung als System auf einem Chip (SoC) integriert sein. In mindestens einer Ausführungsform können die Komponenten des Rechensystems 1900 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 1900 in ein Multi-Chip-Modul (multi-chip module - MCM) integriert sein, das mit anderen Multi-Chip-Modulen zu einem modularen Rechensystem zusammengeschaltet sein kann.

[0278] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 in dem Rechensystem 1900 aus **Fig. 19** für Inferenz- oder Vorhersageoperationen mindestens zum Teil auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder hierin beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

PROZESSOREN

[0279] **Fig. 20A** veranschaulicht einen Parallelprozessor 2000 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform können verschiedene Komponenten des Parallelprozessors 2000 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 2000 eine Variante eines oder mehrerer Parallelprozessoren 1912, die in **Fig. 19** gemäß einer beispielhaften Ausführungsform gezeigt sind.

[0280] In mindestens einer Ausführungsform beinhaltet der Parallelprozessor 2000 eine Parallelverarbeitungseinheit 2002. In mindestens einer Ausführungsform beinhaltet die Parallelverarbeitungseinheit 2002 eine E/A-Einheit 2004, die die Kommunikation mit anderen Geräten ermöglicht, einschließlich anderer Instan-

zen der Parallelverarbeitungseinheit 2002. In mindestens einer Ausführungsform kann die E/A-Einheit 2004 direkt mit anderen Vorrichtungen verbunden sein. In mindestens einer Ausführungsform ist die E/A-Einheit 2004 über die Verwendung einer Hub- oder Switch-Schnittstelle, wie etwa eines Speicher-Hubs 2005, mit anderen Vorrichtungen verbunden. In mindestens einer Ausführungsform bilden Verbindungen zwischen dem Speicher-Hub 2005 und der E/A-Einheit 2004 eine Kommunikationsverknüpfung 2013. In mindestens einer Ausführungsform ist die E/A-Einheit 2004 mit einer Host-Schnittstelle 2006 und einer Speicherkreuzschiene 2016 verbunden, wobei die Hostschnittstelle 2006 Befehle empfängt, die auf das Durchführen von Verarbeitungsoperationen gerichtet sind, und die Speicherkreuzschiene 2016 Befehle empfängt, die auf das Durchführen von Speicheroperationen gerichtet sind.

[0281] In mindestens einer Ausführungsform kann, wenn die Host-Schnittstelle 2006 einen Befehlspuffer über die E/A-Einheit 2004 empfängt, die Host-Schnittstelle 2006 Arbeitsoperationen zum Durchführen dieser Befehle an ein Frontend 2008 richten. In mindestens einer Ausführungsform ist das Frontend 2008 mit einem Planer 2010 gekoppelt, der so konfiguriert ist, dass er Befehle oder andere Arbeitselemente an ein Verarbeitungsclusterarray 2012 verteilt. In mindestens einer Ausführungsform stellt der Planer 2010 sicher, dass das Verarbeitungsclusterarray 2012 geeignet konfiguriert ist und sich in einem gültigen Zustand befindet, bevor Tasks an ein Cluster eines Verarbeitungsclusterarrays 2012 verteilt werden. In mindestens einer Ausführungsform ist der Planer 2010 über Firmware-Logik implementiert, die auf einem Mikrocontroller ausgeführt wird. In mindestens einer Ausführungsform ist der Mikrocontroller-implementierte Planer 2010 so konfigurierbar, dass er komplexe Planungs- und Arbeitsverteilungsoperationen mit grober und feiner Granularität durchführt, was eine schnelle Präemption und Kontextumschaltung von Threads ermöglicht, die auf dem Verarbeitungsarray 2012 ausgeführt werden. In mindestens einer Ausführungsform kann die Hostsoftware Workloads für die Planung auf dem Verarbeitungsclusterarray 2012 über einen von mehreren Grafikverarbeitungs-pfaden nachweisen. In mindestens einer Ausführungsform können die Workloads dann durch die Logik des Planers 2010 innerhalb eines Mikrocontrollers, der den Planer 2010 beinhaltet, automatisch auf das Verarbeitungsarraycluster 2012 verteilt werden.

[0282] In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2012 bis zu „N“ Verarbeitungscluster beinhalten (z. B. Cluster 2014 A, Cluster 2014 B bis Cluster 2014 N), wobei „N“ eine positive ganze Zahl darstellt (die eine andere ganze Zahl „N“ sein kann, als in anderen Figuren verwendet). In mindestens einer Ausführungsform kann jedes Cluster 2014A-2014N des Verarbeitungsclusterarrays 2012 eine große Anzahl von nebenläufigen Threads ausführen. In mindestens einer Ausführungsform kann der Planer 2010 den Clustern 2014A-2014N des Verarbeitungsclusterarrays 2012 Arbeit zuweisen, indem er verschiedene Planungs- und/oder Arbeitsverteilungsalgorithmen verwendet, die in Abhängigkeit vom Workload variieren können, die für jeden Typ von Programm oder Berechnung entsteht. In mindestens einer Ausführungsform kann die Planung dynamisch durch den Planer 2010 gehandhabt werden oder kann teilweise durch Compiler-Logik während der Kompilierung der Programmlogik unterstützt werden, die für die Ausführung durch die Verarbeitungsclusteranordnung 2012 konfiguriert ist. In mindestens einer Ausführungsform können unterschiedliche Cluster 2014A-2014N des Verarbeitungsclusterarrays 2012 zum Verarbeiten unterschiedlicher Programmtypen oder zum Durchführen unterschiedlicher Berechnungstypen zugewiesen sein.

[0283] In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2012 so konfiguriert sein, dass es verschiedene Typen von Parallelverarbeitungsoperationen durchführt. In mindestens einer Ausführungsform ist das Verarbeitungsclusterarray 2012 so konfiguriert, dass es Universal-Parallelberechnungsoperationen durchführt. In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2012 zum Beispiel Logik zum Ausführen von Verarbeitungs-Tasks beinhalten, einschließlich des Filterns von Video- und/oder Audiodaten, des Durchführens von Modellierungsoperationen, einschließlich Physikoperationen, und des Durchführens von Datentransformationen.

[0284] In mindestens einer Ausführungsform ist das Verarbeitungsclusterarray 2012 so konfiguriert, dass es Parallelgrafikverarbeitungsoperationen durchführt. In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2012 zusätzliche Logik beinhalten, um die Ausführung derartiger Grafikverarbeitungsoperationen zu unterstützen, einschließlich, aber nicht beschränkt auf, Texturabastlogik, um Texturoperationen durchzuführen, sowie Tessellierungslogik und andere Scheitelpunktverarbeitungslogik. In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2012 so konfiguriert sein, dass es grafikverarbeitungsbezogene Shader-Programme ausführt, wie etwa, aber nicht beschränkt auf, Scheitelpunkt-Shader, Tessellierungs-Shader, Geometrie-Shader und Pixel-Shader. In mindestens einer Ausführungsform kann die Parallelverarbeitungseinheit 2002 Daten aus dem Systemspeicher über die E/A-Einheit 2004 zur Verarbeitung übermitteln. In mindestens einer Ausführungsform können während der Verarbeitung die übermittelten Daten

in einem chipinternen Speicher (z. B. dem Parallelprozessorspeicher 2022) während der Verarbeitung gespeichert und dann wieder in den Systemspeicher geschrieben werden.

[0285] Wenn die Parallelverarbeitungseinheit 2002 zum Durchführen der Grafikverarbeitung verwendet wird, kann der Planer 2010 in mindestens einer Ausführungsform so konfiguriert sein, dass er eine Verarbeitungsarbeitslast in annähernd gleich große Tasks aufteilt, um eine bessere Verteilung der Grafikverarbeitungsoperationen auf mehrere Cluster 2014A-2014N des Verarbeitungsclusterarrays 2012 zu ermöglichen. In mindestens einer Ausführungsform können Abschnitte des Verarbeitungsclusterarrays 2012 so konfiguriert sein, dass sie unterschiedliche Verarbeitungstypen durchführen. Zum Beispiel kann in mindestens einer Ausführungsform ein erster Abschnitt so konfiguriert sein, dass er Scheitelpunkt-Shading und Topologieerzeugung durchführt, kann ein zweiter Abschnitt so konfiguriert sein, dass er Tesselations- und Geometrie-Shading durchführt, und kann ein dritter Abschnitt so konfiguriert sein, dass er Pixel-Shading oder andere Bildschirmraumvorgänge durchführt, um ein gerendertes Bild zur Anzeige zu produzieren. In mindestens einer Ausführungsform können Zwischendaten, die von einem oder mehreren Clustern 2014A-2014N produziert werden, in Puffern gespeichert werden, um zu ermöglichen, dass die Zwischendaten zur weiteren Verarbeitung zwischen den Clustern 2014A-2014N übertragen werden.

[0286] In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2012 auszuführende Verarbeitungs-Tasks über den Planer 2010 empfangen, der von dem Frontend 2008 Befehle empfängt, die Verarbeitungs-Tasks definieren. In mindestens einer Ausführungsform können die Verarbeitungs-Tasks Indizes der zu verarbeitenden Daten beinhalten, z. B. Oberflächen(-Patch)-Daten, Primitivdaten, Scheitelpunkt-Daten und/oder Pixeldaten, sowie Statusparameter und Befehle, die definieren, wie die Daten verarbeitet werden sollen (z. B. welches Programm ausgeführt werden soll). In mindestens einer Ausführungsform kann der Planer 2010 so konfiguriert sein, dass er den Tasks entsprechende Indizes abrufen, oder er kann Indizes von dem Frontend 2008 empfangen. In mindestens einer Ausführungsform kann das Frontend 2008 so konfiguriert sein, dass es sicherstellt, dass die Verarbeitungsclusteranordnung 2012 in einen gültigen Status konfiguriert wird, bevor eine durch eingehende Befehlspuffer (z. B. Batch-Puffer, Push-Puffer usw.) spezifizierte Arbeitslast initiiert wird.

[0287] In mindestens einer Ausführungsform kann jede von einer oder mehreren Instanzen der Parallelverarbeitungseinheit 2002 mit einem Parallelprozessorspeicher 2022 gekoppelt sein. In mindestens einer Ausführungsform kann auf den Parallelprozessorspeicher 2022 über die Speicherkreuzschiene 2016 zugegriffen werden, die Speicheranforderungen von dem Verarbeitungsclusterarray 2012 sowie von der E/A-Einheit 2004 empfangen kann. In mindestens einer Ausführungsform kann die Speicherkreuzschiene 2016 über eine Speicherschnittstelle 2018 auf den Parallelprozessorspeicher 2022 zugreifen. In mindestens einer Ausführungsform kann die Speicherschnittstelle 2018 mehrere Partitionseinheiten (z. B. Partitionseinheit 2020 A, Partitionseinheit 2020 B bis Partitionseinheit 2020 N) beinhalten, die jeweils an einen Abschnitt (z. B. Speichereinheit) des Parallelprozessorspeichers 2022 gekoppelt werden können. In mindestens einer Ausführungsform ist eine Anzahl der Partitionseinheiten 2020A-2020N so konfiguriert, dass sie gleich einer Anzahl von Speichereinheiten ist, sodass eine erste Partitionseinheit 2020A eine entsprechende erste Speichereinheit 2024A aufweist, eine zweite Partitionseinheit 2020B eine entsprechende Speichereinheit 2024B aufweist und eine N-te Partitionseinheit 2020N eine entsprechende N-te Speichereinheit 2024N aufweist. In mindestens einer Ausführungsform kann eine Anzahl der Partitionseinheiten 2020A-2020N nicht gleich einer Anzahl der Speichereinheiten sein.

[0288] In mindestens einer Ausführungsform können die Speichereinheiten 2024A-2024N verschiedene Typen von Speichervorrichtungen beinhalten, darunter dynamischer Direktzugriffsspeicher (DRAM) oder Grafik-Direktzugriffsspeicher, wie etwa synchroner Grafik-Direktzugriffsspeicher (SGRAM), darunter Grafik-Double-Data-Rate-(GDDR-)Speicher. In mindestens einer Ausführungsform können die Speichereinheiten 2024A-2024N 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 Bildspeicher oder Texturkarten, über die Speichereinheiten 2024A-2024N hinweg gespeichert werden, was es den Partitionseinheiten 2020A-2020N ermöglicht, Abschnitte jedes Rendering-Ziels parallel zu schreiben, um die verfügbare Bandbreite des Parallelprozessorspeichers 2022 effizient zu nutzen. In mindestens einer Ausführungsform kann eine lokale Instanz des Parallelprozessorspeichers 2022 zugunsten einer einheitlichen Speicherausgestaltung ausgeschlossen werden, die Systemspeicher in Verbindung mit lokalem Cachespeicher nutzt.

[0289] In mindestens einer Ausführungsform kann ein beliebiger der Cluster 2014A-2014N des Verarbeitungsclusterarrays 2012 Daten verarbeiten, die in beliebige der Speichereinheiten 2024A-2024N innerhalb

des Parallelprozessorspeichers 2022 geschrieben werden. In mindestens einer Ausführungsform kann die Speicherkreuzschiene 2016 so konfiguriert sein, dass sie eine Ausgabe jedes Clusters 2014A-2014N an eine beliebige Partitionseinheit 2020A-2020N oder an einen anderen Cluster 2014A-2014N übermittelt, der zusätzliche Verarbeitungsoperationen an einer Ausgabe ausführen kann. In mindestens einer Ausführungsform kann jeder Cluster 2014A-2014N durch die Speicherkreuzschiene 2016 mit der Speicherschnittstelle 2018 kommunizieren, um aus verschiedenen externen Speichervorrichtungen zu lesen oder in diese zu schreiben. In mindestens einer Ausführungsform weist die Speicherkreuzschiene 2016 eine Verbindung mit der Speicherschnittstelle 2018 auf, um mit der E/A-Einheit 2004 zu kommunizieren, sowie eine Verbindung mit einer lokalen Instanz des Parallelprozessorspeichers 2022, was es den Verarbeitungseinheiten innerhalb der unterschiedlichen Verarbeitungscluster 2014A-2014N ermöglicht, mit Systemspeicher oder anderem Speicher zu kommunizieren, der nicht lokal zu der Parallelverarbeitungseinheit 2002 ist. In mindestens einer Ausführungsform kann die Speicherkreuzschiene 2016 virtuelle Kanäle verwenden, um Verkehrsströme zwischen Clustern 2014A-2014N und Partitionseinheiten 2020A-2020N zu trennen.

[0290] In mindestens einer Ausführungsform können mehrere Instanzen der Parallelverarbeitungseinheit 2002 auf einer einzelnen Erweiterungskarte bereitgestellt sein oder mehrere Erweiterungskarten zusammengeschaltet sein. In mindestens einer Ausführungsform können unterschiedliche Instanzen der Parallelverarbeitungseinheit 2002 so konfiguriert sein, dass sie zusammenarbeiten, 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 2002 Gleitkommaeinheiten mit höherer Genauigkeit in Bezug auf andere Instanzen beinhalten. In mindestens einer Ausführungsform können Systeme, die eine oder mehrere Instanzen der Parallelverarbeitungseinheit 2002 oder des Parallelprozessors 2000 enthalten, in einer Vielfalt von Konfigurationen und Formfaktoren implementiert sein, einschließlich, aber nicht beschränkt auf Desktop-, Laptop- oder tragbare persönliche Computer, Server, Arbeitsstationen, Spielekonsolen und/oder eingebettete Systeme.

[0291] **Fig. 20 B** ist ein Blockdiagramm einer Partitionseinheit 2020 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist die Partitionseinheit 2020 eine Instanz einer der Partitionseinheiten 2020A-2020N aus **Fig. 20 A**. In mindestens einer Ausführungsform beinhaltet die Partitionseinheit 2020 einen L2-Cache 2021, eine Bildspeicherschnittstelle 2025 und eine ROP 2026 (raster operations unit - Rasteroperationeneinheit). In mindestens einer Ausführungsform ist der L2-Cache 2021 ein Lese-/Schreib-Cache, der so konfiguriert ist, dass er Lade- und Sicherungsoperationen durchführt, die von der Speicherkreuzschiene 2016 und der ROP 2026 empfangen werden. In mindestens einer Ausführungsform werden Lesefehler und dringende Rückschreibanforderungen durch den L2-Cache 2021 an die Bildspeicherschnittstelle 2025 zur Verarbeitung ausgegeben. In mindestens einer Ausführungsform können Aktualisierungen auch über die Bildspeicherschnittstelle 2025 zur Verarbeitung an einen Bildspeicher gesendet werden. In mindestens einer Ausführungsform bildet die Bildspeicherschnittstelle 2025 eine Schnittstelle mit einer der Speichereinheiten in dem Parallelprozessorspeicher, wie etwa mit den Speichereinheiten 2024A-2024N aus **Fig. 20** (z. B. innerhalb des Parallelprozessorspeichers 2022).

[0292] In mindestens einer Ausführungsform ist die ROP 2026 eine Verarbeitungseinheit, die Rasteroperationen durchführt, wie etwa Schablone, Z-Test, Blending usw. In mindestens einer Ausführungsform gibt die ROP 2026 dann verarbeitete Grafikdaten aus, die in Grafikspeicher gespeichert werden. In mindestens einer Ausführungsform beinhaltet die ROP 2026 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 eine verlustfreie Komprimierungslogik sein, die einen oder mehrere von mehreren Komprimierungsalgorithmen verwendet. In mindestens einer Ausführungsform kann ein Typ der Komprimierung, die durch die ROP 2026 durchgeführt wird, 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 einer Kachelbasis durchgeführt.

[0293] In mindestens einer Ausführungsform ist die ROP 2026 innerhalb jedes Verarbeitungsclusters (z. B. Cluster 2014A-2014N aus **Fig. 20A**) statt innerhalb der Partitionseinheit 2020 enthalten. In mindestens einer Ausführungsform werden Lese- und Schreibanforderungen für Pixeldaten anstelle von Pixelfragmentdaten über die Speicherkreuzschiene 2016 übertragen. In mindestens einer Ausführungsform können verarbeitete Grafikdaten auf einem Anzeigegerät angezeigt werden, wie z. B. einem von einem oder mehreren Anzeigegeräten 1910 von **Fig. 19**, zur weiteren Verarbeitung durch Prozessor(en) 1302 weitergeleitet werden oder

zur weiteren Verarbeitung durch eine der Verarbeitungseinheiten innerhalb des Parallelprozessors 2000 von **Fig. 20 A**.

[0294] **Fig. 20C** ist ein Blockdiagramm eines Verarbeitungsclusters 2014 innerhalb einer Parallelverarbeitungseinheit gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist ein Verarbeitungscluster eine Instanz eines der Verarbeitungscluster 2014A-2014N aus **Fig. 20 A**. In mindestens einer Ausführungsform kann das Verarbeitungscluster 2014 so konfiguriert sein, dass es viele Threads parallel ausführt, wobei sich „Thread“ auf eine Instanz eines konkreten Programms bezieht, die an einem konkreten Satz von Eingabedaten ausgeführt wird. In mindestens einer Ausführungsform werden SIMD(Single-Instruction, Multiple-Data)-Anweisungsausgabetechniken 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 so konfiguriert ist, dass sie Anweisungen an einen Satz von Verarbeitungs-Engines innerhalb jedes der Verarbeitungscluster ausgibt.

[0295] In mindestens einer Ausführungsform kann der Betrieb des Verarbeitungsclusters 2014 über einen Pipelineverwalter 2032 gesteuert werden, der die Verarbeitungs-Tasks auf die SIMT-Parallelprozessoren verteilt. In mindestens einer Ausführungsform empfängt der Pipelineverwalter 2032 Anweisungen von dem Planner 2010 aus **Fig. 20 A** und verwaltet die Ausführung dieser Anweisungen über einen Grafik-Multiprozessor 2034 und/oder eine Textureinheit 2036. In mindestens einer Ausführungsform ist der Grafik-Multiprozessor 2034 eine beispielhafte Instanz eines SIMT-Parallelprozessors. In mindestens einer Ausführungsform können jedoch verschiedene Typen von SIMT-Parallelprozessoren mit unterschiedlichen Architekturen innerhalb des Verarbeitungsclusters 2014 enthalten sein. In mindestens einer Ausführungsform können eine oder mehrere Instanzen des Grafik-Multiprozessors 2034 innerhalb eines Verarbeitungsclusters 2014 enthalten sein. In mindestens einer Ausführungsform kann der Grafik-Multiprozessor 2034 Daten verarbeiten und eine Datenkreuzschiene 2040 kann verwendet werden, um verarbeitete Daten an eines von mehreren möglichen Zielen, einschließlich anderer Shader-Einheiten, zu verteilen. In mindestens einer Ausführungsform kann der Pipelineverwalter 2032 die Verteilung von verarbeiteten Daten erleichtern, indem er Ziele für zu verteilende verarbeitete Daten über die Datenkreuzschiene 2040 vorgibt.

[0296] In mindestens einer Ausführungsform kann jeder Grafik-Multiprozessor 2034 innerhalb des Verarbeitungsclusters 2014 einen identischen Satz von funktioneller Ausführungslogik (z. B. arithmetisch-logische Einheiten, Lade-Speicher-Einheiten usw.) beinhalten. In mindestens einer Ausführungsform kann die 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 Vorgängen, darunter Integer- und Fließkommaarithmetik, Vergleichsvorgänge, boolesche Vorgänge, Bitverschiebung und Berechnung verschiedener algebraischer Funktionen. In mindestens einer Ausführungsform kann dieselbe Hardware einer funktionellen Einheit ausgenutzt werden, um unterschiedliche Vorgänge durchzuführen, und eine beliebige Kombination von funktionellen Einheiten vorhanden sein.

[0297] In mindestens einer Ausführungsform stellen die an den Verarbeitungscluster 2014 übertragenen Anweisungen einen Thread dar. In mindestens einer Ausführungsform ist ein Satz von Threads, der über einen Satz von Parallelverarbeitungs-Engines ausgeführt werden, eine Thread-Gruppe. In mindestens einer Ausführungsform führt eine Thread-Gruppe ein gemeinsames Programm an unterschiedlichen Eingabedaten aus. In mindestens einer Ausführungsform kann jeder Thread innerhalb einer Thread-Gruppe einer anderen Verarbeitungs-Engine innerhalb eines Grafik-Multiprozessors 2034 zugeordnet sein. In mindestens einer Ausführungsform kann eine Thread-Gruppe weniger Threads beinhalten als eine Anzahl von Verarbeitungs-Engines innerhalb des Grafik-Multiprozessors 2034. Wenn eine Thread-Gruppe weniger Threads beinhaltet als eine Anzahl von Verarbeitungs-Engines, können in mindestens einer Ausführungsform eine oder mehrere der 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 2034. Wenn eine Thread-Gruppe mehr Threads als eine Anzahl von Verarbeitungs-Engines innerhalb des Grafik-Multiprozessors 2034 beinhaltet, kann das Verarbeiten in mindestens einer Ausführungsform über aufeinanderfolgende Taktzyklen durchgeführt werden. In mindestens einer Ausführungsform können mehrere Thread-Gruppen nebenläufig auf einem Grafik-Multiprozessor 2034 ausgeführt werden.

[0298] In mindestens einer Ausführungsform beinhaltet der Grafik-Multiprozessor 2034 einen internen Cachespeicher zum Durchführen von Lade- und Speicheroperationen. In mindestens einer Ausführungsform kann der Grafik-Multiprozessor 2034 auf einen internen Cache verzichten und einen Cachespeicher (z. B. L1-Cache 2048) innerhalb des Verarbeitungsclusters 2014 verwenden. In mindestens einer Ausführungsform hat jeder Grafik-Multiprozessor 2034 auch Zugriff auf L2-Caches innerhalb von Partitionseinheiten (z. B. Partitionseinheiten 2020A-2020N von **Fig. 20A**), die von allen Verarbeitungsclustern 2014 gemeinsam genutzt werden und zur Datenübertragung zwischen Threads verwendet werden können. In mindestens einer Ausführungsform kann der Grafikmultiprozessor 2034 auch auf den chipexternen globalen Speicher zugreifen, der einen oder mehrere von dem lokalen Parallelprozessorspeicher und/oder Systemspeicher beinhalten kann. In mindestens einer Ausführungsform kann ein beliebiger Speicher, der außerhalb der Parallelverarbeitungseinheit 2002 vorliegt, als globaler Speicher verwendet werden. In mindestens einer Ausführungsform beinhaltet der Verarbeitungscluster 2014 mehrere Instanzen des Grafik-Multiprozessors 2034 und er kann gemeinsame Anweisungen und Daten teilen, die in dem L1-Cache 2048 gespeichert sein können.

[0299] In mindestens einer Ausführungsform kann jeder Verarbeitungscluster 2014 eine MMU 2045 (Speicherverwaltungseinheit) beinhalten, die so konfiguriert ist, dass sie virtuelle Adressen auf physische Adressen abbildet. In mindestens einer Ausführungsform können sich eine oder mehrere Instanzen der MMU 2045 innerhalb der Speicherschnittstelle 2018 aus **Fig. 20 A**. In mindestens einer Ausführungsform beinhaltet die MMU 2045 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 Cachezeilenindex. In mindestens einer Ausführungsform kann die MMU 2045 Adressenübersetzungspuffer (TLB) oder Caches beinhalten, die sich innerhalb des Grafik-Multiprozessors 2034 oder L1-Cache 2048 oder Verarbeitungsclusters 2014 befinden können. In mindestens einer Ausführungsform wird eine physische Adresse verarbeitet, um den Oberflächendatenzugriff lokal zu verteilen, um eine effiziente Anforderungsverschachtelung zwischen den Partitionseinheiten zu ermöglichen. In mindestens einer Ausführungsform kann ein Cachezeilenindex verwendet werden, um zu bestimmen, ob eine Anforderung für eine Cachezeile ein Treffer oder ein Fehler ist.

[0300] In mindestens einer Ausführungsform kann ein Verarbeitungscluster 2014 derart konfiguriert sein, dass jeder Grafik-Multiprozessor 2034 an eine Textureinheit 2036 zum Durchführen von Texturabbildungsoperationen gekoppelt ist, z. B. zum Bestimmen von Texturab tastpositionen, Lesen von Texturdaten und Filtern von Texturdaten. In mindestens einer Ausführungsform werden die Texturdaten aus einem internen Texture-L1-Cache (nicht gezeigt) oder aus einem L1-Cache innerhalb des Grafik-Multiprozessors 2034 gelesen und je nach Bedarf aus einem L2-Cache, dem lokalen Parallelprozessorspeicher oder dem Systemspeicher abgerufen. In mindestens einer Ausführungsform gibt jeder Grafik-Multiprozessor 2034 verarbeitete Tasks an die Datenkreuzschiene 2040 aus, um einen verarbeiteten Task einem anderen Verarbeitungscluster 2014 zur weiteren Verarbeitung bereitzustellen oder um einen verarbeiteten Task über die Speicherkreuzschiene 2016 in einem L2-Cache, lokalen Parallelprozessorspeicher oder Systemspeicher zu speichern. In mindestens einer Ausführungsform ist eine preROP 2042 (Vor-Rasteroperationeneinheit) so konfiguriert, dass sie Daten von dem Grafik-Multiprozessor 2034 empfängt und Daten an ROP-Einheiten leitet, die sich in den hierin beschriebenen Partitionseinheiten befinden können (z. B. Partitionseinheiten 2020A-2020N aus **Fig. 20 A**). In mindestens einer Ausführungsform kann die preROP-Einheit 2042 Optimierungen für die Farbmischung, das Organisieren von Pixelfarbdaten und das Durchführen von Adressübersetzungen durchführen.

[0301] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 in dem Grafikverarbeitungscluster 2014 für Inferenz- oder Vorhersageoperationen mindestens zum Teil auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder hierin beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0302] **Fig. 20D** zeigt einen Grafik-Multiprozessor 2034 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist der Grafik-Multiprozessor 2034 mit dem Pipelineverwalter 2032 des Verarbeitungsclusters 2014 gekoppelt. In mindestens einer Ausführungsform weist der Grafik-Multiprozessor 2034 eine Ausführungspipeline auf, die einen Anweisungscache 2052, eine Anweisungseinheit 2054, eine Adressabbildungseinheit 2056, eine Registerbank 2058, einen oder mehrere Kerne 2062 einer Universal-Grafikverarbeitungseinheit (GPGPU) und eine oder mehrere Lade-/Speichereinheiten 2066 beinhaltet, ohne darauf beschränkt zu sein. In mindestens einer Ausführungsform sind die GPGPU-Kerne 2062 und die Lade-/Spei-

chereinheiten 2066 über eine Speicher- und Cache-Verbindung 2068 an den Cachespeicher 2072 und den gemeinsam genutzten Speicher 2070 gekoppelt.

[0303] In mindestens einer Ausführungsform empfängt der Anweisungs-Cache 2052 einen Strom aus auszuführenden Anweisungen von dem Pipelineverwalter 2032. In mindestens einer Ausführungsform werden die Anweisungen in dem Anweisungscache 2052 zwischengespeichert und durch eine Anweisungseinheit 2054 zur Ausführung zugeteilt. In mindestens einer Ausführungsform kann die Anweisungseinheit 2054 Anweisungen als Thread-Gruppen (z. B. Warps) zuteilen, wobei jeder Thread der Thread-Gruppe einer anderen Ausführungseinheit innerhalb der GPGPU-Kerne 2062 zugeordnet wird. 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 einheitlichen Adressraums spezifiziert. In mindestens einer Ausführungsform kann die Adressabbildungseinheit 2056 verwendet werden, um Adressen in einem einheitlichen Adressraum in eine eindeutige Speicheradresse zu übersetzen, auf die durch die Lade-/Speichereinheiten 2066 zugegriffen werden kann.

[0304] In mindestens einer Ausführungsform stellt die Registerbank 2058 einen Satz von Registern für funktionelle Einheiten des Grafik-Multiprozessors 2034 bereit. In mindestens einer Ausführungsform stellt die Registerbank 2058 einen temporären Datenspeicher für Operanden bereit, die mit Datenpfaden von funktionellen Einheiten (z. B. GPGPU-Kernen 2062, Lade-/Speichereinheiten 2066) des Grafik-Multiprozessors 2034 verbunden sind. In mindestens einer Ausführungsform ist die Registerbank 2058 derart zwischen den einzelnen funktionellen Einheiten aufgeteilt, dass jeder funktionellen Einheit ein dedizierter Abschnitt der Registerbank 2058 zugewiesen ist. In mindestens einer Ausführungsform ist die Registerdatei 2058 auf unterschiedliche Warps aufgeteilt, die vom Grafikmultiprozessor 2034 ausgeführt werden.

[0305] In mindestens einer Ausführungsform können die GPGPU-Kerne 2062 jeweils Gleitkommaeinheiten (FPUs) und/oder Ganzzahlarithmetiklogikeinheiten (ALUs) beinhalten, die verwendet werden, um Anweisungen des Grafikmultiprozessors 2034 auszuführen. In mindestens einer Ausführungsform können die GPGPU-Kerne 2062 eine ähnliche Architektur aufweisen oder sich bezüglich der Architektur unterscheiden. In mindestens einer Ausführungsform beinhaltet ein erster Teil von GPGPU-Kernen 2062 eine FPU mit einfacher Genauigkeit und eine ganzzahlige ALU, während ein zweiter Teil von GPGPU-Kernen 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 2034 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 einer oder mehrere der GPGPU-Kerne 2062 auch Fest- oder Spezialfunktionslogik beinhalten.

[0306] In mindestens einer Ausführungsform beinhalten die GPGPU-Kerne 2062 SIMD-Logik, die dazu in der Lage ist, eine einzelne Anweisung an mehreren Datensätzen durchzuführen. In mindestens einer Ausführungsform können die GPGPU-Kerne 2062 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 von einem 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.

[0307] In mindestens einer Ausführungsform ist die Speicher- und Cache-Zusammenschaltung 2068 ein Zusammenschaltungsnetz, das jede funktionelle Einheit des Grafikmultiprozessors 2034 mit der Registerbank 2058 und dem gemeinsam genutzten Speicher 2070 verbindet. In mindestens einer Ausführungsform ist die Speicher- und Cache-Zusammenschaltung 2068 eine Kreuzschienen-Zusammenschaltung, die es der Lade-/Speichereinheit 2066 ermöglicht, Lade- und Speicheroperationen zwischen dem gemeinsam genutzten Speicher 2070 und der Registerbank 2058 zu implementieren. In mindestens einer Ausführungsform kann die Registerbank 2058 mit der gleichen Frequenz wie die GPGPU-Kerne 2062 arbeiten, sodass die Datenübermittlung zwischen den GPGPU-Kernen 2062 und der Registerbank 2058 eine sehr geringe Latenz aufweisen kann. In mindestens einer Ausführungsform kann der gemeinsam genutzte Speicher 2070 verwendet werden, um die Kommunikation zwischen Threads zu ermöglichen, die auf funktionellen Einheiten inner-

halb des Grafikmultiprozessors 2034 ausgeführt werden. In mindestens einer Ausführungsform kann der Cachespeicher 2072 zum Beispiel als Daten-Cache verwendet werden, um Texturdaten zwischenspeichern, die zwischen funktionellen Einheiten und der Textureinheit 2036 kommuniziert werden. In mindestens einer Ausführungsform kann der gemeinsam genutzte Speicher 2070 auch als programmverwalteter Cache verwendet werden. In mindestens einer Ausführungsform können Threads, die auf den GPGPU-Kernen 2062 ausgeführt werden, zusätzlich zu den automatisch gecachten Daten, die im Cache-Speicher 2072 gespeichert sind, programmatisch Daten im gemeinsam genutzten Speicher speichern.

[0308] In mindestens einer Ausführungsform ist ein Parallelprozessor oder eine GPGPU, wie hierin beschrieben, kommunikativ an Host-/Prozessorkerne gekoppelt, um Grafikoperationen, Operationen des maschinellen Lernens, Musteranalyseoperationen und verschiedene Funktionen einer Universal-GPU (GPGPU) zu beschleunigen. In mindestens einer Ausführungsform kann eine GPU über einen Bus oder eine andere Zusammenschaltung (z. B. eine Hochgeschwindigkeitszusammenschaltung wie etwa PCIe oder NVLink) kommunikativ an den Hostprozessor/die Kerne gekoppelt sein. In mindestens einer Ausführungsform kann eine GPU in einem Gehäuse oder Chip als Kerne integriert sein und kommunikativ über einen internen Prozessorbus/eine Zusammenschaltung, die sich innerhalb eines Gehäuses oder Chip befindet, an Kerne gekoppelt sein. In mindestens einer Ausführungsform können die Prozessorkerne unabhängig von einer Weise, auf welche eine GPU verbunden ist, einer derartigen GPU Arbeit in Form von Sequenzen von Befehlen/Anweisungen zuweisen, die in einem Arbeitsdeskriptor enthalten sind. In mindestens einer Ausführungsform verwendet diese GPU dann dedizierte Schaltung/Logik zum effizienten Verarbeiten dieser Befehle/Anweisungen.

[0309] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 in dem Grafikmultiprozessor 2034 für Inferenz- oder Vorhersageoperationen mindestens zum Teil auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder hierin beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0310] **Fig. 21** veranschaulicht ein Mehr-GPU-Rechensystem 2100 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann das Mehr-GPU-Rechensystem 2100 einen Prozessor 2102 beinhalten, der über einen Hostschnittstellen-Switch 2104 an mehrere Universal-Grafikverarbeitungseinheiten (GPGPUs) 2106A-D gekoppelt ist. In mindestens einer Ausführungsform ist der Hostschnittstellen-Switch 2104 eine PCI-Express-Switch-Vorrichtung, die den Prozessor 2102 an einen PCI-Express-Bus koppelt, über den der Prozessor 2102 mit den GPGPUs 2106A-D kommunizieren kann. In mindestens einer Ausführungsform können die GPGPUs 2106A-D über eine Reihe von Hochgeschwindigkeits-Punkt-zu-Punkt-GPU-zu-GPU-Verbindungen 2116 miteinander verbunden werden. In mindestens einer Ausführungsform sind die GPU-zu-GPU-Verbindungen 2116 mit jeder der GPGPUs 2106A-D über eine eigene GPU-Verbindung verbunden. In mindestens einer Ausführungsform ermöglichen die P2P-GPU-Verbindungen 2116 eine direkte Kommunikation zwischen den einzelnen GPGPUs 2106A-D, ohne dass eine Kommunikation über den Host-Schnittstellenbus 2104 erforderlich ist, an den der Prozessor 2102 angeschlossen ist. In mindestens einer Ausführungsform, bei der der GPU-zu-GPU-Verkehr auf P2P-GPU-Verbindungen 2116 geleitet wird, bleibt der Host-Schnittstellenbus 2104 für den Systemspeicherzugriff oder für die Kommunikation mit anderen Instanzen des Multi-GPU-Computersystems 2100 verfügbar, beispielsweise über ein oder mehrere Netzwerkgeräte. Während in mindestens einer Ausführungsform die GPGPUs 2106A-D mit dem Prozessor 2102 über den Hostschnittstellen-Switch 2104 verbunden sind, beinhaltet der Prozessor 2102 in mindestens einer Ausführungsform eine direkte Unterstützung für P2P-GPU-Verknüpfungen 2116 und kann direkt mit den GPGPUs 2106A-D verbunden sein.

[0311] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 in dem Mehr-GPU-Rechensystem 1500 für Inferenz- oder Vorhersageoperationen mindestens zum Teil auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder hierin beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0312] Fig. 22 ist ein Blockdiagramm eines Grafikprozessors 2200 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2200 eine Ringzusammenschaltung 2202, ein Pipeline-Frontend 2204, eine Medien-Engine 2237 und Grafikkern 2280A-2280N. In mindestens einer Ausführungsform koppelt die Ringzusammenschaltung 2202 den Grafikprozessor 2200 an andere Verarbeitungseinheiten, einschließlich anderer Grafikprozessoren oder eines oder mehrerer Universalprozessorkerne. In mindestens einer Ausführungsform ist der Grafikprozessor 2200 einer von vielen Prozessoren, die in ein Mehrkern-Verarbeitungssystem integriert sind.

[0313] In mindestens einer Ausführungsform empfängt der Grafikprozessor 2200 Batches von Befehlen über die Ringzusammenschaltung 2202. In mindestens einer Ausführungsform werden eingehende Befehle durch einen Befehls-Streamer 2203 in dem Pipeline-Frontend 2204 interpretiert. In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2200 skalierbare Ausführungslogik zum Durchführen von 3D-Geometrieverarbeitung und Medienverarbeitung über den Grafikkern oder die Grafikkern 2280A-2280N. In mindestens einer Ausführungsform führt der Befehls-Streamer 2203 der Geometriepipeline 2236 Befehle für 3D-Geometrieverarbeitungsbefehle zu. In mindestens einer Ausführungsform führt der Befehls-Streamer 2203 für mindestens einige Medienverarbeitungsbefehle Befehle einem Video-Frontend 2234 zu, das mit der Medien-Engine 2237 gekoppelt ist. In mindestens einer Ausführungsform beinhaltet die Medien-Engine 2237 eine Videoqualitäts-Engine (Video Quality Engine - VQE) 2230 für die Video- und Bildnachverarbeitung und eine Engine zum Codieren/Decodieren in mehreren Formaten (multi-format encode/decode - MFX) 2233 zum Bereitstellen von hardwarebeschleunigter Codierung und Decodierung von Mediendaten. In mindestens einer Ausführungsform erzeugen die Geometriepipeline 2236 und die Medien-Engine 2237 jeweils Ausführungs-Threads für Thread-Ausführungsressourcen, die durch mindestens einen Grafikkern 2280 bereitgestellt sind.

[0314] In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2200 skalierbare Thread-Ausführungsressourcen mit den Grafikkernen 2280A-2280N (die modular sein können und mitunter als Kernslices bezeichnet werden), die jeweils mehrere Teilkern 2250A-50N, 2260A-2260N (mitunter als Kernteilslices bezeichnet) aufweisen. In mindestens einer Ausführungsform kann der Grafikprozessor 2200 eine beliebige Anzahl von Grafikkernen 2280 A aufweisen. In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2200 einen Grafikkern 2280 A mit mindestens einem ersten Teilkern 2250 A und einem zweiten Teilkern 2260 A. In mindestens einer Ausführungsform ist der Grafikprozessor 2200 ein Prozessor niedriger Leistung mit einem einzelnen Teilkern (z. B. 2250 A). In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2200 mehrere Grafikkern 2280A-2280N, von denen jeder einen Satz von ersten Teilkernen 2250A-2250N und einen Satz von zweiten Teilkernen 2260A-2260N beinhaltet. In mindestens einer Ausführungsform beinhaltet jeder Teilkern in den ersten Teilkernen 2250A-2250N mindestens einen ersten Satz von Ausführungseinheiten 2252A-2252N und Medien-/Texturabstern 2254A-2254N. In mindestens einer Ausführungsform beinhaltet jeder Teilkern in den zweiten Teilkernen 2260A-2260N mindestens einen zweiten Satz von Ausführungseinheiten 2262A-2262N und Abstern 2264A-2264N. In mindestens einer Ausführungsform nutzen die Teilkern 2250A-2250N, 2260A-2260N jeweils einen Satz von gemeinsam genutzten Ressourcen 2270A-2270N gemeinsam. In mindestens einer Ausführungsform beinhalten die gemeinsam genutzten Ressourcen gemeinsam genutzten Cachespeicher und Pixeloperationslogik.

[0315] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit Fig. 1A und/oder Fig. 1B bereitgestellt werden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 115 in dem Grafikprozessor 2200 für Inferenz- oder Vorhersageoperationen mindestens zum Teil auf Grundlage von Gewichtungsparemtern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder hierin beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0316] Fig. 23 ist ein Blockdiagramm, das eine Mikroarchitektur für einen Prozessor 2300 darstellt, der Logikschaltungen beinhalten kann, um Anweisungen gemäß mindestens einer Ausführungsform auszuführen. In mindestens einer Ausführungsform kann der Prozessor 2300 Anweisungen durchführen, einschließlich x86-Anweisungen, ARM-Anweisungen, spezialisierter Anweisungen für anwendungsspezifische integrierte Schaltungen (ASICs) usw. In mindestens einer Ausführungsform kann der Prozessor 2300 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 Fließkommaform verfügbar sind, mit Paket-Datenelementen 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), solche Paket-Datenoperanden halten. In mindestens einer Ausführungsform kann der Prozessor 2300 Anweisungen zum Beschleunigen von Algorithmen für maschinelles Lernen oder Deep Learning, Training oder Inferenzieren durchführen.

[0317] In mindestens einer Ausführungsform beinhaltet der Prozessor 2300 ein In-Order-Frontend („Frontend“) 2301 zum Abrufen von auszuführenden Anweisungen und Vorbereiten von Anweisungen, die später in einer Prozessorpipeline verwendet werden sollen. In mindestens einer Ausführungsform kann das Frontend 2301 mehrere Einheiten beinhalten. In mindestens einer Ausführungsform ruft ein Anweisungsvorabruffer 2326 Anweisungen aus dem Speicher ab und führt einem Anweisungsdekodierer 2328 Anweisungen zu, der wiederum Anweisungen dekodiert oder interpretiert. Zum Beispiel decodiert in mindestens einer Ausführungsform der Anweisungsdekodierer 2328 eine empfangene Anweisung in eine oder mehrere Operationen, die als „Mikroanweisungen“ oder „Mikrooperationen“ bezeichnet werden (auch als „Mikro-Ops“ oder „μops“ bezeichnet) und die eine Maschine ausführen kann. In mindestens einer Ausführungsform parst der Anweisungsdekodierer 2328 eine Anweisung in einen Opcode und entsprechende Daten- und Steuerfelder, die durch die Mikroarchitektur verwendet werden können, um Operationen gemäß mindestens einer Ausführungsform durchzuführen. In mindestens einer Ausführungsform kann ein Ablaufverfolgungscache 2330 decodierte μops in programmgeordnete Sequenzen oder Ablaufverfolgungen in einer μop-Warteschlange 2334 zur Ausführung zusammenstellen. Wenn der Ablaufverfolgungscache 2330 auf eine komplexe Anweisung stößt, stellt in mindestens einer Ausführungsform ein Mikrocode-ROM 2332 die für den Abschluss einer Operation notwendigen μops bereit.

[0318] 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, falls mehr als vier Mikro-Ops benötigt werden, um eine Anweisung abzuschließen, der Anweisungsdekodierer 2328 auf den Mikrocode-ROM 2332 zugreifen, um diese Anweisung durchzuführen. In mindestens einer Ausführungsform kann eine Anweisung in eine kleine Anzahl von Mikro-Ops zur Verarbeitung in dem Anweisungsdekodierer 2328 decodiert werden. In mindestens einer Ausführungsform kann eine Anweisung im Mikrocode-ROM 2332 gespeichert werden, sollte eine Anzahl von Mikrooperationen erforderlich sein, um eine solche Operation auszuführen. In mindestens einer Ausführungsform bezieht sich der Ablaufverfolgungscache 2330 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 2332 gemäß mindestens einer Ausführungsform abzuschließen. In mindestens einer Ausführungsform kann das Frontend 2301 der Maschine, nachdem der Mikrocode-ROM 2332 die Sequenzierung von Mikro-Ops für eine Anweisung beendet hat, das Abrufen von Mikro-Ops aus dem Ablaufverfolgungszwischenspeicher 2330 wieder aufnehmen.

[0319] In mindestens einer Ausführungsform kann die Engine zur Ausführung in einer anderen Reihenfolge („Out-of-Order-Engine“) 2303 Anweisungen zur Ausführung vorbereiten. In mindestens einer Ausführungsform weist die Out-of-Order-Ausführungslogik eine Anzahl von Puffern auf, um den Fluss der Anweisungen zu glätten und neu zu ordnen, um die Rechenleistung zu optimieren, während sie eine Pipeline durchlaufen und zur Ausführung eingeplant werden. In mindestens einer Ausführungsform beinhaltet die Engine 2303 zur Out-of-Order-Ausführung ohne Einschränkung einen Zuteiler/Registerumbenenner 2340, eine Speicher-μop-Warteschlange 2342, eine Ganzzahl-/Gleitkomma-μop-Warteschlange 2344, einen Speicherplaner 2346, einen schnellen Planer 2302, einen langsamen/allgemeinen Gleitkomma-Planer („langsamer/allgemeiner FP-Planer“) 2304 und einen einfachen Gleitkomma-Planer („einfacher FP-Planer“) 2306. In mindestens einer Ausführungsform werden der schnelle Planer 2302, der langsame/allgemeine Gleitkomma-Planer 2304 und der einfache Gleitkomma-Planer 2306 in dieser Schrift auch gemeinsam als „μop-Planer 2302, 2304, 2306“ bezeichnet. In mindestens einer Ausführungsform weist der Zuweiser/Registerumbenenner 2340 Maschinenpuffer und Ressourcen zu, die jede μop für die Ausführung benötigt. In mindestens einer Ausführungsform benennt der Zuweiser/Registerumbenenner 2340 logische Register in Einträge in einer Registerbank um. In mindestens einer Ausführungsform weist der Zuordner/Registerumbenenner 2340 auch einen Eintrag für jede μop in einer von zwei μop-Warteschlangen, der Speicher-μop-Warteschlange 2342 für Speicheroperationen und der Ganzzahl-/Fließkomma-μop-Warteschlange 2344 für Nicht-Speicheroperationen vor dem Speicherplaner 2346 und den μop-Planern 2302, 2304, 2306 zu. In mindestens einer Ausführungsform bestimmen die μop-Planer 2302, 2304, 2306 auf Grundlage der Bereitschaft ihrer abhängigen Eingaberegister-Operandenquellen und der Verfügbarkeit der Ausführungsressourcen, die μops benötigen, um ihre Operation abzuschließen, wann eine μop zur Ausführung bereit ist. In mindestens einer Ausführungsform kann

der schnelle Planer 2302 auf jeder Hälfte des Haupttaktzyklus planen, während der langsame/allgemeine Gleitkommaplaner 2304 und der einfache Gleitkommaplaner 2306 einmal pro Hauptprozessortaktzyklus planen können. In mindestens einer Ausführungsform vermitteln die μ op-Planer 2302, 2304, 2306 für Versandports, um μ ops für die Ausführung zu planen.

[0320] In mindestens einer Ausführungsform beinhaltet der Ausführungsblock 2311 ohne Einschränkung eine Ganzzahlregisterdatei/ein Umgehungsnetz 2308, eine Gleitkommaregisterdatei/ein Umgehungsnetz („FP-Registerdatei/Umgehungsnetz“) 2310, Adresserzeugungseinheiten (address generation units - „AGUs“) 2312 und 2314, schnelle arithmetische Logikeinheiten (ALUs) („schnelle ALUs“) 2316 und 2318, eine langsame arithmetische Logikeinheit („langsame ALU“) 2320, eine Gleitkomma-ALU („FP“) 2322 und eine Gleitkommabewegungseinheit („FP-Bewegung“) 2324. In mindestens einer Ausführungsform werden die Ganzzahlregisterdatei/das Umgehungsnetz 2308 und Gleitkommaregisterdatei/Umgehungsnetz 2310 in dieser Schrift auch als „Registerdateien 2308, 2310“ bezeichnet. In mindestens einer Ausführungsform werden AGUs 2312 und 2314, schnelle ALUs 2316 und 2318, die langsame ALU 2320, die Gleitkomma-ALU 2322 und die Gleitkommabewegungseinheit 2324 in dieser Schrift auch als „Ausführungseinheiten 2312, 2314, 2316, 2318, 2320, 2322 und 2324“ bezeichnet. In mindestens einer Ausführungsform kann der Ausführungsblock 2311 ohne Einschränkung eine beliebige Anzahl (einschließlich null) und einen beliebigen Typ von Registerbänken, Umgehungsnetzen, Adresserzeugungseinheiten und Ausführungseinheiten in beliebiger Kombination beinhalten.

[0321] In mindestens einer Ausführungsform können die Registernetze 2308, 2310 zwischen den μ op-Planern 2302, 2304, 2306 und den Ausführungseinheiten 2312, 2314, 2316, 2318, 2320, 2322 und 2324 angeordnet sein. In mindestens einer Ausführungsform führt die Ganzzahlregisterbank/das Umgehungsnetz 2308 Ganzzahloperationen durch. In mindestens einer Ausführungsform führt die Gleitkommaregisterbank/das Umgehungsnetz 2310 Gleitkommaoperationen durch. In mindestens einer Ausführungsform kann jedes der Registernetze 2308, 2310 ohne Einschränkung ein Umgehungsnetz beinhalten, das gerade abgeschlossene Ergebnisse, die noch nicht in eine Registerbank geschrieben wurden, umgehen oder zu neuen abhängigen μ ops weiterleiten kann. In mindestens einer Ausführungsform können die Registernetze 2308, 2310 miteinander Daten austauschen. In mindestens einer Ausführungsform kann die Ganzzahlregisterbank/das Umgehungsnetz 2308 ohne Einschränkung zwei separate Registerbänke beinhalten, eine Registerbank für Daten niedriger Ordnung mit zweiunddreißig Bit und eine zweite Registerbank für Daten hoher Ordnung mit zweiunddreißig Bit. In mindestens einer Ausführungsform kann die Gleitkommaregisterbank/das Umgehungsnetz 2310 ohne Einschränkung 128 Bit breite Einträge beinhalten, da Gleitkommaanweisungen typischerweise Operanden mit einer Breite von 64 bis 128 Bit aufweisen.

[0322] In mindestens einer Ausführungsform können die Ausführungseinheiten 2312, 2314, 2316, 2318, 2320, 2322, 2324 Anweisungen ausführen. In mindestens einer Ausführungsform speichern die Registernetze 2308, 2310 Integer- und Gleitkommadaten-Operandenwerte, die Mikroanweisungen benötigen, um ausgeführt zu werden. In mindestens einer Ausführungsform kann der Prozessor 2300 ohne Einschränkung eine beliebige Anzahl und Kombination der Ausführungseinheiten 2312, 2314, 2316, 2318, 2320, 2322, 2324 beinhalten. In mindestens einer Ausführungsform können die Gleitkomma-ALU 2322 und die Gleitkommabewegungseinheit 2324 Gleitkomma-, MMX-, SIMD-, AVX- und SSE-Operationen oder andere Operationen ausführen, was spezialisierte Anweisungen zum maschinellen Lernen beinhaltet. In mindestens einer Ausführungsform kann die Gleitkomma-ALU 2322 ohne Einschränkung einen 64 Bit mal 64 Bit großen Gleitkommadividerer zum Ausführen von Divisions-, Quadratwurzel- und Rest-Mikro-Ops beinhalten. In mindestens einer Ausführungsform können Befehle, die einen Gleitkommawert beinhalten, mit Gleitkomma-Hardware gehandhabt werden. In mindestens einer Ausführungsform können ALU-Operationen an schnelle ALUs 2316, 2318 weitergegeben werden. In mindestens einer Ausführungsform können schnelle ALUs 2316, 2318 schnelle Operationen mit einer effektiven Latenz von einem halben Taktzyklus ausführen. In mindestens einer Ausführungsform gehen die meisten komplexen Ganzzahloperationen zur langsamen ALU 2320, da die langsame ALU 2320 ohne Einschränkung Ganzzahlausführungshardware für Operationen mit langer Latenzzeit beinhalten kann, wie etwa eine Multiplikation, Verschiebungen, Kennzeichenlogik und Verzweigungsverarbeitung. In mindestens einer Ausführungsform können Speicherlade-/speicheroperationen von AGUs 2312, 2314 ausgeführt werden. In mindestens einer Ausführungsform können die schnelle ALU 2316, die schnelle ALU 2318 und die langsame ALU 2320 Ganzzahloperationen an 64-Bit-Datenoperanden ausführen. In mindestens einer Ausführungsform können die schnelle ALU 2316, die schnelle ALU 2318 und die langsame ALU 2320 so implementiert sein, dass sie eine Vielfalt von Datenbitgrößen unterstützen, darunter sechzehn, zweiunddreißig, 128, 256 usw. In mindestens einer Ausführungsform können die Gleitkomma-ALU 2322 und die Gleitkommabewegungseinheit 2324 so implementiert sein, dass sie einen Bereich von Operanden unterstüt-

zen, der Bits mit verschiedenen Breiten aufweist, wie etwa 128 Bit breite Operanden mit gepackten Daten in Verbindung mit SIMD- und Multimedia-Anweisungen.

[0323] In mindestens einer Ausführungsform teilen die μ op-Planer 2302, 2304, 2306 abhängige Operationen zu, bevor die Ausführung einer übergeordneten Last beendet ist. In mindestens einer Ausführungsform kann der Prozessor 2300, da μ ops spekulativ geplant und im Prozessor 2300 ausgeführt werden können, auch eine Logik beinhalten, um Speicherfehler zu handhaben. In mindestens einer Ausführungsform können, falls eine Datenlast in einem Datencache 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 kann es sein, dass abhängige Operationen wiederholt werden müssen und es unabhängigen ermöglicht werden kann, 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.

[0324] In mindestens einer Ausführungsform können sich „Register“ auf bordeigene Prozessorspeicherorte beziehen, die als Teil von Anweisungen zum Identifizieren von Operanden verwendet werden können. In mindestens einer Ausführungsform kann es sich bei den Registern um diejenigen handeln, die von außerhalb eines Prozessors (aus der Sicht eines Programmierers) verwendbar sein können. 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 hierin beschriebenen Funktionen ausführen. In mindestens einer Ausführungsform können die hierin beschriebenen Register durch Schaltungen innerhalb eines Prozessors unter Verwendung einer beliebigen Anzahl unterschiedlicher Techniken implementiert werden, wie z. B. dedizierte physische Register, dynamisch zugewiesene physische Register unter Verwendung von Registerumbenennung, Kombinationen aus dedizierten und dynamisch zugewiesenen physischen Registern usw. In mindestens einer Ausführungsform speichern Integerregister 32-Bit-Integerdaten. Eine Registerbank aus mindestens einer Ausführungsform beinhaltet zudem acht Multimedia-SIMD-Register für gepackte Daten.

[0325] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform können Abschnitte der oder die gesamte Inferenz- und/oder Trainingslogik 115 in den Ausführungsblock 2311 und andere gezeigte oder nicht gezeigte Speicher oder Register einbezogen sein. Zum Beispiel können in mindestens einer Ausführungsform die hierin beschriebenen Trainings- und/oder Inferenztechniken eine oder mehrere der in dem Ausführungsblock 2311 veranschaulichten ALUs verwenden. Darüber hinaus können Gewichtungparameter im chipinternem oder chipexternem Speicher und/oder in Registern (gezeigt oder nicht gezeigt) gespeichert werden, die ALUs des Ausführungsblocks 2311 konfigurieren, um eine(n) oder mehrere hierin beschriebene Algorithmen des maschinellen Lernens, neuronale Netzarchitekturen, Anwendungsfälle oder Trainingstechniken durchzuführen.

[0326] **Fig. 24** veranschaulicht einen Deep-Learning-Anwendungsprozessor 2400 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform verwendet der Deep-Learning-Anwendungsprozessor 2400 Anweisungen, die bei Ausführung durch den Deep-Learning-Anwendungsprozessor 2400 bewirken, dass der Deep-Learning-Anwendungsprozessor 2400 einige oder alle der in dieser Offenbarung beschriebenen Prozesse und Techniken ausführt. In mindestens einer Ausführungsform ist der Deep-Learning-Anwendungsprozessor 2400 eine anwendungsspezifische integrierte Schaltung (ASIC). In mindestens einer Ausführungsform führt der Anwendungsprozessor 2400 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 2400 ohne Einschränkung Verarbeitungcluster 2410(1)-2410(12), chipübergreifende Verknüpfungen (Inter-Chip Links - „ICLs“) 2420(1)-2420(12), chipübergreifende Steuerungen (Inter-Chip Controllers - „ICCs“) 2430(1)-2430(2), Speicher mit hoher Bandbreite der zweiten Generation (high-bandwidth memory second generation - „HBM2“) 2440(1)-2440(4), Speichersteuerungen (memory controllers - „Mem Ctrlrs“) 2442(1)-2442(4), eine Bitübertragungsschicht mit Speicher mit hoher Bandbreite (high bandwidth memory physical layer - „HBM PHY“) 2444(1)-2444(4), eine zentrale Verwaltungssteuerungs-Verarbeitungseinheit („Verwaltungssteuerungs-CPU“) 2450, eine serielle Peripherieschnittstelle, eine zwischenintegrierte Schaltung und einen Universal-Eingabe/Ausgabe-Block („SPI, I²C, GPIO“) 2460, eine Interconnect-Express-Steuerung für Peri-

perierikomponenten und einen Direktspeicherzugriffsblock („PCIe-Steuerung und DMA“) 2470 und einen sechzehnspurigen Interconnect-Express-Port für Peripheriekomponenten („PCI-Express x 16“) 2480.

[0327] In mindestens einer Ausführungsform können die Verarbeitungscluster 2410 Deep-Learning-Operationen durchführen, einschließlich Inferenz- oder Vorhersageoperationen auf Grundlage von Gewichtungsparametern, die mit einer oder mehreren Trainingstechniken, einschließlich der hierin beschriebenen, berechnet wurden. In mindestens einer Ausführungsform kann jeder Verarbeitungscluster 2410 ohne Einschränkung eine beliebige Anzahl und einen beliebigen Typ von Prozessoren beinhalten. In mindestens einer Ausführungsform kann der Deep-Learning-Anwendungsprozessor 2400 eine beliebige Anzahl und Art von Verarbeitungsclustern beinhalten. In mindestens einer Ausführungsform sind die Inter-Chip-Verknüpfungen 2420 bidirektional. In mindestens einer Ausführungsform ermöglichen die Inter-Chip-Verknüpfungen 2420 und die Inter-Chip-Steuerung 2430 mehreren Deep-Learning-Anwendungsprozessoren 2400 den Austausch von Informationen, die Aktivierungsinformationen beinhalten, die sich aus der Ausführung eines oder mehrerer Algorithmen des maschinellen Lernens ergeben, die in einem oder mehreren neuronalen Netzen verkörpert sind. In mindestens einer Ausführungsform kann der Deep-Learning-Anwendungsprozessor 2400 eine beliebige Anzahl (einschließlich null) und Art von ICLs 2420 und ICCs 2430 beinhalten.

[0328] In mindestens einer Ausführungsform stellen HBM2s 2440 insgesamt 32 Gigabyte (GB) Speicher bereit. In mindestens einer Ausführungsform ist HBM2 2440(i) sowohl mit der Speichersteuerung 2442(i) als auch der HBM PHY 2444(i) assoziiert, wobei „i“ eine willkürliche ganze Zahl ist. In mindestens einer Ausführungsform kann eine beliebige Anzahl von HBM2s 2440 eine beliebige Art und Gesamtmenge von Speicher mit hoher Bandbreite bereitstellen und kann einer beliebigen Anzahl (die null beinhaltet) und Art von Speichersteuerungen 2442 und HBM PHYs 2444 zugeordnet sein. In mindestens einer Ausführungsform können SPI, I²C, GPIO 2460, PCIe-Steuerung und DMA 2470 und/oder PCIe 2480 durch eine beliebige Anzahl und Art von Blöcken ersetzt werden, die eine beliebige Anzahl und Art von Kommunikationsstandards auf eine beliebige technisch machbare Weise ermöglichen.

[0329] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. 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 Deep-Learning-Anwendungsprozessor 2400 bereitgestellte Informationen vorherzusagen oder zu inferenzieren. In mindestens einer Ausführungsform wird der Deep-Learning-Anwendungsprozessor 2400 verwendet, um Informationen auf Grundlage eines trainierten Modells des maschinellen Lernens (z. B. eines neuronalen Netzes) zu inferenzieren oder vorherzusagen, das durch einen anderen Prozessor oder ein anderes System oder durch den Deep-Learning-Anwendungsprozessor 2400 trainiert wurde. In mindestens einer Ausführungsform kann der Prozessor 2400 verwendet werden, um einen oder mehrere der hierin beschriebenen Anwendungsfälle für neuronale Netze durchzuführen.

[0330] **Fig. 25** ist ein Blockdiagramm eines neuromorphen Prozessors 2500 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann der neuromorphe Prozessor 2500 eine oder mehrere Eingaben von Quellen außerhalb des neuromorphen Prozessors 2500 empfangen. In mindestens einer Ausführungsform können diese Eingaben an ein oder mehrere Neuronen 2502 innerhalb des neuromorphen Prozessors 2500 übermittelt werden. In mindestens einer Ausführungsform können die Neuronen 2502 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 2500 ohne Einschränkung Tausende oder Millionen von Instanzen von Neuronen 2502 beinhalten, aber es kann eine beliebige geeignete Anzahl von Neuronen 2502 verwendet werden. In mindestens einer Ausführungsform kann jede Instanz des Neurons 2502 einen Neuroneneingang 2504 und einen Neuronenausgang 2506 beinhalten. In mindestens einer Ausführungsform können die Neuronen 2502 Ausgaben erzeugen, die an Eingänge anderer Instanzen von Neuronen 2502 übermittelt werden können. Zum Beispiel können in mindestens einer Ausführungsform die Neuroneneingänge 2504 und die Neuronenausgänge 2506 über Synapsen 2508 miteinander verbunden sein.

[0331] In mindestens einer Ausführungsform können die Neuronen 2502 und die Synapsen 2508 derartig miteinander verbunden sein, dass der neuromorphe Prozessor 2500 betrieben wird, um von dem neuromorphen Prozessor 2500 empfangene Informationen zu verarbeiten oder zu analysieren. In mindestens einer Ausführungsform können die Neuronen 2502 einen Ausgangsimpuls (oder „Feuer“ oder „Spitze“) übermitteln, wenn durch den Neuroneneingang 2504 empfangene Eingaben einen Schwellenwert überschreiten. In min-

destens einer Ausführungsform können die Neuronen 2502 an den Neuroneneingängen 2504 empfangene Signale summieren oder integrieren. Zum Beispiel können die Neuronen 2502 in mindestens einer Ausführungsform als undichte integrate-and-fire-Neuronen umgesetzt sein, wobei, wenn eine Summe (als ein „Membranpotential“ bezeichnet) einen Schwellenwert überschreitet, das Neuron 2502 eine Ausgabe (oder „Feuer“) unter Verwendung einer Übertragungsfunktion überschreitet, wie etwa einer Sigmoid- oder Schwellenfunktion. In mindestens einer Ausführungsform kann ein leckendes Integrate-and-Fire-Neuron Signale, die an den Neuroneneingängen 2504 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 2504 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 2502 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 2502 in mindestens einer Ausführungsform ohne Einschränkung Vergleicherschaltungen oder Logik beinhalten, die eine Ausgangsspitze am Neuronenausgang 2506 erzeugen, wenn das Ergebnis des Anwendens einer Übertragungsfunktion auf den Neuroneneingang 2504 einen Schwellenwert überschreitet. In mindestens einer Ausführungsform kann das Neuron 2502, 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 2502 nach einem geeigneten Zeitraum (oder Refraktärzeitraum) den normalen Betrieb wieder aufnehmen, sobald das Membranpotential auf 0 zurückgesetzt ist.

[0332] In mindestens einer Ausführungsform können die Neuronen 2502 durch Synapsen 2508 miteinander verbunden sein. In mindestens einer Ausführungsform können die Synapsen 2508 arbeiten, um Signale von einem Ausgang eines ersten Neurons 2502 an einen Eingang eines zweiten Neurons 2502 zu übermitteln. In mindestens einer Ausführungsform können die Neuronen 2502 Informationen über mehr als eine Instanz der Synapse 2508 übermitteln. In mindestens einer Ausführungsform können eine oder mehrere Instanzen der Neuronenausgabe 2506 über eine Instanz der Synapse 2508 mit einer Instanz der Neuroneneingabe 2504 in demselben Neuron 2502 verbunden sein. In mindestens einer Ausführungsform kann eine Instanz des Neurons 2502, die eine über eine Instanz der Synapse 2508 zu übermittelnde Ausgabe erzeugt, in Bezug auf diese Instanz der Synapse 2508 als ein „präsynaptisches Neuron“ bezeichnet werden. In mindestens einer Ausführungsform kann eine Instanz des Neurons 2502, die eine über eine Instanz der Synapse 2508 übermittelte Eingabe empfängt, in Bezug auf diese Instanz der Synapse 2508 als ein „postsynaptisches Neuron“ bezeichnet werden. Da eine Instanz des Neurons 2502 Eingaben von einer oder mehreren Instanzen der Synapse 2508 empfangen kann und auch Ausgaben über eine oder mehrere Instanzen der Synapse 2508 übertragen kann, kann in mindestens einer Ausführungsform eine einzelne Instanz des Neurons 2502 daher sowohl ein „präsynaptisches Neuron“ als auch ein „postsynaptisches Neuron“ in Bezug auf verschiedene Instanzen der Synapsen 2508 sein.

[0333] In mindestens einer Ausführungsform können die Neuronen 2502 in eine oder mehrere Schichten organisiert sein. In mindestens einer Ausführungsform kann jede Instanz des Neurons 2502 einen Neuronenausgang 2506 aufweisen, der sich durch eine oder mehrere Synapsen 2508 zu einem oder mehreren Neuroneneingängen 2504 auffächern kann. In mindestens einer Ausführungsform können Neuronenausgänge 2506 von Neuronen 2502 in einer ersten Schicht 2510 mit Neuroneneingängen 2504 von Neuronen 2502 in einer zweiten Schicht 2512 verbunden sein. In mindestens einer Ausführungsform kann die Schicht 2510 als „vorwärtsgekoppelte Schicht“ bezeichnet werden. In mindestens einer Ausführungsform kann jede Instanz des Neurons 2502 in einer Instanz der ersten Schicht 2510 zu jeder Instanz des Neurons 2502 in der zweiten Schicht 2512 auffächern. In mindestens einer Ausführungsform kann die erste Schicht 2510 als „vollständig verbundene vorwärtsgekoppelte Schicht“ bezeichnet werden. In mindestens einer Ausführungsform kann jede Instanz des Neurons 2502 in einer Instanz der zweiten Schicht 2512 auf weniger als alle Instanzen des Neurons 2502 in einer dritten Schicht 2514 auffächern. In mindestens einer Ausführungsform kann die zweite Schicht 2512 als „spärlich verbundene vorwärtsgekoppelte Schicht“ bezeichnet werden. In mindestens einer Ausführungsform können sich die Neuronen 2502 in der zweiten Schicht 2512 zu den Neuronen 2502 in mehreren anderen Schichten auffächern, einschließlich zu den Neuronen 2502, die sich ebenfalls in der zweiten Schicht 2512 befinden. In mindestens einer Ausführungsform kann die zweite Schicht 2512 als „rekurrente Schicht“ bezeichnet werden. In mindestens einer Ausführungsform kann der neuromorphe Prozessor 2500 ohne Einschränkung eine beliebige geeignete Kombination von rekurrenten Schichten und vor-

wärtsgekoppelten Schichten beinhalten, einschließlich ohne Einschränkung sowohl spärlich verbundener vorwärtsgekoppelter Schichten als auch vollständig verbundener vorwärtsgekoppelter Schichten.

[0334] In mindestens einer Ausführungsform kann der neuromorphe Prozessor 2500 ohne Einschränkung eine rekonfigurierbare Zusammenschaltungsarchitektur oder dedizierte fest verdrahtete Zusammenschaltungen beinhalten, um die Synapse 2508 mit den Neuronen 2502 zu verbinden. In mindestens einer Ausführungsform kann der neuromorphe Prozessor 2500 ohne Einschränkung eine Schaltung oder Logik beinhalten, die es Synapsen ermöglicht, unterschiedlichen Neuronen 2502 nach Bedarf auf Grundlage der Topologie des neuronalen Netzwerks und dem Eingangs-/Ausgangslastfaktor von Neuronen zugewiesen zu werden. Zum Beispiel können die Synapsen 2508 in mindestens einer Ausführungsform mit Neuronen 2502 unter Verwendung einer Verbindungsstruktur, wie etwa Network-on-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.

[0335] Fig. 26 ist ein Blockdiagramm eines Verarbeitungssystems gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform umfasst das System 2600 einen oder mehrere Prozessoren 2602 und einen oder mehrere Grafikprozessoren 2608 und kann ein Einzelprozessor-Desktop-System, ein Multiprozessor-Arbeitsplatzrechnersystem oder ein Serversystem mit einer großen Anzahl von Prozessoren 2602 oder Prozessorkernen 2607 sein. In mindestens einer Ausführungsform ist das System 2600 eine Verarbeitungsplattform, die in eine integrierte Schaltung als System auf einem Chip (SoC) zur Verwendung in mobilen, tragbaren oder eingebetteten Vorrichtungen integriert ist.

[0336] In mindestens einer Ausführungsform kann das System 2600 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 2600 ein Mobiltelefon, ein Smartphone, eine Tablet-Rechenvorrichtung oder eine mobile Internet-Vorrichtung. In mindestens einer Ausführungsform kann das Verarbeitungssystem 2600 auch eine tragbare Vorrichtung beinhalten, an diese gekoppelt oder in diese integriert sein, wie etwa eine Smartwatch-Wearable-Vorrichtung, eine Smart-Eyewear-Vorrichtung, eine Augmented-Reality-Vorrichtung oder eine Virtual-Reality-Vorrichtung. In mindestens einer Ausführungsform ist das Verarbeitungssystem 2600 eine Fernseh- oder Set-Top-Box-Vorrichtung mit einem oder mehreren Prozessoren 2602 und einer grafischen Schnittstelle, die von einem oder mehreren Grafikprozessoren 2608 erzeugt wird.

[0337] In mindestens einer Ausführungsform beinhalten ein oder mehrere Prozessoren 2602 jeweils einen oder mehrere Prozessorkerne 2607, um Anweisungen zu verarbeiten, die, wenn sie ausgeführt werden, Operationen für System- und Benutzersoftware ausführen. In mindestens einer Ausführungsform ist jeder von einem oder mehreren Prozessorkernen 2607 so konfiguriert, dass er eine spezifische Anweisungssequenz 2609 verarbeitet. In mindestens einer Ausführungsform kann die Anweisungssequenz 2609 Complex Instruction Set Computing (CISC), Reduced Instruction Set Computing (RISC) oder Rechnen über ein Very Long Instruction Word (VLIW) ermöglichen. In mindestens einer Ausführungsform können die Prozessorkerne 2607 jeweils eine andere Anweisungssequenz 2609 verarbeiten, die Anweisungen beinhalten kann, um die Emulation anderer Anweisungssequenzen zu erleichtern. In mindestens einer Ausführungsform kann der Prozessorkern 2607 auch andere Verarbeitungsvorrichtungen beinhalten, wie etwa einen digitalen Signalprozessor (DSP).

[0338] In mindestens einer Ausführungsform beinhaltet der Prozessor 2602 einen Cache-Speicher 2604. In mindestens einer Ausführungsform kann der Prozessor 2602 einen einzelnen internen Cache oder mehrere Ebenen von internem Cache aufweisen. In mindestens einer Ausführungsform wird der Cachespeicher von verschiedenen Komponenten des Prozessors 2602 gemeinsam genutzt. In mindestens einer Ausführungsform verwendet der Prozessor 2602 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 2607 unter Verwendung bekannter Cachekohärenztechniken gemeinsam genutzt werden kann. In mindestens einer Ausführungsform ist zusätzlich eine Registerdatei 2606 im Prozessor 2602 enthalten, die verschiedene Typen von Registern zum Speichern verschiedener Datentypen beinhalten kann (z. B. Ganzzahlregister, Gleitkommaregister, Statusregister und ein Anweisungszeigerregister). In mindestens einer Ausführungsform kann die Registerbank 2606 Universalregister oder andere Register beinhalten.

[0339] In mindestens einer Ausführungsform sind ein oder mehrere Prozessoren 2602 mit einem oder mehreren Schnittstellenbussen 2610 gekoppelt, um Kommunikationssignale, wie etwa Adress-, Daten- oder Steuersignale, zwischen dem Prozessor 2602 und anderen Komponenten im System 2600 zu übermitteln. In

mindestens einer Ausführungsform kann der Schnittstellenbus 2610 in einer Ausführungsform ein Prozessorbus sein, wie etwa eine Version eines Mediendirektsschnittstellen(Direct Media Interface - DMI)-Busses. In mindestens einer Ausführungsform ist der Schnittstellenbus 2610 nicht auf einen DMI-Bus beschränkt, sondern kann einen oder mehrere Peripheral-Component-Interconnect-Busse (z. B. PCI, PCI Express), Speicherbusse oder andere Typen von Schnittstellenbussen beinhalten. In mindestens einer Ausführungsform beinhalten Prozessor(en) 2602 eine integrierte Speichersteuerung 2616 und einen Plattformsteuerungs-Hub 2630. In mindestens einer Ausführungsform ermöglicht die Speichersteuerung 2616 die Kommunikation zwischen einer Speichervorrichtung und anderen Komponenten des Systems 2600, während der Plattformsteuer-Hub (platform controller hub - PCH) 2630 Verbindungen zu E/A-Vorrichtungen über einen lokalen E/A-Bus bereitstellt.

[0340] In mindestens einer Ausführungsform kann eine Speichervorrichtung 2620 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 2620 als Systemspeicher für das System 2600 arbeiten, um Daten 2622 und Anweisungen 2621 zur Verwendung zu speichern, wenn ein oder mehrere Prozessoren 2602 eine Anwendung oder einen Prozess ausführen. In mindestens einer Ausführungsform ist die Speichersteuerung 2616 auch an einen optionalen externen Grafikprozessor 2612 gekoppelt, der mit einem oder mehreren Grafikprozessoren 2608 in den Prozessoren 2602 kommunizieren kann, um Grafik- und Medienoperationen durchzuführen. In mindestens einer Ausführungsform kann eine Anzeigevorrichtung 2611 mit Prozessor(en) 2602 verbunden sein. In mindestens einer Ausführungsform kann die Anzeigevorrichtung 2611 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 2611 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.

[0341] In mindestens einer Ausführungsform ermöglicht der Plattformsteuerungshub 2630 den Peripheriegeräten, sich über einen Hochgeschwindigkeits-E/A-Bus mit der Speichervorrichtung 2620 und dem Prozessor 2602 zu verbinden. In mindestens einer Ausführungsform beinhalten E/A-Peripheriegeräte, ohne darauf beschränkt zu sein, eine Audiosteuerung 2646, eine Netzwerksteuerung 2634, eine Firmwareschnittstelle 2628, einen drahtlosen Sendeempfänger 2626, Berührungssensoren 2625, eine Datenspeichervorrichtung 2624 (z. B. Festplatte, Flash-Speicher usw.). In mindestens einer Ausführungsform kann die Datenspeichervorrichtung 2624 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 2625 Touchscreen-Sensoren, Drucksensoren oder Fingerabdrucksensoren beinhalten. In mindestens einer Ausführungsform kann der drahtlose Sendeempfänger 2626 ein Wi-Fi-Sendeempfänger, ein Bluetooth-Sendeempfänger oder ein Sendeempfänger für ein mobiles Netz, wie etwa ein 3G-, 4G- oder Long-Term-Evolution-(LTE-)Sendeempfänger sein. In mindestens einer Ausführungsform ermöglicht die Firmware-Schnittstelle 2628 die Kommunikation mit der System-Firmware und kann zum Beispiel eine Unified Extensible Firmware Interface (UEFI) sein. In mindestens einer Ausführungsform kann die Netzsteuerung 2634 eine Netzverbindung zu einem drahtgebundenen Netz ermöglichen. In mindestens einer Ausführungsform ist eine Netzsteuerung mit hoher Rechenleistung (nicht gezeigt) mit dem Schnittstellenbus 2610 gekoppelt. In mindestens einer Ausführungsform ist die Audiosteuerung 2646 eine Mehrkanal-High-Definition-Audiosteuerung. In mindestens einer Ausführungsform beinhaltet das System 2600 eine optionale Alt-E/A-Steuerung 2640 zum Koppeln von Alt-Vorrichtungen (z. B. Personal System 2 (PS/2)) an das System 2600. In mindestens einer Ausführungsform kann der Plattformsteuer-Hub 2630 auch mit einer oder mehreren Universal-Serial-Bus(USB)-Steuerungen 2642 verbunden sein, die mit Eingabevorrichtungen, wie zum Beispiel Kombinationen aus Tastatur und Maus 2643, einer Kamera 2644 oder anderen USB-Eingabevorrichtungen, verbunden sind.

[0342] In mindestens einer Ausführungsform kann eine Instanz der Speichersteuerung 2616 und des Plattformsteuerungs-Hubs 2630 in einen diskreten externen Grafikprozessor, wie etwa den externen Grafikprozessor 2612, integriert sein. In mindestens einer Ausführungsform können sich der Plattformsteuerungs-Hub 2630 und/oder die Speichersteuerung 2616 außerhalb eines oder mehrerer Prozessoren 2602 befinden. Zum Beispiel kann das System 2600 in mindestens einer Ausführungsform eine externe Speichersteuerung 2616 und einen Plattformsteuer-Hub 2630 enthalten, der als Speichersteuer-Hub und Peripheriesteuer-Hub inner-

halb eines Systemchipsatzes konfiguriert sein kann, der mit dem oder den Prozessoren 2602 in Kommunikation steht.

[0343] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform können Abschnitte der oder die gesamte Inferenz- und/oder Trainingslogik 115 in die Thread-Ausführungslogik 2600 einbezogen sein. Zum Beispiel können in mindestens einer Ausführungsform die hierin beschriebenen Trainings- und/oder Inferenztechniken eine oder mehrere der in einer 3D-Pipeline verkörpert ALUs verwenden. Darüber hinaus können in mindestens einer Ausführungsform die hierin beschriebenen Inferenz- und/oder Trainingsoperationen unter Verwendung von anderer Logik als der in **Fig. 1A** oder **Fig. 1B** dargestellten 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 2600 konfigurieren, um eine(n) oder mehrere hierin beschriebene Algorithmen des maschinellen Lernens, neuronale Netzarchitekturen, Anwendungsfälle oder Trainingstechniken durchzuführen.

[0344] **Fig. 27** ist ein Blockdiagramm eines Prozessors 2700, der einen oder mehrere Prozessorkerne 2702A-2702N, eine integrierte Speichersteuerung 2714 und einen integrierten Grafikprozessor 2708 gemäß mindestens einer Ausführungsform aufweist. In mindestens einer Ausführungsform kann der Prozessor 2700 zusätzliche Kerne bis zu und einschließlich des zusätzlichen Kerns 2702N beinhalten, der durch gestrichelte Kästen dargestellt ist. In mindestens einer Ausführungsform beinhaltet jeder der Prozessorkerne 2702A-2702N eine oder mehrere Einheiten des internen Cache 2704A-2704N. In mindestens einer Ausführungsform weist jeder Prozessorkern auch Zugriff auf eine oder mehrere gemeinsam genutzte Cacheeinheiten 2706 auf.

[0345] In mindestens einer Ausführungsform stellen die Einheiten des internen Cache 2704A-2704N und die Einheiten des gemeinsam genutzten Cache 2706 eine Cachespeicherhierarchie innerhalb des Prozessors 2700 dar. In mindestens einer Ausführungsform können die Cachespeichereinheiten 2704A-2704N mindestens eine Ebene des Anweisungs- und Datencache innerhalb jedes Prozessorkerns und eine oder mehrere Ebenen eines gemeinsam genutzten Cache der mittleren Ebene beinhalten, wie etwa eine Ebene 2 (L2), Ebene 3 (L3), Ebene 4 (L4) oder andere Cacheebenen, wobei die höchste Cacheebene vor dem externen Speicher als eine LLC klassifiziert ist. In mindestens einer Ausführungsform hält die Cachekohärenzlogik die Kohärenz zwischen verschiedenen Cacheeinheiten 2706 und 2704A-2704N aufrecht.

[0346] In mindestens einer Ausführungsform kann der Prozessor 2700 auch einen Satz von einer oder mehreren Bussteuerungseinheiten 2716 und einem Systemagentenkern 2710 beinhalten. In mindestens einer Ausführungsform verwalten Bussteuerungseinheiten 2716 einen Satz von Peripheriebussen, wie etwa einen oder mehrere PCI- oder PCI-Express-Busse. In mindestens einer Ausführungsform stellt der Systemagentenkern 2710 Verwaltungsfunktionen für verschiedene Prozessorkomponenten bereit. In mindestens einer Ausführungsform beinhaltet der Systemagentenkern 2710 eine oder mehrere integrierte Speichersteuerungen 2714, um den Zugriff auf verschiedene externe Speichervorrichtungen (nicht gezeigt) zu verwalten.

[0347] In mindestens einer Ausführungsform beinhalten einer oder mehrere der Prozessorkerne 2702A-2702N Unterstützung für simultanes Multithreading. In mindestens einer Ausführungsform beinhaltet der Systemagentenkern 2710 Komponenten zum Koordinieren und Betreiben der Kerne 2702A-2702N während der Multithread-Verarbeitung. In mindestens einer Ausführungsform kann der Systemagentenkern 2710 zusätzlich eine Leistungssteuereinheit (power control unit - PCU) beinhalten, die Logik und Komponenten zur Regulierung eines oder mehrerer Leistungsstatus der Prozessorkerne 2702A-2702N und des Grafikprozessors 2708 beinhaltet.

[0348] In mindestens einer Ausführungsform beinhaltet der Prozessor 2700 zusätzlich den Grafikprozessor 2708, um Grafikverarbeitungsoperationen auszuführen. In mindestens einer Ausführungsform ist der Grafikprozessor 2708 mit Einheiten des gemeinsam genutzten Cache 2706 und dem Systemagentenkern 2710 gekoppelt, was eine oder mehrere integrierte Speichersteuerungen 2714 beinhaltet. In mindestens einer Ausführungsform beinhaltet der Systemagentenkern 2710 außerdem eine Anzeigesteuerung 2711, um die Grafikprozessorausgabe an eine oder mehrere gekoppelte Anzeigen zu lenken. In mindestens einer Ausführungsform kann die Anzeigesteuerung 2711 auch ein separates Modul sein, das über mindestens eine Zusammenschaltung mit dem Grafikprozessor 2708 gekoppelt ist, oder sie kann in den Grafikprozessor 2708 integriert sein.

[0349] In mindestens einer Ausführungsform wird eine ringbasierte Zusammenschaltungseinheit 2712 zum Koppeln interner Komponenten des Prozessors 2700 verwendet. In mindestens einer Ausführungsform kann eine alternative Zusammenschaltungseinheit verwendet werden, wie z. B. eine Punkt-zu-Punkt-Zusammenschaltung, eine geschaltete Zusammenschaltung oder andere Techniken. In mindestens einer Ausführungsform ist der Grafikprozessor 2708 über eine E/A-Verknüpfung 2713 mit der Ringzusammenschaltung 2712 gekoppelt.

[0350] In mindestens einer Ausführungsform stellt die E/A-Verknüpfung 2713 mindestens eine von mehreren Sorten von E/A-Zusammenschaltungen dar, einschließlich einer gehäuseinternen E/A-Zusammenschaltung, die Kommunikation zwischen verschiedenen Prozessorkomponenten und einem eingebetteten Speichermodul 2718 mit hoher Rechenleistung, wie etwa einem eDRAM-Modul, erleichtert. In mindestens einer Ausführungsform verwenden jeder der Prozessorkerne 2702A-2702N und der Grafikprozessor 2708 eingebettete Speichermodule 2718 als gemeinsam genutzten Cache der letzten Ebene.

[0351] In mindestens einer Ausführungsform sind die Prozessorkerne 2702A-2702N homogene Kerne, die eine gemeinsame Anweisungssatzarchitektur ausführen. In mindestens einer Ausführungsform sind die Prozessorkerne 2702A-2702N in Bezug auf die Anweisungssatzarchitektur (instruction set architecture - ISA) heterogen, wobei einer oder mehrere der Prozessorkerne 2702A-2702N einen gemeinsamen Anweisungssatz ausführen, während ein oder mehrere andere Kerne der Prozessorkerne 2702A-2702N eine Teilmenge eines gemeinsamen Anweisungssatzes oder einen anderen Anweisungssatz ausführen. In mindestens einer Ausführungsform sind die Prozessorkerne 2702A-2702N hinsichtlich der Mikroarchitektur heterogen, wobei ein oder mehrere Kerne, die einen verhältnismäßig höheren Leistungsverbrauch aufweisen, mit einem oder mehreren Leistungskernen gekoppelt sind, die einen geringeren Leistungsverbrauch aufweisen. In mindestens einer Ausführungsform kann der Prozessor 2700 auf einem oder mehreren Chips oder als SoC-integrierte Schaltung implementiert sein.

[0352] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform können Teile oder die Gesamtheit der Inferenz- und/oder Trainingslogik 115 in den Grafikprozessor 2708 einbezogen sein. Zum Beispiel können in mindestens einer Ausführungsform die hierin beschriebenen Trainings- und/oder Inferenztechniken eine oder mehrere der in einer 3D-Pipeline verkörpert ALUs, Grafikern(e) 2702, gemeinsam genutzte Logik oder andere Logik in **Fig. 27**. Darüber hinaus können in mindestens einer Ausführungsform die hierin beschriebenen Inferenz- und/oder Trainingsoperationen unter Verwendung von anderer Logik als der in **Fig. 1A** oder **Fig. 1B** dargestellten Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtungsparmeter in chipinternem oder chip externem Speicher und/oder in Registern (gezeigt oder nicht gezeigt) gespeichert werden, die ALUs des Prozessors 2700 konfigurieren, um eine(n) oder mehrere hierin beschriebene Algorithmen des maschinellen Lernens, neuronale Netzarchitekturen, Anwendungsfälle oder Trainingstechniken durchzuführen.

[0353] **Fig. 28** ist ein Blockdiagramm eines Grafikprozessors 2800, bei dem es sich um eine diskrete Grafikverarbeitungseinheit oder um einen mit einer Vielzahl von Verarbeitungskernen integrierten Grafikprozessor handeln kann. In mindestens einer Ausführungsform kommuniziert der Grafikprozessor 2800 über eine auf Speicher abgebildete E/A-Schnittstelle mit Registern auf dem Grafikprozessor 2800 und mit in Speicher abgelegten Befehlen. In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2800 eine Speicherschnittstelle 2814 zum Zugreifen auf Speicher. In mindestens einer Ausführungsform ist die Speicherschnittstelle 2814 eine Schnittstelle zu lokalem Speicher, einem oder mehreren internen Caches, einem oder mehreren gemeinsam genutzten externen Caches und/oder zu Systemspeicher.

[0354] In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2800 zudem eine Anzeigesteuerung 2802, um Anzeigegabedaten zu einer Anzeigevorrichtung 2820 zu treiben. In mindestens einer Ausführungsform beinhaltet die Anzeigesteuerung 2802 Hardware für eine oder mehrere Überlagerungsebenen für die Anzeigevorrichtung 2820 und die Komposition mehrerer Schichten von Video- oder Benutzerschnittstellenelementen. In mindestens einer Ausführungsform kann die Anzeigevorrichtung 2820 eine interne oder externe Anzeigevorrichtung sein. In mindestens einer Ausführungsform ist die Anzeigevorrichtung 2820 eine am Kopf befestigte Anzeigevorrichtung, wie etwa eine Virtual-Reality(VR)-Anzeigevorrichtung oder eine Augmented-Reality(AR)-Anzeigevorrichtung. In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2800 eine Videocodier-Engine 2806 zum Codieren, Decodieren oder Transcodieren von Medien in, aus oder zwischen einem oder mehreren Mediencodierformaten, einschließlich, aber nicht beschränkt auf,

Moving-Picture-Experts-Group-(MPEG-)Formate wie etwa MPEG-2, Advanced-Video-Coding-(AVC)-Formate wie etwa H.264/MPEG-4 AVC sowie die Society of Motion Picture & Television Engineers (SMPTE) 421M/VC-1 und Joint-Photographic-Experts-Group-(JPEG-)Formate wie etwa JPEG und Motion JPEG (MJPEG).

[0355] In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2800 eine Block-Image-Transfer-(BLIT-)Engine 2804, um zweidimensionale (2D) Rasteroperationen durchzuführen, einschließlich zum Beispiel Bit-Boundary Block Transfers. In mindestens einer Ausführungsform werden jedoch 2D-Grafikoperationen unter Verwendung einer oder mehrerer Komponenten einer Grafikverarbeitungsengine (graphics processing engine - GPE) 2810 durchgeführt. In mindestens einer Ausführungsform ist die GPE 2810 eine Rechen-Engine zum Durchführen von Grafikoperationen, einschließlich dreidimensionaler (3D) Grafikoperationen und Medienoperationen.

[0356] In mindestens einer Ausführungsform beinhaltet die GPE 2810 eine 3D-Pipeline 2812 zum Durchführen von 3D-Operationen, wie etwa Rendern dreidimensionaler Bilder und Szenen unter Verwendung von Verarbeitungsfunktionen, die an 3D-Primitivformen (z. B. Rechteck, Dreieck usw.) wirken. In mindestens einer Ausführungsform beinhaltet die 3D-Pipeline 2812 programmierbare Elemente und Festfunktionselemente, die verschiedene Tasks durchführen und/oder Ausführungs-Threads für ein 3D-/Medienteilsystem 2815 erzeugen. Während die 3D-Pipeline 2812 zum Durchführen von Medienoperationen verwendet werden kann, beinhaltet die GPE 2810 in mindestens einer Ausführungsform auch eine Medienpipeline 2816, die zum Durchführen von Medienoperationen, wie etwa Videonachverarbeitung und Bildverbesserung, verwendet wird.

[0357] In mindestens einer Ausführungsform beinhaltet die Medienpipeline 2816 Festfunktions- oder programmierbare Logikeinheiten, um eine oder mehrere spezialisierte Medienoperationen durchzuführen, wie etwa Beschleunigung von Videodecodierung, Videoentschachtelung und Beschleunigung von Videocodierung anstelle oder im Auftrag der Videocodec-Engine 2806. In mindestens einer Ausführungsform beinhaltet die Medienpipeline 2816 zusätzlich eine Thread-Erzeugungseinheit, um Threads zum Ausführen auf dem 3D-/Medienteilsystem 2815 zu erzeugen. In mindestens einer Ausführungsform führen erzeugte Threads Berechnungen für Medienoperationen auf einer oder mehreren Grafikausführungseinheiten aus, die in dem 3D-/Medienteilsystem 2815 enthalten sind.

[0358] In mindestens einer Ausführungsform beinhaltet das 3D-/Medienteilsystem 2815 Logik zum Ausführen von Threads, die durch die 3D-Pipeline 2812 und die Medienpipeline 2816 erzeugt werden. In mindestens einer Ausführungsform senden die 3D-Pipeline 2812 und die Medienpipeline 2816 Thread-Ausführungsanforderungen an das 3D-/Medienteilsystem 2815, das Thread-Zuteilungslogik zum Vermitteln und Zuteilen verschiedener Anforderungen an verfügbare Thread-Ausführungsressourcen beinhaltet. In mindestens einer Ausführungsform beinhalten die Ausführungsressourcen ein Array von Grafikausführungseinheiten zum Verarbeiten von 3D- und Medien-Threads. In mindestens einer Ausführungsform beinhaltet das 3D-/Medienteilsystem 2815 einen oder mehrere interne Caches für Thread-Anweisungen und Daten. In mindestens einer Ausführungsform beinhaltet das Teilsystem 2815 auch gemeinsam genutzten Speicher, einschließlich Registern und adressierbaren Speichers, um Daten zwischen Threads gemeinsam zu nutzen und Ausgabedaten zu speichern.

[0359] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform können Teile oder die Gesamtheit der Inferenz- und/oder Trainingslogik 115 in den Grafikprozessor 2800 einbezogen sein. Zum Beispiel können in mindestens einer Ausführungsform die hierin beschriebenen Trainings- und/oder Inferenztechniken eine oder mehrere der in der 3D-Pipeline 2812 verkörpert ALUs verwenden. Darüber hinaus können in mindestens einer Ausführungsform die hierin beschriebenen Inferenz- und/oder Trainingsoperationen unter Verwendung von anderer Logik als der in **Fig. 1A** oder **Fig. 1B** dargestellten 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 2800 konfigurieren, um eine(n) oder mehrere hierin beschriebene Algorithmen des maschinellen Lernens, neuronale Netzarchitekturen, Anwendungsfälle oder Trainingstechniken durchzuführen.

[0360] **Fig. 29** ist ein Blockdiagramm einer Grafikverarbeitungsengine 2910 eines Grafikprozessors gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist die Grafikverarbeitungs-Engine

(GPE) 2910 eine Version der in **Fig. 28** gezeigten GPE 2810. In mindestens einer Ausführungsform ist eine Medienpipeline 2916 optional und möglicherweise nicht explizit in der GPE 2910 enthalten. In mindestens einer Ausführungsform ist ein separater Medien- und/oder Bildprozessor an die GPE 2910 gekoppelt.

[0361] In mindestens einer Ausführungsform ist die GPE 2910 an einen Befehls-Streamer 2903 gekoppelt oder sie beinhaltet einen solchen, der einer 3D-Pipeline 2912 und/oder der Medienpipeline 2916 einen Befehlsstrom bereitstellt. In mindestens einer Ausführungsform ist der Befehls-Streamer 2903 an Speicher gekoppelt, bei dem es sich um Systemspeicher oder um einen oder mehrere von internem Cachespeicher und gemeinsam genutztem Cachespeicher handeln kann. In mindestens einer Ausführungsform empfängt der Befehls-Streamer 2903 Befehle aus Speicher und sendet Befehle an die 3D-Pipeline 2912 und/oder die Medienpipeline 2916. In mindestens einer Ausführungsform handelt es sich bei den Befehlen um Anweisungen, Primitive oder Mikrooperationen, die aus einem Ringpuffer abgerufen werden, der Befehle für die 3D-Pipeline 2912 und die Medienpipeline 2916 speichert. In mindestens einer Ausführungsform kann ein Ringpuffer zusätzlich Batch-Befehlspeicher beinhalten, die Batches von mehreren Befehlen speichern. In mindestens einer Ausführungsform können die Befehle für die 3D-Pipeline 2912 auch Bezugnahmen auf Daten beinhalten, die in Speicher gespeichert sind, wie etwa, aber nicht beschränkt auf, Scheitelpunkt- und Geometriedaten für die 3D-Pipeline 2912 und/oder Bilddaten und Speicherobjekte für die Medienpipeline 2916. In mindestens einer Ausführungsform verarbeiten die 3D-Pipeline 2912 und die Medienpipeline 2916 Befehle und Daten, indem sie Operationen durchführen oder einen oder mehrere Ausführungs-Threads einem Grafikkernarray 2914 zuteilen. In mindestens einer Ausführungsform beinhaltet das Grafikkernarray 2914 einen oder mehrere Blöcke von Grafikkernen (z. B. Grafikkern(e) 2915 A, Grafikkern(e) 2915 B), wobei jeder Block einen oder mehrere Grafikkerne beinhaltet. In mindestens einer Ausführungsform beinhaltet jeder Grafikkern einen Satz von Grafikausführungsressourcen, der eine Universal- und eine grafikspezifische Ausführungslogik zum Durchführen von Grafik- und Rechenoperationen sowie Logik zur Texturverarbeitung mit fester Funktion und/oder zur Beschleunigung des maschinellen Lernens und der künstlichen Intelligenz, einschließlich der Inferenz- und/oder Trainingslogik 115 in **Fig. 1A** und **Fig. 1B**, beinhaltet.

[0362] In mindestens einer Ausführungsform beinhaltet die 3D-Pipeline 2912 Festfunktionslogik und programmierbare Logik, um ein oder mehrere Shader-Programme, wie etwa Vertex-Shader, Geometrie-Shader, Pixel-Shader, Fragment-Shader, Rechen-Shader oder andere Shader-Programme, zu verarbeiten, indem Anweisungen verarbeitet und Ausführungs-Threads dem Grafikkernarray 2914 zugeteilt werden. In mindestens einer Ausführungsform stellt das Grafikkernarray 2914 einen einheitlichen Block von Ausführungsressourcen zur Verwendung beim Verarbeiten von Shader-Programmen bereit. In mindestens einer Ausführungsform beinhaltet eine Mehrzweck-Ausführungslogik (z. B. Ausführungseinheiten) innerhalb des Grafikkerns/der Grafikkerne 2915A-2915B des Grafikkernarrays 2914 Unterstützung für verschiedene 3D-API-Shader-Sprachen und sie kann mehrere simultane Ausführungs-Threads ausführen, die mit mehreren Shadern assoziiert sind.

[0363] In mindestens einer Ausführungsform beinhaltet das Grafikkernarray 2914 auch Ausführungslogik zum Durchführen von Medienfunktionen, wie etwa Video- und/oder Bildverarbeitung. In mindestens einer Ausführungsform beinhalten die Ausführungseinheiten zusätzlich Universallogik, die so programmiert sein kann, dass sie zusätzlich zu Grafikverarbeitungsoperationen parallele Universalrechenoperationen durchführt.

[0364] In mindestens einer Ausführungsform können Ausgabedaten, die durch Threads erzeugt werden, die auf dem Grafikkernarray 2914 ausgeführt werden, Daten in einem einheitlichen Rückgabepuffer (unified return buffer - URB) 2918 an Speicher ausgeben. In mindestens einer Ausführungsform kann der URB 2918 Daten für mehrere Threads speichern. In mindestens einer Ausführungsform kann der URB 2918 verwendet werden, um Daten zwischen unterschiedlichen Threads zu senden, die auf dem Grafikkernarray 2914 ausgeführt werden. In mindestens einer Ausführungsform kann der URB 2918 zusätzlich für die Synchronisation zwischen Threads auf dem Grafikkernarray 2914 und der Festfunktionslogik innerhalb der gemeinsam genutzten Funktionslogik 2920 verwendet werden.

[0365] In mindestens einer Ausführungsform ist das Grafikkernarray 2914 skalierbar, sodass das Grafikkernarray 2914 eine variable Anzahl von Grafikkernen beinhaltet, die jeweils eine variable Anzahl von Ausführungseinheiten auf Grundlage eines angestrebten Leistungs- und Rechenleistungslevels der GPE 2910 aufweisen. In mindestens einer Ausführungsform sind die Ausführungsressourcen dynamisch skalierbar, sodass die Ausführungsressourcen nach Bedarf aktiviert oder deaktiviert werden können.

[0366] In mindestens einer Ausführungsform ist das Grafikkernarray 2914 an die gemeinsam genutzte Funktionslogik 2920 gekoppelt, die mehrere Ressourcen beinhaltet, die von den Grafikkernen in dem Grafikkernarray 2914 gemeinsam genutzt werden. In mindestens einer Ausführungsform sind die gemeinsam genutzten Funktionen, die durch die gemeinsam genutzte Funktionslogik 2920 durchgeführt werden, in Hardware-Logikeinheiten verkörpert, die spezialisierte Ergänzungsfunktionen für das Grafikkernarray 2914 bereitstellen. In mindestens einer Ausführungsform beinhaltet die gemeinsam genutzte Funktionslogik 2920 eine Abtastereinheit 2921, eine Mathematikeinheit 2922 und Logik 2929 zur Zwischen-Thread-Kommunikation (inter-thread communication - ITC), ist aber nicht darauf beschränkt. In mindestens einer Ausführungsform sind ein oder mehrere Cache(s) 2925 in der gemeinsam genutzten Funktionslogik 2920 enthalten oder an diese gekoppelt.

[0367] In mindestens einer Ausführungsform wird eine gemeinsam genutzte Funktion verwendet, falls die Nachfrage nach einer spezialisierten Funktion für die Aufnahme in das Grafikkernarray 2914 nicht ausreicht. In mindestens einer Ausführungsform wird eine einzelne Instanziierung einer spezialisierten Funktion in der gemeinsam genutzten Funktionslogik 2920 verwendet und von anderen Ausführungsressourcen innerhalb des Grafikkernarrays 2914 gemeinsam genutzt. In mindestens einer Ausführungsform können spezifische gemeinsam genutzte Funktionen innerhalb der gemeinsam genutzten Funktionslogik 2920, die durch das Grafikkernarray 2914 intensiv genutzt werden, in der gemeinsam genutzten Funktionslogik 2920 innerhalb des Grafikkernarrays 2914 enthalten sein. In mindestens einer Ausführungsform kann die gemeinsam genutzte Funktionslogik 2920 innerhalb des Grafikkernarrays 2914 einen Teil der oder die gesamte Logik innerhalb der gemeinsam genutzten Funktionslogik 2920 beinhalten. In mindestens einer Ausführungsform können alle Logikelemente innerhalb der gemeinsam genutzten Funktionslogik 2920 innerhalb der gemeinsam genutzten Funktionslogik 2926 des Grafikkernarrays 2914 dupliziert werden. In mindestens einer Ausführungsform ist die gemeinsam genutzte Funktionslogik 2920 zugunsten der gemeinsam genutzten Funktionslogik 2926 innerhalb des Grafikkernarrays 2914 ausgeschlossen.

[0368] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform können Teile oder die Gesamtheit der Inferenz- und/oder Trainingslogik 115 in den Grafikprozessor 2910 einbezogen sein. Zum Beispiel können in mindestens einer Ausführungsform die hierin beschriebenen Trainings- und/oder Inferenztechniken eine oder mehrere der in der 3D-Pipeline 2912 verkörpert ALUs, Grafikkern(e) 2915, gemeinsam genutzte Funktionslogik 2926, gemeinsam genutzte Funktionslogik 2920 oder andere Logik in **Fig. 29**. Darüber hinaus können in mindestens einer Ausführungsform die hierin beschriebenen Inferenz- und/oder Trainingsoperationen unter Verwendung von anderer Logik als der in **Fig. 1A** oder **Fig. 1B** dargestellten Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtungsparemeter in chipinternem oder chipexternem Speicher und/oder in Registern (gezeigt oder nicht gezeigt) gespeichert werden, die ALUs des Grafikprozessors 2910 konfigurieren, um eine(n) oder mehrere hierin beschriebene Algorithmen des maschinellen Lernens, neuronale Netzarchitekturen, Anwendungsfälle oder Trainingstechniken durchzuführen.

[0369] **Fig. 30** ist ein Blockdiagramm von Hardware-Logik eines Grafikprozessorkerns 3000 gemäß mindestens einer hierin beschriebenen Ausführungsform. In mindestens einer Ausführungsform ist der Grafikprozessorkern 3000 in einem Grafikkernarray enthalten. In mindestens einer Ausführungsform kann es sich bei dem Grafikprozessorkern 3000, mitunter als Kernslice bezeichnet, um einen oder mehrere Grafikkerne innerhalb eines modularen Grafikprozessors handeln. In mindestens einer Ausführungsform ist der Grafikprozessorkern 3000 beispielhaft für eine Grafikkernslice und ein Grafikprozessor, wie hierin beschrieben, kann auf Grundlage der angestrebten Leistungs- und Rechenleistungshüllkurven mehrere Grafikkernslices beinhalten. In mindestens einer Ausführungsform kann jeder Grafikkern 3000 einen Festfunktionsblock 3030 beinhalten, der mit mehreren Teilkernen 3001A-3001F gekoppelt ist, die auch als Teilslices bezeichnet werden und modulare Blöcke von Universal- und Festfunktionslogik beinhalten.

[0370] In mindestens einer Ausführungsform umfasst der Festfunktionsblock 3030 eine Geometrie- und Festfunktionspipeline 3036, die von allen Unterkernen im Grafikprozessor 3000 gemeinsam genutzt werden kann, z. B. in Grafikprozessorimplementierungen mit geringerer Leistung und/oder geringerem Energieverbrauch. In mindestens einer Ausführungsform umfasst die Geometrie- und Festfunktionspipeline 3036 eine 3D-Festfunktionspipeline, eine Video-FrontEnd-Einheit, einen Thread-Spawner und Thread-Dispatcher sowie einen Unified-Return-Puffer-Manager, der Unified-Return-Puffer verwaltet.

[0371] In mindestens einer Ausführungsform beinhaltet der Festfunktionsblock 3030 auch eine Grafik-SoC-Schnittstelle 3037, einen Grafik-Mikrocontroller 3038 und eine Medienpipeline 3039. In mindestens einer Ausführungsform stellt die Grafik-SoC-Schnittstelle 3037 eine Schnittstelle zwischen dem Grafikkern 3000 und anderen Prozessorkernen innerhalb einer integrierten Schaltung als System auf einem Chip bereit. In mindestens einer Ausführungsform ist der Grafik-Mikrocontroller 3038 ein programmierbarer Teilprozessor, der so konfiguriert werden kann, dass er verschiedene Funktionen des Grafikprozessors 3000 verwaltet, einschließlich Thread-Zuteilung, -Planung und -Präemption. In mindestens einer Ausführungsform beinhaltet die Medienpipeline 3039 Logik zum Erleichtern der Decodierung, Codierung, Vorverarbeitung und/oder Nachverarbeitung von Multimedia-Daten, einschließlich Bild- und Videodaten. In mindestens einer Ausführungsform implementiert die Medienpipeline 3039 Medienvorgänge über Anforderungen an Rechen- oder Abtastlogik innerhalb der Teilkern 3001A-3001F.

[0372] In mindestens einer Ausführungsform ermöglicht die SoC-Schnittstelle 3037 es dem Grafikkern 3000, mit Universal-Anwendungsprozessorkernen (z. B. CPUs) und/oder anderen Komponenten innerhalb eines SoC zu kommunizieren, einschließlich Speicherhierarchieelementen, wie etwa einem gemeinsam genutzten Last-Level-Cachespeicher, System-RAM und/oder eingebettetem chipinternem oder gehäuseinternem DRAM. In mindestens einer Ausführungsform kann die SoC-Schnittstelle 3037 auch Kommunikation mit Festfunktionsvorrichtungen innerhalb eines SoC ermöglichen, wie etwa Kamera-Bildgebungspipelines, und sie ermöglicht die Verwendung von globalem atomarem Speicher und/oder implementiert diesen, der von dem Grafikprozessorkern 3000 und den CPUs innerhalb eines SoCs gemeinsam genutzt werden kann. In mindestens einer Ausführungsform kann die Grafik-SoC-Schnittstelle 3037 auch Leistungsverwaltungssteuerelemente für den Grafikprozessorkern 3000 implementieren und eine Schnittstelle zwischen einer Taktdomäne des Grafikprozessorkerns 3000 und anderen Taktdomänen innerhalb eines SoC ermöglichen. In mindestens einer Ausführungsform ermöglicht die SoC-Schnittstelle 3037 den Empfang von Befehlspuffern von einem Befehls-Streamer und einem globalen Thread-Zuteiler, die so konfiguriert sind, dass sie jedem von einem oder mehreren Grafikkernen innerhalb eines Grafikprozessors Befehle und Anweisungen bereitstellen. In mindestens einer Ausführungsform können Befehle und Anweisungen der Medienpipeline 3039 zugeteilt werden, wenn Medienoperationen durchgeführt werden sollen, oder einer Geometrie- und Festfunktionspipeline (z. B. der Geometrie- und Festfunktionspipeline 3036 und/oder der Geometrie- und Festfunktionspipeline 3014), wenn Grafikverarbeitungsoperationen durchgeführt werden sollen.

[0373] In mindestens einer Ausführungsform kann der Grafik-Mikrocontroller 3038 so konfiguriert sein, dass er verschiedene Planungs- und Verwaltungs-Tasks für den Grafikkern 3000 durchführt. In mindestens einer Ausführungsform kann der Grafik-Mikrocontroller 3038 die Planung der Grafik- und/oder Rechenarbeitslast auf verschiedenen Grafikparallel-Engines innerhalb der Arrays 3002A-3002F, 3004A-3004F der Ausführungseinheiten (execution unit - EU) innerhalb der Teilkern 3001A-3001F durchführen. In mindestens einer Ausführungsform kann Hostsoftware, die auf einem CPU-Kern eines SoC, einschließlich des Grafikkerns 3000, ausgeführt wird, Workloads an einen von mehreren Grafikprozessorpfeilen absenden, der eine Planungsoperation auf einer zweckmäßigen Grafik-Engine aufruft. In mindestens einer Ausführungsform umfassen die Planungsvorgänge das Bestimmen, welche Arbeitslast als Nächstes laufen soll, das Übermitteln einer Arbeitslast an einen Befehls-Streamer, das Vorwegnehmen vorhandener 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 3038 auch Zustände mit niedriger Leistung oder inaktive Zustände für den Grafikkern 3000 ermöglichen, wobei dem Grafikkern 3000 die Fähigkeit verliehen wird, Register innerhalb des Grafikkerns 3000 über Zustandsübergänge mit niedriger Leistung unabhängig von einem Betriebssystem und/oder einer Grafiktreibersoftware auf einem System zu sichern und wiederherzustellen.

[0374] In mindestens einer Ausführungsform kann der Grafikprozessorkern 3000 mehr oder weniger als die veranschaulichten Teilkern 3001A-3001F aufweisen, bis zu N modulare Teilkern. Für jeden Satz von N Teilkernen kann der Grafikprozessorkern 3000 in mindestens einer Ausführungsform auch gemeinsam genutzte Funktionslogik 3010, gemeinsam genutzten und/oder Cachespeicher 3012, eine Geometrie-/Festfunktionspipeline 3014 sowie zusätzliche Festfunktionslogik 3016 zum Beschleunigen verschiedener Grafik- und Rechenverarbeitungsoperationen beinhalten. In mindestens einer Ausführungsform kann die gemeinsam genutzte Funktionslogik 3010 Logikeinheiten (z. B. Abtaster-, Mathematik- und/oder Zwischen-Thread-Kommunikationslogik) beinhalten, die von N Teilkernen innerhalb des Grafikprozessorkerns 3000 gemeinsam genutzt werden können. In mindestens einer Ausführungsform kann der gemeinsam genutzte und/oder Cachespeicher 3012 Cache der letzten Ebene für N Teilkern 3001A-3001F innerhalb des Grafikprozessorkerns 3000 sein und auch als gemeinsam genutzter Speicher dienen, auf den mehrere Teilkern zugreifen können. In mindestens einer Ausführungsform kann die Geometrie-/Festfunktionspipeline 3014 anstelle der

Geometrie-/Festfunktionspipeline 3036 innerhalb des Festfunktionsblocks 3030 enthalten sein und ähnliche Logikeinheiten beinhalten.

[0375] In mindestens einer Ausführungsform beinhaltet der Grafikprozessorkern 3000 zusätzliche Festfunktionslogik 3016, die verschiedene Festfunktionsbeschleunigungslogik zur Verwendung durch den Grafikprozessorkern 3000 beinhalten kann. In mindestens einer Ausführungsform beinhaltet die zusätzliche Festfunktionslogik 3016 eine zusätzliche Geometriepipeline für die Verwendung beim Shading von nur der Position. Beim Shading von nur der Position existieren mindestens zwei Geometriepipelines, wohingegen eine vollständige Geometriepipeline innerhalb der Geometrie- und Festfunktionspipelines 3014, 3036 und eine Culling-Pipeline, die eine zusätzliche Geometriepipeline ist, die innerhalb der zusätzlichen Festfunktionslogik 3016 enthalten sein kann. In mindestens einer Ausführungsform ist eine Culling-Pipeline eine abgespeckte Version einer vollständigen Geometriepipeline. In mindestens einer Ausführungsform können eine vollständige Pipeline und eine Culling-Pipeline unterschiedliche Instanzen einer Anwendung ausführen, wobei jede Instanz einen separaten Kontext aufweist. In mindestens einer Ausführungsform kann eine Schattierung von lediglich der Position lange Ausleseläufe von verworfenen Dreiecken verbergen, wodurch die Schattierung in einigen Fällen früher abgeschlossen werden kann. Zum Beispiel kann in mindestens einer Ausführungsform die Culling-Pipeline-Logik innerhalb der zusätzlichen Festfunktionslogik 3016 Positions-Shader parallel zu einer Hauptanwendung ausführen und sie erzeugt im Allgemeinen kritische Ergebnisse schneller als eine vollständige Pipeline, da eine Culling-Pipeline Positionsattribute von Vertices abrufen und schattiert, ohne Rasterung und Rendering von Pixeln in einem Bildspeicher durchzuführen. In mindestens einer Ausführungsform kann eine 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 eine 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.

[0376] In mindestens einer Ausführungsform kann die zusätzliche Festfunktionslogik 3016 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.

[0377] In mindestens einer Ausführungsform beinhaltet jeder Grafikteilkern 3001A-3001F 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 Grafikteilkern 3001A-3001F mehrere EU-Arrays 3002A-3002F, 3004A-3004F, Logik 3003A-3003F zur Thread-Zuteilung und Zwischen-Thread-Kommunikation (thread dispatch/inter-thread communication-TD/IC), einen 3D- (z. B. Textur-) Abtaster 3005A-3005F, einen Medienabtaster 3006A-3006F, einen Shader-Prozessor 3007A-3007F und gemeinsam genutzten lokalen Speicher (shared local memory - SLM) 3008A-3008F. In mindestens einer Ausführungsform beinhalten die EU-Arrays 3002A-3002F, 3004A-3004F jeweils mehrere Ausführungseinheiten, bei denen es sich um Universal-Grafikverarbeitungseinheiten handelt, die dazu 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 3003A-3003F lokale Thread-Zuteilungs- und Thread-Steueroperationen für Ausführungseinheiten innerhalb eines Teilkerns aus und ermöglicht die Kommunikation zwischen Threads, die auf Ausführungseinheiten eines Teilkerns ausgeführt werden. In mindestens einer Ausführungsform können die 3D-Abtaster 3005A-3005F Textur- oder andere mit 3D-Grafik verwandte Daten in Speicher lesen. In mindestens einer Ausführungsform können die 3D-Abtaster Texturdaten auf Grundlage eines konfigurierten Abtastzustands und eines Texturformats, das mit einer gegebenen Textur assoziiert ist, unterschiedlich lesen. In mindestens einer Ausführungsform können die Medienabtaster 3006A-3006F ähnliche Leseoperationen auf Grundlage eines Typs und Formats, die mit den Mediendaten assoziiert sind, durchführen. In mindestens einer Ausführungsform kann jeder Grafikteilkern 3001A-3001F alternativ einen einheitlichen 3D-Abtaster und Medienabtaster beinhalten. In mindestens einer Ausführungsform können Threads, die auf Ausführungseinheiten in jedem der Teilkern 3001A-3001F ausgeführt werden, den gemeinsam genutzten lokalen Speicher 3008A-3008F in jedem Teilkern nutzen, um es Threads, die innerhalb einer Thread-Gruppe ausgeführt werden, zu ermöglichen, unter Verwendung eines gemeinsamen Pools von chipinternem Speicher ausgeführt zu werden.

[0378] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform können Teile oder die Gesamtheit der Inferenz- und/oder Trainingslogik 115 in den Grafikprozessorkern 3000 einbezogen sein. Zum Beispiel können in mindestens einer Ausführungsform die hierin beschriebenen Trainings- und/oder Inferenztechniken eine oder mehrere der in einer 3D-Pipeline verkörpert ALUs, den Grafik-Mikrocontroller 3038, die Geometrie- und Festfunktionspipeline 3014 und 3036 oder andere Logik in **Fig. 30** verwenden. Darüber hinaus können in mindestens einer Ausführungsform die hierin beschriebenen Inferenz- und/oder Trainingsoperationen unter Verwendung von anderer Logik als der in **Fig. 1A** oder **Fig. 1B** dargestellten 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 Grafikprozessorkerns 3000 konfigurieren, um eine(n) oder mehrere hierin beschriebene Algorithmen des maschinellen Lernens, neuronale Netzarchitekturen, Anwendungsfälle oder Trainingstechniken durchzuführen.

[0379] **Fig. 31A-31B** veranschaulichen Thread-Ausführungslogik 3100, die ein Array von Verarbeitungselementen eines Grafikprozessorkerns beinhaltet, gemäß mindestens einer Ausführungsform. **Fig. 31A** veranschaulicht mindestens eine Ausführungsform, in der die Thread-Ausführungslogik 3100 verwendet wird. **Fig. 31B** veranschaulicht beispielhafte interne Details einer Grafikausführungseinheit 3108 gemäß mindestens einer Ausführungsform.

[0380] Wie in **Fig. 31A** dargestellt, beinhaltet die Thread-Ausführungslogik 3100 in mindestens einer Ausführungsform einen Shader-Prozessor 3102, einen Thread-Zuteiler 3104, einen Anweisungs-Cache 3106, ein skalierbares Ausführungseinheitenarray einschließlich einer Vielzahl von Ausführungseinheiten 3107A-3107N und 3108A-3108N, einen Abtaster 3110, einen Datencache 3112 und einen Datenport 3114. In mindestens einer Ausführungsform kann ein skalierbares Ausführungseinheitenarray dynamisch skalieren, indem eine oder mehrere Ausführungseinheiten (z. B. beliebige der Ausführungseinheiten 3108A-N oder 3107A-N) zum Beispiel auf Grundlage von Rechenanforderungen eines Workloads aktiviert oder deaktiviert werden. In mindestens einer Ausführungsform sind die skalierbaren Ausführungseinheiten über eine Zusammenschaltungsstruktur zusammengeschaltet, die mit jeder Ausführungseinheit verknüpft ist. In mindestens einer Ausführungsform beinhaltet die Thread-Ausführungslogik 3100 eine oder mehrere Verbindungen mit Speicher, wie etwa Systemspeicher oder Cachespeicher, durch eines oder mehrere des Anweisungs-Cache 3106, des Datenports 3114, des Abtasters 3110 und der Ausführungseinheiten 3107 oder 3108. In mindestens einer Ausführungsform ist jede Ausführungseinheit (z. B. 3107A) 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 3107 und/oder 3108 so skalierbar, dass es eine beliebige Anzahl einzelner Ausführungseinheiten beinhaltet.

[0381] In mindestens einer Ausführungsform werden die Ausführungseinheiten 3107 und/oder 3108 hauptsächlich zum Ausführen von Shader-Programmen verwendet. In mindestens einer Ausführungsform kann der Shader-Prozessor 3102 verschiedene Shader-Programme verarbeiten und mit den Shader-Programmen assoziierte Ausführungs-Threads über einen Thread-Zuteiler 3104 zuteilen. In mindestens einer Ausführungsform beinhaltet der Thread-Zuteiler 3104 Logik zum Vermitteln von Thread-Initiierungsanforderungen von Grafik- und Medienpipelines und zum Instanzieren angeforderter Threads auf einer oder mehreren Ausführungseinheiten in den Ausführungseinheiten 3107 und/oder 3108. Zum Beispiel kann in mindestens einer Ausführungsform eine Geometriepipeline Scheitelpunkt-, Tesselations- oder Geometrie-Shader zur Verarbeitung an die Thread-Ausführungslogik versenden. In mindestens einer Ausführungsform kann der Thread-Zuteiler 3104 auch Laufzeit-Thread-Erzeugungsanforderungen von ausführenden Shader-Programmen verarbeiten.

[0382] In mindestens einer Ausführungsform unterstützen die Ausführungseinheiten 3107 und/oder 3108 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 Ausführungseinheiten Scheitelpunkt- und Geometrieverarbeitung (z. B. Scheitelpunktprogramme, Geometrieprogramme und/oder Scheitelpunkt-Shader), Pixelverarbeitung (z. B. Pixel-Shader, Fragment-Shader) und Allzweckverarbeitung (z. B. Rechen- und Medien-Shader). In mindestens einer Ausführungsform ist jede der Ausführungseinheiten 3107 und/oder 3108, 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 assoziierten unabhängigen Thread-Status auf. In mindestens einer Ausführungsform erfolgt die Ausführung mit mehreren Ausgaben pro Takt an Pipelines, die für Integer- und Gleitkommaoperationen mit einfacher oder doppelter Genauigkeit, SIMD-Verzweigungsfähigkeit, logische Operationen, transzendente Operationen und andere verschiedene Operationen in der Lage sind. In mindestens einer Ausführungsform bewirkt die Abhängigkeitslogik innerhalb der Ausführungseinheiten 3107 und/oder 3108, dass ein wartender Thread schläft, während er auf Daten aus dem Speicher oder einer der gemeinsam genutzten Funktionen wartet, bis die angeforderten Daten zurückgegeben wurden. In mindestens einer Ausführungsform können, während ein wartender Thread schläft, Hardware-Ressourcen der Verarbeitung anderer Threads gewidmet werden. Zum Beispiel kann in mindestens einer Ausführungsform eine Ausführungseinheit während einer Verzögerung, die mit einem Scheitelpunkt-Shader-Vorgang assoziiert ist, Vorgänge für einen Pixel-Shader, Fragment-Shader oder eine andere Art von Shader-Programm, einschließlich eines anderen Scheitelpunkt-Shaders, durchführen.

[0383] In mindestens einer Ausführungsform arbeitet jede Ausführungseinheit in den Ausführungseinheiten 3107 und/oder 3108 an Arrays von Datenelementen. In mindestens einer Ausführungsform ist die Anzahl der Datenelemente eine „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 3107 und/oder 3108 Integer- und Gleitkommatentypen.

[0384] In mindestens einer Ausführungsform beinhaltet ein Anweisungssatz einer Ausführungseinheit SIMD-Anweisungen. In mindestens einer Ausführungsform können verschiedene Datenelemente als eine Paketdatenart in einem Register gespeichert werden und die Ausführungseinheit verarbeitet verschiedene Elemente basierend auf der Datengröße der Elemente. Zum Beispiel werden in mindestens einer Ausführungsform bei dem Betrieb an einem 256 Bit breiten Vektor 256 Bit eines Vektors in einem Register gespeichert und wird eine Ausführungseinheit an einem Vektor als vier separate 64-Bit-Paket-Datenelemente (Datenelemente der Größe Quad-Word (QW)), acht separate 32-Bit-Paket-Datenelemente (Datenelemente der Größe Double Word (DW)), sechzehn separate 16-Bit-Paket-Datenelemente (Datenelemente der Größe Word (W)) oder zweiunddreißig separate 8-Bit-Datenelemente (Datenelemente der Größe Byte (B)) betrieben. In mindestens einer Ausführungsform sind jedoch andere Vektorbreiten und Registergrößen möglich.

[0385] In mindestens einer Ausführungsform können eine oder mehrere Ausführungseinheiten zu einer fusionierten Ausführungseinheit 3109A-3109N kombiniert werden, die Thread-Steuerlogik (3111A-3111N) aufweist, die fusionierten EUs gemeinsam ist, wie etwa die Ausführungseinheit 3107A, die mit der Ausführungseinheit 3108A zu der fusionierten Ausführungseinheit 3109A fusioniert wird. In mindestens einer Ausführungsform können mehrere EUs zu einer EU-Gruppe fusioniert werden. In mindestens einer Ausführungsform kann jede EU in einer fusionierten EU-Gruppe so konfiguriert sein, dass sie einen separaten SIMD-Hardware-Thread ausführt, wobei eine Anzahl der EUs in einer fusionierten EU-Gruppe gemäß verschiedenen Ausführungsformen möglicherweise variiert. 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 3109A-3109N mindestens zwei Ausführungseinheiten. Zum Beispiel beinhaltet in mindestens einer Ausführungsform die fusionierte Ausführungseinheit 3109 A eine erste EU 3107 A, eine zweite EU 3108 A und Thread-Steuerlogik 3111 A, die der ersten EU 3107 A und der zweiten EU 3108 A gemeinsam ist. In mindestens einer Ausführungsform steuert die Thread-Steuerlogik 3111A Threads, die auf der fusionierten Grafikausführungseinheit 3109A ausgeführt werden, sodass jede EU innerhalb der fusionierten Ausführungseinheiten 3109A-3109N unter Verwendung eines gemeinsamen Anweisungszeigerregisters ausgeführt werden kann.

[0386] In mindestens einer Ausführungsform sind ein oder mehrere interne Anweisungs-Caches (z.B. 3106) in der Thread-Ausführungslogik 3100 enthalten, um Thread-Anweisungen für Ausführungseinheiten zwischenspeichern. In mindestens einer Ausführungsform sind ein oder mehrere Datencaches (z. B. 3112) enthalten, um Thread-Daten während der Thread-Ausführung zwischenspeichern. In mindestens einer Ausführungsform ist ein Abtaster 3110 enthalten, um Texturabtastung für 3 D-Operationen und Medienabtastung für Medienoperationen bereitzustellen. In mindestens einer Ausführungsform beinhaltet der Abtaster

3110 eine spezialisierte Textur- oder Medienabtafstunktionalität, um Textur- oder Mediendaten während eines Abtastprozesses zu verarbeiten, bevor die abgetasteten Daten einer Ausführungseinheit bereitgestellt werden.

[0387] Während der Ausführung senden Grafik- und Medienpipelines in mindestens einer Ausführungsform Thread-Initiierungsanforderungen über die Thread-Erzeugungs- und -Zuteilungslogik an die Thread-Ausführungslogik 3100. 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 3102 aufgerufen, um Ausgabeinformationen weiter zu berechnen und zu veranlassen, dass die Ergebnisse in Ausgabebereichen (z. B. Farbpuffer, Tiefenpuffer, Stencil-Puffer usw.) geschrieben werden. In mindestens einer Ausführungsform berechnet ein Pixel-Shader oder Fragment-Shader die Werte verschiedener Scheitelpunktattribute, die über ein gerastertes Objekt interpoliert werden sollen. In mindestens einer Ausführungsform führt die Pixelprozessorlogik innerhalb des Shader-Prozessors 3102 dann ein über eine Anwendungsprogrammierschnittstelle (application programming interface - API) zugeführtes Pixel- oder Fragment-Shader-Programm aus. In mindestens einer Ausführungsform teilt der Shader-Prozessor 3102 zum Ausführen eines Shader-Programms Threads über den Thread-Zuteiler 3104 einer Ausführungseinheit (z.B. 3108A) zu. In mindestens einer Ausführungsform verwendet der Shader-Prozessor 3102 die Texturabtafstlogik in dem Abtaster 3110, um auf Texturdaten in Texturkarten zuzugreifen, die in Speicher gespeichert sind. In mindestens einer Ausführungsform werden durch arithmetische Vorgänge an Texturdaten und Eingabegeometriedaten Pixelfarbdaten für jedes geometrische Fragment berechnet oder ein oder mehrere Pixel von der weiteren Verarbeitung ausgeschlossen.

[0388] In mindestens einer Ausführungsform stellt der Datenport 3114 einen Speicherzugriffsmechanismus für die Thread-Ausführungslogik 3100 bereit, um verarbeitete Daten zur weiteren Verarbeitung an einer Grafikprozessor-Ausgabepipeline an Speicher auszugeben. In mindestens einer Ausführungsform beinhaltet der Datenport 3114 einen oder mehrere Cache-Speicher (z. B. den Daten-Cache 3112) oder ist mit diesen gekoppelt, um Daten für den Speicherzugriff über einen Datenport zu cachieren.

[0389] Wie in **Fig. 31B** veranschaulicht, kann eine Grafikausführungseinheit 3108 in mindestens einer Ausführungsform eine Anweisungsabrufeinheit 3137, ein Array von allgemeinen Registerbänken (general register file - GRF) 3124, ein Array von architektonischen Registerbänken (architectural register file - ARF) 3126, einen Thread-Vermittler 3122, eine Sendeeinheit 3130, eine Verzweigungseinheit 3132, einen Satz SIMD-Gleitkommaeinheiten (FPUs) 3134 und einen Satz dedizierter Ganzzahl-SIMD-ALUs 3135 beinhalten. In mindestens einer Ausführungsform beinhalten die GRF 3124 und die ARF 3126 einen Satz allgemeiner Registerbänke und Architekturregisterbänke, die mit jedem simultanen Hardware-Thread assoziiert sind, der in der Grafikausführungseinheit 3108 aktiv sein kann. In mindestens einer Ausführungsform wird der architektonische Zustand pro Thread in der ARF 3126 verwaltet, während die während der Thread-Ausführung verwendeten Daten in der GRF 3124 gespeichert werden. In mindestens einer Ausführungsform kann der Ausführungszustand jedes Threads, einschließlich der Anweisungszeiger für jeden Thread, in Thread-spezifischen Registern in der ARF 3126 aufbewahrt werden.

[0390] In mindestens einer Ausführungsform weist die Grafikausführungseinheit 3108 eine Architektur auf, die eine Kombination aus simultanem Multi-Threading (Simultaneous Multi-Threading - SMT) und feinkörnigem verschachteltem Multi-Threading (Interleaved Multi-Threading - IMT) ist. In mindestens einer Ausführungsform weist die Architektur eine modulare Konfiguration auf, die zur Ausgestaltungszeit basierend auf einer Zielanzahl von simultanen Threads und der Anzahl von Registern pro Ausführungseinheit fein abgestimmt werden kann, wobei die Ressourcen der Ausführungseinheit auf die Logik aufgeteilt werden, die zur Ausführung mehrerer simultaner Threads verwendet wird.

[0391] In mindestens einer Ausführungsform kann die Grafikausführungseinheit 3108 mehrere Anweisungen gleichzeitig ausgeben, die jeweils unterschiedliche Anweisungen sein können. In mindestens einer Ausführungsform kann der Thread-Vermittler 3122 des Grafikausführungseinheits-Threads 3108 Anweisungen einer der Sendeeinheit 3130, der Verzweigungseinheit 3132 oder der SIMD-FPU(s) 3134 zur Ausführung zuteilen. In mindestens einer Ausführungsform kann jeder Ausführungs-Thread auf 128 Universalregister innerhalb der GRF 3124 zugreifen, wobei jedes Register 32 Byte speichern kann, die als SIMD-8-Elementvektor von 32-Bit-Datenelementen zugänglich sind. In mindestens einer Ausführungsform weist jeder Thread der Ausführungseinheit Zugriff auf 4 Kilobyte innerhalb der GRF 3124 auf, obwohl Ausführungsformen nicht derart beschränkt sind und in anderen Ausführungsformen mehr oder weniger Registerressourcen bereitgestellt sein können. In mindestens einer Ausführungsform können bis zu sieben Threads simultan ausgeführt werden, obwohl die Anzahl der Threads pro Ausführungseinheit gemäß Ausführungsform auch variieren

kann. In mindestens einer Ausführungsform, in der sieben Threads auf 4 Kilobyte zugreifen können, kann die GRF 3124 insgesamt 28 Kilobyte speichern. In mindestens einer Ausführungsform können flexible Adressierungsmodi erlauben, dass Register zusammen adressiert werden, um effektiv breitere Register zu bilden oder um geschichtete rechteckige Blockdatenstrukturen darzustellen.

[0392] In mindestens einer Ausführungsform werden Speicheroperationen, Abtasteroperationen und andere Systemkommunikation mit längerer Latenz über „Sende“-Anweisungen zugeteilt, die durch Nachrichtenweitergabe an die Sendeeinheit 3130 ausgeführt werden. In mindestens einer Ausführungsform werden Verzweigungsanweisungen der Verzweigungseinheit 3132 zugeteilt, um SIMD-Divergenz und schließlich -Konvergenz zu erleichtern.

[0393] In mindestens einer Ausführungsform beinhaltet die Grafikausführungseinheit 3108 eine oder mehrere SIMD-Gleitkommaeinheiten (FPU(s)) 3134 zum Durchführen von Gleitkommaoperationen. In mindestens einer Ausführungsform unterstützen die FPU(s) 3134 auch Ganzzahlberechnung. In mindestens einer Ausführungsform können die FPU(s) 3134 bis zu einer Anzahl von M 32-Bit-Gleitkomma- (oder -Ganzzahl-) Operationen über SIMD ausführen oder bis zu 2M 16-Bit-Ganzzahl- oder 16-Bit-Gleitkommaoperationen über SIMD ausführen. In mindestens einer Ausführungsform stellt mindestens eine FPU 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 3135 vorhanden, der speziell für die Durchführung von Vorgängen, die mit Berechnungen für maschinelles Lernen assoziiert sind, optimiert sein kann.

[0394] In mindestens einer Ausführungsform können Arrays von mehreren Instanzen der Grafikausführungseinheit 3108 in einer Grafikteilkern-Gruppierung (z. B. einem Teilslice) instanziiert sein. In mindestens einer Ausführungsform kann die Ausführungseinheit 3108 Anweisungen über eine Vielzahl von Ausführungskanälen hinweg ausführen. In mindestens einer Ausführungsform wird jeder Thread, der auf der Grafikausführungseinheit 3108 ausgeführt wird, auf einem anderen Kanal ausgeführt.

[0395] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. In mindestens einer Ausführungsform können Teile oder die Gesamtheit der Inferenz- und/oder Trainingslogik 115 in die Thread-Ausführungslogik 3100 einbezogen sein. Darüber hinaus können in mindestens einer Ausführungsform die hierin beschriebenen Inferenz- und/oder Trainingsoperationen unter Verwendung von anderer Logik als der in **Fig. 1A** oder **Fig. 1B** dargestellten 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 der Thread-Ausführungslogik 3100 konfigurieren, um eine(n) oder mehrere hierin beschriebene Algorithmen des maschinellen Lernens, neuronale Netzarchitekturen, Anwendungsfälle oder Trainingstechniken durchzuführen.

[0396] **Fig. 32** veranschaulicht eine Parallelverarbeitungseinheit („PPU“) 3200 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist die PPU 3200 mit maschinenlesbarem Code konfiguriert, der bei Ausführung durch die PPU 3200 die PPU 3200 dazu veranlasst, einige oder alle der in dieser gesamten Offenbarung beschriebenen Prozesse und Techniken durchzuführen. In mindestens einer Ausführungsform ist die PPU 3200 ein MultiThread-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 3200 ausgeführt zu werden. In mindestens einer Ausführungsform ist die PPU 3200 eine Grafikverarbeitungseinheit („GPU“), die so konfiguriert ist, dass sie eine Grafik-Rendering-Pipeline zum Verarbeiten dreidimensionaler („3D“) Grafikdaten implementiert, um zweidimensionale („2D“) Bilddaten zur Anzeige auf einer Anzeigevorrichtung, wie etwa einer Vorrichtung mit Flüssigkristallanzeige („LCD“), zu erzeugen. In mindestens einer Ausführungsform wird die PPU 3200 genutzt, um Berechnungen, wie etwa lineare Algebraoperationen und Operationen des maschinellen Lernens, durchzuführen. **Fig. 32** veranschaulicht ein Beispiel für einen Parallelprozessor, der nur zu Veranschaulichungszwecken dient und als nicht einschränkendes Beispiel für Prozessorarchitekturen auszulegen ist, 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.

[0397] In mindestens einer Ausführungsform sind eine oder mehrere PPU 3200 so konfiguriert, dass sie Anwendungen für Hochleistungsrechnen (High Performance Computing - „HPC“), für ein Rechenzentrum und für maschinelles Lernen beschleunigen. In mindestens einer Ausführungsform ist die PPU 3200 so konfiguriert, dass sie Deep-Learning-Systeme und -Anwendungen beschleunigt, einschließlich der folgenden nicht einschränkenden Beispiele: Plattformen für autonome Fahrzeuge, Deep Learning, Sprach-, Bild- und Texterkennungssysteme mit hoher Genauigkeit, intelligente Videoanalytik, molekulare Simulationen, Arzneimittelentdeckung, Krankheitsdiagnose, Wettervorhersage, Big-Data-Analytik, Astronomie, Molekulardynamiksimulation, Finanzmodellierung, Robotik, Fabrikautomatisierung, Echtzeit-Sprachübersetzung, Online-Suchoptimierung und personalisierte Benutzerempfehlungen und mehr.

[0398] In mindestens einer Ausführungsform beinhaltet die PPU 3200 ohne Einschränkung eine Eingabe-/Ausgabe- („E/A“-)Einheit 3206, eine Frontend-Einheit 3210, eine Planer-Einheit 3212, eine Arbeitsverteilungseinheit 3214, einen Hub 3216, eine Kreuzschiene (crossbar - „XBar“) 3220, einen oder mehrere Universalverarbeitungscluster („GPCs“) 3218 und eine oder mehrere Partitionseinheiten („Speicherpartitionseinheiten“) 3222. In mindestens einer Ausführungsform ist die PPU 3200 mit einem Hostprozessor oder anderen PPU 3200 über eine oder mehrere Hochgeschwindigkeits-GPU-Zusammenschaltungen („GPU-Zusammenschaltungen“) 3208 verbunden. In mindestens einer Ausführungsform ist die PPU 3200 über einen Systembus 3202 mit einem Hostprozessor oder anderen Peripherievorrichtungen verbunden. In mindestens einer Ausführungsform ist die PPU 3200 mit einem lokalen Speicher verbunden, der eine oder mehrere Speichervorrichtungen („Speicher“) 3204 umfasst. In mindestens einer Ausführungsform beinhalten die Speichervorrichtungen 3204 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.

[0399] In mindestens einer Ausführungsform kann sich die Hochgeschwindigkeits-GPU-Zusammenschaltung 3208 auf eine drahtbasierte mehrspurige Kommunikationsverknüpfung beziehen, die durch Systeme zum Skalieren verwendet wird und eine oder mehrere PPU 3200 in Kombination mit einer oder mehreren zentralen Verarbeitungseinheiten („CPUs“) beinhaltet und die Cache-Kohärenz zwischen PPU 3200 und CPUs sowie CPU-Mastering unterstützt. In mindestens einer Ausführungsform werden Daten und/oder Befehle von der Hochgeschwindigkeits-GPU-Zusammenschaltung 3208 über den Hub 3216 zu/von anderen Einheiten der PPU 3200 übertragen, wie z. B. einer/einem oder mehreren Kopier-Engines, Videocodierern, Videodecodierern, Leistungsverwaltungseinheiten und anderen Komponenten, die in **Fig. 32** möglicherweise nicht explizit veranschaulicht sind.

[0400] In mindestens einer Ausführungsform ist die E/A-Einheit 3206 so konfiguriert, dass sie Kommunikation (z. B. Befehle, Daten) von einem Hostprozessor (in **Fig. 32** nicht dargestellt) über den Systembus 3202 überträgt und empfängt. In mindestens einer Ausführungsform kommuniziert die E/A-Einheit 3206 mit dem Hostprozessor direkt über den Systembus 3202 oder durch eine oder mehrere Zwischenvorrichtungen wie etwa eine Speicherbrücke. In mindestens einer Ausführungsform kann die E/A-Einheit 3206 über den Systembus 3202 mit einem oder mehreren anderen Prozessoren kommunizieren, wie etwa einer oder mehreren der PPU 3200. In mindestens einer Ausführungsform implementiert die E/A-Einheit 3206 eine Peripheral-Component-Interconnect-Express- („PCIe“-)Schnittstelle für die Kommunikation über einen PCIe-Bus. In mindestens einer Ausführungsform implementiert die E/A-Einheit 3206 Schnittstellen zum Kommunizieren mit externen Vorrichtungen.

[0401] In mindestens einer Ausführungsform decodiert die E/A-Einheit 3206 über den Systembus 3202 empfangene Pakete. In mindestens einer Ausführungsform stellen mindestens einige Pakete Befehle dar, die so konfiguriert sind, dass sie die PPU 3200 dazu veranlassen, verschiedene Operationen durchzuführen. In mindestens einer Ausführungsform überträgt die E/A-Einheit 3206 decodierte Befehle an verschiedene andere Einheiten der PPU 3200, wie durch Befehle vorgegeben. In mindestens einer Ausführungsform werden Befehle an die Frontend-Einheit 3210 übertragen und/oder an den Hub 3216 oder andere Einheiten der PPU 3200 übertragen, wie etwa eine oder mehrere Kopier-Engines, einen Videocodierer, einen Videodecodierer, eine Leistungsverwaltungseinheit usw. (in **Fig. 32** nicht explizit dargestellt). In mindestens einer Ausführungsform ist die E/A-Einheit 3206 so konfiguriert, dass sie die Kommunikation zwischen und unter verschiedenen logischen Einheiten der PPU 3200 routet.

[0402] In mindestens einer Ausführungsform kodiert ein vom Hostprozessor ausgeführtes Programm einen Befehlsstrom in einem Puffer, welcher der PPU 3200 Workloads zur Verarbeitung bereitstellt. In mindestens einer Ausführungsform umfasst ein Workload Anweisungen und Daten, die von diesen Anweisungen verar-

beitet werden sollen. In mindestens einer Ausführungsform ist ein Puffer eine Region in einem Speicher, auf die sowohl ein Hostprozessor als auch die PPU 3200 zugreifen können (z. B. Lesen/Schreiben) - eine Host-schnittstelleneinheit kann so konfiguriert sein, dass sie auf diesen Puffer in einem Systemspeicher, der mit dem Systembus 3202 verbunden ist, über Speichieranforderungen zugreift, die über den Systembus 3202 durch die E/A-Einheit 3206 übertragen werden. In mindestens einer Ausführungsform schreibt ein Host-Prozessor einen Befehlsstrom in einen Puffer und überträgt dann einen Zeiger für einen Start eines Befehlsstroms an die PPU 3200, sodass die Frontend-Einheit 3210 Zeiger für einen oder mehrere Befehlsströme empfängt und einen oder mehrere Befehlsströme verwaltet, indem sie Befehle aus Befehlsströmen liest und Befehle an verschiedene Einheiten der PPU 3200 weiterleitet.

[0403] In mindestens einer Ausführungsform ist die Frontend-Einheit 3210 an die Planer-Einheit 3212 gekoppelt, die verschiedene GPCs 3218 zum Verarbeiten von Tasks konfiguriert, die durch einen oder mehrere Befehlsströme definiert sind. In mindestens einer Ausführungsform ist die Planer-Einheit 3212 so konfiguriert, dass sie Zustandsinformationen in Bezug auf verschiedene durch die Planer-Einheit 3212 verwaltete Tasks verfolgt, wobei die Zustandsinformationen angeben können, welchem der GPCs 3218 ein Task zugeordnet ist, ob der Task aktiv oder inaktiv ist, welche Prioritätsstufe mit dem Task assoziiert ist, und so weiter. In mindestens einer Ausführungsform verwaltet die Planer-Einheit 3212 die Ausführung einer Vielzahl von Tasks auf einem oder mehreren GPCs 3218.

[0404] In mindestens einer Ausführungsform ist die Planer-Einheit 3212 an die Arbeitsverteilungseinheit 3214 gekoppelt, die so konfiguriert ist, dass sie Tasks zur Ausführung auf den GPCs 3218 zuteilt. In mindestens einer Ausführungsform verfolgt die Arbeitsverteilungseinheit 3214 eine Anzahl geplanter Tasks nach, die von der Planer-Einheit 3212 empfangen wurde, und die Arbeitsverteilungseinheit 3214 verwaltet einen Pool ausstehender Tasks und einen Pool aktiver Tasks für jeden der GPCs 3218. In mindestens einer Ausführungsform umfasst der Pool ausstehender Tasks eine Anzahl von Slots (z. B. 32 Slots), die Tasks enthalten, die zur Verarbeitung durch einen konkreten GPC 3218 zugeordnet sind; ein Pool aktiver Tasks kann eine Anzahl von Slots (z. B. 4 Slots) für Tasks umfassen, die aktiv durch die GPCs 3218 verarbeitet werden, sodass, wenn einer der GPCs 3218 die Ausführung eines Tasks abschließt, dieser Task aus diesem Pool aktiver Tasks für den GPC 3218 entfernt wird und ein anderer Task aus einem Pool ausstehender Tasks ausgewählt und zur Ausführung auf dem GPC 3218 eingeplant wird. Falls ein aktiver Task auf dem GPC 3218 inaktiv ist, etwa während er darauf wartet, dass eine Datenabhängigkeit aufgelöst wird, dann wird in mindestens einer Ausführungsform dieser aktive Task aus dem GPC 3218 entfernt und in diesen Pool ausstehender Tasks zurückgeführt, während ein anderer Task in diesem Pool ausstehender Tasks ausgewählt und zur Ausführung auf dem GPC 3218 eingeplant wird.

[0405] In mindestens einer Ausführungsform kommuniziert die Arbeitsverteilungseinheit 3214 mit einem oder mehreren GPCs 3218 über die XBar 3220. In mindestens einer Ausführungsform ist die XBar 3220 ein Zusammenschaltungsnetz, das viele Einheiten der PPU 3200 an andere Einheiten der PPU 3200 koppelt und dazu konfiguriert sein kann, die Arbeitsverteilungseinheit 3214 an einen konkreten GPC 3218 zu koppeln. In mindestens einer Ausführungsform können auch eine oder mehrere andere Einheiten der PPU 3200 über den Hub 3216 mit der XBar 3220 verbunden sein.

[0406] In mindestens einer Ausführungsform werden Tasks durch die Planer-Einheit 3212 verwaltet und durch die Arbeitsverteilungseinheit 3214 einem der GPCs 3218 zugeteilt. Der GPC 3218 ist so konfiguriert, dass er die Task verarbeitet und Ergebnisse erzeugt. In mindestens einer Ausführungsform können die Ergebnisse durch andere Tasks innerhalb des GPC 3218 verbraucht, über die XBar 3220 an einen anderen GPC 3218 geroutet oder in dem Speicher 3204 gespeichert werden. In mindestens einer Ausführungsform können die Ergebnisse über die Partitionseinheiten 3222, die eine Speicherschnittstelle zum Lesen und Schreiben von Daten in den/aus dem Speicher 3204 implementieren, in den Speicher 3204 geschrieben werden. In mindestens einer Ausführungsform können die Ergebnisse über die Hochgeschwindigkeits-GPU-Zusammenschaltung 3208 an eine andere PPU 3200 oder CPU übertragen werden. In mindestens einer Ausführungsform beinhaltet die PPU 3200 ohne Einschränkung eine Anzahl U von Partitionseinheiten 3222, die gleich einer Anzahl von separaten und unterschiedlichen Speichervorrichtungen 3204 ist, die an die PPU 3200 gekoppelt sind, wie hierin in Verbindung mit **Fig. 34** detaillierter beschrieben.

[0407] In mindestens einer Ausführungsform führt ein Hostprozessor einen Treiberkernel aus, der eine Anwendungsprogrammierschnittstelle („API“) implementiert, die es einer oder mehreren auf einem Hostprozessor ausgeführten Anwendungen ermöglicht, Operationen zur Ausführung auf der PPU 3200 zu planen. In mindestens einer Ausführungsform werden mehrere Rechenanwendungen simultan durch die PPU 3200 ausgeführt und die PPU 3200 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 einen Treiberkernel dazu veranlassen, einen oder mehrere Tasks zur Ausführung durch die PPU 3200 zu erzeugen, und dieser Treiberkernel gibt Tasks an einen oder mehrere Ströme aus, die durch die PPU 3200 verarbeitet werden. In mindestens einer Ausführungsform umfasst jeder Task eine oder mehrere Gruppen zugehöriger 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 zur Durchführung von Tasks enthalten und die Daten über einen gemeinsam genutzten Speicher austauschen. In mindestens einer Ausführungsform werden Threads und kooperierende Threads in Verbindung mit **Fig. 34** detaillierter beschrieben.

[0408] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. 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 3200 bereitgestellte Informationen vorherzusagen oder zu inferenzieren. In mindestens einer Ausführungsform wird die PPU 3200 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 3200 trainiert wurde. In mindestens einer Ausführungsform kann die PPU 3200 verwendet werden, um einen oder mehrere der hierin beschriebenen Anwendungsfälle für neuronale Netze durchzuführen.

[0409] **Fig. 33** veranschaulicht ein allgemeines Verarbeitungscluster („GPC“) 3300 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform handelt es sich bei dem GPC 3300 um den GPC 3218 aus **Fig. 32**. In mindestens einer Ausführungsform beinhaltet jeder GPC 3300 ohne Einschränkung eine Anzahl von Hardware-Einheiten zum Verarbeiten von Tasks und beinhaltet jeder GPC 3300 ohne Einschränkung einen Pipelineverwalter 3302, eine Vor-Rasteroperationeneinheit (pre-raster operations unit - „preROP“) 3304, eine Raster-Engine 3308, eine Arbeitsverteilungskreuzschiene (work distribution crossbar - „WDX“) 3316, eine Speicherverwaltungseinheit („MMU“) 3318, ein oder mehrere Datenverarbeitungscluster (Data Processing Clusters - „DPCs“) 3306 und eine beliebige geeignete Kombination von Teilen.

[0410] In mindestens einer Ausführungsform wird der Betrieb des GPC 3300 durch den Pipelineverwalter 3302 gesteuert. In mindestens einer Ausführungsform verwaltet der Pipelineverwalter 3302 die Konfiguration eines oder mehrerer DPCs 3306 für die Verarbeitung von Tasks, die dem GPC 3300 zugewiesen sind. In mindestens einer Ausführungsform konfiguriert der Pipelineverwalter 3302 mindestens einen von einem oder mehreren DPCs 3306 dazu, mindestens einen Teil einer Grafik-Rendering-Pipeline zu implementieren. In mindestens einer Ausführungsform ist der DPC 3306 dazu konfiguriert, ein Scheitelpunkt-Shader-Programm auf einem programmierbaren Streaming-Multiprozessor („SM“) 3314 auszuführen. In mindestens einer Ausführungsform ist der Pipelineverwalter 3302 dazu konfiguriert, die von einer Arbeitsverteilungseinheit empfangenen Pakete an geeignete logische Einheiten innerhalb des GPC 3300 routet, und in mindestens einer Ausführungsform können einige Pakete an Festfunktions-Hardwareeinheiten in dem preROP 3304 und/oder der Raster-Engine 3308 geroutet werden, während andere Pakete zum Verarbeiten durch eine Primitiv-Engine 3312 oder den SM 3314 an die DPCs 3306 geroutet werden können. In mindestens einer Ausführungsform konfiguriert der Pipelineverwalter 3302 mindestens einen der DPCs 3306 zur Implementierung eines Modells eines neuronalen Netzwerks und/oder einer Rechenpipeline.

[0411] In mindestens einer Ausführungsform ist die preROP-Einheit 3304 dazu konfiguriert, in mindestens einer Ausführungsform die durch die Raster-Engine 3308 und die DPCs 3306 erzeugten Daten an eine Einheit für Rasteroperationen („ROP“) in der Partitionseinheit 3222 zu routen, die vorstehend in Verbindung mit **Fig. 32** detaillierter beschrieben wurde. In mindestens einer Ausführungsform ist die preROP-Einheit 3304 dazu konfiguriert, unter anderem Optimierungen für die Farbmischung durchzuführen, Pixelfarbdaten zu organisieren und Adressübersetzungen durchzuführen. In mindestens einer Ausführungsform beinhaltet die Raster-Engine 3308 ohne Einschränkung eine Anzahl von Festfunktions-Hardwareeinheiten, die so konfiguriert sind, dass sie verschiedene Rastervorgänge durchführen, und in mindestens einer Ausführungsform beinhaltet die Raster-Engine 3308 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 Setup-Engine transformierte Scheitelpunkte und erzeugt Ebenengleichungen, die mit dem durch die Scheitelpunkte

definierten geometrischen Primitiv assoziiert sind; die Ebenengleichungen werden an eine Grobraster-Engine übertragen, um Abdeckungsinformationen (z. B. eine x-, y-Abdeckungsmaske für eine Kachel) für das Primitiv zu erzeugen; die Ausgabe einer Grobraster-Engine wird an eine Culling-Engine übertragen, in der Fragmente, die mit einem Primitiv assoziiert sind und einen z-Test nicht bestehen, Culling unterzogen werden und an eine Clipping-Engine übertragen werden, in der Fragmente, die außerhalb eines Sichtkegelstumpfes liegen, Clipping unterzogen werden. In mindestens einer Ausführungsform werden Fragmente, die das Clipping und Culling überstehen, an eine Feinraster-Engine übergeben, um Attribute für Pixelfragmente auf Grundlage von Ebenengleichungen zu erzeugen, die durch eine Setup-Engine erzeugt werden. In mindestens einer Ausführungsform umfasst eine Ausgabe der Raster-Engine 3308 Fragmente, die durch eine beliebige geeignete Entität, wie etwa durch einen innerhalb des DPC 3306 implementierten Fragment-Shader, verarbeitet werden sollen.

[0412] In mindestens einer Ausführungsform umfasst jeder DPC 3306, der in dem GPC 3300 enthalten ist, ohne Einschränkung eine M-Pipe-Steuerung (M-Pipe Controller - „MPC“) 3310; die Primitiv-Engine 3312; einen oder mehrere SMs 3314 und eine beliebige geeignete Kombination davon. In mindestens einer Ausführungsform steuert die MPC 3310 den Betrieb des DPC 3306 und routet von dem Pipelineverwalter 3302 empfangene Pakete an die entsprechenden Einheiten in dem DPC 3306. In mindestens einer Ausführungsform werden Pakete, die mit einem Scheitelpunkt assoziiert sind, an die Primitiv-Engine 3312 geroutet, die so konfiguriert ist, dass sie Scheitelpunktattribute, die mit einem Scheitelpunkt assoziiert sind, aus dem Speicher abruft; im Gegensatz dazu können Pakete, die mit einem Shader-Programm assoziiert sind, an den SM 3314 übertragen werden.

[0413] In mindestens einer Ausführungsform umfasst der SM 3314 ohne Einschränkung einen programmierbaren Streaming-Prozessor, der dazu konfiguriert ist, Tasks zu verarbeiten, die durch eine Anzahl von Threads dargestellt sind. In mindestens einer Ausführungsform weist der SM 3314 mehrere Threads auf und ist dazu konfiguriert, eine Vielzahl von Threads (z. B. 32 Threads) aus einer konkreten Gruppe von Threads nebenläufig ausführt und eine Single-Instruction-Multiple-Data („SIMD“-) Architektur implementiert, 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 einen gemeinsamen Satz von Anweisungen aus. In mindestens einer Ausführungsform implementiert der SM 3314 eine Single-Instruction-Multiple-Thread („SIMT“-) Architektur, bei der jeder Thread in einer Gruppe von Threads dazu konfiguriert ist, einen anderen Datensatz auf Grundlage dieses gemeinsamen Anweisungssatzes zu verarbeiten, wobei jedoch zugelassen wird, dass die einzelnen Threads in einer Gruppe von Threads während der Ausführung divergieren. 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 werden, wenn Threads innerhalb eines Warps divergieren. In einer anderen Ausführungsform werden ein Programmzähler, ein Aufrufstapel und ein Ausführungsstatus für jeden einzelnen Thread geführt, 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 Threads, die gemeinsame Anweisungen ausführen, können zur besseren Effizienz konvergiert und parallel ausgeführt werden. Mindestens eine Ausführungsform des SM 3314 wird hierin detaillierter beschrieben.

[0414] In mindestens einer Ausführungsform stellt die MMU 3318 eine Schnittstelle zwischen dem GPC 3300 und einer Speicherpartitionseinheit (z. B. der Partitionseinheit 3222 in **Fig. 32**) bereit, und die MMU 3318 sorgt für die Übersetzung virtueller Adressen in physikalische Adressen, den Speicherschutz und die Arbitrierung von Speicheranforderungen. In mindestens einer Ausführungsform stellt die MMU 3318 einen oder mehrere Übersetzungspuffer („TLBs“) zur Durchführung der Übersetzung von virtuellen Adressen in physische Adressen im Speicher bereit.

[0415] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. 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 3300 bereitgestellte Informationen vorherzusagen oder zu inferenzieren. In mindestens einer Ausführungsform wird das GPC 3300 verwendet, um Informationen auf Grundlage eines trainierten Modells des maschinellen Lernens (z. B. eines neuronalen Netzes) zu inferenzieren oder vorherzusagen, das durch einen anderen Prozessor oder ein anderes System oder durch das GPC 3300 trainiert wurde. In mindestens

einer Ausführungsform kann das GPC 3300 verwendet werden, um einen oder mehrere der hierin beschriebenen Anwendungsfälle für neuronale Netze durchzuführen.

[0416] Fig. 34 veranschaulicht eine Speicherpartitionseinheit 3400 einer Parallelverarbeitungseinheit („PPU“) gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform beinhaltet die Speicherpartitionseinheit 3400 ohne Einschränkung eine Einheit 3402 für Rasteroperationen („ROP“), einen Level-Zwei („L2“)-Cache 3404, eine Speicherschnittstelle 3406 und eine beliebige geeignete Kombination davon. In mindestens einer Ausführungsform ist die Speicherschnittstelle 3406 an Speicher gekoppelt. In mindestens einer Ausführungsform kann die Speicherschnittstelle 3406 32-, 64-, 134-, 1024-Bit-Datenbusse oder dergleichen für die Hochgeschwindigkeitsdatenübermittlung implementieren. In mindestens einer Ausführungsform beinhaltet die PPU U Speicherschnittstellen 3406, wobei U eine positive ganze Zahl ist, mit einer Speicherschnittstelle 3406 pro Paar Partitionseinheiten 3400, wobei jedes Paar Partitionseinheiten 3400 mit einer entsprechenden Speichervorrichtung verbunden ist. In mindestens einer Ausführungsform kann die PPU beispielsweise mit bis zu Y Speichervorrichtungen verbunden sein, z. B. mit Speicherstapeln mit hoher Bandbreite oder mit einem synchronen dynamischen Grafik-Double-Data-Rate(Version 5)-Direktzugriffsspeicher („GDDR5 SDRAM“).

[0417] In mindestens einer Ausführungsform implementiert die Speicherschnittstelle 3406 eine Speicherschnittstelle mit Speicher mit hoher Bandbreite der zweiten Generation („HBM2“) und Y ist gleich der Hälfte von U. In mindestens einer Ausführungsform befinden sich HBM2-Speicherstapel auf einem physischen Gehäuse mit einer 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 Speicher-Chips mit Y=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 dieser Speicher Single-Error-Correcting-Double-Error-Detecting („SECEDED“-) Fehlerkorrekturcode (Error Correction Code - „ECC“) zum Schützen von Daten. In mindestens einer Ausführungsform kann der ECC eine höhere Zuverlässigkeit für Rechenanwendungen bereitstellen, die für Datenkorruption empfindlich sind.

[0418] In mindestens einer Ausführungsform implementiert die PPU eine Speicherhierarchie mit mehreren Ebenen. In mindestens einer Ausführungsform unterstützt die Speicherpartitionseinheit 3400 einen einheitlichen Speicher, um einen einzelnen einheitlichen 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 durch eine PPU auf einen 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 3208 Adressübersetzungsdienste, die es der PPU ermöglichen, direkt auf die Seitentabellen einer CPU zuzugreifen, und einer PPU vollen Zugriff auf den CPU-Speicher bereitstellen.

[0419] In mindestens einer Ausführungsform übertragen Kopierengines Daten zwischen mehreren PPU oder zwischen PPU und CPUs. In mindestens einer Ausführungsform können Kopier-Engines Seitenfehler für Adressen erzeugen, die nicht in Seitentabellen abgebildet sind, und die Speicherpartitionseinheit 3400 bedient dann Seitenfehler, indem sie die Adressen in die Seitentabelle abbildet, woraufhin die Kopier-Engine eine Übermittlung durchführt. In mindestens einer Ausführungsform ist der Speicher für mehrere Kopier-Engine-Vorgänge 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 ein Kopierprozess transparent ist.

[0420] Daten aus dem Speicher 3204 aus **Fig. 32** oder einem anderen Systemspeicher werden gemäß mindestens einer Ausführungsform durch die Speicherpartitionseinheit 3400 abgerufen und in L2-Cache 3404 gespeichert, der sich auf dem Chip befindet und von verschiedenen GPCs gemeinsam genutzt wird. Jede Speicherpartitionseinheit 3400 beinhaltet in mindestens einer Ausführungsform ohne Einschränkung mindestens einen Abschnitt des L2-Cache, der mit einer entsprechenden Speichervorrichtung assoziiert ist. In mindestens einer Ausführungsform sind die Caches der unteren Ebenen in verschiedenen Einheiten innerhalb der GPCs implementiert. In mindestens einer Ausführungsform kann jeder der SMs 2714 aus **Fig. 33** einen Level-1 („L1“)-Cache implementieren, wobei dieser L1-Cache ein privater Speicher ist, der für einen konkreten SM 2714 dediziert ist, und Daten aus dem L2-Cache 3404 werden abgerufen und in jedem L1-Cache zum

Verarbeiten in funktionellen Einheiten der SMs 2714 gespeichert. In mindestens einer Ausführungsform ist der L2-Cache 3404 an die Speicherschnittstelle 3406 und die in **Fig. 32** gezeigte XBar 3220 gekoppelt.

[0421] führt die ROP-Einheit 3402 Grafikrasteroperationen durch, die mit der Pixelfarbe in Bezug stehen, wie etwa Farbkomprimierung, Pixel-Blending und mehr in mindestens einer Ausführungsform. In mindestens einer Ausführungsform implementiert die ROP-Einheit 3402 die Tiefenprüfung in Verbindung mit der Raster-Engine 3308, wobei sie eine Tiefe für eine Abtaststelle, die mit einem Pixelfragment assoziiert ist, von einer Culling-Engine der Raster-Engine 3308 empfängt. In mindestens einer Ausführungsform wird die Tiefe mit einer entsprechenden Tiefe in einem Tiefenpuffer für eine mit einem Fragment assoziierte Abtaststelle abgeglichen. Falls dieses Fragment die Tiefenprüfung für diese Abtaststelle besteht, aktualisiert die ROP-Einheit 3402 dann in mindestens einer Ausführungsform den Tiefenpuffer und überträgt ein Ergebnis dieser Tiefenprüfung an die Raster-Engine 3308. Es versteht sich, dass sich eine Anzahl der Partitionseinheiten 3400 von einer Anzahl der GPCs unterscheiden kann, und daher kann jede ROP-Einheit 3402 in mindestens einer Ausführungsform an jeden GPC gekoppelt sein. In mindestens einer Ausführungsform verfolgt die ROP-Einheit 3402 die von verschiedenen GPCs empfangenen Pakete und bestimmt, ob ein durch die ROP-Einheit 3402 erzeugtes Ergebnis zu der XBar 3220 durchgeroutet werden soll.

[0422] **Fig. 35** veranschaulicht einen Streaming-Multiprozessor („SM“) 3500 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist der SM 3500 der SM aus **Fig. 33**. In mindestens einer Ausführungsform beinhaltet der SM 3500 ohne Einschränkung einen Anweisungscache 3502, eine oder mehrere Planer-Einheiten 3504, eine Registerbank 3508, einen oder mehrere Verarbeitungskerne („Kerne“) 3510, eine oder mehrere Spezialfunktionseinheiten (special function units - „SFUs“) 3512, eine oder mehrere Lade-/Speichereinheiten (load/store units - „LSUs“) 3514, ein Zusammenschaltungsnetz 3516, einen gemeinsam genutzten Speicher/Level-Eins („L1“-)Cache 3518 und/oder eine beliebige geeignete Kombination davon.

[0423] In mindestens einer Ausführungsform verteilt eine Arbeitsverteilungseinheit Tasks zur Ausführung auf allgemeinen Verarbeitungsclustern („GPCs“) von Parallelverarbeitungseinheiten („PPUs“) und jeder Task ist einem bestimmten Datenverarbeitungsknoten („DPC“) innerhalb eines GPC zugewiesen, und wenn ein Task mit einem Shader-Programm verbunden ist, so wird dieser Task einem der SMs 3500 zugewiesen. In mindestens einer Ausführungsform empfängt die Planereinheit 3504 Tasks von einer Arbeitsverteilungseinheit und verwaltet die Anweisungsplanung für einen oder mehrere Thread-Blöcke, die dem SM 3500 zugeordnet sind. In mindestens einer Ausführungsform plant die Planereinheit 3504 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 Planereinheit 3504 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 3510, SFUs 3512 und LSUs 3514) zuteilt.

[0424] 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 stellen Anwendungen herkömmlicher Programmiermodelle ein einzelnes, einfaches Konstrukt zum Synchronisieren von kooperierenden Threads bereit: 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 auf Teilblock- (d. h. so klein wie ein einzelner Thread) und Mehrblock-Granularität zu definieren und kollektive Vorgänge, wie zum Beispiel Synchronisation, an Threads in einer kooperativen Gruppe durchzuführen. In mindestens einer Ausführungsform unterstützt dieses Programmiermodell eine saubere Komposition über Software-Grenzen hinweg, sodass sich Bibliotheken und Dienstprogrammfunktionen 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.

[0425] In mindestens einer Ausführungsform ist eine Zuteilungseinheit 3506 so konfiguriert, dass sie Anweisungen an eine oder mehrere funktionelle Einheiten überträgt, und die Planer-Einheit 3504 beinhaltet ohne Einschränkung zwei Zuteilungseinheiten 3506, die es ermöglichen, dass zwei unterschiedliche Anweisungen aus einem gemeinsamen Warp während jedes Taktzyklus zugeteilt werden. In mindestens einer Ausführungsform beinhaltet jede Planer-Einheit 3504 eine einzelne Zuteilungseinheit 3506 oder zusätzliche Zuteilungseinheiten 3506.

[0426] In mindestens einer Ausführungsform beinhaltet jeder SM 3500 in mindestens einer Ausführungsform ohne Einschränkung die Registerbank 3508, die einen Satz von Registern für funktionelle Einheiten des SM 3500 bereitstellt. In mindestens einer Ausführungsform ist die Registerbank 3508 auf jede funktionelle Einheit aufgeteilt, sodass jeder funktionellen Einheit ein dedizierter Abschnitt der Registerbank 3508 zugewiesen ist. In mindestens einer Ausführungsform ist die Registerbank 3508 auf unterschiedliche Warps aufgeteilt, die durch den SM 3500 ausgeführt werden, und die Registerbank 3508 stellt temporären Datenspeicher für Operanden bereit, die mit Datenpfaden von funktionellen Einheiten verbunden sind. In mindestens einer Ausführungsform umfasst jeder SM 3500 ohne Einschränkung eine Vielzahl von L Verarbeitungskernen 3510, wobei L eine positive ganze Zahl ist. In mindestens einer Ausführungsform beinhaltet der SM 3500 ohne Einschränkung eine große Anzahl (z. B. 128 oder mehr) von unterschiedlichen Verarbeitungskernen 3510. In mindestens einer Ausführungsform beinhaltet jeder Verarbeitungskern 3510 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 Ganzzahlen beinhaltet. In mindestens einer Ausführungsform implementieren die arithmetischen Fließkomma-Logikeinheiten den IEEE-754-2008-Standard für Fließkommaarithmetik. In mindestens einer Ausführungsform beinhalten die Verarbeitungskerne 3510 ohne Einschränkung 64 Gleitkommakeme mit einfacher Genauigkeit (32 Bit), 64 Integerkerne, 32 Gleitkommakeme mit doppelter Genauigkeit (64 Bit) und 8 Tensorkerne.

[0427] Tensorkerne sind gemäß mindestens einer Ausführungsform dazu konfiguriert, Matrixoperationen durchzuführen. In mindestens einer Ausführungsform sind ein oder mehrere Tensorkerne in den Verarbeitungskernen 3510 enthalten. In mindestens einer Ausführungsform sind die Tensorkerne dazu konfiguriert, Deep-Learning-Matrixarithmetik durchzuführen, wie z. B. Faltungsvorgänge für das Training und die Inferenzierung neuronaler Netzwerke. In mindestens einer Ausführungsform arbeitet jeder Tensorkern an einer 4x4-Matrix und führt eine Matrixmultiplikations- und -akkumulationsoperation $D = A \times B + C$ durch, wobei A, B, C und D 4x4-Matrizen sind.

[0428] In mindestens einer Ausführungsform sind die Matrixmultiplikationseingaben A und B 16-Bit-Gleitkommamatrizen und die Akkumulationsmatrizen C und D 16-Bit-Gleitkomma- oder 32-Bit-Gleitkommamatrizen. In mindestens einer Ausführungsform arbeiten die Tensorkerne an 16-Bit-Fließkommaeingabedaten mit 32-Bit-Fließkommaakkumulation. In mindestens einer Ausführungsform verwendet die 16-Bit-Fließkomma-multiplikation 64 Vorgänge und ergibt ein Produkt mit voller Genauigkeit, das dann unter Verwendung der 32-Bit-Fließkommaaddition mit anderen Zwischenprodukten für eine 4x4x4-Matrixmultiplikation akkumuliert wird. In mindestens einer Ausführungsform werden die Tensorkerne verwendet, um viel größere zweidimensionale oder höherdimensionale Matrixvorgänge 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 wird auf einem CUDA-Level auf einer Warp-Level-Schnittstelle von Matrizen der Größe 16x16 ausgegangen, die sich über alle 32 Threads des Warp erstrecken.

[0429] In mindestens einer Ausführungsform umfasst jeder SM 3500 ohne Einschränkung M SFUs 3512, die Spezialfunktionen durchführen (z. B. Attributbewertung, reziproke Quadratwurzel und dergleichen). In mindestens einer Ausführungsform beinhalten die SFUs 3512 ohne Einschränkung eine Baumtraversierungseinheit, die dazu konfiguriert ist, eine hierarchische Baumdatenstruktur zu traversieren. In mindestens einer Ausführungsform beinhalten die SFUs 3512 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 produzieren, die durch den SM 3500 ausgeführt werden. In mindestens einer Ausführungsform werden die Texturkarten in dem gemeinsam genutzten Speicher/L1-Cache 3518 gespeichert. In mindestens einer Ausführungsform implementieren die Textureinheiten Texturvorgänge, wie z. B. Filterungsvorgänge unter Verwendung von MIP-Karten (z. B.

Texturkarten mit variierenden Detaillevels), gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform beinhaltet jeder SM 3500 ohne Einschränkung zwei Textureinheiten.

[0430] Jeder SM 3500 umfasst in mindestens einer Ausführungsform ohne Einschränkung N LSUs 3514, die Lade- und Speicheroperationen zwischen dem gemeinsam genutzten Speicher/L1-Cache 3518 und der Registerbank 3508 implementieren. Das Zusammenschaltungsnetz 3516 verbindet in mindestens einer Ausführungsform jede funktionelle Einheit mit der Registerbank 3508 und die LSU 3514 mit der Registerbank 3508 und dem gemeinsam genutzten Speicher/L1-Cache 3518. In mindestens einer Ausführungsform ist das Zusammenschaltungsnetz 3516 eine Kreuzschiene, die dazu konfiguriert sein kann, beliebige funktionelle Einheiten mit beliebigen Registern in der Registerbank 3508 zu verbinden und LSUs 3514 mit der Registerbank 3508 und Speicherorten in dem gemeinsam genutzten Speicher/L1-Cache 3518 zu verbinden.

[0431] In mindestens einer Ausführungsform ist der gemeinsam genutzte Speicher/L1-Cache 3518 ein Array von chipinternem Speicher, das in mindestens einer Ausführungsform die Datenspeicherung und die Kommunikation zwischen dem SM 3500 und der Primitiv-Engine sowie zwischen Threads in dem SM 3500 ermöglicht. In mindestens einer Ausführungsform umfasst der gemeinsam genutzte Speicher/L1-Cache 3518 ohne Einschränkung eine Speicherkapazität von 128 KB und er befindet sich in einem Pfad von dem SM 3500 zu einer Partitionseinheit. In mindestens einer Ausführungsform wird der gemeinsam genutzte Speicher/L1-Cache 3518 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 3518, L2-Cache und Speicher Ergänzungsspeicher.

[0432] 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 kann so verwendet werden, wie etwa, falls der gemeinsam genutzte Speicher dazu konfiguriert ist, die Hälfte einer Kapazität zu verwenden, und Textur- und Lade-/Speicheroperationen die verbleibende Kapazität verwenden können. Die Integration in den gemeinsam genutzten Speicher/L1-Cache 3518 ermöglicht dem gemeinsam genutzten Speicher/L1-Cache 3518 gemäß mindestens einer Ausführungsform das Fungieren als Leitung mit hohem Durchsatz für Streaming-Daten, während simultan Zugriff mit hoher Bandbreite und niedriger Latenz auf häufig wiederverwendete Daten bereitgestellt wird. In mindestens einer Ausführungsform kann bei einer Konfiguration für Universalparallelberechnungen eine einfachere Konfiguration im Vergleich zur Grafikverarbeitung verwendet werden. In mindestens einer Ausführungsform werden Festfunktions-Grafikverarbeitungseinheiten umgangen, wodurch ein wesentlich einfacheres Programmiermodell entsteht. Bei einer Konfiguration für Universalparallelberechnungen ordnet eine Arbeitsverteilungseinheit in mindestens einer Ausführungsform Blöcke von Threads direkt den DPCs zu und verteilt diese. In mindestens einer Ausführungsform führen Threads in einem Block ein gemeinsames Programm aus, wobei eine eindeutige Thread-ID bei der Berechnung verwendet wird, um sicherzustellen, dass jeder Thread eindeutige Ergebnisse erzeugt, wobei der SM 3500 zum Ausführen des Programms und Durchführen von Berechnungen verwendet wird, der gemeinsam genutzte Speicher/L1-Cache 3518 zum Kommunizieren zwischen den Threads verwendet wird und die LSU 3514 zum Lesen des und Schreiben in den globalen Speicher durch den gemeinsam genutzten Speicher/L1-Cache 3518 und die Speicherpartitionseinheit verwendet wird. In mindestens einer Ausführungsform schreibt, bei der Konfiguration für Universalparallelberechnungen, der SM 3500 Befehle, welche die Planer-Einheit 3504 verwenden kann, um neue Arbeit in den DPCs zu starten.

[0433] In mindestens einer Ausführungsform ist die PPU in Folgendem beinhaltet: 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 tragbaren elektronischen Vorrichtung und mehr. In mindestens einer Ausführungsform ist eine PPU auf einem einzelnen Halbleitersubstrat verkörpert. In mindestens einer Ausführungsform ist eine 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.

[0434] In mindestens einer Ausführungsform kann die PPU auf einer Grafikkarte beinhaltet sein, die ein oder mehrere Speichervorrichtungen beinhaltet. In mindestens einer Ausführungsform kann diese Grafikkarte dazu konfiguriert sein, mit einem PCIe-Steckplatz auf einer Hauptplatine eines Desktop-Computers eine

Schnittstelle zu bilden. In mindestens einer Ausführungsform kann diese PPU eine integrierte Grafikverarbeitungseinheit („iGPU“) sein, die im Chipsatz einer Hauptplatine enthalten ist.

[0435] Die Ableitungs- und/oder Trainingslogik 115 wird verwendet, um Ableitungs- und/oder Trainingsvorgänge durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 115 werden nachfolgend in Verbindung mit **Fig. 1A** und/oder **Fig. 1B** bereitgestellt werden. 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 SM 3500 bereitgestellte Informationen vorherzusagen oder zu inferenzieren. In mindestens einer Ausführungsform wird der SM 3500 verwendet, um Informationen auf Grundlage eines trainierten Modells des maschinellen Lernens (z. B. eines neuronalen Netzes) zu inferenzieren oder vorherzusagen, das durch einen anderen Prozessor oder ein anderes System oder durch den SM 3500 trainiert wurde. In mindestens einer Ausführungsform kann der SM 3500 verwendet werden, um einen oder mehrere der hierin beschriebenen Anwendungsfälle für neuronale Netze durchzuführen.

[0436] Es werden Ausführungsformen in Bezug auf eine virtualisierte Rechenplattform für weiterentwickeltes Rechnen offenbart, wie etwa Bildableitung und Bildverarbeitung.

[0437] Mit Bezug auf **Fig. 36** ist **Fig. 36** ein beispielhaftes Datenflussdiagramm für einen Prozess 3600 zum Erzeugen und Einsetzen einer Verarbeitungs- und Inferenzpipeline gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann der Prozess 3600 eingesetzt werden, um eine Spielnamenkennungsanalyse und Inferenzen an Benutzerfeedbackdaten in einer oder mehreren Einrichtungen 3602, wie etwa einem Rechenzentrum, durchzuführen.

[0438] In mindestens einer Ausführungsform kann der Prozess 3600 innerhalb eines Trainingssystems 3604 und/oder eines Einsatzsystems 3606 ausgeführt werden. In mindestens einer Ausführungsform kann das Trainingssystem 3604 verwendet werden, um das Training, den Einsatz und die Implementierung von Modellen des maschinellen Lernens (z. B. neuronale Netzwerke, Objekterkennungsalgorithmen, Algorithmen für maschinelles Sehen usw.) zur Verwendung im Einsatzsystem 3606 durchzuführen. In mindestens einer Ausführungsform kann das Einsatzsystem 3606 so konfiguriert sein, dass es Verarbeitungs- und Rechenressourcen in einer verteilten Rechenumgebung ablädt, um die Infrastrukturanforderungen in der Einrichtung 3602 zu reduzieren. In mindestens einer Ausführungsform kann das Einsatzsystem 3606 eine gestraffte Plattform zum Auswählen, individuellen Anpassen und Implementieren virtueller Instrumente zur Verwendung mit Rechenvorrichtungen in der Einrichtung 3602 bereitstellen. In mindestens einer Ausführungsform können virtuelle Instrumente softwaredefinierte Anwendungen zum Durchführen einer oder mehrerer Verarbeitungsoperationen in Bezug auf Feedbackdaten beinhalten. In mindestens einer Ausführungsform können eine oder mehrere Anwendungen in einer Pipeline Dienste (z. B. Inferenz, Visualisierung, Berechnung, KI usw.) des Einsatzsystems 3606 während der Ausführung von Anwendungen verwenden oder aufrufen.

[0439] In mindestens einer Ausführungsform können einige der Anwendungen, die in weiterentwickelten Verarbeitungs- und Inferenzpipelines verwendet werden, Modelle des maschinellen Lernens oder andere KI verwenden, um einen oder mehrere Verarbeitungsschritte durchzuführen. In mindestens einer Ausführungsform können Modelle des maschinellen Lernens in der Einrichtung 3602 unter Verwendung von in der Einrichtung 3602 gespeicherten Feedbackdaten 3608 (wie beispielsweise Feedbackdaten) oder Feedbackdaten 3608 von einer anderen Einrichtung oder Einrichtungen oder einer Kombination davon trainiert werden. In mindestens einer Ausführungsform kann das Trainingssystem 3604 verwendet werden, um Anwendungen, Dienste und/oder andere Ressourcen zur Erzeugung von funktionierenden, einsatzfähigen Modellen des maschinellen Lernens für das Einsatzsystem 3606 bereitzustellen.

[0440] In mindestens einer Ausführungsform kann eine Modellregistrierungsdatenbank 3624 durch Objektspeicher unterstützt werden, der Versionierung und Objekt-Metadaten unterstützen kann. In mindestens einer Ausführungsform kann der Zugriff auf den Objektspeicher zum Beispiel durch eine mit Cloud-Speicher (z. B. einer Cloud 3726 aus **Fig. 37**) kompatible Anwendungsprogrammierschnittstelle (API) von innerhalb einer Cloud-Plattform erfolgen. In mindestens einer Ausführungsform können Modelle des maschinellen Lernens innerhalb der Modellregistrierungsdatenbank 3624 durch Entwickler oder Partner eines Systems, das mit einer API interagiert, hochgeladen, aufgelistet, modifiziert oder gelöscht werden. In mindestens einer Ausführungsform kann eine API Zugriff auf Verfahren bereitstellen, die es Benutzern mit zweckmäßigen Anmeldeinformationen ermöglichen, Modelle mit Anwendungen zu assoziieren, sodass Modelle als Teil der Ausführung von containerisierten Instanzierungen von Anwendungen ausgeführt werden können.

[0441] In mindestens einer Ausführungsform kann eine Trainingspipeline 3704 (**Fig. 37**) ein Szenario beinhalten, in dem die Einrichtung 3602 ihr eigenes Modell des maschinellen Lernens trainiert oder ein bestehendes Modell des maschinellen Lernens aufweist, das optimiert oder aktualisiert werden muss. In mindestens einer Ausführungsform können Feedbackdaten 3608 von verschiedenen Kanälen empfangen werden, wie beispielsweise Foren, Webformularen oder ähnlichen Kanälen. In mindestens einer Ausführungsform kann, sobald Bildgebungsdaten 3608 empfangen werden, die KI-gestützte Annotation 3610 verwendet werden, um bei der Erzeugung von Annotationen zu helfen, die den Bildgebungsdaten 3608 entsprechen und als Ground-Truth-Daten für ein Modell des maschinellen Lernens verwendet werden. In mindestens einer Ausführungsform kann die KI-gestützte Annotation 3610 ein oder mehrere Modelle des maschinellen Lernens (z. B. neuronale Faltungsnetze (CNNs)) beinhalten, die darauf trainiert werden können, Annotationen zu erzeugen, die bestimmten Typen von Feedbackdaten 3608 (z. B. von bestimmten Vorrichtungen) und/oder bestimmten Typen von Anomalien in den Feedbackdaten 3608 entsprechen. In mindestens einer Ausführungsform können die KI-gestützten Annotationen 3610 dann direkt verwendet oder unter Verwendung eines Annotationswerkzeugs eingestellt oder fein abgestimmt werden, um Ground-Truth-Daten zu erzeugen. In mindestens einer Ausführungsform können in einigen Beispielen gekennzeichnete Daten 3612 als Ground-Truth-Daten für das Training eines Modells des maschinellen Lernens verwendet werden. In mindestens einer Ausführungsform können die KI-gestützten Annotationen 3610, gekennzeichneten Daten 3612 oder eine Kombination davon als Ground-Truth-Daten für das Training eines Modells des maschinellen Lernens verwendet werden. In mindestens einer Ausführungsform kann ein trainiertes Modell des maschinellen Lernens als Ausgabemodell 3616 bezeichnet werden und durch das Einsatzsystem 3606 verwendet werden, wie hierin beschrieben.

[0442] In mindestens einer Ausführungsform kann die Trainingspipeline 3704 (**Fig. 37**) ein Szenario beinhalten, in dem die Einrichtung 3602 ein Modell des maschinellen Lernens zur Verwendung beim Durchführen eines oder mehrerer Verarbeitungs-Tasks für eine oder mehrere Anwendungen in dem Einsatzsystem 3606 benötigt, die Einrichtung 3602 aber möglicherweise derzeit kein derartiges Modell des maschinellen Lernens aufweist (oder möglicherweise kein Modell aufweist, das für derartige Zwecke optimiert, effizient oder effektiv ist). In mindestens einer Ausführungsform kann ein bestehendes Modell des maschinellen Lernens aus der Modellregistrierungsdatenbank 3624 ausgewählt werden. In mindestens einer Ausführungsform kann die Modellregistrierungsdatenbank 3624 Modelle des maschinellen Lernens beinhalten, die zum Durchführen einer Vielfalt von unterschiedlichen Inferenz-Tasks an Bildgebungsdaten trainiert sind. In mindestens einer Ausführungsform können die Modelle des maschinellen Lernens in der Modellregistrierungsdatenbank 3624 an Bildgebungsdaten von anderen Einrichtungen als der Einrichtung 3602 trainiert worden sein (z. B. Einrichtungen, die sich an einem anderen Ort befinden). In mindestens einer Ausführungsform können die Modelle des maschinellen Lernens an Bildgebungsdaten von einem Ort, zwei Orten oder einer beliebigen Anzahl von Orten trainiert worden sein. In mindestens einer Ausführungsform kann das Training an Bildgebungsdaten eines spezifischen Ortes an diesem Ort oder mindestens auf eine Weise stattfinden, welche die Vertraulichkeit der Bildgebungsdaten schützt oder die Übertragung von Bildgebungsdaten außerhalb des Gebäudes einschränkt (z. B. zur Einhaltung von HIPAA-Vorschriften, Datenschutzbestimmungen usw.). In mindestens einer Ausführungsform kann ein Modell des maschinellen Lernens, sobald es an einem Ort trainiert - oder teilweise trainiert - wurde, zur Modellregistrierungsdatenbank 3624 hinzugefügt werden. In mindestens einer Ausführungsform kann ein Modell des maschinellen Lernens dann in einer beliebigen Anzahl von anderen Einrichtungen neu trainiert oder aktualisiert werden, und ein neu trainiertes oder aktualisiertes Modell kann in der Modellregistrierungsdatenbank 3624 verfügbar gemacht werden. In mindestens einer Ausführungsform kann dann ein Modell des maschinellen Lernens aus der Modellregistrierungsdatenbank 3624 ausgewählt werden - und als Ausgabemodell 3616 bezeichnet werden - und in dem Einsatzsystem 3606 verwendet werden, um einen oder mehrere Verarbeitungs-Tasks für eine oder mehrere Anwendungen eines Einsatzsystems durchzuführen.

[0443] In mindestens einer Ausführungsform kann die Trainingspipeline 3704 (**Fig. 37**) ein Szenario beinhalten, in dem die Einrichtung 3602 ein Modell des maschinellen Lernens zur Verwendung beim Durchführen eines oder mehrerer Verarbeitungs-Tasks für eine oder mehrere Anwendungen in dem Einsatzsystem 3606 benötigt, die Einrichtung 3602 aber möglicherweise derzeit kein derartiges Modell des maschinellen Lernens aufweist (oder möglicherweise kein Modell aufweist, das für derartige Zwecke optimiert, effizient oder effektiv ist). In mindestens einer Ausführungsform könnte ein aus der Modellregistrierungsdatenbank 3624 ausgewähltes Modell des maschinellen Lernens aufgrund von Unterschieden bei den Populationen, genetischen Variationen, der Robustheit der zum Trainieren eines Modells des maschinellen Lernens verwendeten Trainingsdaten, der Verschiedenartigkeit der Anomalien der Trainingsdaten und/oder anderer Probleme mit den Trainingsdaten nicht für die in der Einrichtung 3602 erzeugten Feedbackdaten 3608 fein abgestimmt oder optimiert sein. In mindestens einer Ausführungsform kann die KI-gestützte Annotation 3610 verwendet wer-

den, um beim Erzeugen von Annotationen zu helfen, die den Feedbackdaten 3608 entsprechen, die als Ground-Truth-Daten für das erneute Trainieren oder Aktualisieren eines Modells des maschinellen Lernens verwendet werden sollen. In mindestens einer Ausführungsform können gekennzeichnete Daten 3612 als Ground-Truth-Daten für das Training eines Modells des maschinellen Lernens verwendet werden. In mindestens einer Ausführungsform kann das erneute Trainieren oder Aktualisieren eines Modells des maschinellen Lernens als Modelltraining 3614 bezeichnet werden. In mindestens einer Ausführungsform kann das Modelltraining 3614 - z. B. KI-gestützte Annotationen 3610, gekennzeichnete Daten 3612 oder eine Kombination davon - als Ground-Truth-Daten für das erneute Training oder Aktualisieren eines Modells des maschinellen Lernens verwendet werden.

[0444] In mindestens einer Ausführungsform kann das Einsatzsystem 3606 Software 3618, Dienste 3620, Hardware 3622 und/oder andere Komponenten, Merkmale und Funktionalitäten beinhalten. In mindestens einer Ausführungsform kann das Einsatzsystem 3606 einen Software-„Stapel“ beinhalten, sodass die Software 3618 auf den Diensten 3620 aufgebaut sein kann und die Dienste 3620 verwenden kann, um einige oder alle Verarbeitungs-Tasks durchzuführen, und die Dienste 3620 und die Software 3618 können auf der Hardware 3622 aufgebaut sein und die Hardware 3622 verwenden, um Verarbeitungs-, Speicher- und/oder andere Rechen-Tasks des Einsatzsystems 3606 auszuführen.

[0445] In mindestens einer Ausführungsform kann die Software 3618 eine beliebige Anzahl von unterschiedlichen Containern beinhalten, wobei jeder Container eine Instanziierung einer Anwendung ausführen kann. In mindestens einer Ausführungsform kann jede Anwendung eine oder mehrere Verarbeitungs-Tasks in einer weiterentwickelten Verarbeitungs- und Ableitungspipeline durchführen (z. B. Ableitung, Objekterkennung, Merkmalerkennung, Segmentierung, Bildverbesserung, Kalibrierung usw.). In mindestens einer Ausführungsform kann es für jeden Typ von Computervorrichtung eine beliebige Anzahl von Containern geben, die eine Datenverarbeitungs-Task in Bezug auf Feedbackdaten 3608 (oder andere Datentypen, wie die hierin beschriebenen) ausführen können. In mindestens einer Ausführungsform kann eine erweiterte Verarbeitungs- und Inferenz-Pipeline basierend auf einer Auswahl verschiedener Container definiert werden, die für die Verarbeitung von Feedbackdaten 3608 gewünscht oder erforderlich sind, zusätzlich zu Containern, die Bildungsdaten zur Verwendung durch jeden Container empfangen und konfigurieren, und/oder zur Verwendung durch die Einrichtung 3602 nach der Verarbeitung durch eine Pipeline (z. B. um Ausgaben zurück in einen verwendbaren Datentyp zum Speichern und Anzeigen in der Einrichtung 3602 umzuwandeln). In mindestens einer Ausführungsform kann eine Kombination von Containern innerhalb der Software 3618 (die z. B. eine Pipeline bilden) als virtuelles Instrument bezeichnet werden (wie hierin detaillierter beschrieben) und ein virtuelles Instrument kann Dienste 3620 und Hardware 3622 ausnutzen, um einige oder alle Verarbeitungs-Tasks von in Containern instanziierten Anwendungen auszuführen.

[0446] . . In mindestens einer Ausführungsform können die Daten als Teil der Datenverarbeitungspipeline einer Vorverarbeitung unterzogen werden, um die Daten für die Verarbeitung durch eine oder mehrere Anwendungen vorzubereiten. In mindestens einer Ausführungsform kann eine Nachverarbeitung an einer Ausgabe einer oder mehrerer Inferenz-Tasks oder anderen Verarbeitungs-Tasks einer Pipeline durchgeführt werden, um Ausgabedaten für eine nächste Anwendung vorzubereiten und/oder um Ausgabedaten für die Übertragung und/oder Verwendung durch einen Benutzer vorzubereiten (z. B. als Reaktion auf eine Inferenzanforderung). In mindestens einer Ausführungsform können Inferenz-Tasks von einem oder mehreren Modellen des maschinellen Lernens durchgeführt werden, z.B. von trainierten oder eingesetzten neuronalen Netzen, die Ausgabemodelle 3616 des Trainingssystems 3604 enthalten können.

[0447] In mindestens einer Ausführungsform können Aufgaben der Datenverarbeitungspipeline in einem oder mehreren Containern gekapselt sein, die jeweils eine separate, voll funktionsfähige Instanziierung einer Anwendung und einer virtualisierten Computerumgebung darstellen, die in der Lage ist, auf Modelle des maschinellen Lernens zu verweisen. In mindestens einer Ausführungsform können Container oder Anwendungen in einem privaten (z. B. zugriffsbeschränkten) Bereich einer Containerregistrierungsdatenbank (hierin detaillierter beschrieben) veröffentlicht werden und trainierte oder eingesetzte Modelle können in der Modellregistrierungsdatenbank 3624 gespeichert und mit einer oder mehreren Anwendungen assoziiert werden. In mindestens einer Ausführungsform können Abbilder von Anwendungen (z. B. Container-Abbilder) in einer Container-Registrierungsdatenbank verfügbar sein und sobald sie von einem Benutzer aus einer Container-Registrierungsdatenbank für den Einsatz in einer Pipeline ausgewählt wurden, kann ein Abbild verwendet werden, um einen Container für eine Instanziierung einer Anwendung zur Verwendung durch das System eines Benutzers zu erzeugen.

[0448] In mindestens einer Ausführungsform können Entwickler Anwendungen (z. B. als Container) zum Durchführen von Verarbeitung und/oder Inferenzierung an zugeführten Daten entwickeln, veröffentlichen und speichern. In mindestens einer Ausführungsform kann die Entwicklung, Veröffentlichung und/oder Speicherung unter Verwendung eines Softwareentwicklungskits (software development kit - SDK) durchgeführt werden, das einem System zugeordnet ist (z. B. um sicherzustellen, dass eine Anwendung und/oder ein entwickelter Container mit einem System konform oder kompatibel ist). In mindestens einer Ausführungsform kann eine Anwendung, die entwickelt wird, lokal (z. B. in einer ersten Einrichtung, an Daten von einer ersten Einrichtung) mit einem SDK geprüft werden, das mindestens einige der Dienste 3620 als System (z. B. System 3700 aus **Fig. 37**) unterstützen kann.. In mindestens einer Ausführungsform kann eine Anwendung, sobald sie durch das System 3700 validiert wurde (z. B. bezüglich Genauigkeit usw.), in einer Containerregistrierungsdatenbank zur Auswahl und/oder Implementation durch einen Benutzer (z. B. ein Krankenhaus, eine Klinik, ein Labor, einen Gesundheitsdienstleister usw.) verfügbar sein, um einen oder mehrere Verarbeitungstasks in Bezug auf Daten in einer Einrichtung (z. B. einer zweiten Einrichtung) eines Benutzers durchzuführen.

[0449] In mindestens einer Ausführungsform können Entwickler dann Anwendungen oder Container durch ein Netz für den Zugriff und die Verwendung durch Benutzer eines Systems (z. B. des Systems 3700 aus **Fig. 37**) teilen. In mindestens einer Ausführungsform können abgeschlossene und validierte Anwendungen oder Container in einer Containerregistrierungsdatenbank gespeichert werden und damit assoziierte Modelle des maschinellen Lernens in der Modellregistrierungsdatenbank 3624 gespeichert werden. In mindestens einer Ausführungsform kann eine anfordernde Entität - die eine Inferenz- oder Bildverarbeitungsanforderung bereitstellt - eine Containerregistrierungsdatenbank und/oder Modellregistrierungsdatenbank 3624 nach einer Anwendung, einem Container, einem Datensatz, einem Modell des maschinellen Lernens usw. durchsuchen, eine gewünschte Kombination von Elementen zur Aufnahme in die Datenverarbeitungspipeline auswählen und eine Verarbeitungsanforderung absenden. In mindestens einer Ausführungsform kann eine Anforderung Eingabedaten beinhalten, die zum Durchführen einer Anforderung notwendig sind, und/oder eine Auswahl von Anwendung(en) und/oder Modellen des maschinellen Lernens beinhalten, die bei der Verarbeitung einer Anforderung ausgeführt werden sollen. In mindestens einer Ausführungsform kann eine Anforderung dann an eine oder mehrere Komponenten des Einsatzsystems 3606 (z. B. eine Cloud) übergeben werden, um die Verarbeitung der Datenverarbeitungspipeline durchzuführen. In mindestens einer Ausführungsform kann die Verarbeitung durch das Einsatzsystem 3606 die Referenzierung ausgewählter Elemente (z. B. Anwendungen, Container, Modelle usw.) aus einer Containerregistrierungsdatenbank und/oder Modellregistrierungsdatenbank 3624 beinhalten. In mindestens einer Ausführungsform können, sobald Ergebnisse durch eine Pipeline erzeugt wurden, die Ergebnisse als Referenz an einen Benutzer zurückgegeben werden (z. B. zur Ansicht in einer Ansichtsanwendungssuite, die auf einer lokalen Arbeitsstation oder einem lokalen Endgerät in den Räumlichkeiten ausgeführt wird).

[0450] In mindestens einer Ausführungsform können zur Unterstützung der Verarbeitung oder Ausführung von Anwendungen oder Containern in Pipelines die Dienste 3620 ausgenutzt werden. In mindestens einer Ausführungsform können die Dienste 3620 Rechendienste, Dienste für künstliche Intelligenz (KI), Visualisierungsdienste und/oder andere Dienstypen beinhalten. In mindestens einer Ausführungsform können die Dienste 3620 Funktionen bereitstellen, die eine oder mehrere Anwendungen in der Software 3618 gemeinsam haben, sodass die Funktionen zu einem Dienst abstrahiert werden können, der durch Anwendungen aufgerufen oder ausgenutzt werden kann. In mindestens einer Ausführungsform kann die durch die Dienste 3620 bereitgestellte Funktionalität dynamisch und effizienter laufen, während sie ebenfalls gut skalierbar ist, indem es Anwendungen erlaubt wird, Daten parallel zu verarbeiten (z. B. unter Verwendung einer Parallelrechenplattform 3730 (**Fig. 37**)). Anstatt dass jede Anwendung, die eine gleiche Funktion nutzt, die durch einen Dienst 3620 angeboten wird, eine entsprechende Instanz des Dienstes 3620 aufweisen muss, kann der Dienst 3620 in mindestens einer Ausführungsform von verschiedenen Anwendungen gemeinsam genutzt werden. In mindestens einer Ausführungsform können die Dienste einen Inferenzserver oder eine Inferenz-Engine beinhalten, der/die als nicht einschränkende Beispiele für die Ausführung von Erkennungs- oder Segmentierungstasks verwendet werden kann. In mindestens einer Ausführungsform kann ein Modelltrainingsdienst enthalten sein, der die Fähigkeit bereitstellen kann, Modelle des maschinellen Lernens zu trainieren und/oder erneut zu trainieren...

[0451] In mindestens einer Ausführungsform, in der ein Dienst 3620 einen KI-Dienst (z. B. einen Inferenzdienst) beinhaltet, können ein oder mehrere Modelle des maschinellen Lernens, die mit einer Anwendung zur Anomaliedetektion (z. B. Tumoren, Wachstumsauffälligkeiten, Narbenbildung usw.) assoziiert sind, ausgeführt werden, indem ein Inferenzdienst (z. B. ein Inferenzserver) aufgerufen wird (z. B. als API-Aufruf), um ein Modell oder Modelle des maschinellen Lernens oder deren Verarbeitung als Teil der Anwendungsausführung

rung auszuführen. In mindestens einer Ausführungsform, in der eine andere Anwendung ein oder mehrere Modelle des maschinellen Lernens für Segmentierungs-Tasks enthält, kann eine Anwendung einen Inferenzdienst aufrufen, um Modelle des maschinellen Lernens zur Durchführung eines oder mehrerer mit Segmentierungs-Tasks assoziierten Verarbeitungsvorgängen auszuführen. In mindestens einer Ausführungsform kann die Software 3618, die eine weiterentwickelte Verarbeitungs- und Inferenzpipeline implementiert, gestrafft werden, da jede Anwendung einen gleichen Inferenzdienst zum Durchführen eines oder mehrerer Inferenz-Tasks aufrufen kann.

[0452] In mindestens einer Ausführungsform kann die Hardware 3622 GPUs, CPUs, Grafikkarten, ein KI-/Deep-Learning-System (z. B. einen KI-Supercomputer wie etwa das DGX-Supercomputersystem von NVIDIA), eine Cloud-Plattform oder eine Kombination davon beinhalten. In mindestens einer Ausführungsform können unterschiedliche Arten von Hardware 3622 verwendet werden, um eine effiziente, zweckmäßige Unterstützung für Software 3618 und Dienste 3620 im Einsatzsystem 3606 bereitzustellen. In mindestens einer Ausführungsform kann die Verwendung der GPU-Verarbeitung zur lokalen Verarbeitung (z. B. in der Einrichtung 3602), innerhalb eines KI-/Deep-Learning-Systems, in einem Cloud-System und/oder in anderen Verarbeitungskomponenten des Bereitstellungssystems 3606 implementiert werden, um Effizienz, Genauigkeit und Wirksamkeit der Erkennung von Spielnamen zu verbessern.

[0453] In mindestens einer Ausführungsform können die Software 3618 und/oder die Dienste 3620 als nicht einschränkende Beispiele für die GPU-Verarbeitung in Bezug auf Deep Learning, maschinelles Lernen und/oder Datenverarbeitung mit hoher Rechenleistung optimiert sein. In mindestens einer Ausführungsform kann mindestens ein Teil der Rechenumgebung des Einsatzsystems 3606 und/oder des Trainingssystems 3604 in einem Rechenzentrum auf einem oder mehreren Supercomputern oder Rechensystemen mit hoher Rechenleistung mit GPU-optimierter Software (z. B. Hardware- und Software-Kombination des DGX-Systems von NVIDIA) ausgeführt werden. In mindestens einer Ausführungsform kann die Hardware 3622 eine beliebige Anzahl von GPUs beinhalten, die zur parallelen Verarbeitung von Daten, wie hierin beschrieben, aufgerufen werden können. In mindestens einer Ausführungsform kann die Cloud-Plattform ferner GPU-Verarbeitung für die GPU-optimierte Ausführung von Deep-Learning-Tasks, Tasks des maschinellen Lernens oder anderen Rechen-Tasks beinhalten. In mindestens einer Ausführungsform kann die Cloud-Plattform (z. B. NGC von NVIDIA) unter Verwendung von KI-/Deep-Learning-Supercomputer(n) und/oder GPU-optimierter Software (z. B. wie auf DGX-Systemen von NVIDIA bereitgestellt) als Hardware-Abstraktions- und -Skalierungsplattform ausgeführt werden. In mindestens einer Ausführungsform kann die Cloud-Plattform ein Anwendungscontainer-Clustering-System oder Orchestrierungssystem (z. B. KUBERNETES) auf mehreren GPUs integrieren, um eine nahtlose Skalierung und Lastverteilung zu ermöglichen.

[0454] **Fig. 37** ist eine Systemdarstellung für ein beispielhaftes System 3700 zum Erzeugen und Einsetzen einer Einsatzpipeline gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann das System 3700 verwendet werden, um den Prozess 3600 von **Fig. 36** und/oder andere Prozesse einschließlich fortgeschrittener Verarbeitungs- und Inferenzpipelines zu implementieren. In mindestens einer Ausführungsform kann das System 3700 das Trainingssystem 3604 und das Einsatzsystem 3606 beinhalten. In mindestens einer Ausführungsform können das Trainingssystem 3604 und das Einsatzsystem 3606 unter Verwendung von Software 3618, Diensten 3620 und/oder Hardware 3622, wie hierin beschrieben, implementiert werden.

[0455] In mindestens einer Ausführungsform kann das System 3700 (z. B. das Trainingssystem 3604 und/oder das Einsatzsystem 3606) in einer Cloud-Computing-Umgebung implementiert sein (z. B. unter Verwendung der Cloud 3726). In mindestens einer Ausführungsform kann das System 3700 lokal in Bezug auf eine Einrichtung oder als Kombination aus sowohl Cloud- als auch lokalen Rechenressourcen implementiert werden. In mindestens einer Ausführungsform kann der Zugriff auf APIs in der Cloud 3726 durch beschlossene Sicherheitsmaßnahmen oder -protokolle auf autorisierte Benutzer beschränkt sein. In mindestens einer Ausführungsform kann ein Sicherheitsprotokoll Web-Token beinhalten, die von einem Authentifizierungsdienst (z. B. AuthN, AuthZ, Gluecon usw.) signiert sein können und eine entsprechende Autorisierung tragen können. In mindestens einer Ausführungsform können APIs von virtuellen Instrumenten (hierin beschrieben) oder anderen Instanziierungen des Systems 3700 auf einen Satz von öffentlichen IPs beschränkt sein, die für die Interaktion sicherheitsüberprüft oder autorisiert wurden.

[0456] In mindestens einer Ausführungsform können verschiedene Komponenten des Systems 3700 unter Verwendung beliebiger einer Vielfalt von unterschiedlichen Netztypen, einschließlich, aber nicht beschränkt auf, lokaler Netze (LANs) und/oder Weitverkehrsnetze (WANs), über drahtgebundene und/oder drahtlose Kommunikationsprotokolle miteinander kommunizieren. In mindestens einer Ausführungsform kann die Kom-

munikation zwischen Einrichtungen und Komponenten des Systems 3700 (z. B. zum Übertragen von Inferenzanforderungen, zum Empfangen von Ergebnissen von Inferenzanforderungen usw.) über einen Datenbus oder Datenbusse, drahtlose Datenprotokolle (Wi-Fi), drahtgebundene Datenprotokolle (z. B. Ethernet) usw. übermittelt werden.

[0457] In mindestens einer Ausführungsform kann das Trainingssystem 3604 Trainingspipelines 3704 ausführen, die denjenigen ähnlich sind, die hierin in Bezug auf **Fig. 36** beschrieben sind. In mindestens einer Ausführungsform, in der ein oder mehrere Modelle des maschinellen Lernens in Einsatzpipelines 3710 durch das Einsatzsystem 3606 verwendet werden sollen, können Trainingspipelines 3704 verwendet werden, um ein oder mehrere (z. B. vorab trainierte) Modelle zu trainieren und/oder eines oder mehrere der vorab trainierten Modelle 3706 zu implementieren (z. B. ohne eine Notwendigkeit eines Neutrainings oder einer Aktualisierung). In mindestens einer Ausführungsform können als Ergebnis der Trainingspipelines 3704 ein Ausgabe-Modell oder Ausgabemodelle 3616 erzeugt werden. In mindestens einer Ausführungsform können die Trainingspipelines 3704 eine beliebige Anzahl von Verarbeitungsschritten 37, KI-unterstützte Annotation 3610, Kennzeichnen oder Annotieren von Feedbackdaten 3608, um gekennzeichnete Daten 3612 zu generieren, Modellauswahl aus einem Modellregister, Modelltraining 3614, Training, erneutes Training oder Aktualisieren von Modellen und/oder andere Verarbeitungsschritte umfassen. In mindestens einer Ausführungsform können für unterschiedliche Modelle des maschinellen Lernens, die durch das Einsatzsystem 3606 verwendet werden, unterschiedliche Trainingspipelines 3704 verwendet werden. In mindestens einer Ausführungsform kann für ein erstes maschinelles Lernmodell eine Trainingspipeline 3704 verwendet werden, die einem ersten Beispiel ähnlich ist, das in Bezug auf **Fig. 36** beschrieben ist, für ein zweites maschinelles Lernmodell kann eine Trainingspipeline 3704 verwendet werden, die einem zweiten Beispiel ähnlich ist, das in Bezug auf **Fig. 36** beschrieben ist, und für ein drittes maschinelles Lernmodell kann eine Trainingspipeline 3704 verwendet werden, die einem dritten Beispiel ähnlich ist, das in Bezug auf **Fig. 36** beschrieben ist. In mindestens einer Ausführungsform kann eine beliebige Kombination von Tasks innerhalb des Trainingssystems 3604 verwendet werden, je nachdem, was für das jeweilige Modell des maschinellen Lernens erforderlich ist. In mindestens einer Ausführungsform können eines oder mehrere der Modelle des maschinellen Lernens bereits trainiert und einsatzbereit sein, sodass die Modelle des maschinellen Lernens möglicherweise keiner Verarbeitung durch das Trainingssystem 3604 unterzogen werden und durch das Einsatzsystem 3606 implementiert werden können.

[0458] In mindestens einer Ausführungsform können die Ausgabemodell(e) 3616 und/oder die vorab trainierte(n) Modell(e) 3706 in Abhängigkeit von der Implementation oder Ausführungsform beliebige Typen von Modellen des maschinellen Lernens beinhalten. In mindestens einer Ausführungsform und ohne Einschränkung können durch das System 3700 verwendete Modelle des maschinellen Lernens (ein) Modell(e) des maschinellen Lernens unter Verwendung von linearer Regression, logistischer Regression, Entscheidungsbäumen, Stützvektormaschinen (support vector machines - SVM), naivem Bayes-Klassifikator, k-nächstem Nachbarn (k-nearest neighbor - Knn), k-Means-Clustering, Random Forest, Dimensionsreduktionsalgorithmen, Gradientenverstärkungsalgorithmen, neuronalen Netzwerken (z. B. Autocodierern, faltenden, rekurrenten, Perzeptronen, langem Kurzzeitgedächtnis (Long/Short Term Memory - LSTM), Bi-LSTM, Hopfield, Boltzmann, Deep Belief, entfaltenden, erzeugenden gegnerischen, Flüssigkeitszustandsmaschine usw.) und/oder andere Arten von Modellen des maschinellen Lernens beinhalten.

[0459] In mindestens einer Ausführungsform können die Trainingspipelines 3704 eine KI-unterstützte Annotation umfassen. In mindestens einer Ausführungsform können gekennzeichnete Daten 3612 (z. B. herkömmliche Annotation) durch eine beliebige Anzahl von Techniken erzeugt werden. In mindestens einer Ausführungsform können Kennzeichnungen oder andere Annotationen in einem Zeichenprogramm (z. B. einem Annotationsprogramm), einem Programm zur computergestützten Konstruktion (Computer Aided Design - CAD), einem Kennzeichnungsprogramm, einer anderen Art von Programm, das zur Erzeugung von Annotationen oder Kennzeichnungen für Ground Truth geeignet ist, erzeugt werden und/oder in einigen Beispielen von Hand gezeichnet werden. In mindestens einer Ausführungsform können die Ground-Truth-Daten synthetisch produziert (z. B. aus Computermodellen oder Renderings), real produziert (z. B. aus Daten der realen Welt konstruiert und produziert), maschinell automatisiert (z. B. unter Verwendung von Merkmalsanalyse und Lernen, um Merkmale aus den Daten zu extrahieren und dann Markierungen zu erzeugen), von Menschen annotiert (z. B. definiert ein Markierer oder Annotationsexperte die Position der Kennzeichnungen) und/oder eine Kombination davon sein. In mindestens einer Ausführungsform können für jede Instanz der Feedbackdaten 3608 (oder eines anderen Datentyps, der durch Modelle des maschinellen Lernens verwendet wird) entsprechende Ground-Truth-Daten vorhanden sein, die durch das Trainingssystem 3604 erzeugt werden. In mindestens einer Ausführungsform kann die KI-gestützte Annotation als Teil der Einsatzpipelines 3710 durchgeführt werden; entweder zusätzlich zu oder anstelle der KI-gestützten Annotation, die in den

Trainingspipelines 3704 enthalten ist. In mindestens einer Ausführungsform kann das System 3700 eine mehrschichtige Plattform beinhalten, die eine Softwareschicht (z. B. Software 3618) von Diagnoseanwendungen (oder anderen Anwendungstypen) beinhalten kann, die eine oder mehrere medizinische Bildgebungs- und Diagnosefunktionen ausführen können..

[0460] In mindestens einer Ausführungsform kann eine Software-Schicht als sichere, verschlüsselte und/oder authentifizierte API implementiert sein, durch die Anwendungen oder Container aus einer externen Umgebung(en) (z. B. Einrichtung 3602) ausgewählt (z. B. aufgerufen) werden können. In mindestens einer Ausführungsform können Anwendungen dann einen oder mehrere Dienste 3620 aufrufen oder ausführen, um Rechen-, KI- oder Visualisierungs-Tasks auszuführen, die mit den entsprechenden Anwendungen assoziiert sind, und können die Software 3618 und/oder die Dienste 3620 die Hardware 3622 nutzen, um die Verarbeitungstasks auf effektive und effiziente Weise durchzuführen.

[0461] In mindestens einer Ausführungsform kann das Einsatzsystem 3606 Einsatzpipelines 3710 ausführen. In mindestens einer Ausführungsform können Einsatzpipelines 3710 eine beliebige Anzahl von Anwendungen beinhalten, die sequenziell, nicht sequenziell oder auf andere Weise auf Feedbackdaten (und/oder andere Datentypen) angewendet werden können - einschließlich KI-gestützter Annotation, wie vorstehend beschrieben. In mindestens einer Ausführungsform, wie hierin beschrieben, kann eine Einsatzpipeline 3710 für eine einzelne Vorrichtung als virtuelles Instrument für eine Vorrichtung bezeichnet werden. In mindestens einer Ausführungsform kann für eine einzelne Vorrichtung in Abhängigkeit von Informationen, die von Daten gewünscht sind, die durch eine Vorrichtung erzeugt wurden, mehr als eine Einsatzpipeline 3710 vorhanden sein.

[0462] In mindestens einer Ausführungsform können Anwendungen, die für Einsatzpipelines 3710 verfügbar sind, eine beliebige Anwendung beinhalten, die zum Ausführen von Verarbeitungsaufgaben bei Feedbackdaten oder anderen Daten von Vorrichtungen verwendet werden kann..... Da verschiedene Anwendungen in einigen Ausführungsformen Bildoperationen gemeinsam nutzen können, kann in mindestens einer Ausführungsform eine Datenerweiterungsbibliothek (z. B. als einer der Dienste 3620) verwendet werden, um diese Operationen zu beschleunigen. In mindestens einer Ausführungsform kann zur Vermeidung von Engpässen herkömmlicher Verarbeitungsansätze, die sich auf CPU-Verarbeitung stützen, die Parallelrechenplattform 3730 zur GPU-Beschleunigung dieser Verarbeitungstasks verwendet werden.

[0463] In mindestens einer Ausführungsform kann das Einsatzsystem 3606 eine Benutzerschnittstelle 3714 (z. B. eine grafische Benutzerschnittstelle, eine Webschnittstelle usw.) beinhalten, die verwendet werden kann, um Anwendungen für die Aufnahme in die Einsatzpipeline(s) 3710 auszuwählen, Anwendungen anzuordnen, Anwendungen oder Parameter oder Konstrukte davon zu modifizieren oder zu ändern, die Einsatzpipeline(s) 3710 während des Setups und/oder des Einsatzes zu verwenden und damit zu interagieren und/oder anderweitig mit dem Einsatzsystem 3606 zu interagieren. In mindestens einer Ausführungsform, obwohl nicht in Bezug auf das Trainingssystem 3604 veranschaulicht, kann die Benutzerschnittstelle 3714 (oder eine andere Benutzerschnittstelle) zur Auswahl von Modellen für die Verwendung im Einsatzsystem 3606, zur Auswahl von Modellen für das Training oder das erneute Training im Trainingssystem 3604 und/oder zur anderweitigen Interaktion mit dem Trainingssystem 3604 verwendet werden.

[0464] In mindestens einer Ausführungsform kann der Pipeline-Verwalter 3712 zusätzlich zu einem Anwendungsortchestrierungssystem 3728 verwendet werden, um die Interaktion zwischen Anwendungen oder Containern der Einsatzpipeline(s) 3710 und Diensten 3620 und/oder Hardware 3622 zu verwalten. In mindestens einer Ausführungsform kann der Pipelineverwalter 3712 dazu konfiguriert sein, Interaktionen von Anwendung zu Anwendung, von Anwendung zu Dienst 3620 und/oder von Anwendung oder Dienst zu Hardware 3622 zu ermöglichen. In mindestens einer Ausführungsform soll dies, obwohl es als in der Software 3618 beinhaltet veranschaulicht ist, nicht einschränkend sein, und in einigen Beispielen kann der Pipeline-Verwalter 3712 in den Diensten 3620 beinhaltet sein. In mindestens einer Ausführungsform kann das Anwendungsortchestrierungssystem 3728 (z. B. Kubernetes, DOCKER usw.) ein Container-Orchestrierungssystem beinhalten, das Anwendungen zu Containern als logische Einheiten zur Koordination, Verwaltung, Skalierung und zum Einsatz gruppieren kann. In mindestens einer Ausführungsform kann durch das Assoziieren von Anwendungen aus der oder den Einsatzpipeline(s) 3710 (z. B. einer Rekonstruktionsanwendung, einer Segmentierungsanwendung usw.) mit einzelnen Containern jede Anwendung in einer in sich geschlossenen Umgebung (z. B. auf Kernel-Level) ausgeführt werden, um die Geschwindigkeit und Effizienz zu erhöhen.

[0465] In mindestens einer Ausführungsform kann jede Anwendung und/oder jeder Container (oder ein Abbild davon) einzeln entwickelt, modifiziert und eingesetzt werden (z. B. kann ein erster Benutzer oder Ent-

wickler eine erste Anwendung entwickeln, modifizieren und einsetzen und ein zweiter Benutzer oder Entwickler eine zweite Anwendung separat von einem ersten Benutzer oder Entwickler entwickeln, modifizieren und einsetzen), was es ermöglichen kann, sich auf einen Task einer einzelnen Anwendung und/oder von einem Container oder Containern zu fokussieren und zu konzentrieren, ohne durch Tasks einer anderen Anwendung oder anderer Anwendungen oder eines anderen Containers oder anderer Container behindert zu werden. In mindestens einer Ausführungsform kann die Kommunikation und Kooperation zwischen unterschiedlichen Containern oder Anwendungen durch den Pipelineverwalter 3712 und das Anwendungsorchestrierungssystem 3728 unterstützt werden. In mindestens einer Ausführungsform kann, solange eine erwartete Eingabe und/oder Ausgabe jedes Containers oder jeder Anwendung einem System bekannt ist (z. B. auf Grundlage von Konstrukten von Anwendungen oder Containern), das Anwendungsorchestrierungssystem 3728 und/oder der Pipelineverwalter 3712 die Kommunikation unter und zwischen jedem der Anwendungen oder Container sowie die gemeinsame Nutzung von Ressourcen unter und zwischen diesen erleichtern. Da eine oder mehrere Anwendungen oder Container in der oder den Einsatzpipelines 3710 dieselben Dienste und Ressourcen gemeinsam nutzen können, kann das Anwendungsorchestrierungssystem 3728 in mindestens einer Ausführungsform die gemeinsame Nutzung von Diensten oder Ressourcen zwischen und unter verschiedenen Anwendungen oder Containern orchestrieren, verteilen und bestimmen. In mindestens einer Ausführungsform kann ein Planer verwendet werden, um den Ressourcenbedarf von Anwendungen oder Containern, die aktuelle Nutzung oder geplante Nutzung dieser Ressourcen und die Ressourcenverfügbarkeit zu verfolgen. In mindestens einer Ausführungsform kann ein Planer somit Ressourcen unterschiedlichen Anwendungen zuweisen und Ressourcen zwischen und unter Anwendungen im Hinblick auf Bedarf und Verfügbarkeit eines Systems verteilen. In einigen Beispielen kann ein Planer (und/oder eine andere Komponente des Anwendungsorchestrierungssystems 3728) die Ressourcenverfügbarkeit und -verteilung auf Grundlage von Beschränkungen bestimmen, die einem System auferlegt werden (z. B. Benutzerbeschränkungen), wie etwa Dienstgüte (QoS), Dringlichkeit des Bedarfs an Datenausgaben (um z. B. zu bestimmen, ob Echtzeitverarbeitung oder verzögerte Verarbeitung ausgeführt werden soll) usw.

[0466] In mindestens einer Ausführungsform können die Dienste 3620, die durch Anwendungen oder Container in dem Einsatzsystem 3606 ausgenutzt und gemeinsam genutzt werden, Rechendienste 3716, KI-Dienste 3718, Visualisierungsdienste 3720 und/oder andere Diensttypen beinhalten. In mindestens einer Ausführungsform können Anwendungen einen oder mehrere der Dienste 3620 aufrufen (z. B. ausführen), um Verarbeitungsoperationen für eine Anwendung durchzuführen. In mindestens einer Ausführungsform können die Rechendienste 3716 durch Anwendungen ausgenutzt werden, um Superrechnen oder andere Tasks des Hochleistungsrechnens (HPC) durchzuführen. In mindestens einer Ausführungsform können die der oder die Rechendienste 3716 ausgenutzt werden, um Parallelverarbeitung (z. B. unter Verwendung einer Parallelrechenplattform 3730) zum Verarbeiten von Daten durch eine oder mehrere Anwendungen und/oder einen oder mehrere Tasks einer einzelnen Anwendung im Wesentlichen simultan durchzuführen. In mindestens einer Ausführungsform kann die Parallelrechenplattform 3730 (z. B. CUDA von NVIDIA) Universaldatenverarbeitung auf GPUs (GPGPU) ermöglichen (z. B. GPUs 3722). In mindestens einer Ausführungsform kann eine Software-Schicht der Parallelrechenplattform 3730 Zugriff auf virtuelle Anweisungssätze und Parallelberechnungselemente von GPUs zur Ausführung von Rechenkerneln bereitstellen. In mindestens einer Ausführungsform kann die Parallelrechenplattform 3730 Speicher beinhalten und in einigen Ausführungsformen kann ein Speicher zwischen und unter mehreren Containern und/oder zwischen und unter unterschiedlichen Verarbeitungs-Tasks innerhalb eines einzelnen Containers gemeinsam genutzt werden. In mindestens einer Ausführungsform können Aufrufe für Interprozesskommunikation (inter-process communication - IPC) für mehrere Container und/oder für mehrere Prozesse innerhalb eines Containers erzeugt werden, um gleiche Daten aus einem gemeinsam genutzten Speichersegment der Parallelrechenplattform 3730 zu verwenden (z. B., wenn mehrere unterschiedliche Stufen einer Anwendung oder mehrere Anwendungen gleiche Informationen verarbeiten). In mindestens einer Ausführungsform können dieselben Daten an demselben Speicherort für eine beliebige Anzahl von Verarbeitungs-Tasks verwendet werden (z. B. zur selben Zeit, zu unterschiedlichen Zeiten usw.), anstatt eine Kopie der Daten zu erstellen und die Daten an unterschiedliche Orte im Speicher zu verschieben (z. B. ein Lese-/Schreibvorgang). In mindestens einer Ausführungsform können, da Daten verwendet werden, um neue Daten als ein Ergebnis der Verarbeitung zu erzeugen, diese Informationen über einen neuen Ort der Daten gespeichert und von verschiedenen Anwendungen gemeinsam genutzt werden. In mindestens einer Ausführungsform kann ein Ort der Daten und ein Ort von aktualisierten oder modifizierten Daten Teil einer Definition sein, wie Nutzdaten innerhalb von Containern zu verstehen sind.

[0467] In mindestens einer Ausführungsform können die KI-Dienste 3718 ausgenutzt werden, um Inferenzdienste zum Ausführen von einem Modell oder Modellen des maschinellen Lernens durchzuführen, die mit Anwendungen assoziiert sind (z. B. mit dem Durchführen eines oder mehrerer Verarbeitungs-Tasks einer

Anwendung beauftragt). In mindestens einer Ausführungsform können die KI-Dienste 3718 das KI-System 3724 ausnutzen, um ein Modell oder Modelle des maschinellen Lernens (z. B. neuronale Netze wie etwa CNNs) für Segmentierung, Rekonstruktion, Objektdetektion, Merkmalsdetektion, Klassifizierung und/oder andere Inferenz-Tasks auszuführen. In mindestens einer Ausführungsform können die Anwendungen der Einsatzpipeline(s) 3710 eines oder mehrere der Ausgabemodelle 3616 aus dem Trainingssystem 3604 und/oder andere Modelle der Anwendungen verwenden, um Inferenz an Bildgebungsdaten (z. B. DICOM-Daten, RIS-Daten, CIS-Daten, REST-konformen Daten, RPC-Daten, Rohdaten usw.) durchzuführen. In mindestens einer Ausführungsform können zwei oder mehr Beispiele für das Inferenzieren unter Verwendung des Anwendungsorchestrierungssystems 3728 (z. B. eines Planers) verfügbar sein. In mindestens einer Ausführungsform kann eine erste Kategorie einen Pfad mit hoher Priorität/geringer Latenz beinhalten, der höhere Dienstgütevereinbarungen erreichen kann, z. B. für die Durchführung von Ableitungen bei dringenden Anforderungen während eines Notfalls oder für einen Radiologen während der Diagnose. In mindestens einer Ausführungsform kann eine zweite Kategorie einen Pfad mit Standardpriorität beinhalten, der für Anfragen verwendet werden kann, die nicht dringend sind oder bei denen die Analyse zu einem späteren Zeitpunkt durchgeführt werden kann. In mindestens einer Ausführungsform kann das Anwendungsorchestrierungssystem 3728 Ressourcen (z. B. Dienste 3620 und/oder Hardware 3622) basierend auf Prioritätspfaden für unterschiedliche Ableitungs-Tasks der KI-Dienste 3718 verteilen.

[0468] In mindestens einer Ausführungsform kann der gemeinsam genutzte Datenspeicher mit den KI-Diensten 3718 innerhalb des Systems 3700 verbunden sein. In mindestens einer Ausführungsform kann der gemeinsam genutzte Datenspeicher als ein Cache (oder eine andere Speichervorrichtungsart) arbeiten und zur Verarbeitung von Inferenzanforderungen von Anwendungen verwendet werden. In mindestens einer Ausführungsform kann, wenn eine Ableitungsanforderung übermittelt wird, eine Anforderung von einem Satz von API-Instanzen des Einsatzsystems 3606 empfangen werden und können eine oder mehrere Instanzen ausgewählt werden (z. B. für beste Anpassung, für Lastverteilung usw.), um eine Anforderung zu verarbeiten. In mindestens einer Ausführungsform kann zum Verarbeiten einer Anforderung eine Anforderung in eine Datenbank eingegeben werden, ein Modell des maschinellen Lernens kann aus der Modellregistrierungsdatenbank 3624 lokalisiert werden, falls es sich nicht bereits in einem Cache befindet, ein Validierungsschritt kann sicherstellen, dass ein zweckmäßiges Modell des maschinellen Lernens in einen Cache (z. B. einen gemeinsam genutzten Datenspeicher) geladen wird, und/oder eine Kopie eines Modells kann in einem Cache gesichert werden. In mindestens einer Ausführungsform kann ein Planer (z. B. des Pipelineverwalters 3712) verwendet werden, um eine Anwendung zu starten, auf die in einer Anforderung verwiesen wird, falls eine Anwendung nicht bereits läuft oder falls nicht genügend Instanzen einer Anwendung vorhanden sind. In mindestens einer Ausführungsform kann ein Inferenzserver gestartet werden, wenn ein Inferenzserver zur Ausführung eines Modells nicht bereits gestartet ist. In mindestens einer Ausführungsform kann pro Modell eine beliebige Anzahl von Inferenzservern gestartet werden. In mindestens einer Ausführungsform können in einem Pull-Modell, in dem Inferenzserver geclustert sind, Modelle gecacht werden, wenn eine Lastverteilung vorteilhaft ist. In mindestens einer Ausführungsform können die Inferenzserver statisch in entsprechende, verteilte Server geladen werden.

[0469] In mindestens einer Ausführungsform kann das Inferenzieren unter Verwendung eines Inferenzservers durchgeführt werden, der in einem Container läuft. In mindestens einer Ausführungsform kann eine Instanz eines Inferenzservers mit einem Modell (und optional einer Vielzahl von Versionen eines Modells) assoziiert sein. In mindestens einer Ausführungsform kann, falls eine Instanz eines Inferenzservers nicht existiert, wenn eine Anforderung zum Durchführen einer Inferenz an einem Modell empfangen wird, eine neue Instanz geladen werden. In mindestens einer Ausführungsform kann beim Starten eines Inferenzservers ein Modell an einen Inferenzserver übergeben werden, sodass der gleiche Container zum Bedienen unterschiedlicher Modelle verwendet werden kann, solange der Inferenzserver als andere Instanz läuft.

[0470] In mindestens einer Ausführungsform kann während der Anwendungsausführung eine Inferenzanforderung für eine gegebene Anwendung empfangen werden und ein Container (der z. B. eine Instanz eines Inferenzservers hostet) geladen werden (falls nicht bereits geschehen) und eine Startprozedur aufgerufen werden. In mindestens einer Ausführungsform kann die Vorverarbeitungslogik in einem Container eingehende Daten laden, decodieren und/oder eine beliebige zusätzliche Vorverarbeitung an diesen durchführen (z. B. unter Verwendung von CPU(s) und/oder GPU(s)). In mindestens einer Ausführungsform kann ein Container, sobald die Daten für die Inferenz vorbereitet sind, die Inferenz nach Bedarf an den Daten durchführen. In mindestens einer Ausführungsform kann dies einen einzelnen Inferenzaufruf an einem Bild (z. B. ein Handröntgenbild) beinhalten oder eine Ableitung für Hunderte von Bildern (z. B. ein Brust-CT) erfordern. In mindestens einer Ausführungsform kann eine Anwendung die Ergebnisse vor dem Abschluss zusammenfassen, was ohne Einschränkung eine einzelne Konfidenzbewertung, eine Segmentierung auf Pixellevel, eine Seg-

mentierung auf Voxellevel, die Erzeugung einer Visualisierung oder die Erzeugung von Text zur Zusammenfassung von Befunden beinhalten kann. In mindestens einer Ausführungsform können unterschiedlichen Modellen oder Anwendungen unterschiedliche Prioritäten zugewiesen werden. Zum Beispiel können einige Modelle eine Echtzeitpriorität (TAT weniger als eine Minute) aufweisen, während andere eine niedrigere Priorität aufweisen können (z. B. TAT weniger als 10 Minuten). In mindestens einer Ausführungsform können die Modellausführungszeiten von der anfordernden Institution oder Entität gemessen werden und können die Zeit für die Durchquerung des Partnernetzes sowie die Ausführung auf einem Inferenzdienst beinhalten.

[0471] In mindestens einer Ausführungsform kann die Übermittlung von Anforderungen zwischen den Diensten 3620 und den Inferenzanwendungen hinter einem Software-Entwicklungskit (SDK) verborgen sein und der robuste Transport durch eine Warteschlange bereitgestellt sein. In mindestens einer Ausführungsform wird eine Anforderung über eine API für eine individuelle Anwendungs-/Tenant-ID-Kombination in eine Warteschlange gestellt und zieht ein SDK eine Anforderung aus einer Warteschlange und gibt eine Anforderung an eine Anwendung weiter. In mindestens einer Ausführungsform kann ein Name einer Warteschlange in einer Umgebung bereitgestellt sein, aus der ein SDK diesen aufnimmt. In mindestens einer Ausführungsform kann die asynchrone Kommunikation durch eine Warteschlange nützlich sein, da sie es einer beliebigen Instanz einer Anwendung ermöglicht, die Arbeit aufzunehmen, sobald sie verfügbar wird. In mindestens einer Ausführungsform können die Ergebnisse durch eine Warteschlange zurück übermittelt werden, um sicherzustellen, dass keine Daten verloren gehen. In mindestens einer Ausführungsform können Warteschlangen auch eine Fähigkeit bereitstellen, Arbeit zu segmentieren, da Arbeit mit höchster Priorität an eine Warteschlange gehen kann, mit der die meisten Instanzen einer Anwendung verbunden sind, während Arbeit mit niedrigster Priorität an eine Warteschlange gehen kann, mit der nur eine einzelne Instanz verbunden ist, die Tasks in der empfangenen Reihenfolge verarbeitet. In mindestens einer Ausführungsform kann eine Anwendung auf einer GPU-beschleunigten Instanz laufen, die in der Cloud 3726 erzeugt wurde, und ein Inferenzdienst kann das Inferenzieren auf einer GPU durchführen.

[0472] In mindestens einer Ausführungsform können die Visualisierungsdienste 3720 ausgenutzt werden, um Visualisierungen zur Ansicht von Ausgaben von Anwendungen und/oder Einsatzpipeline(s) 3710 zu erzeugen. In mindestens einer Ausführungsform können die GPUs 3722 durch die Visualisierungsdienste 3720 ausgenutzt werden, um Visualisierungen zu erzeugen. In mindestens einer Ausführungsform können Rendering-Effekte, wie etwa Strahlverfolgung, durch die Visualisierungsdienste 3720 implementiert werden, um Visualisierungen mit höherer Qualität zu erzeugen. In mindestens einer Ausführungsform können Visualisierungen ohne Einschränkung 2D-Bild-Renderings, 3D-Volumen-Renderings, 3D-Volumenrekonstruktionen, 2D-Tomographieslices, Virtual-Reality-Anzeigen, Augmented-Reality-Anzeigen usw. beinhalten. In mindestens einer Ausführungsform können virtualisierte Umgebungen verwendet werden, um eine virtuelle interaktive Anzeige oder Umgebung (z. B. eine virtuelle Umgebung) für die Interaktion durch Benutzer eines Systems (z. B. Ärzte, Pfleger, Radiologen usw.) zu erzeugen. In mindestens einer Ausführungsform können die Visualisierungsdienste 3720 einen internen Visualizer, Cinematics- und/oder andere Rendering- oder Bildverarbeitungsfähigkeiten oder -funktionen (z. B. Strahlverfolgung, Rasterung, interne Optik usw.) beinhalten.

[0473] In mindestens einer Ausführungsform kann die Hardware 3622 GPUs 3722, das KI-System 3724, die Cloud 3726 und/oder beliebige andere Hardware beinhalten, die zum Ausführen des Trainingssystems 3604 und/oder des Einsatzsystems 3606 verwendet wird. In mindestens einer Ausführungsform können die GPUs 3722 (z. B. TESLA- und/oder QUADRO-GPUs von NVIDIA) eine beliebige Anzahl von GPUs beinhalten, die zum Ausführen von Verarbeitungs-Tasks von Rechendiensten 3716, KI-Diensten 3718, Visualisierungsdiensten 3720, anderen Diensten und/oder beliebigen von Merkmalen oder Funktionen der Software 3618 verwendet werden können. In Bezug auf die KI-Dienste 3718 können die GPUs 3722 zum Beispiel zum Durchführen von Vorverarbeitung an Bildgebungsdaten (oder anderen Datentypen, die durch Modelle des maschinellen Lernens verwendet werden), Nachverarbeitung an Ausgaben der Modelle des maschinellen Lernens und/oder zum Durchführen von Inferenzieren (z. B. zum Ausführen von Modellen des maschinellen Lernens) verwendet werden. In mindestens einer Ausführungsform können die Cloud 3726, das KI-System 3724 und/oder andere Komponenten des Systems 3700 die GPUs 3722 verwenden. In mindestens einer Ausführungsform kann die Cloud 3726 eine GPU-optimierte Plattform für Deep-Learning-Tasks beinhalten. In mindestens einer Ausführungsform kann das KI-System 3724 GPUs verwenden und kann die Cloud 3726 - oder zumindest ein Abschnitt, der mit Deep Learning oder Ableitungen beauftragt ist - unter Verwendung eines oder mehrerer KI-Systeme 3724 ausgeführt werden. Obwohl es sich bei der Hardware 3622 der Veranschaulichung nach um diskrete Komponenten handelt, soll dies demnach nicht einschränkend sein und beliebige Komponenten der Hardware 3622 können mit beliebigen anderen Komponenten der Hardware 3622 kombiniert oder durch diese ausgenutzt werden.

[0474] In mindestens einer Ausführungsform kann das KI-System 3724 ein speziell entwickeltes Rechensystem (z. B. einen Supercomputer oder einen HPC) beinhalten, das für Inferenzieren, Deep Learning, maschinelles Lernen und/oder andere Tasks der künstlichen Intelligenz konfiguriert ist. In mindestens einer Ausführungsform kann das KI-System 3724 (z. B. DGX von NVIDIA) GPU-optimierte Software (z. B. einen Software-Stapel) beinhalten, die unter Verwendung einer Vielzahl von GPUs 3722 zusätzlich zu CPUs, RAM, Datenspeicher und/oder anderen Komponenten, Merkmalen oder Funktionen ausgeführt werden kann. In mindestens einer Ausführungsform können ein oder mehrere KI-Systeme 3724 in der Cloud 3726 (z. B. in einem Rechenzentrum) implementiert sein, um einige oder alle der KI-basierten Verarbeitungs-Tasks des Systems 3700 durchzuführen.

[0475] In mindestens einer Ausführungsform kann die Cloud 3726 eine GPU-beschleunigte Infrastruktur (z. B. NGC von NVIDIA) beinhalten, die eine GPU-optimierte Plattform zum Ausführen von Verarbeitungs-Tasks des Systems 3700 bereitstellen kann. In mindestens einer Ausführungsform kann die Cloud 3726 ein oder mehrere KI-Systeme 3724 zur Durchführung einer oder mehrerer KI-basierter Tasks des Systems 3700 beinhalten (z. B. als Hardware-Abstraktions- und -skalierungsplattform). In mindestens einer Ausführungsform kann die Cloud 3726 in das Anwendungsorchestrierungssystem 3728 integriert sein, das mehrere GPUs ausnutzt, um eine nahtlose Skalierung und Lastverteilung zwischen und unter den Anwendungen und Diensten 3620 zu ermöglichen. In mindestens einer Ausführungsform kann die Cloud 3726 damit beauftragt sein, mindestens einige der Dienste 3620 des Systems 3700 auszuführen, einschließlich der Rechendienste 3716, der KI-Dienste 3718 und/oder der Visualisierungsdienste 3720, wie hierin beschrieben. In mindestens einer Ausführungsform kann die Cloud 3726 Inferenz mit kleinen und großen Batches durchführen (z. B. unter Ausführung von TENSOR RT von NVIDIA), eine beschleunigte Parallelrechen-API und -Plattform 3730 bereitstellen (z. B. CUDA von NVIDIA), ein Anwendungsorchestrierungssystem 3728 ausführen (z. B. KUBERNETES), eine Grafik-Rendering-API und -Plattform bereitstellen (z. B. für Strahlverfolgung, 2D-Grafik, 3D-Grafik und/oder andere Rendering-Techniken, um Kinematik mit höherer Qualität zu erzeugen) und/oder sie kann andere Funktionen für das System 3700 bereitstellen.

[0476] In mindestens einer Ausführungsform kann die Cloud 3726 in dem Bestreben, die Vertraulichkeit von Patientendaten zu wahren (z. B., wenn Patientendaten oder -akten außerhalb der Räumlichkeiten verwendet werden sollen), eine Registrierungsdatenbank beinhalten - wie etwa eine Deep-Learning-Containerregistrierungsdatenbank. In mindestens einer Ausführungsform kann eine Registrierungsdatenbank Container für Instanzierungen von Anwendungen speichern, die Vorverarbeitungs-, Nachverarbeitungs- oder andere Verarbeitungs-Tasks an Patientendaten durchführen können. In mindestens einer Ausführungsform kann die Cloud 3726 Daten empfangen, die Patientendaten sowie Sensordaten in Containern beinhalten, die angeforderte Verarbeitung nur für die Sensordaten in diesen Containern durchführen und dann eine resultierende Ausgabe und/oder Visualisierungen an zweckmäßige Parteien und/oder Vorrichtungen weiterleiten (z. B. medizinische Vorrichtungen in den Räumlichkeiten, die zur Visualisierung oder Diagnose verwendet werden), ohne dass Patientendaten extrahiert oder gespeichert müssen oder anderweitig darauf zugegriffen werden muss. In mindestens einer Ausführungsform wird die Vertraulichkeit der Patientendaten in Übereinstimmung mit HIPAA- und/oder anderen Datenvorschriften gewahrt.

[0477] Mindestens eine Ausführungsform der Offenbarung kann im Hinblick auf die folgenden Klauseln beschrieben werden:

[0478] In Klausel 1 Prozessor umfassend: einen oder mehrere Schaltkreise zur Verwendung eines oder mehrerer neuronaler Netze, um Betrug durch einen oder mehrere Benutzer eines Computerspiels zumindest teilweise auf der Grundlage eines oder mehrerer durch das Computerspiel erzeugter Bilder zu erkennen.

[0479] In Klausel 2 Prozessor nach Klausel 1, wobei die eine oder mehreren Schaltungen ferner dazu dienen: das eine oder die mehreren Bilder in einem Puffer zu speichern, wobei das gespeicherte eine oder die mehreren Bilder als Eingabe für das eine oder die mehreren neuronalen Netze bereitgestellt und auf einer Anzeigeeinheit wiedergegeben werden sollen.

[0480] In Klausel 3 Prozessor nach Klausel 2, wobei der eine oder die mehreren Schaltkreise weiterhin dazu dienen: einen Bericht zu erzeugen, der die Erkennung von Betrug anzeigt; und den Bericht an einen Spieler-Server zu übermitteln.

[0481] In Klausel 4 Prozessor nach Klausel 3, wobei der eine oder die mehreren Schaltkreise weiterhin dazu dienen: unter Verwendung des einen oder der mehreren neuronalen Netze eine Zuverlässigkeitsstufe zu

erzeugen, die die Zuverlässigkeit charakterisiert, dass das eine oder die mehreren Bilder Betrugsinformationen enthalten.

[0482] In Klausel 5 Prozessor nach Klausel 4, wobei der Bericht an den Spielservers als Reaktion auf die Feststellung, dass die Zuverlässigkeitsstufe bei oder über einem Schwellenwert liegt, übermittelt wird.

[0483] In Klausel 6 Prozessor nach Klausel 3, wobei der eine oder die mehreren Schaltkreise weiterhin dazu dienen: vom Spielservers aktualisierte Parameter für das eine oder die mehreren neuronalen Netzwerke zu empfangen, wobei die aktualisierten Parameter auf der Grundlage eines Satzes von Neutrainingbildern erzeugt werden.

[0484] In Klausel 7 Prozessor nach Klausel 1, wobei der eine oder die mehreren Schaltkreise ein Zertifizierungssignal für einen Spielservers erzeugen sollen, wobei das Zertifizierungssignal dem Spielservers bescheinigen soll, dass der eine oder die mehreren Schaltkreise in der Lage sind, Betrug im Zusammenhang mit dem Computerspiel zu erkennen.

[0485] In Klausel 8 Prozessor, der Folgendes umfasst: eine oder mehrere Schaltungen zur Durchführung des Trainings eines oder mehrerer neuronaler Netze, um Betrug durch einen oder mehrere Benutzer eines Computerspiels zu erkennen, wobei das Training zumindest teilweise auf einem oder mehreren Betrugsbildern basiert.

[0486] In Klausel 9 Prozessor nach Klausel 8, wobei das eine oder die mehreren Betrugsbilder unter Verwendung einer mit dem Computerspiel verbundenen Betrugssoftware erzeugt werden.

[0487] In Klausel 10 Prozessor nach Klausel 8, wobei das Training des einen oder der mehreren neuronalen Netze zumindest teilweise auf einem oder mehreren Nichtbetrugsbildern basiert, wobei jedes der einen oder mehreren Betrugsbilder Betrugsinformationen enthält und jedes der einen oder mehreren Nichtbetrugsbilder keine Betrugsinformationen enthält.

[0488] In Klausel 11 Prozessor nach Klausel 10, wobei mindestens eine Teilmenge des einen oder der mehreren Betrugsbilder Bilder umfasst, die mit Informationen aus Nichtbetrugsbildern angereichert sind, die von einer mit dem Computerspiel verbundenen Spielsoftware erzeugt werden.

[0489] In Klausel 12 Prozessor nach Klausel 11, wobei jedes der Teilmenge des einen oder der mehreren Betrugsbilder einen Teil umfasst, der durch einen Teil eines Nichtbetrugsbildes ersetzt wird.

[0490] In Klausel 13 Prozessor nach Klausel 10, wobei mindestens eine Teilmenge des einen oder der mehreren Nichtbetrugsbilder Nichtbetrugsbilder umfasst, die mit Informationen aus Betrugsbildern angereichert sind, die von einer mit dem Computerspiel verbundenen Spielsoftware erzeugt wurden.

[0491] In Klausel 14 Prozessor nach Klausel 13, wobei jedes der Teilmenge des einen oder der mehreren Nichtbetrugsbilder einen Teil umfasst, der durch einen Teil eines Betrugsbildes ersetzt wird, das eine Nichtbetrugsinformation enthält.

[0492] In Klausel 15 Prozessor nach Klausel 8, wobei die eine oder mehreren Schaltungen ferner dazu dienen, ein Training des einen oder der mehreren neuronalen Netze gegen gegnerische Angriffe durchzuführen.

[0493] In Klausel 16 Prozessor nach Klausel 15, wobei das Training des einen oder der mehreren neuronalen Netze gegen gegnerische Angriffe zumindest teilweise auf einer Teilmenge des einen oder der mehreren Betrugsbilder basiert, die mit gegnerischen Störungen modifiziert wurden.

[0494] In Klausel 17 System umfassend: einen oder mehrere Prozessoren zur Verwendung eines oder mehrerer neuronaler Netze, um Betrug durch einen oder mehrere Benutzer eines Computerspiels zumindest teilweise auf der Grundlage eines oder mehrerer durch das Computerspiel erzeugter Bilder zu erkennen; und einen oder mehrere Speicher zum Speichern von Parametern, die mit dem einen oder den mehreren neuronalen Netzen assoziiert sind.

[0495] In Klausel 18 System nach Klausel 17, wobei die eine oder mehreren Prozessoren ferner dazu dienen: das eine oder die mehreren Bilder in einem Puffer zu speichern, wobei das gespeicherte eine oder die

mehreren Bilder als Eingabe für das eine oder die mehreren neuronalen Netze bereitgestellt und auf einer Anzeigeeinheit wiedergegeben werden sollen.

[0496] In Klausel 19 System nach Klausel 18, wobei der eine oder die mehreren Prozessoren weiterhin dazu dienen: einen Bericht zu erzeugen, der die Erkennung von Betrug anzeigt; und den Bericht an einen Spielserver zu übermitteln.

[0497] In Klausel 20 System nach Klausel 19, wobei zur Übermittlung des Berichts an den Spielserver der eine oder die mehreren Prozessoren: unter Verwendung des einen oder der mehreren neuronalen Netze eine Zuverlässigkeitsstufe erzeugen, die die Zuverlässigkeit kennzeichnet, dass das eine oder die mehreren Bilder eine Betrugsinformation enthalten; und feststellen, dass die Zuverlässigkeitsstufe bei oder über einem Schwellenwert liegt.

[0498] In Klausel 21 System umfassend: einen oder mehrere Prozessoren zur Durchführung des Trainings eines oder mehrerer neuronaler Netze zur Erkennung von Betrug durch einen oder mehrere Benutzer eines Computerspiels, zumindest teilweise auf der Grundlage von einem oder mehreren Betrugsbildern, die von einer mit dem Computerspiel verbundenen Betrugssoftware erzeugt werden; und einen oder mehrere Speicher zur Speicherung von Parametern, die mit dem einen oder den mehreren neuronalen Netzen verbunden sind.

[0499] In Klausel 22 Prozessor nach Klausel 21, wobei das eine oder die mehreren Betrugsbilder unter Verwendung einer mit dem Computerspiel verbundenen Betrugssoftware erzeugt werden, und das Training des einen oder der mehreren neuronalen Netze zumindest teilweise auf einem oder mehreren Nichtbetrugsbildern basiert, wobei jedes der einen oder mehreren Betrugsbilder Betrugsinformationen enthält und jedes der einen oder mehreren Nichtbetrugsbilder keine Betrugsinformationen enthält.

[0500] In Klausel 23 System nach Klausel 22, wobei mindestens eine Teilmenge des einen oder der mehreren Betrugsbilder Bilder umfasst, die mit Informationen aus Nichtbetrugsbildern angereichert sind, die von einer mit dem Computerspiel verbundenen Spielsoftware erzeugt werden.

[0501] In Klausel 24 System nach Klausel 23, wobei jedes der Teilmenge des einen oder der mehreren Betrugsbilder einen Teil umfasst, der durch einen Teil eines Nichtbetrugsbildes ersetzt wird.

[0502] In Klausel 25 System nach Klausel 22, wobei mindestens eine Teilmenge des einen oder der mehreren Nichtbetrugsbilder Nichtbetrugsbilder umfasst, die mit Informationen aus Betrugsbildern angereichert sind, die von einer mit dem Computerspiel verbundenen Spielsoftware erzeugt wurden.

[0503] In Klausel 26 System nach Klausel 25, wobei jedes der Teilmenge des einen oder der mehreren Nichtbetrugsbilder einen Teil umfasst, der durch einen Teil eines Betrugsbildes ersetzt wird, das eine Nichtbetrugsinformation enthält.

[0504] In Klausel 27 System nach Klausel 21, wobei die eine oder mehreren Prozessoren ferner dazu dienen, ein Training des einen oder der mehreren neuronalen Netze gegen gegnerische Angriffe durchzuführen.

[0505] In Klausel 28 Verfahren, das Folgendes umfasst: Empfangen einer Darstellung einer mit einem Computerspiel verbundenen Grafik von einer Rechenvorrichtung; Erzeugen eines oder mehrerer Bilder durch einen oder mehrere Schaltkreise auf der Grundlage der empfangenen Darstellung; und Verarbeiten des einen oder der mehreren Bilder durch den einen oder die mehreren Schaltkreise und unter Verwendung eines oder mehrerer neuronaler Netze, um Betrug durch einen oder mehrere Benutzer des Computerspiels zu erkennen.

[0506] In Klausel 29 Verfahren nach Klausel 28, ferner umfassend: Erzeugen eines Berichts, der die Erkennung von Betrug anzeigt; und Übermitteln des Berichts an einen Spielserver.

[0507] In Klausel 30 Verfahren nach Klausel 28, wobei die Übermittlung des Berichts an einen Spielserver erfolgt in Reaktion auf: Erzeugen, durch den einen oder die mehreren Schaltkreise und unter Verwendung des einen oder der mehreren neuronalen Netze, einer Zuverlässigkeitsstufe, die die Zuverlässigkeit kennzeichnet, dass das eine oder die mehreren Bilder eine Betrugsinformation enthalten; und Bestimmen, dass die Zuverlässigkeitsstufe bei oder über einem Schwellenwert liegt.

[0508] In Klausel 31 Verfahren von Klausel 30 ferner umfassend: Empfangen aktualisierter Parameter für das eine oder die mehreren neuronalen Netze, wobei die aktualisierten Parameter auf der Grundlage von Neutrainingbildern erzeugt werden.

[0509] In Klausel 32 Verfahren nach Klausel 28, wobei das eine oder die mehreren neuronalen Netze trainiert werden unter Verwendung von: einem oder mehreren Betrugsbildern, die unter Verwendung einer mit dem Computerspiel verbundenen Betrugssoftware erzeugt werden; und einem oder mehreren Nichtbetrugsbildern, die frei von Betrugsinformationen sind.

[0510] In Klausel 33 Verfahren nach Klausel 32, wobei das eine oder die mehreren neuronalen Netze weiter gegen gegnerische Angriffe trainiert werden, wobei das Training gegen gegnerische Angriffe zumindest teilweise auf einer Teilmenge des einen oder der mehreren Betrugsbilder basiert, die mit gegnerischen Störungen modifiziert sind.

[0511] 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 Zentraleinheit („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.

[0512] In mindestens einer Ausführungsform, unter erneuter Bezugnahme auf **Fig. 13**, sind Computerprogramme in Form von maschinenlesbarem ausführbarem Code oder Computersteuerlogik-Algorithmen in dem Hauptspeicher 1304 und/oder Sekundärspeicher gespeichert. Falls sie durch einen oder mehrere Prozessoren ausgeführt werden, ermöglichen es Computerprogramme dem System 1300, verschiedene Funktionen gemäß mindestens einer Ausführungsform durchzuführen. In mindestens einer Ausführungsform sind Arbeitsspeicher 1304, Datenspeiche und/oder beliebiger anderer Speicher 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 z. B. 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 1302, des Parallelverarbeitungssystems 1312, einer integrierten Schaltung, die mindestens zu einem Abschnitt der Fähigkeiten sowohl der CPU 1302 als auch des Parallelverarbeitungssystems 1312 in der Lage ist, eines Chipsatzes (z. B. einer Gruppe integrierter Schaltungen, die so ausgestaltet ist, dass sie als eine Einheit zur Durchführung zugehöriger Funktionen arbeitet und verkauft wird, usw.) und einer beliebigen geeigneten Kombination integrierter Schaltungen implementiert.

[0513] In mindestens einer Ausführungsform werden Architektur und/oder Funktion verschiedener vorheriger Figuren im Kontext eines allgemeinen Computersystems, eines Leiterplattensystems, eines Spielkonsolensystems für Unterhaltungszwecke, eines anwendungsspezifischen Systems und mehr umgesetzt. In mindestens einer Ausführungsform kann das Computersystem 1300 die Form eines Desktop-Computers, eines Laptop-Computers, eines Tablet-Computers, von Servern, Supercomputern, eines Smartphones (z. B. einer drahtlosen tragbaren Vorrichtung), eines persönlichen digitalen Assistenten („PDA“), einer Digitalkamera, eines Fahrzeugs, einer am Kopf befestigten Anzeige, einer elektronischen tragbaren Vorrichtung, einer Mobiltelefonvorrichtung, eines Fernsehgeräts, einer Arbeitsstation, von Spielekonsolen, eines eingebetteten Systems und/oder einer beliebigen anderen Art von Logik annehmen.

[0514] In mindestens einer Ausführungsform beinhaltet das Parallelverarbeitungssystem 1312 ohne Einschränkung eine Vielzahl von Parallelverarbeitungseinheiten („PPUs“) 1314 und damit assoziierte Speicher 1316. In mindestens einer Ausführungsform sind die PPUs 1314 mit einem Hostprozessor oder anderen Peripherievorrichtungen über eine Verbindung 1318 und einen Switch 1320 oder einen Multiplexer verbunden. In mindestens einer Ausführungsform verteilt das Parallelverarbeitungssystem 712 Berechnungsaufgaben über die PPUs 1314, die parallelisierbar sein können - zum Beispiel als Teil der Verteilung von Berechnungsaufgaben über mehrere Thread-Blöcke von Grafikverarbeitungseinheiten („GPUs“). In mindestens einer Ausführungsform wird der Speicher von einigen oder allen PPUs 1314 gemeinsam genutzt und ist für diese zugänglich (z. B. für den Lese- und/oder Schreibzugriff), obwohl ein derartiger gemeinsam genutzter Speicher Leistungseinbußen in Bezug auf die Verwendung von lokalem Speicher und in einer PPU 1314 residenten

Registern nach sich ziehen kann. In mindestens einer Ausführungsform wird der Betrieb der PPU's 1314 durch Verwendung eines Befehls wie `_syncthreads()` synchronisiert, wobei alle Threads in einem Block (z. B. über mehrere PPU's 1314 ausgeführt) einen bestimmten Punkt der Codeausführung erreichen müssen, bevor sie fortfahren.

[0515] Andere Variationen liegen innerhalb des Wesens der vorliegenden Offenbarung. Somit können zwar bezüglich der offenbarten Techniken diverse Modifikationen und alternative Konstruktionen vorgenommen werden, doch sind bestimmte veranschaulichte Ausführungsformen in den Zeichnungen gezeigt und wurden vorangehend ausführlich beschrieben. Allerdings versteht es sich, dass nicht die Absicht verfolgt wird, die Offenbarung auf die spezifische(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.

[0516] Die Verwendung der Ausdrücke „ein“ und „eine“ und „der/die/das“ und ähnlicher Referenten im Kontext des Beschreibens offenbarer Ausführungsformen (insbesondere im Kontext der folgenden Ansprüche) ist so auszulegen, dass sie sowohl den Singular als auch den Plural abdeckt, sofern hierin nichts anderes angegeben ist oder der Kontext dem eindeutig widerspricht, und nicht als Definition eines Ausdrucks. Die Begriffe „umfassend“, „aufweisend“, „beinhaltend“ und „enthaltend“ sind als offene Begriffe auszulegen (d. h. „einschließlich, aber nicht begrenzt auf“), sofern nicht anderweitig angegeben. Wenn es unmodifiziert vorliegt und sich auf physische Verbindungen bezieht, ist „verbunden“ als teilweise oder vollständig ineinander enthalten, aneinander angebracht oder aneinander angefügt auszulegen, auch wenn ein Element dazwischenliegt. Die Nennung von Wertebereichen hierin soll lediglich als schnelles Verfahren des einzelnen Bezugnehmens auf jeden separaten Wert dienen, der in den Bereich fällt, es sei denn, hierin ist etwas anderes angegeben, und jeder separate Wert ist in die Beschreibung eingeschlossen, als ob er einzeln hierin wiedergegeben wäre. In mindestens einer Ausführungsform ist die Verwendung des Ausdrucks „Satz“ (z. B. „ein Satz von Gegenständen“) oder „Teilmenge“ als eine nicht leere Sammlung auszulegen, die ein oder mehrere Elemente umfasst, es sei denn, es ist etwas anderes angemerkt oder der Kontext widerspricht dem. 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.

[0517] Sofern nicht spezifisch etwas anderes genannt ist oder der Kontext dem eindeutig widerspricht, sind verbindende Ausdrucksweisen, 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, B und C“ auf einen beliebigen der folgenden Sätze: {A}, {B}, {C}, {A, B}, {A, C}, {B, C}, {A, B, C}. Somit sollen solche verbindenden Ausdrucksweisen im Allgemeinen nicht implizieren, dass bestimmte Ausführungen es 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 Begriff „Vielzahl“ einen Status der Pluralität (z. B. gibt „eine Vielzahl von Objekten“ mehrere Objekte an). In mindestens einer Ausführungsform beträgt die Anzahl der Gegenstände in einer Vielzahl mindestens zwei, es können aber auch mehr sein, wenn dies entweder explizit oder durch den Kontext angegeben ist. Sofern nichts anderes genannt ist oder es anderweitig aus dem Kontext ersichtlich ist, bedeutet die Formulierung „auf Grundlage von“ „mindestens zum Teil auf Grundlage von“ und nicht „ausschließlich auf Grundlage von“.

[0518] Hierin beschriebene Operationen von Prozessen können in einer beliebigen geeigneten Reihenfolge durchgeführt werden, sofern hierin nichts anderes angegeben ist oder der Kontext dem eindeutig widerspricht. In mindestens einer Ausführungsform wird ein Prozess, wie zum Beispiel die hierin 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 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 Speicherme-

dium, 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, hierin beschriebene Vorgänge durchzuführen. Ein Satz von nichttransitorischen computerlesbaren Speichermedien umfasst in mindestens einer Ausführungsform mehrere nichttransitorische computerlesbare Speichermedien und einem oder mehreren der einzelnen nichttransitorischen Speichermedien mehrerer nichttransitorischer computerlesbarer Speichermedien fehlt der gesamte Code, während mehrere nichttransitorische computerlesbare Speichermedien zusammen 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 unterschiedliche Prozessoren führen unterschiedliche Teilmengen von Anweisungen aus.

[0519] Dementsprechend sind in mindestens einer Ausführungsform Computersysteme so konfiguriert, dass sie einen oder mehrere Dienste implementieren, die einzeln oder zusammen Operationen der hierin beschriebenen Prozesse durchführen, und derartige Computersysteme sind mit geeigneter Hardware und/oder Software konfiguriert, die eine Durchführung der Operationen ermöglichen. 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 hierin beschriebenen Vorgänge durchführt und sodass eine einzelne Vorrichtung nicht alle Vorgänge durchführt.

[0520] Die Verwendung von Beispielen oder beispielhafter Wortwahl (z. B. „wie etwa“), die hierin bereitgestellt ist, soll lediglich die Ausführungsformen der Offenbarung besser verdeutlichen 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.

[0521] Jegliche Bezugnahmen, einschließlich Veröffentlichungen, Patentanmeldungen und Patenten, die hierin genannt werden, sind hiermit durch Bezugnahme in demselben Maße aufgenommen, als wäre jede Bezugnahme einzeln und spezifisch als durch Bezugnahme eingeschlossen angegeben und in ihrer Gesamtheit hierin dargelegt.

[0522] In der Beschreibung und den Ansprüchen können die Ausdrücke „gekoppelt“ und „verbunden“ sowie deren Ableitungen verwendet werden. Es versteht sich, dass diese Begriffe nicht unbedingt 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.

[0523] Sofern nicht ausdrücklich etwas anderes genannt ist, versteht es sich, dass sich Ausdrücke wie etwa „Verarbeiten“, „Rechnen“, „Berechnen“, „Bestimmen“ oder dergleichen in der gesamten Beschreibung auf Handlungen und/oder Prozesse eines Computers oder Rechensystems oder einer ähnlichen elektronischen Rechenvorrichtung beziehen, die Daten, die als physische, wie etwa elektronische, Größen innerhalb der Register und/oder Speicher des Rechensystems dargestellt sind, manipulieren und/oder in andere Daten umwandeln, die auf ähnliche Weise als physische Größen innerhalb der Speicher, Register oder anderen derartigen Informationsspeicher-, -übertragungs- oder -anzeigevorrichtungen des Rechensystems dargestellt sind.

[0524] Auf ähnliche Weise kann sich der Ausdruck „Prozessor“ auf eine beliebige Vorrichtung oder einen beliebigen Abschnitt einer Vorrichtung beziehen, die/der elektronische Daten aus Registern und/oder Speicher verarbeitet und diese elektronischen Daten in andere elektronische Daten umwandelt, die in Registern

und/oder 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. Wie hierin verwendet, können „Software“-Prozesse zum Beispiel Software- und/oder Hardware-Entitäten beinhalten, die im Verlauf der Zeit Arbeit verrichten, wie etwa Tasks, Threads und intelligente Agenten. Außerdem kann sich jeder Prozess auf mehrere Prozesse beziehen, um Anweisungen nacheinander oder parallel, kontinuierlich oder intermittierend auszuführen. In mindestens einer Ausführungsform werden die Ausdrücke „System“ und „Verfahren“ hierin insofern austauschbar verwendet, als ein System ein oder mehrere Verfahren verkörpern kann und die Verfahren als System betrachtet werden können.

[0525] Im vorliegenden Dokument kann auf das Erlangen, Erfassen, Empfangen oder Eingeben von analogen oder digitalen Daten in ein Teilsystem, ein Computersystem oder eine computerimplementierte Maschine Bezug genommen werden. In mindestens einer Ausführungsform kann der Prozess des Erlangens, Erfassens, Empfangens oder Eingebens von analogen und digitalen Daten auf eine Vielfalt von Weisen erzielt werden, wie etwa durch das Empfangen von Daten als Parameter eines Funktionsaufrufs oder eines Aufrufs einer Anwendungsprogrammierschnittstelle. In mindestens einer Ausführungsform kann der Prozess des Erlangens, Erfassens, Empfangens oder Eingebens von analogen oder digitalen Daten durch das Übermitteln von Daten über eine serielle oder parallele Schnittstelle erfolgen. In mindestens einer Ausführungsform kann der Prozess des Erlangens, Erfassens, Empfangens oder Eingebens von analogen oder digitalen Daten durch das Übermitteln von Daten über ein Computernetz von der bereitstellenden Entität zu der erfassenden Entität erfolgen. In mindestens einer Ausführungsform kann auch auf das Bereitstellen, Ausgeben, Übertragen, Senden oder Darstellen von analogen oder digitalen Daten Bezug genommen werden. In verschiedenen Beispielen können Prozesse des Bereitstellens, Ausgebens, Übertragens, Sendens oder Darstellens von analogen oder digitalen Daten durch das Übermitteln von Daten als Eingabe- oder Ausgabeparameter eines Funktionsaufrufs, eines Parameters einer Anwendungsprogrammierschnittstelle oder eines Interprozesskommunikationsmechanismus erfolgen.

[0526] Obwohl die vorstehenden Beschreibungen beispielhafte Implementationen der beschriebenen Techniken darlegen, können auch andere Architekturen verwendet werden, um die beschriebenen Funktionen zu implementieren, und sie sollen im Umfang dieser Offenbarung liegen. Darüber hinaus könnten, obwohl spezifische Verteilungen von Zuständigkeiten vorstehend zum Zwecke der Beschreibung definiert wurden, verschiedene Funktionen und Zuständigkeiten in Abhängigkeit von den Umständen anders verteilt und aufgeteilt werden.

[0527] 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 spezifischen 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

- WO 3016201806 [0085]

Patentansprüche**1. Prozessor umfassend:**

einen oder mehrere Schaltkreise zur Verwendung eines oder mehrerer neuronaler Netze, um Betrug durch einen oder mehrere Benutzer eines Computerspiels zumindest teilweise auf der Grundlage eines oder mehrerer durch das Computerspiel erzeugter Bilder zu erkennen.

2. Prozessor nach Anspruch 1, wobei der eine oder die mehreren Schaltkreise weiterhin dazu dienen: das eine oder die mehreren Bilder in einem Puffer zu speichern, wobei das gespeicherte eine oder die mehreren Bilder als Eingabe für das eine oder die mehreren neuronalen Netze bereitzustellen und auf einer Anzeigeeinheit wiederzugeben sind.

3. Prozessor nach Anspruch 2, wobei der eine oder die mehreren Schaltkreise weiterhin dazu dienen: einen Bericht zu erstellen, der die Erkennung von Betrug anzeigt; und den Bericht an einen Spieleserver zu übermitteln.

4. Prozessor nach Anspruch 3, wobei der eine oder die mehreren Schaltkreise weiterhin zu Folgendem dienen:
Erzeugen, unter Verwendung des einen oder der mehreren neuronalen Netze, einer Zuverlässigkeitsstufe, die die Zuverlässigkeit charakterisiert, dass das eine oder die mehreren Bilder Betrugsinformationen enthalten.

5. Prozessor nach Anspruch 4, wobei der Bericht an den Spieleserver als Reaktion auf die Feststellung, dass die Zuverlässigkeitsstufe bei oder über einem Schwellenwert liegt, übermittelt wird.

6. Prozessor nach einem der Ansprüche 3 bis 5, wobei die eine oder die mehreren Schaltungen ferner zu Folgendem dienen:
vom Spieleserver aktualisierte Parameter für das eine oder die mehreren neuronalen Netzwerke zu empfangen, wobei die aktualisierten Parameter auf der Grundlage eines Satzes von Neutrainingbildern erzeugt werden.

7. Prozessor nach einem der vorhergehenden Ansprüche, wobei der eine oder die mehreren Schaltkreise ein Zertifizierungssignal für einen Spieleserver erzeugen sollen, wobei das Zertifizierungssignal dem Spieleserver bescheinigen soll, dass der eine oder die mehreren Schaltkreise in der Lage sind, Betrug im Zusammenhang mit dem Computerspiel zu erkennen.

8. Prozessor umfassend:

eine oder mehrere Schaltungen zur Durchführung des Trainings eines oder mehrerer neuronaler Netze, um Betrug durch einen oder mehrere Benutzer eines Computerspiels zu erkennen, wobei das Training zumindest teilweise auf einem oder mehreren Betrugsbildern basiert.

9. Prozessor nach Anspruch 8, wobei das eine oder die mehreren Betrugsbilder unter Verwendung einer mit dem Computerspiel verbundenen Betrugssoftware erzeugt werden.

10. Prozessor nach Anspruch 8 oder 9, wobei das Training des einen oder der mehreren neuronalen Netze zumindest teilweise auf einem oder mehreren Nichtbetrugsbildern basiert, wobei jedes der einen oder mehreren Betrugsbilder Betrugsinformationen enthält und jedes der einen oder mehreren Nichtbetrugsbilder keine Betrugsinformationen enthält.

11. Prozessor nach Anspruch 10, wobei mindestens eine Teilmenge des einen oder der mehreren Betrugsbilder Bilder umfasst, die mit Informationen aus Nichtbetrugsbildern angereichert sind, die von einer mit dem Computerspiel verbundenen Spielsoftware erzeugt werden.

12. Prozessor nach Anspruch 11, wobei jedes der Teilmenge des einen oder der mehreren Betrugsbilder einen Teil umfasst, der durch einen Teil eines Nichtbetrugsbildes ersetzt wird.

13. Prozessor nach einem der Ansprüche 10 bis 12, wobei mindestens eine Teilmenge des einen oder der mehreren Nichtbetrugsbilder Nichtbetrugsbilder umfasst, die mit Informationen aus Betrugsbildern angereichert sind, die von einer mit dem Computerspiel verbundenen Spielsoftware erzeugt wurden.

14. Prozessor nach Anspruch 13, wobei jedes der Teilmenge des einen oder der mehreren Nichtbetrugsbilder einen Teil umfasst, der durch einen Teil eines Betrugsbildes ersetzt wird, das eine Nichtbetrugsinformation enthält.

15. Prozessor nach einem der Ansprüche 8 bis 14, wobei die eine oder mehreren Schaltungen ferner dazu dienen, ein Training des einen oder der mehreren neuronalen Netze gegen gegnerische Angriffe durchzuführen.

16. Prozessor nach Anspruch 15, wobei das Training des einen oder der mehreren neuronalen Netze gegen gegnerische Angriffe zumindest teilweise auf einer Teilmenge des einen oder der mehreren Betrugsbilder basiert, die mit gegnerischen Störungen modifiziert wurden.

17. System, umfassend:
einen oder mehrere Prozessoren zur Verwendung eines oder mehrerer neuronaler Netze, um Betrug durch einen oder mehrere Benutzer eines Computerspiels zumindest teilweise auf der Grundlage eines oder mehrerer durch das Computerspiel erzeugter Bilder zu erkennen; und
einen oder mehrere Speicher zum Speichern von Parametern, die mit dem einen oder den mehreren neuronalen Netzen verbunden sind.

18. System nach Anspruch 17, wobei der eine oder die mehreren Prozessoren ferner zu Folgendem dienen:
das eine oder die mehreren Bilder in einem Puffer zu speichern, wobei das gespeicherte eine oder die mehreren Bilder als Eingabe für das eine oder die mehreren neuronalen Netze bereitzustellen sind und auf einer Anzeigeeinheit wiederzugeben sind.

19. System nach Anspruch 18, wobei die eine oder mehreren Prozessoren ferner zu Folgendem dienen: einen Bericht zu erstellen, der die Erkennung von Betrug anzeigt; und den Bericht an einen Spieleserver zu übermitteln.

20. System nach Anspruch 19, wobei der eine oder die mehreren Prozessoren zur Übermittlung des Berichts an den Spieleserver Folgendes tun:
Erzeugen, unter Verwendung des einen oder der mehreren neuronalen Netze, einer Zuverlässigkeitsstufe, die die Zuverlässigkeit charakterisiert, dass das eine oder die mehreren Bilder eine Betrugsinformation enthalten; und
bestimmen, dass die Zuverlässigkeitsstufe bei oder über einem Schwellenwert liegt.

21. System, umfassend:
einen oder mehrere Prozessoren zum Durchführen des Trainings eines oder mehrerer neuronaler Netze, um Betrug durch einen oder mehrere Benutzer eines Computerspiels zumindest teilweise auf der Grundlage eines oder mehrerer durch das Computerspiel erzeugter Betrugsbilder zu erkennen; und
einen oder mehrere Speicher zum Speichern von Parametern, die mit dem einen oder den mehreren neuronalen Netzen verbunden sind.

22. Prozessor nach Anspruch 21, wobei das eine oder die mehreren Betrugsbilder unter Verwendung einer mit dem Computerspiel verbundenen Betrugssoftware erzeugt werden, und das Training des einen oder der mehreren neuronalen Netze zumindest teilweise auf einem oder mehreren Nichtbetrugsbildern basiert, wobei jedes der einen oder mehreren Betrugsbilder Betrugsinformationen enthält und jedes der einen oder mehreren Nichtbetrugsbilder keine Betrugsinformationen enthält.

23. System nach Anspruch 22, wobei mindestens eine Teilmenge des einen oder der mehreren Betrugsbilder Bilder umfasst, die mit Informationen aus Nichtbetrugsbildern angereichert sind, die von einer mit dem Computerspiel verbundenen Spielsoftware erzeugt werden.

24. System nach Anspruch 23, wobei jedes der Teilmenge des einen oder der mehreren Betrugsbilder einen Teil umfasst, der durch einen Teil eines Nichtbetrugsbildes ersetzt wird.

25. System nach einem der Ansprüche 22 bis 24, wobei mindestens eine Teilmenge des einen oder der mehreren Nichtbetrugsbilder Nichtbetrugsbilder umfasst, die mit Informationen aus Betrugsbildern angereichert sind, die von einer mit dem Computerspiel verbundenen Spielsoftware erzeugt wurden.

26. System nach Anspruch 25, wobei jedes der Teilmenge des einen oder der mehreren Nichtbetrugsbilder einen Teil umfasst, der durch einen Teil eines Betrugsbildes ersetzt wird, das eine Nichtbetrugsinformation enthält.

27. System nach einem der Ansprüche 21 bis 26, wobei der eine oder die mehreren Prozessoren ferner dazu dienen, ein Training des einen oder der mehreren neuronalen Netze gegen gegnerische Angriffe durchzuführen.

28. Verfahren umfassend:
Empfangen einer Darstellung einer Grafik, die mit einem Computerspiel verbunden ist, von einem Rechen-
gerät;
Erzeugen eines oder mehrerer Bilder durch eine oder mehrere Schaltungen auf der Grundlage der emp-
fangenen Darstellung; und
Verarbeitung des einen oder der mehreren Bilder durch den einen oder die mehreren Schaltkreise und
unter Verwendung eines oder mehrerer neuronaler Netze, um Betrug durch einen oder mehrere Benutzer
des Computerspiels zu erkennen.

29. Verfahren nach Anspruch 28, ferner umfassend:
Erstellen eines Berichts, der die Erkennung von Betrug anzeigt; und
Übermittlung des Berichts an einen Spieleserver.

30. Verfahren nach Anspruch 28 oder 29, wobei die Übermittlung des Berichts an einen Spieleserver in
Reaktion auf Folgendes erfolgt:
Erzeugen durch den einen oder die mehreren Schaltkreise, unter Verwendung des einen oder der mehre-
ren neuronalen Netze, einer Zuverlässigkeitsstufe, die die Zuverlässigkeit charakterisiert, dass das eine
oder die mehreren Bilder eine Betrugsinformation enthalten; und
Bestimmen, dass die Zuverlässigkeitsstufe bei oder über einem Schwellenwert liegt.

31. Verfahren nach Anspruch 30, ferner umfassend:
Empfangen aktualisierter Parameter für das eine oder die mehreren neuronalen Netze, wobei die aktuali-
sierten Parameter auf der Grundlage von Neutrainingsbildern erzeugt werden.

32. Verfahren nach einem der Ansprüche 28 bis 31, wobei das eine oder die mehreren neuronalen
Netze trainiert werden unter Verwendung von:
einem oder mehreren Betrugsbildern, die unter Verwendung einer mit dem Computerspiel verbundenen
Betrugssoftware erzeugt werden; und
einem oder mehreren Nicht-Betrugsbildern, die keine Betrugsinformationen enthalten.

33. Verfahren nach Anspruch 32, wobei das eine oder die mehreren neuronalen Netze weiter gegen
gegnerische Angriffe trainiert werden, wobei das Training gegen gegnerische Angriffe zumindest teilweise
auf einer Teilmenge des einen oder der mehreren Betrugsbilder basiert, die mit gegnerischen Störungen
modifiziert sind.

Es folgen 51 Seiten Zeichnungen

Anhängende Zeichnungen

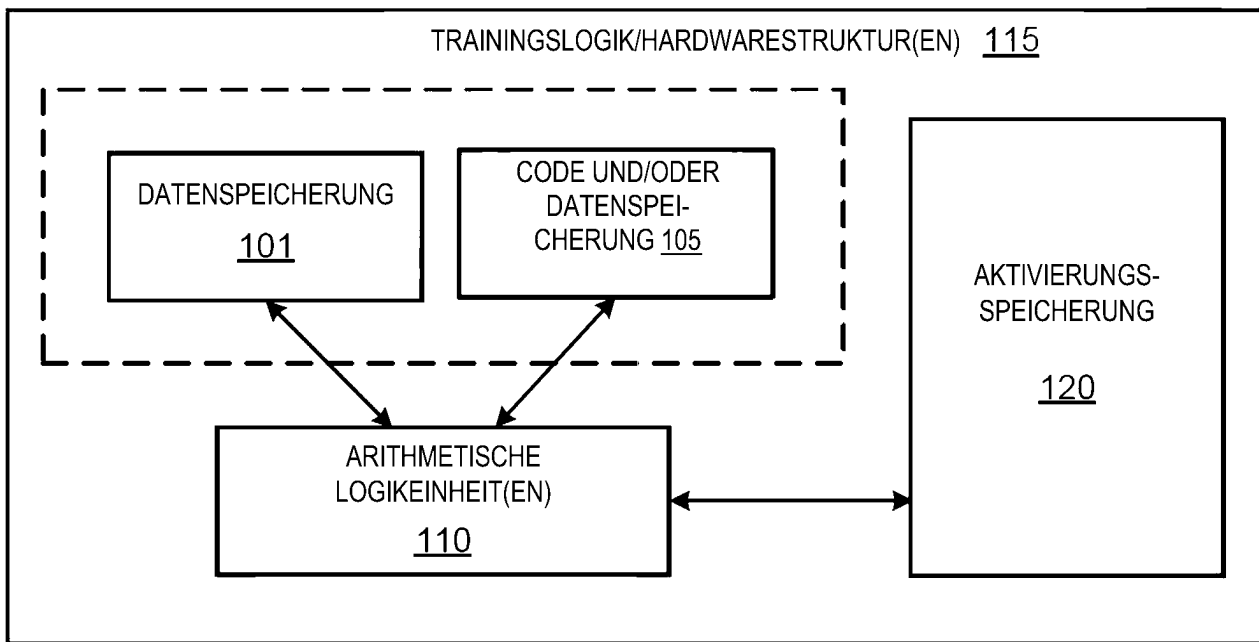


FIG. 1A

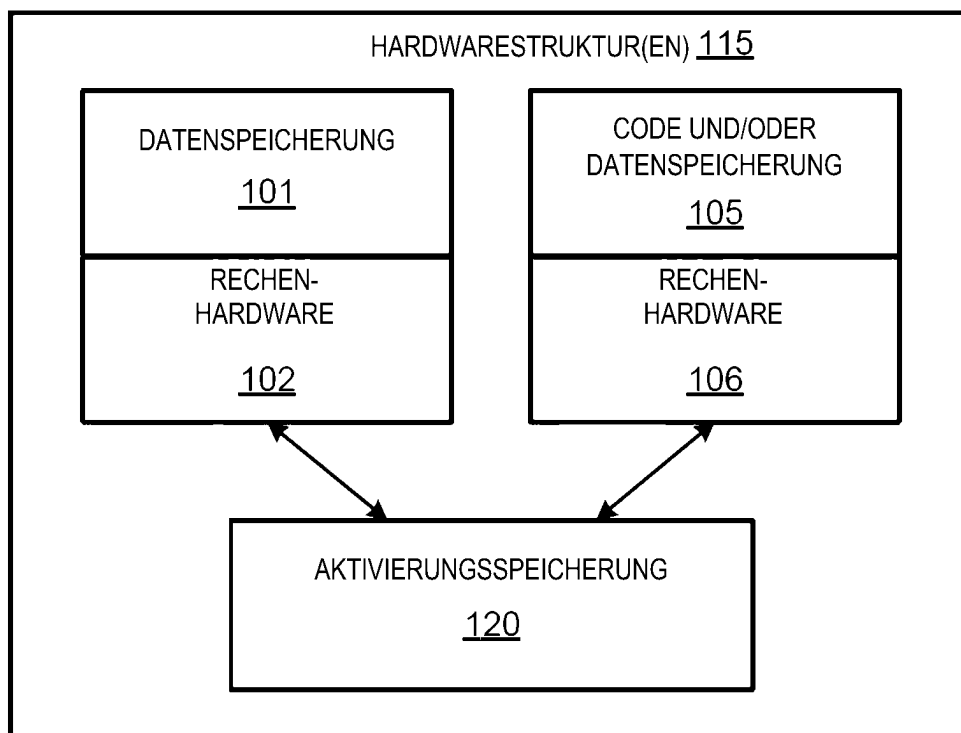


FIG. 1B

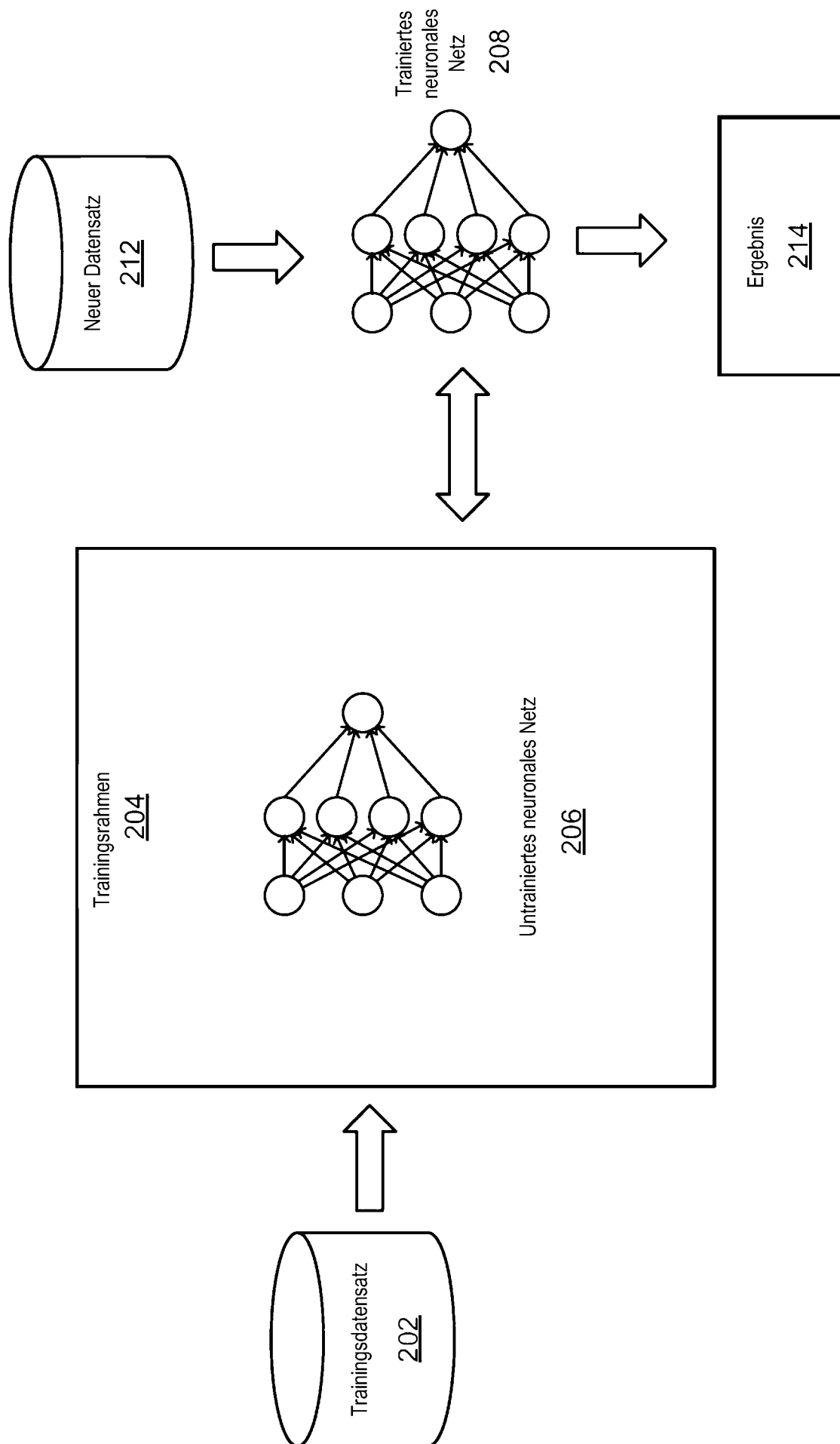


FIG. 2

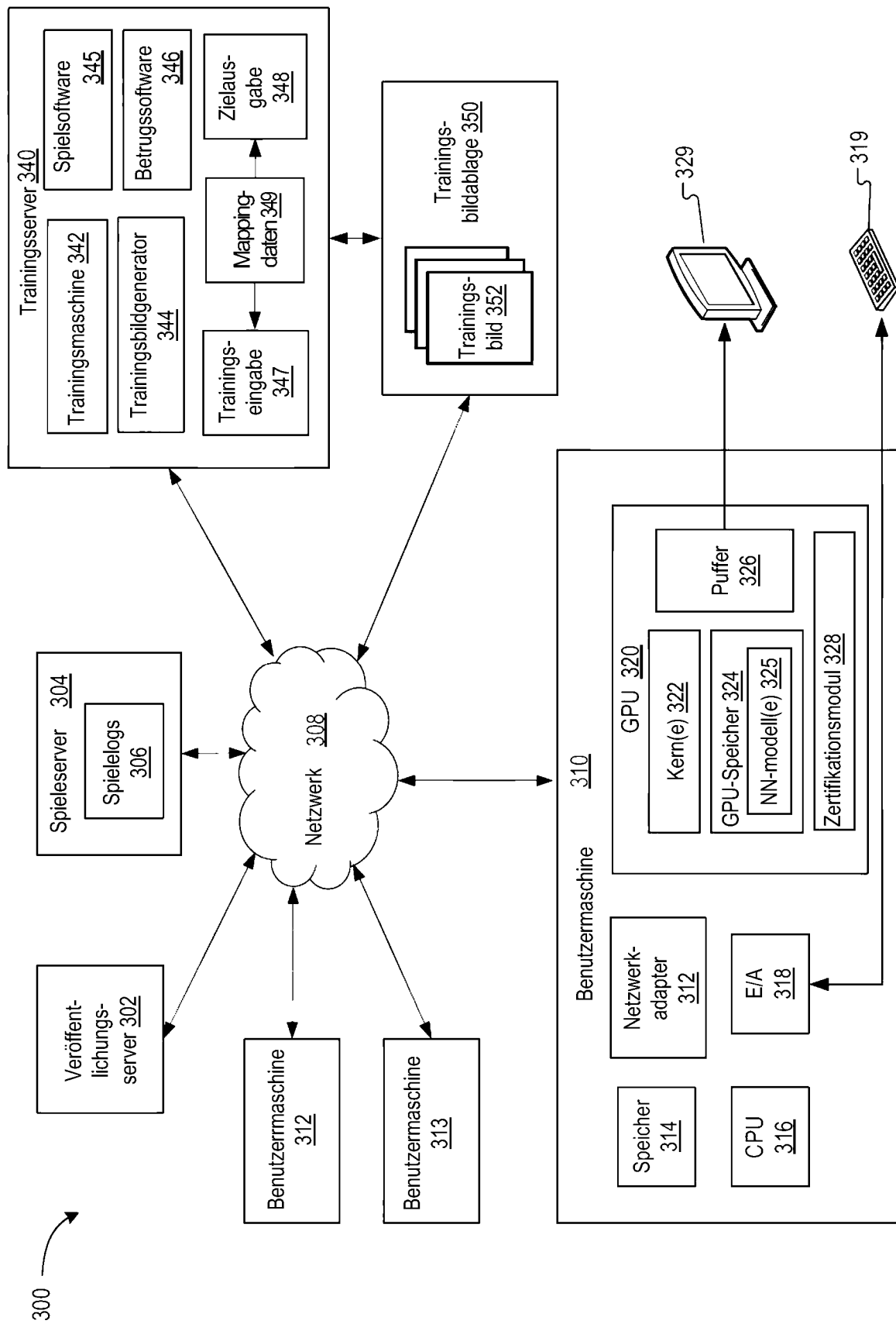


FIG. 3

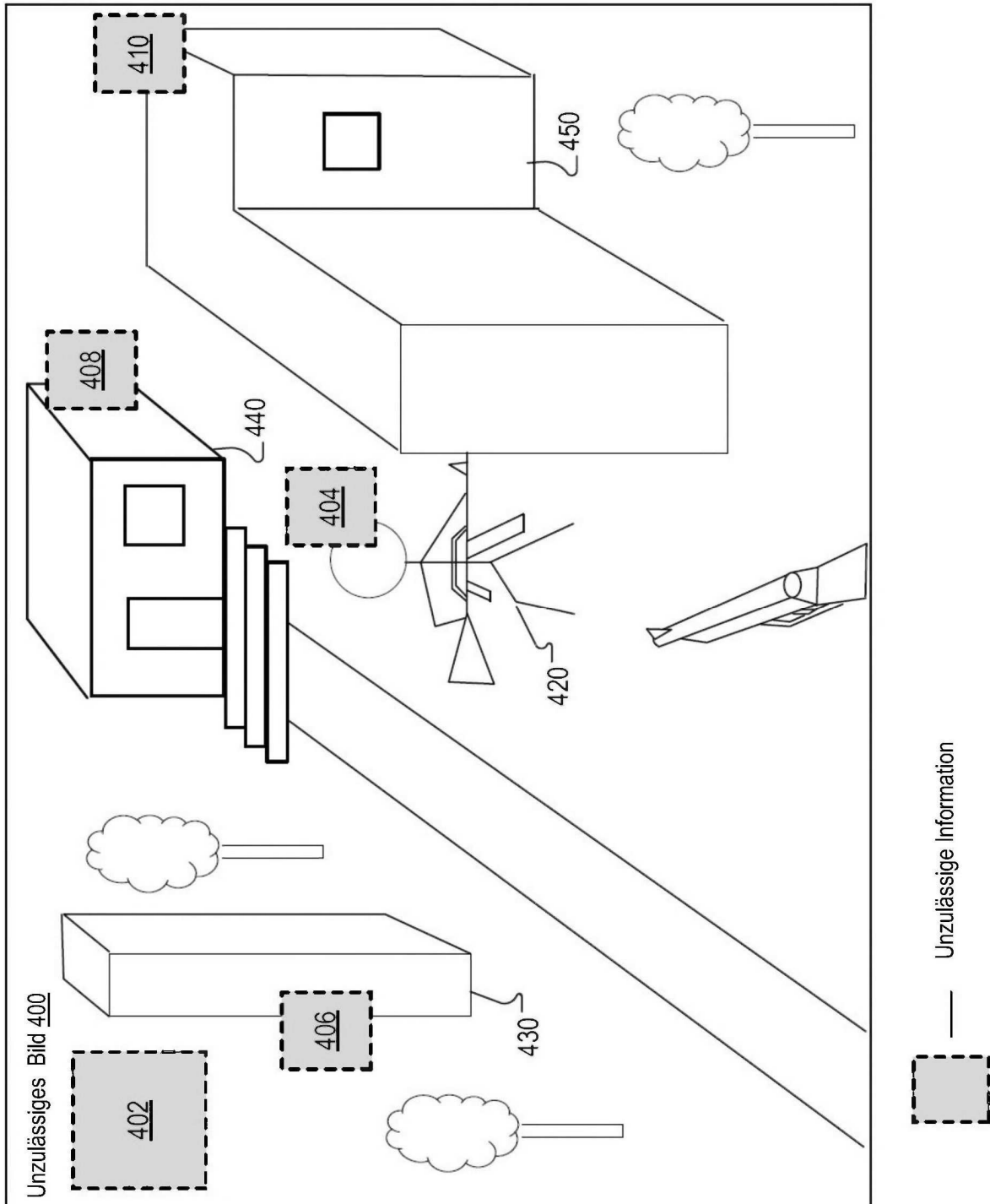


FIG. 4

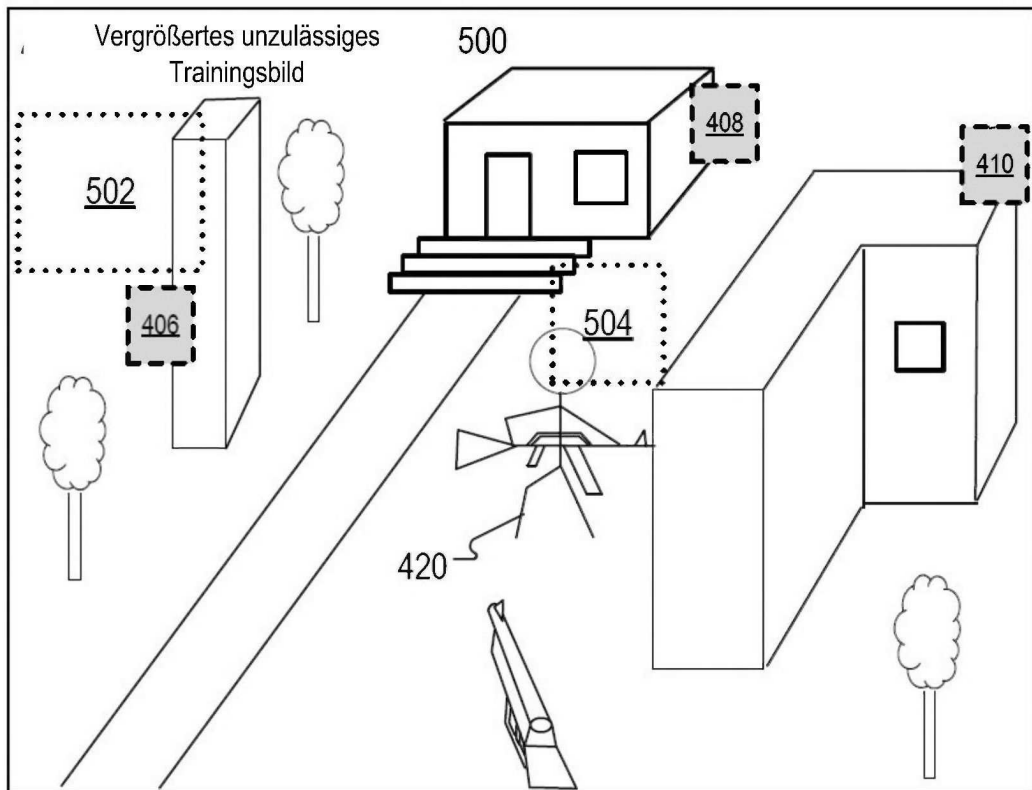


FIG. 5A

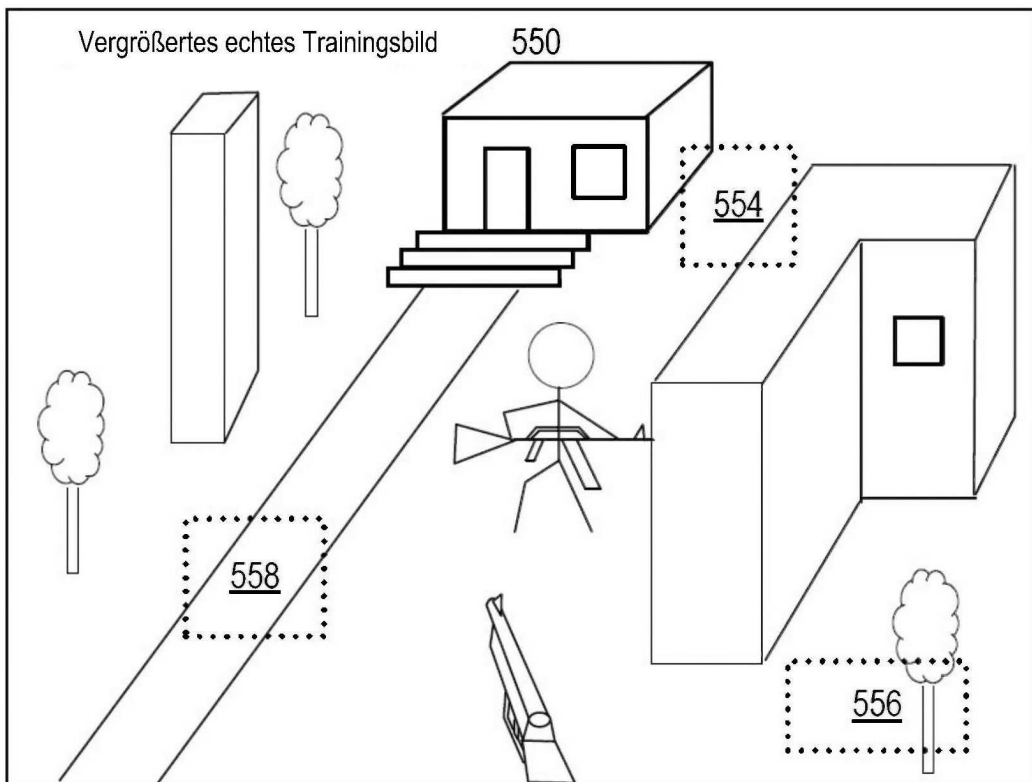


FIG. 5B

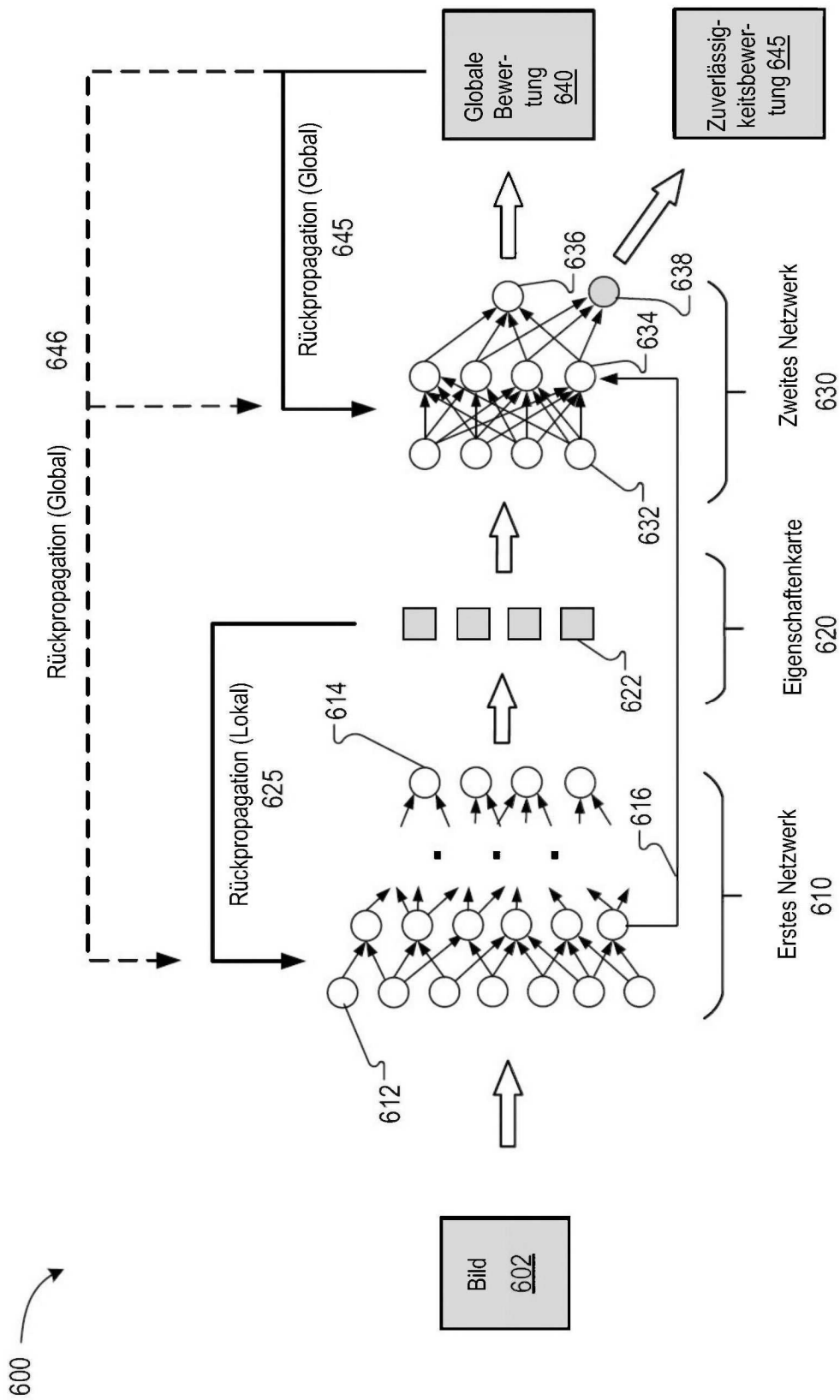


FIG. 6

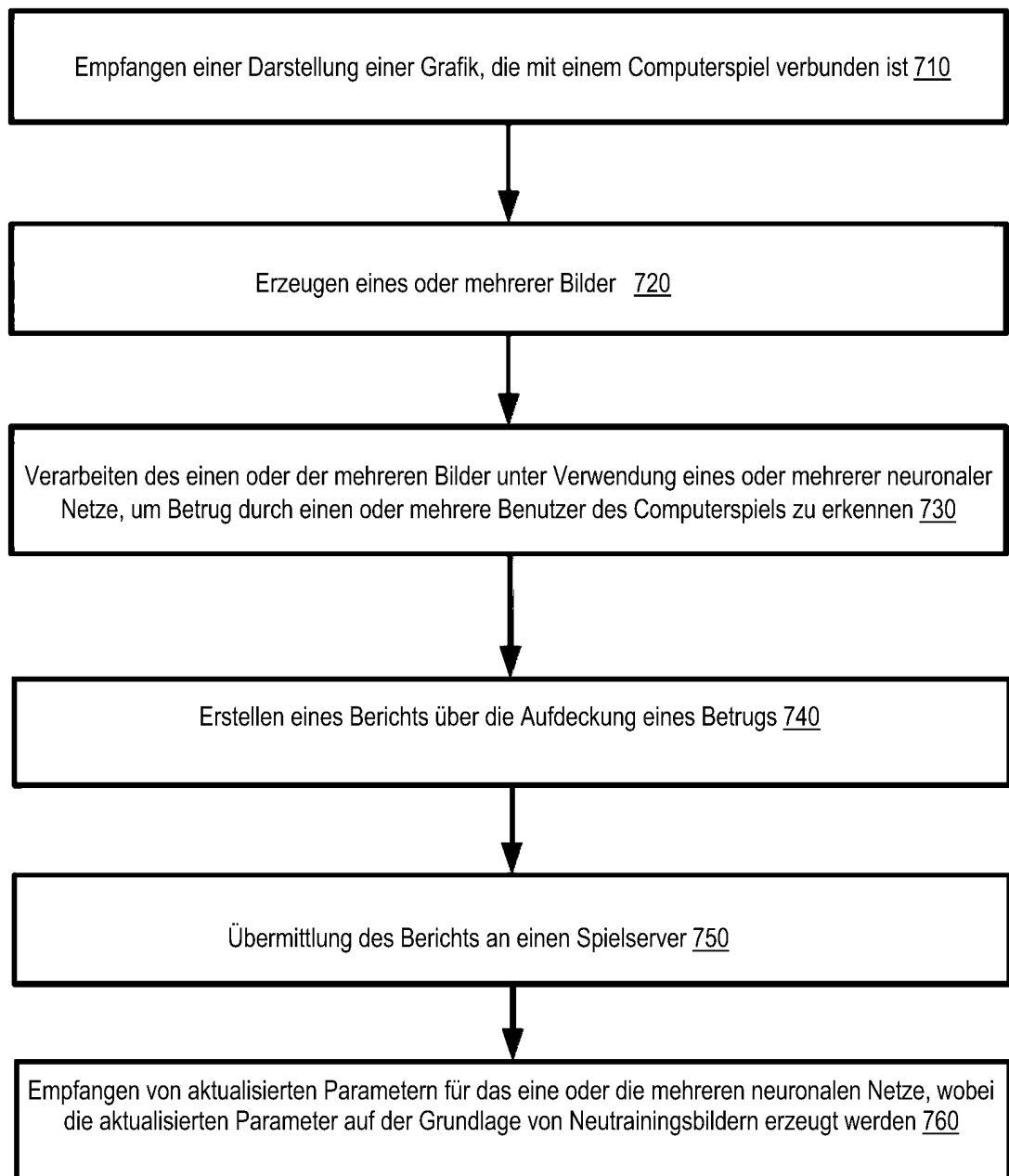
700 

FIG. 7

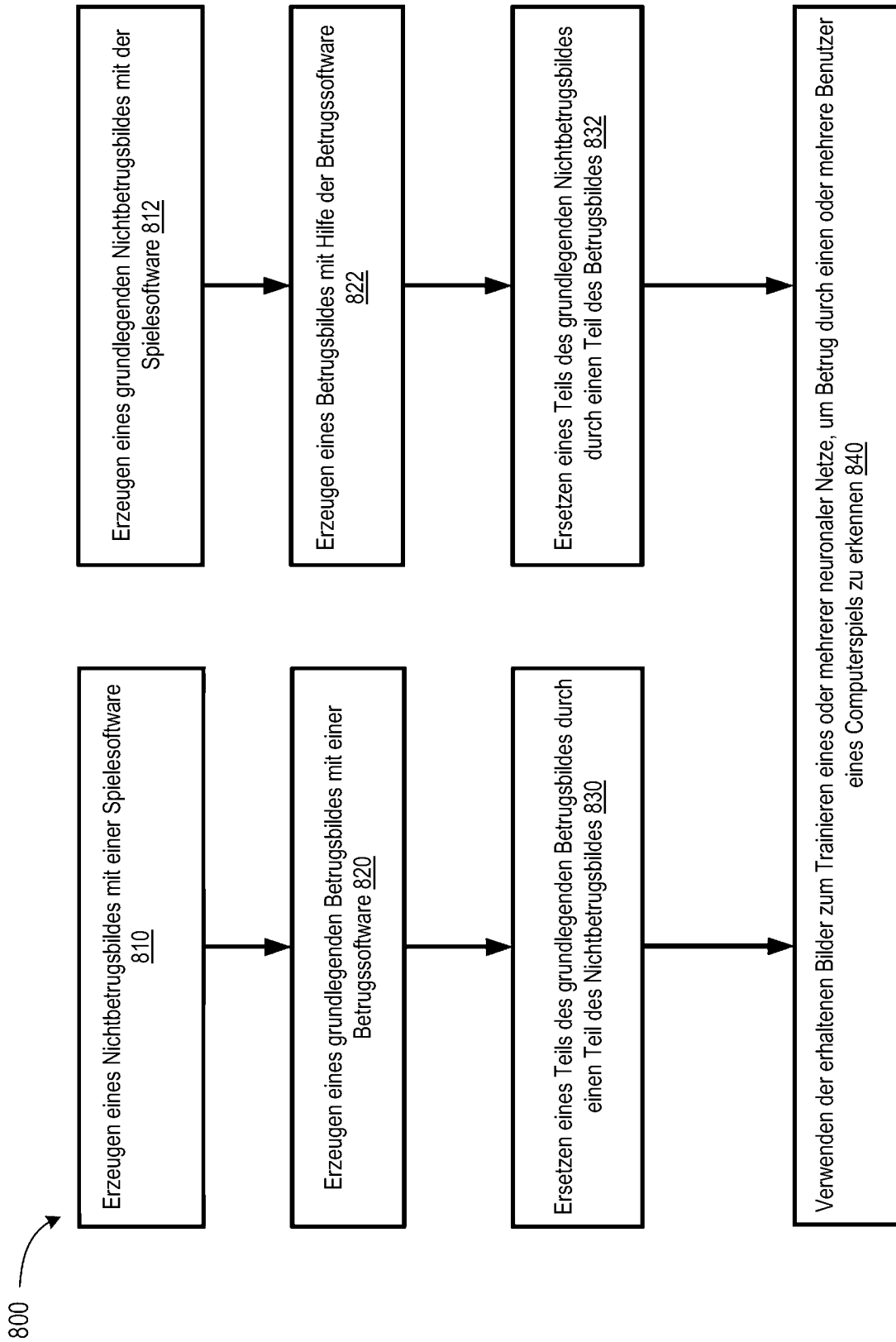
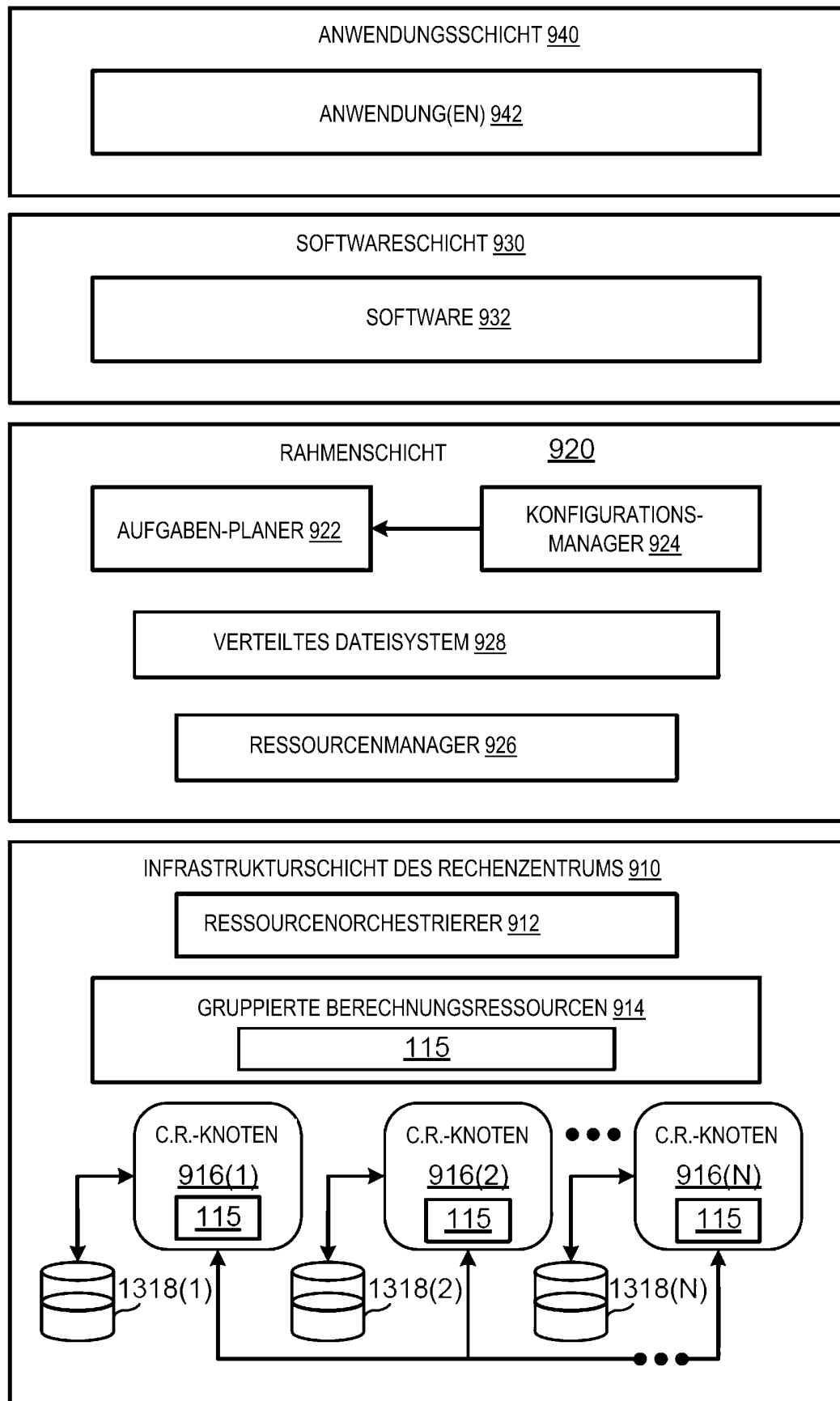


FIG. 8

RECHENZENTRUM
900**FIG. 9**

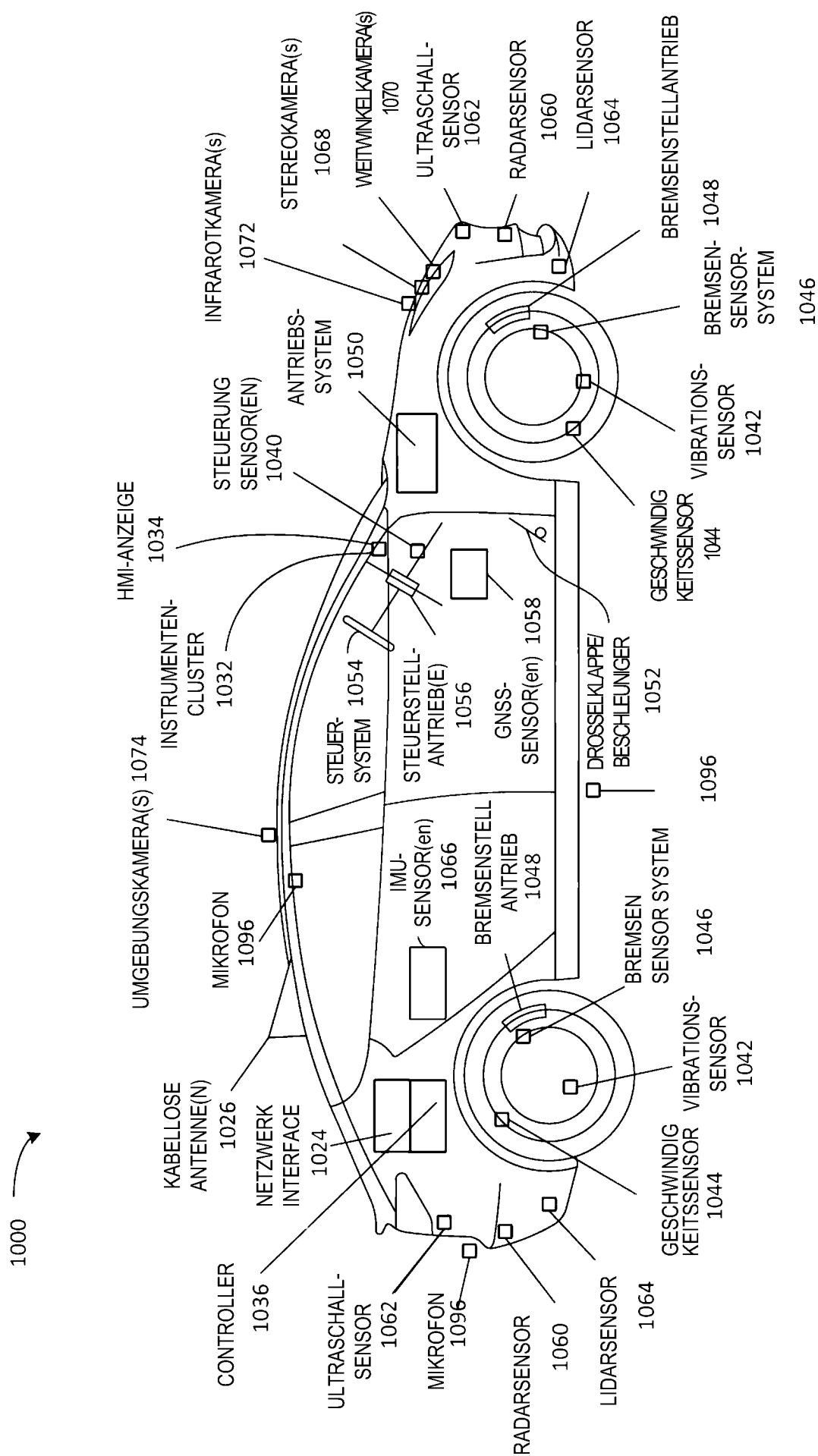


FIG. 10 A

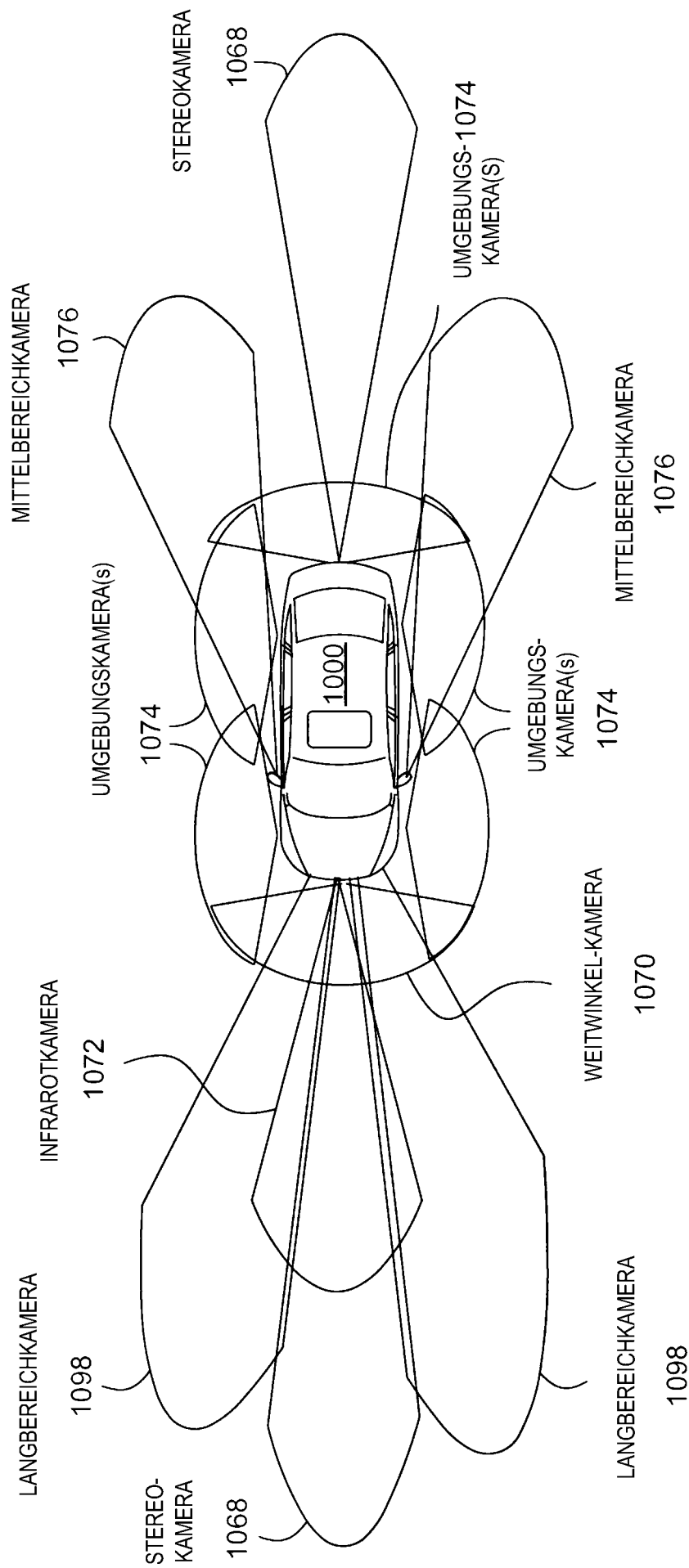


FIG. 10 B

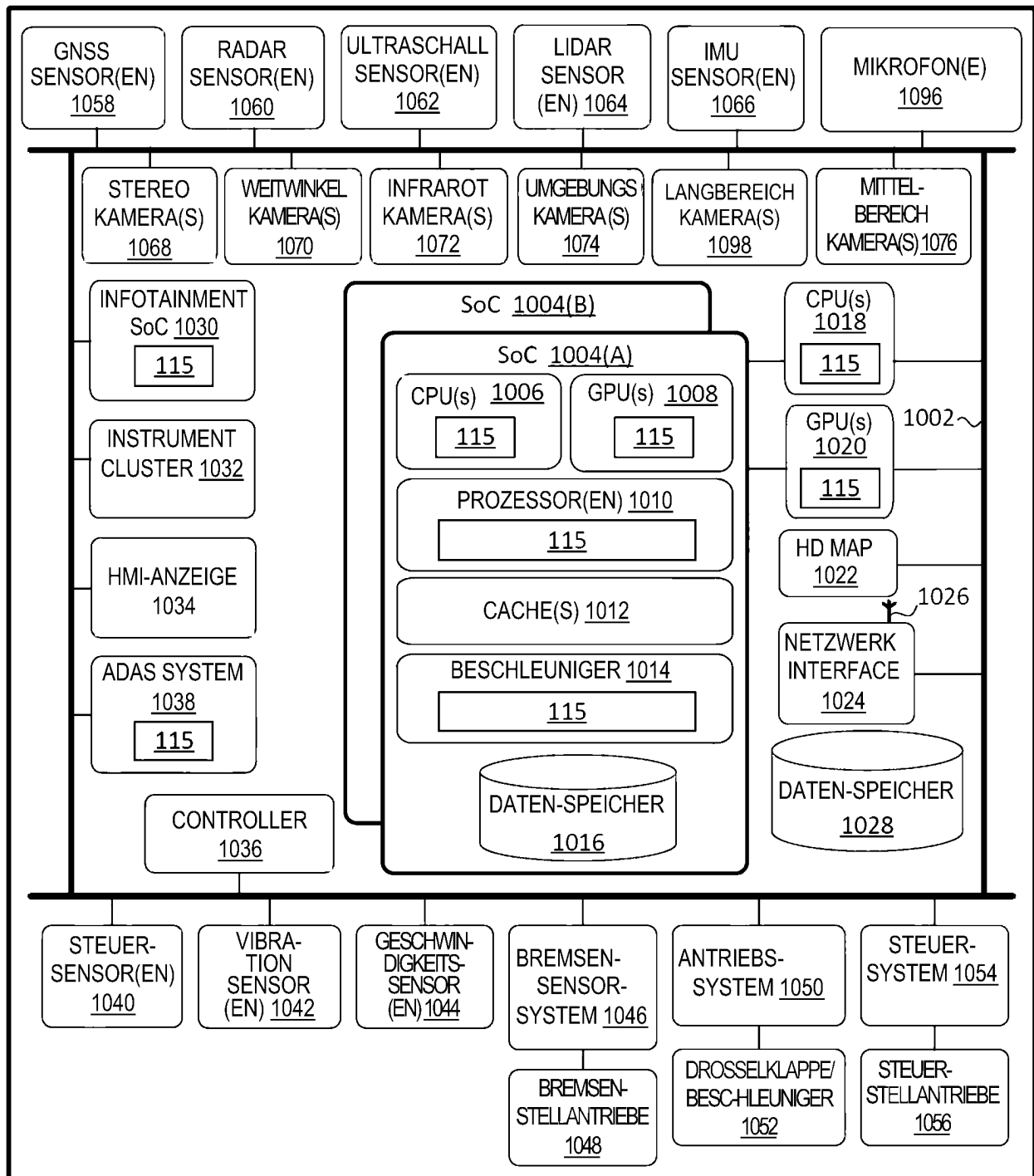


FIG. 10 C

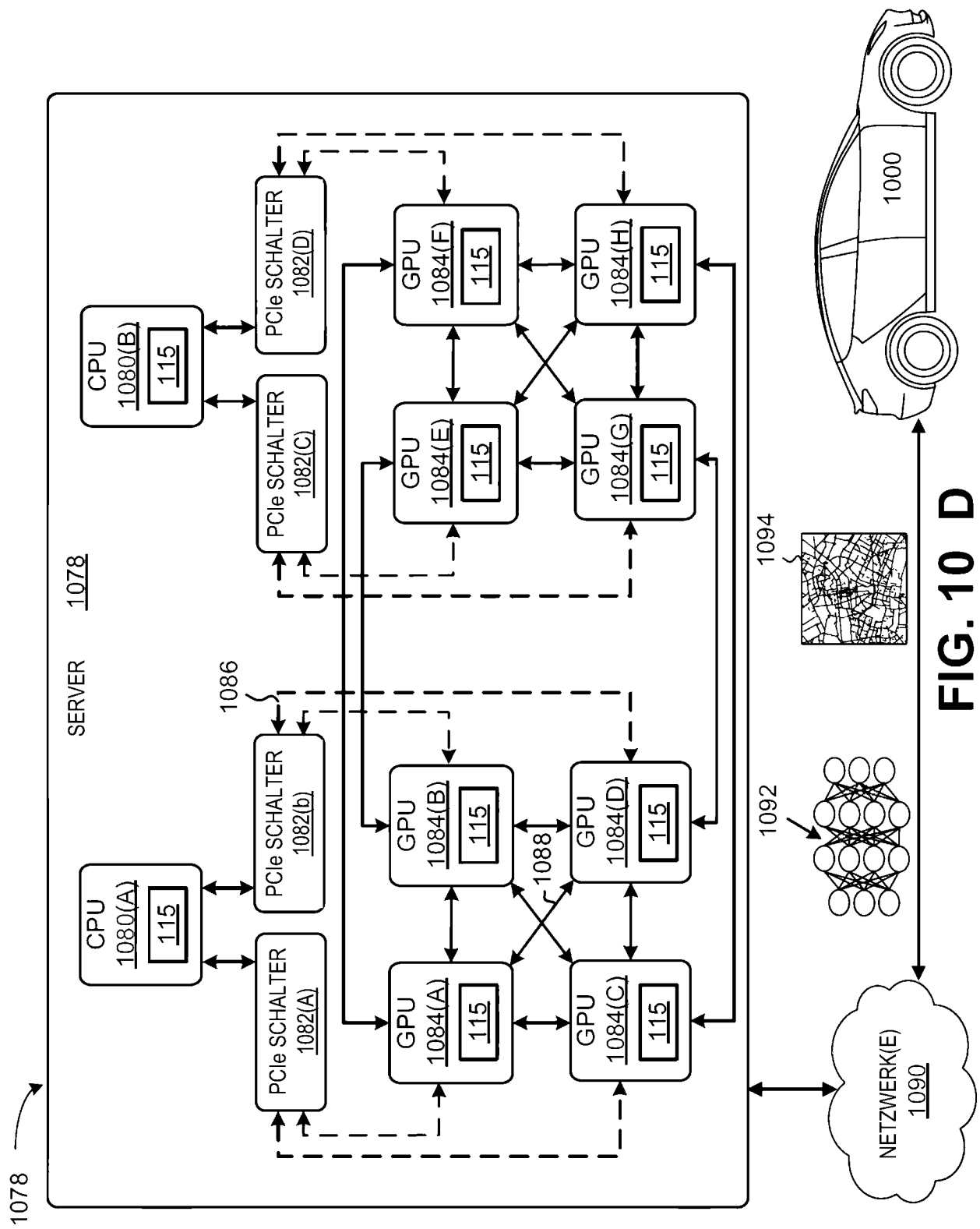
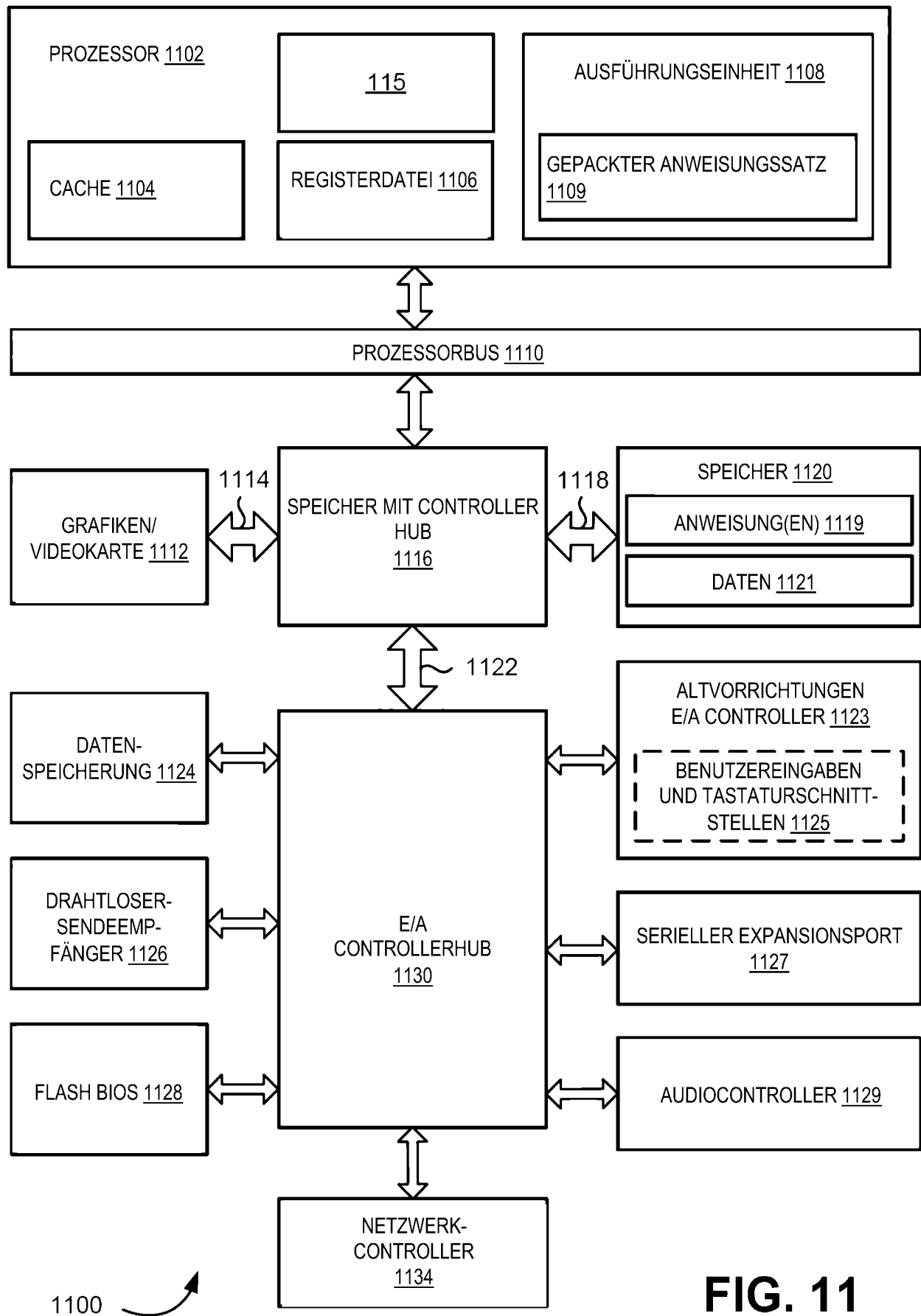
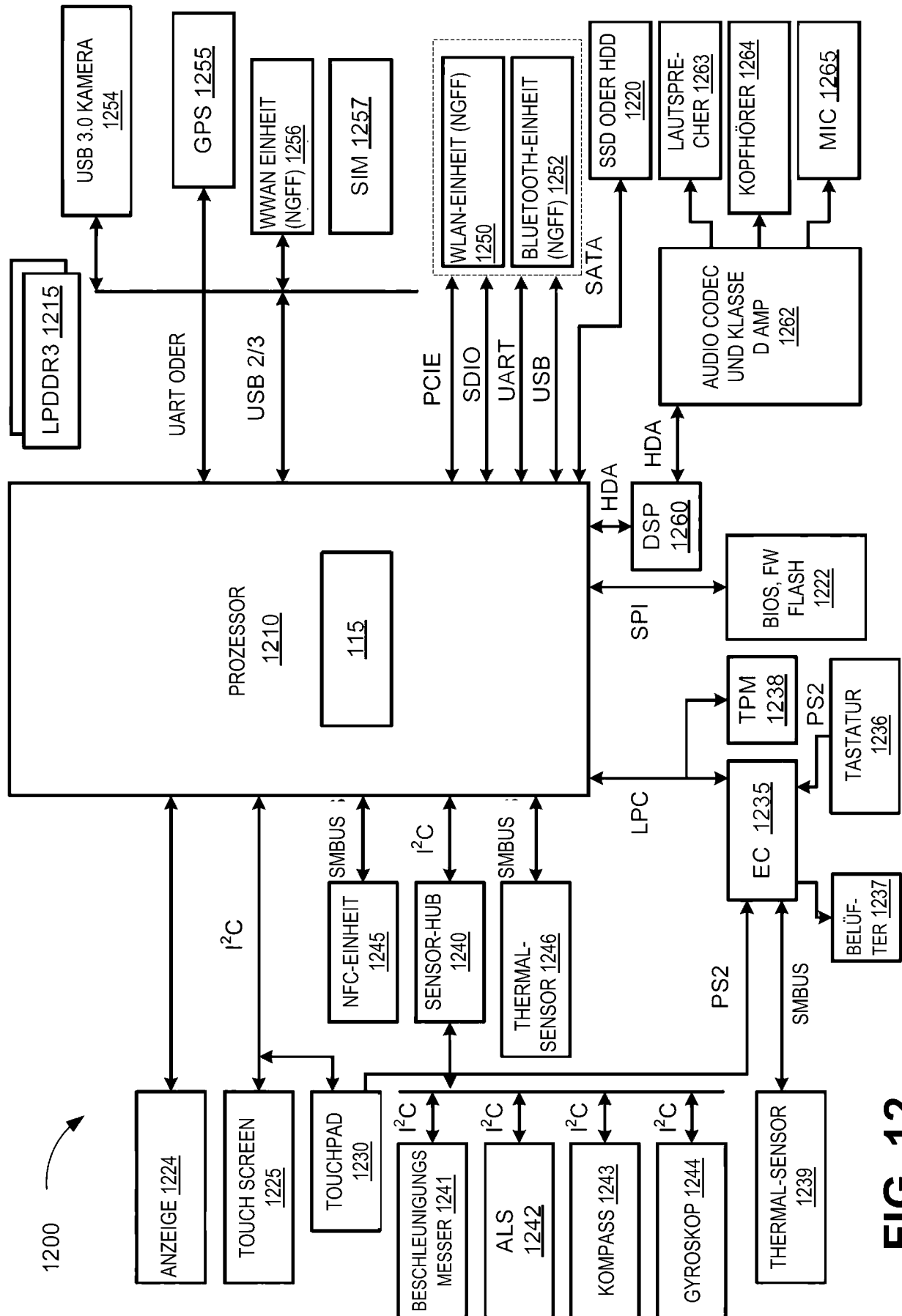
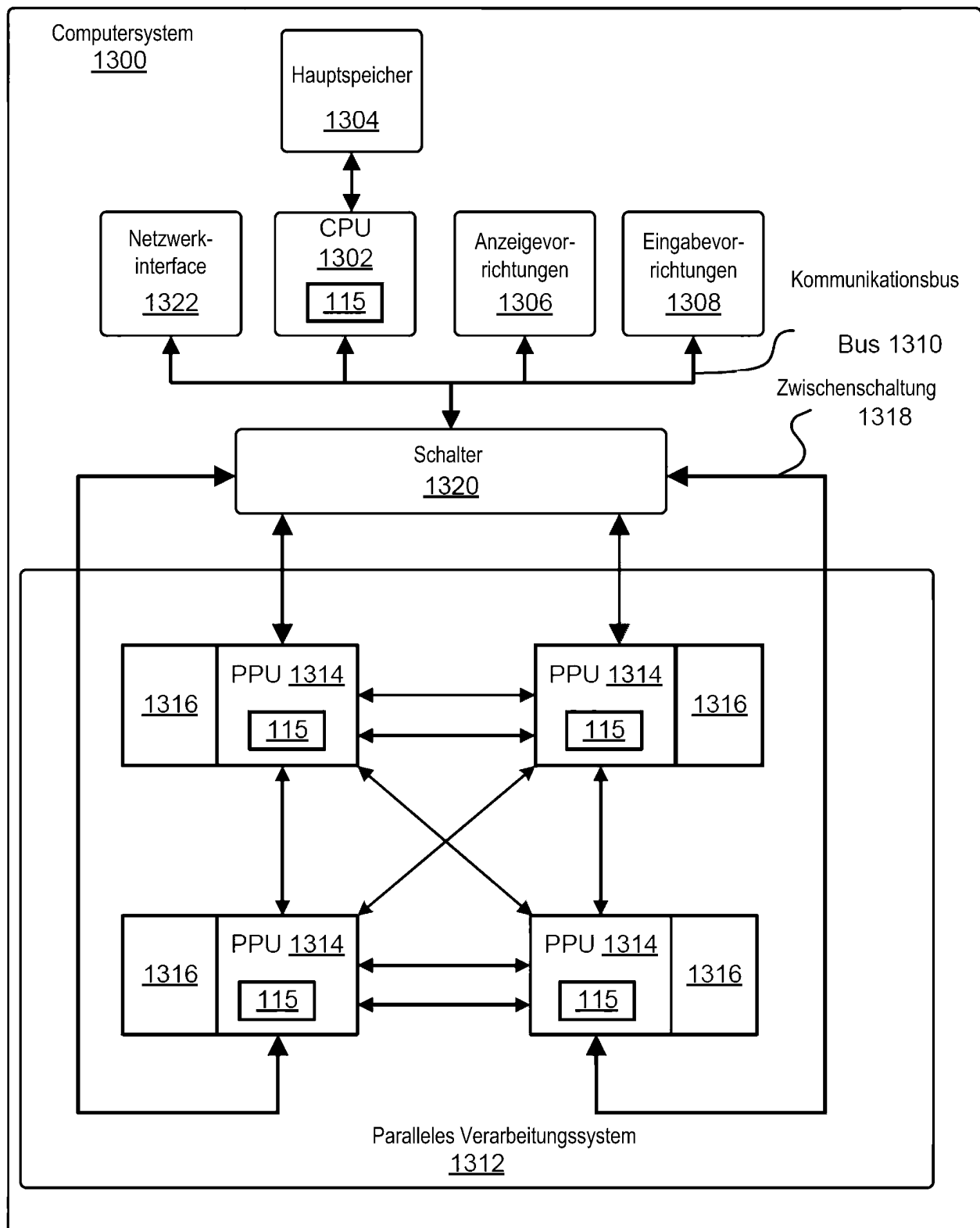


FIG. 10 D

**FIG. 11**

**FIG. 12**

**FIG. 13**

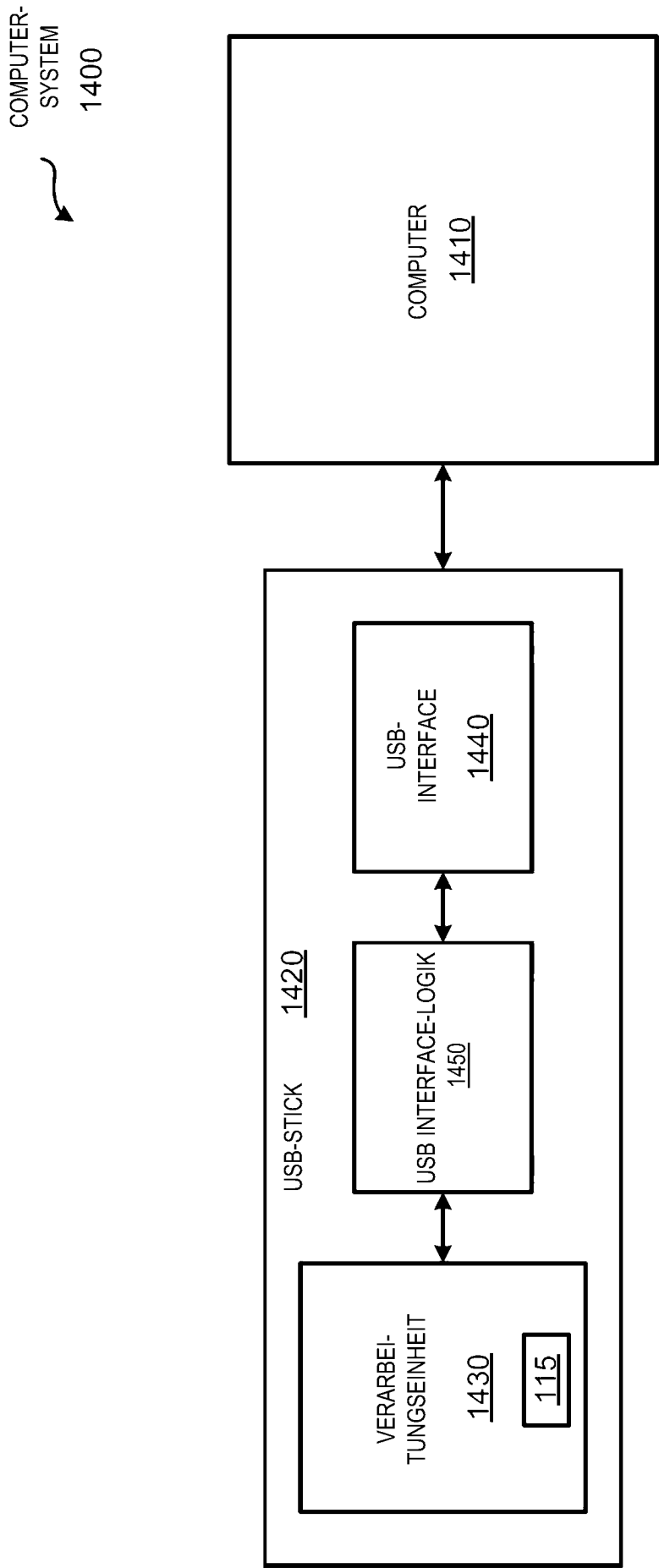


FIG. 14

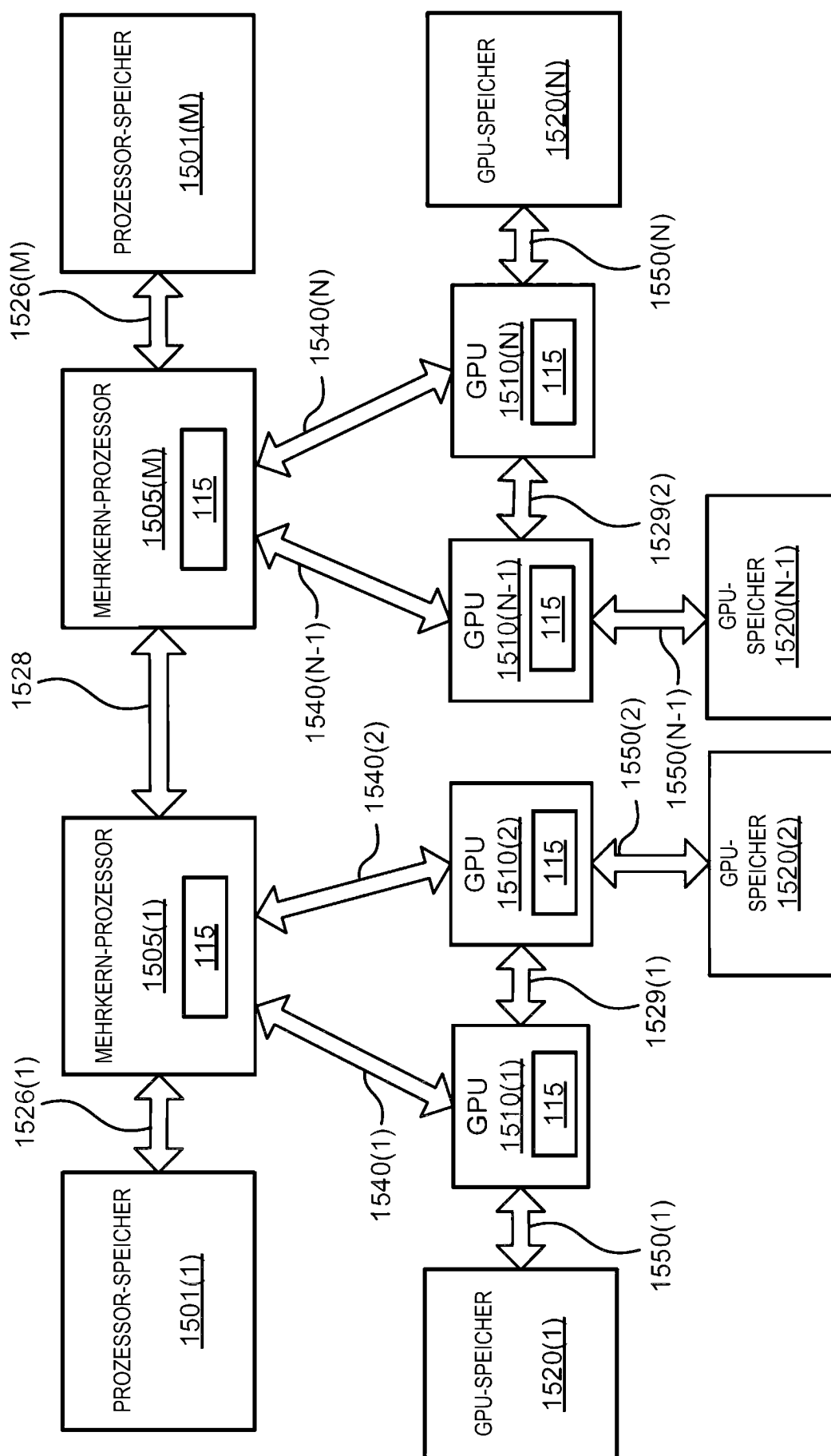


FIG. 15 A

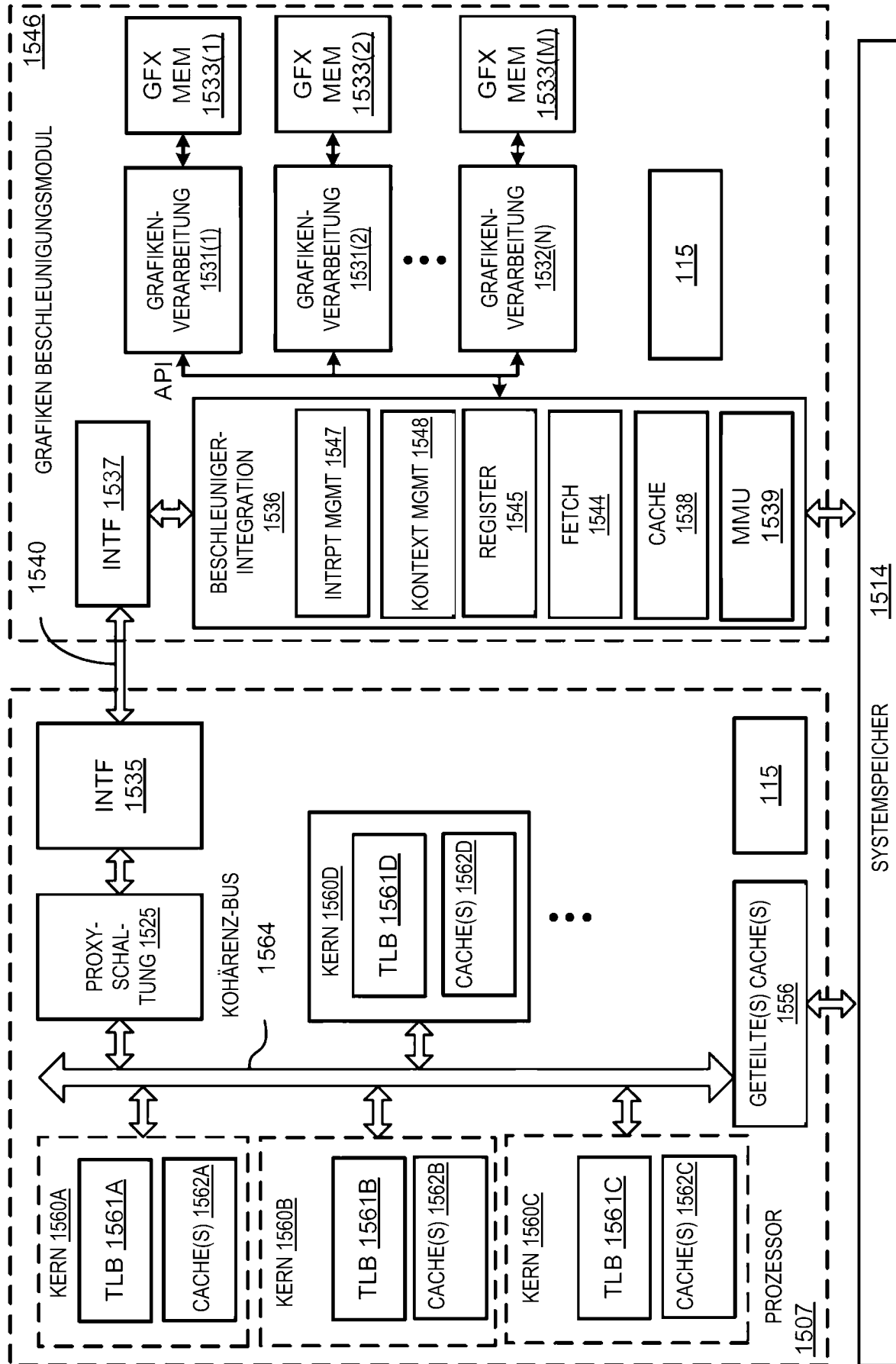


FIG. 15 B

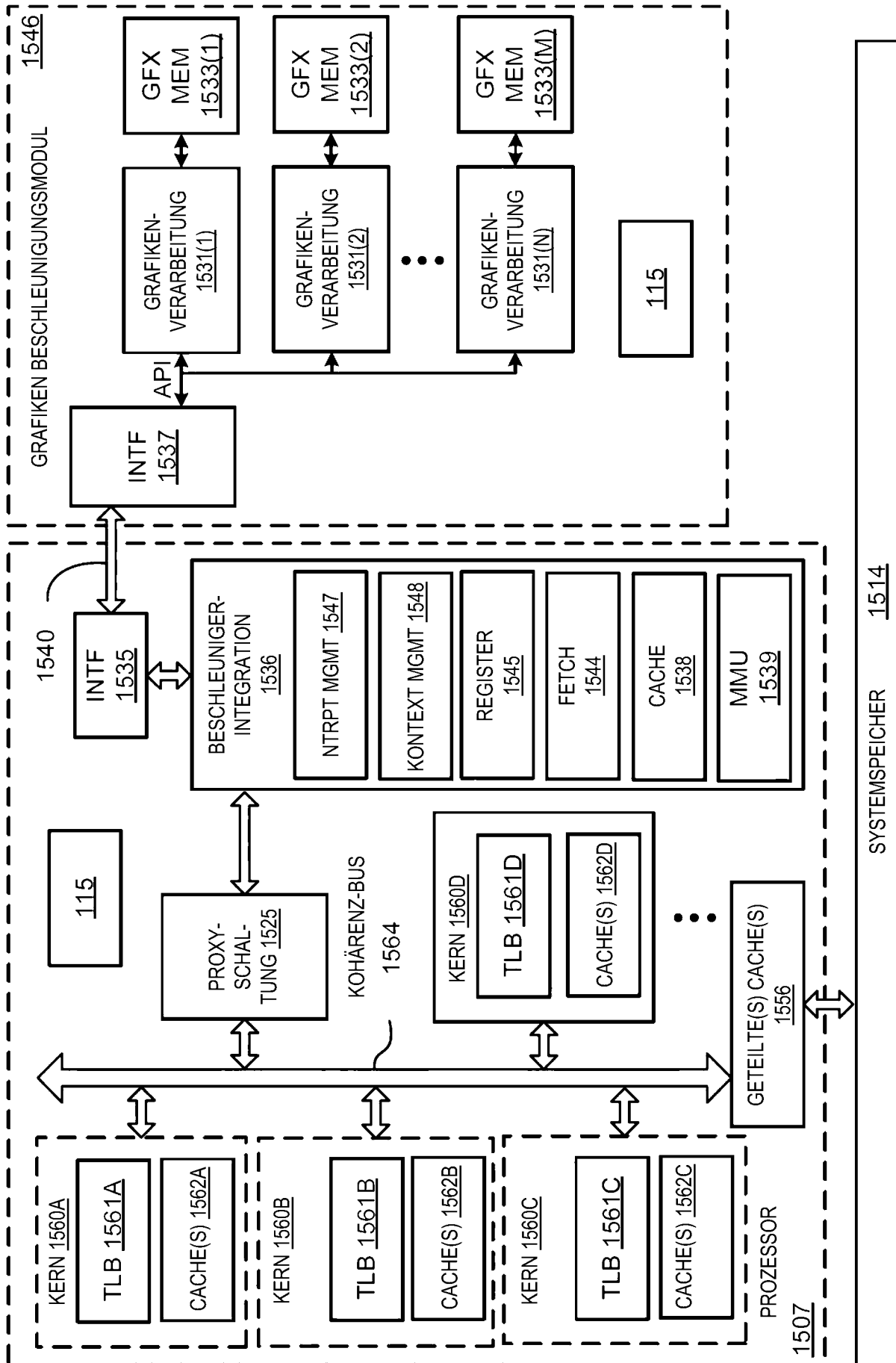
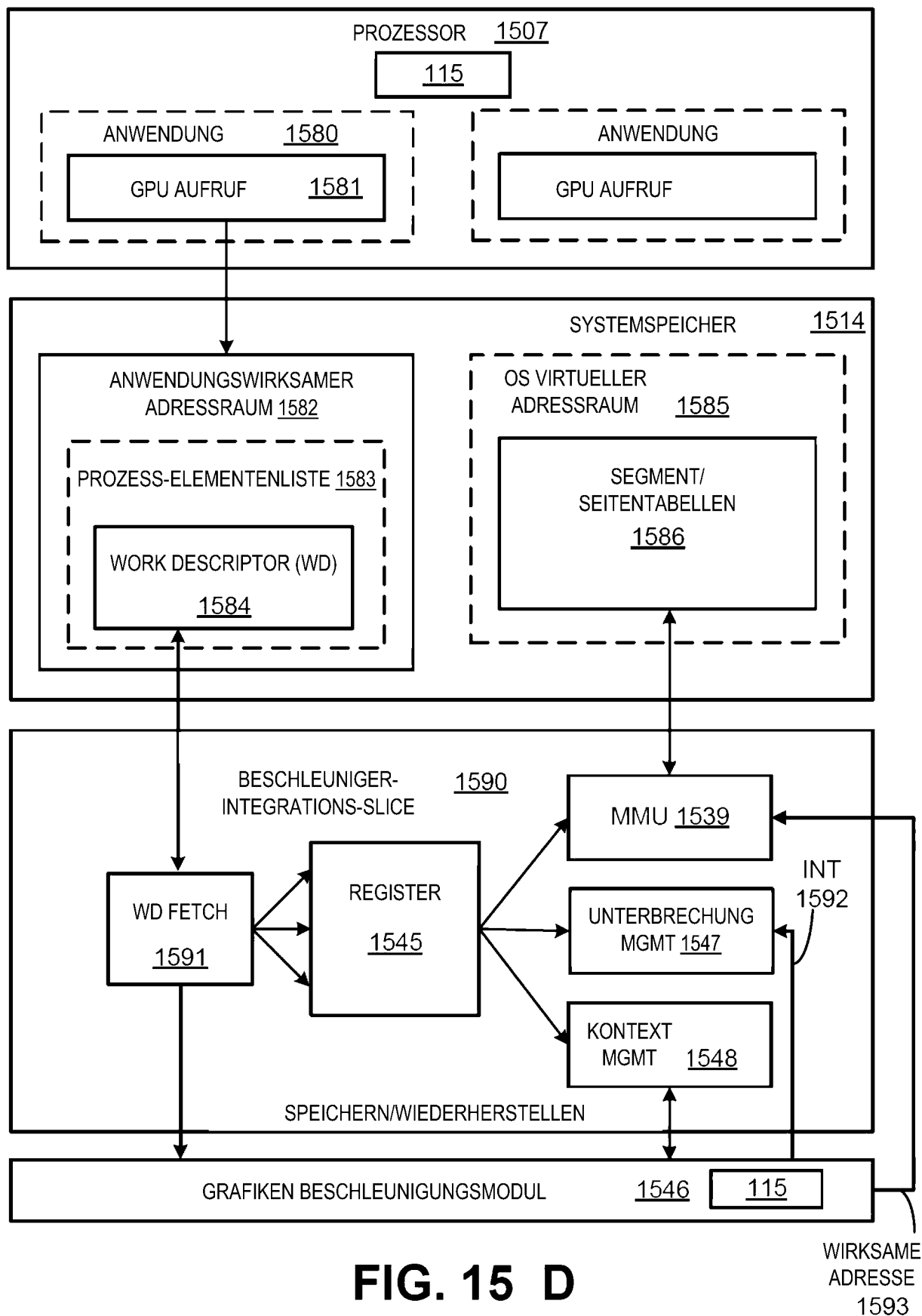
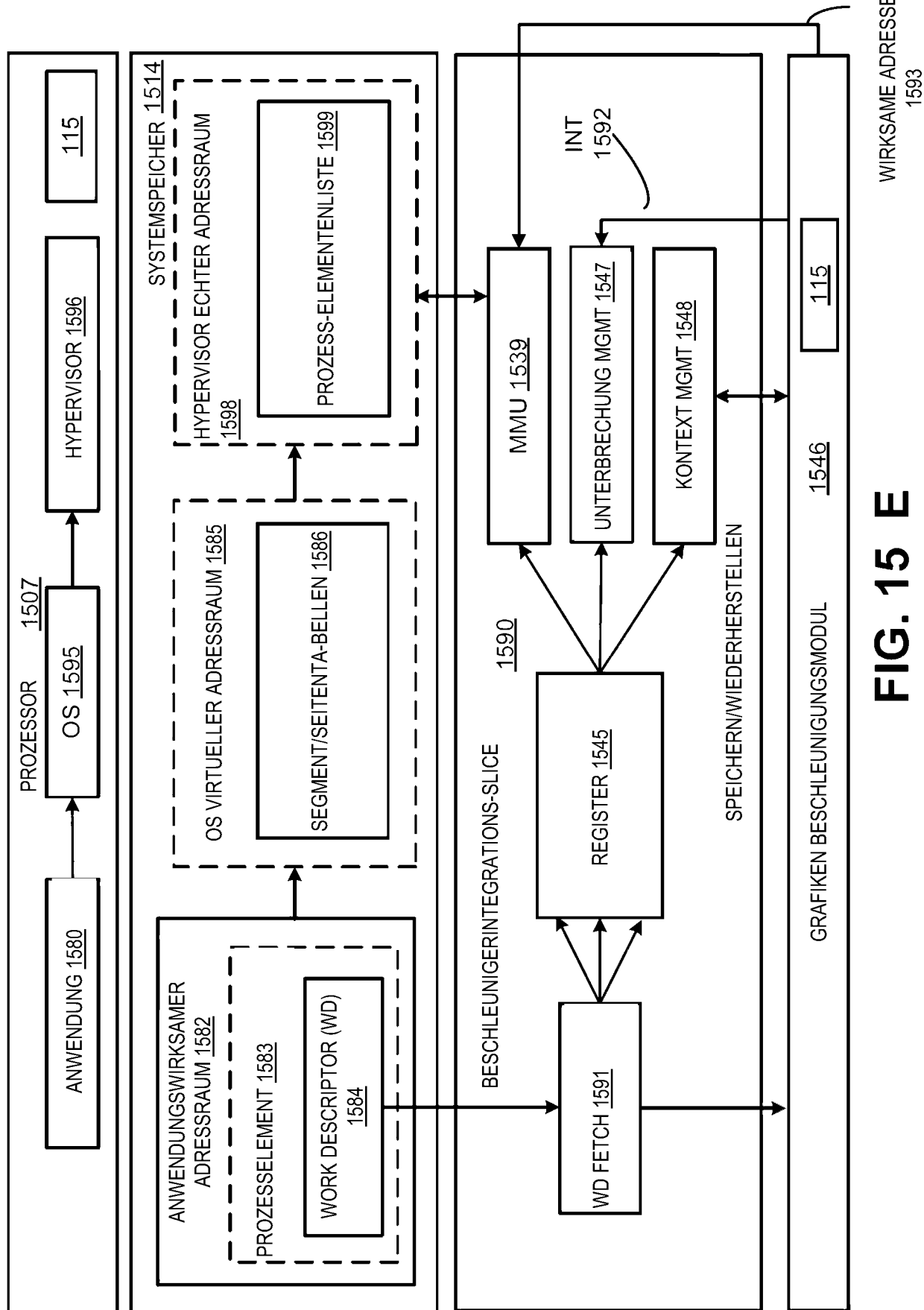


FIG. 15 C

**FIG. 15 D**



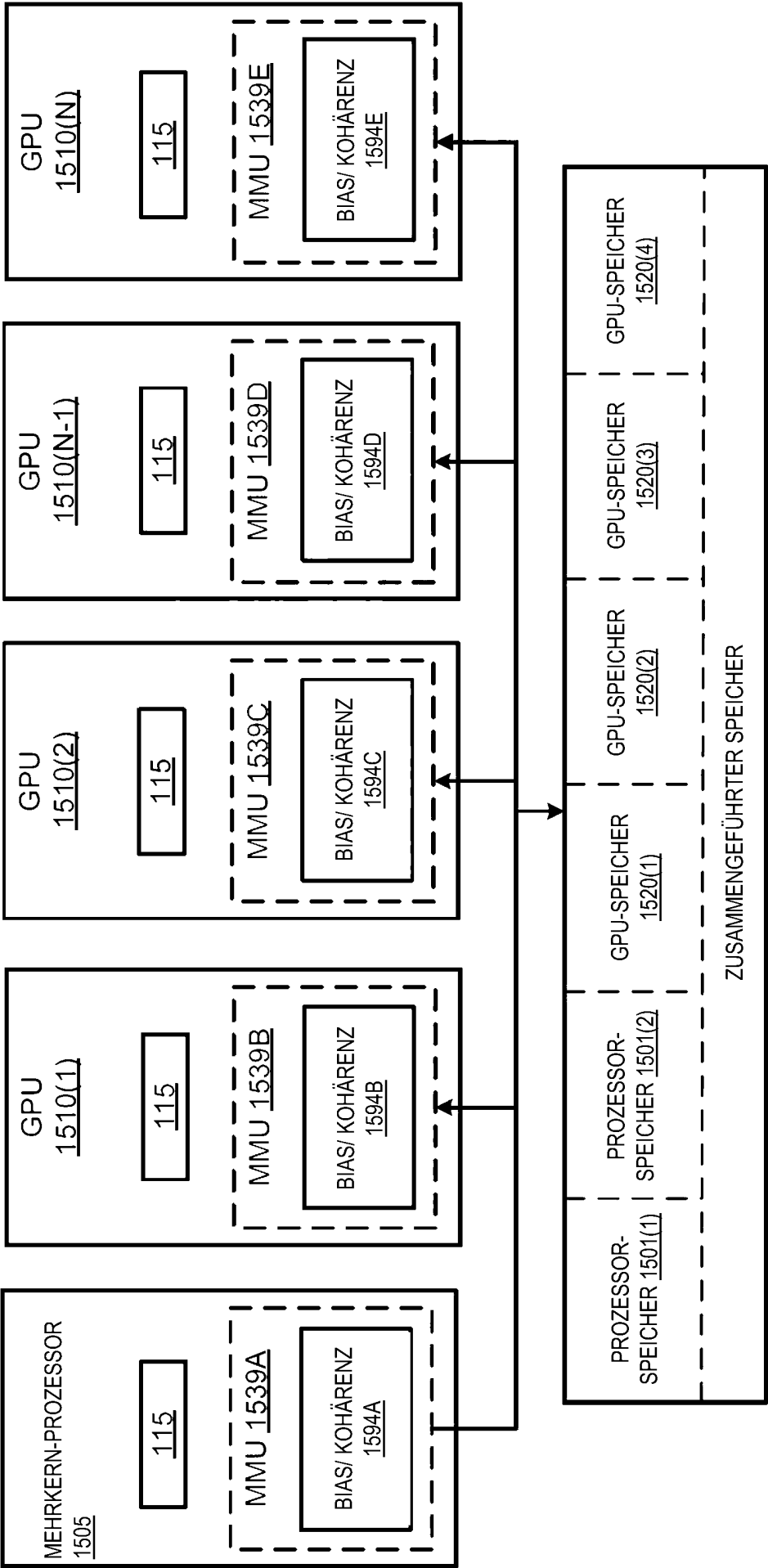


FIG. 15 F

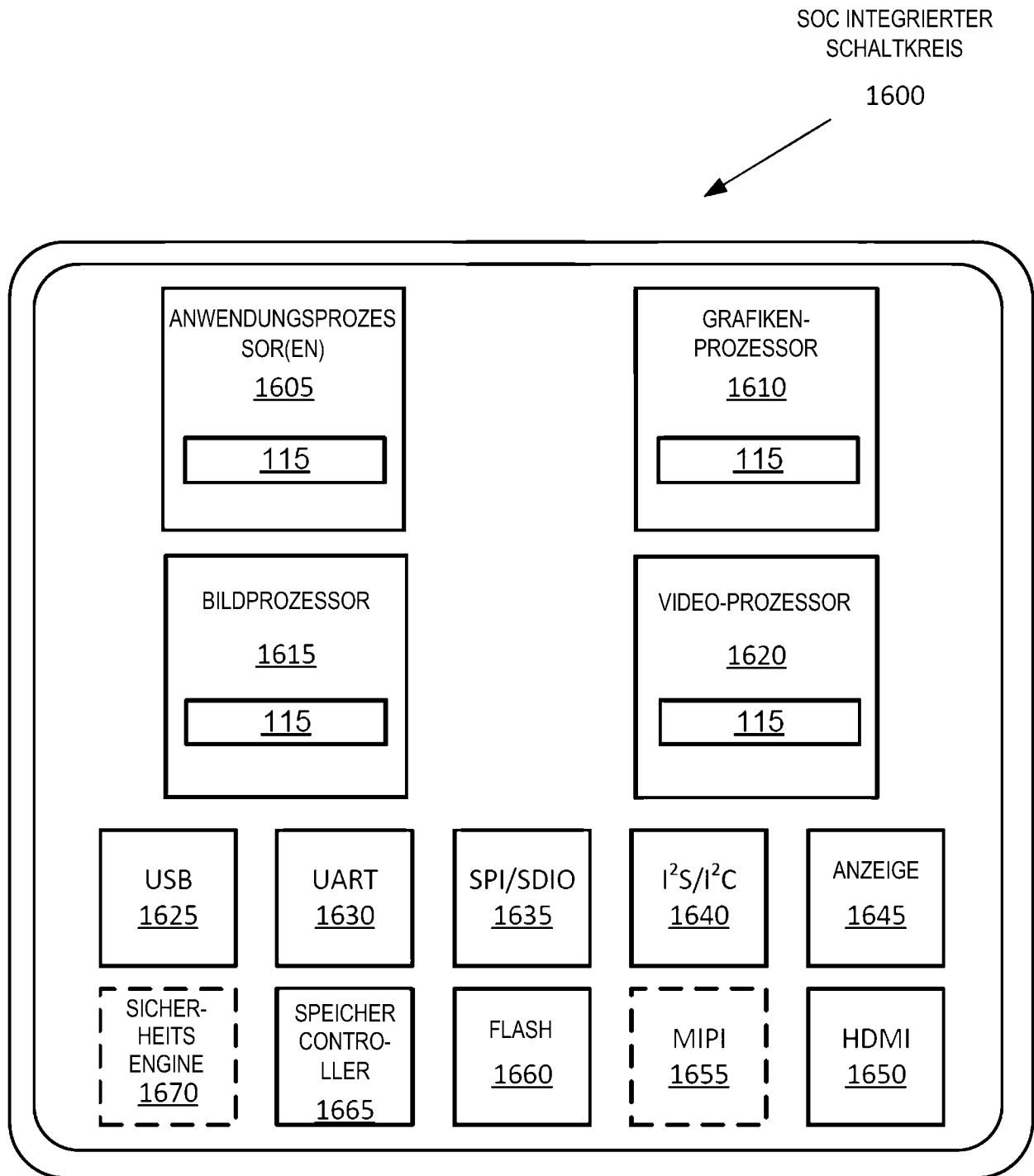


FIG. 16

GRAFIKEN-
PROZESSOR
1710

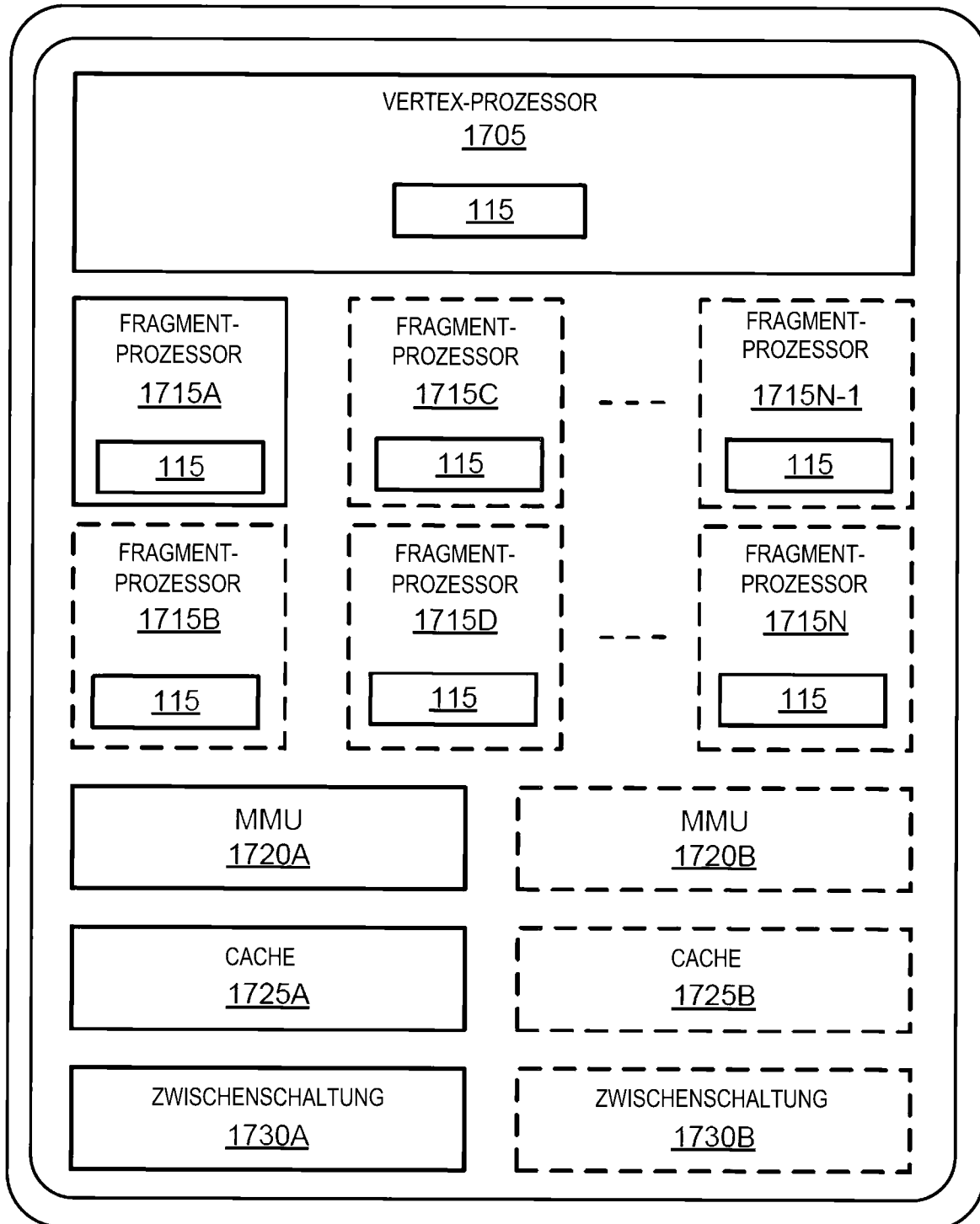
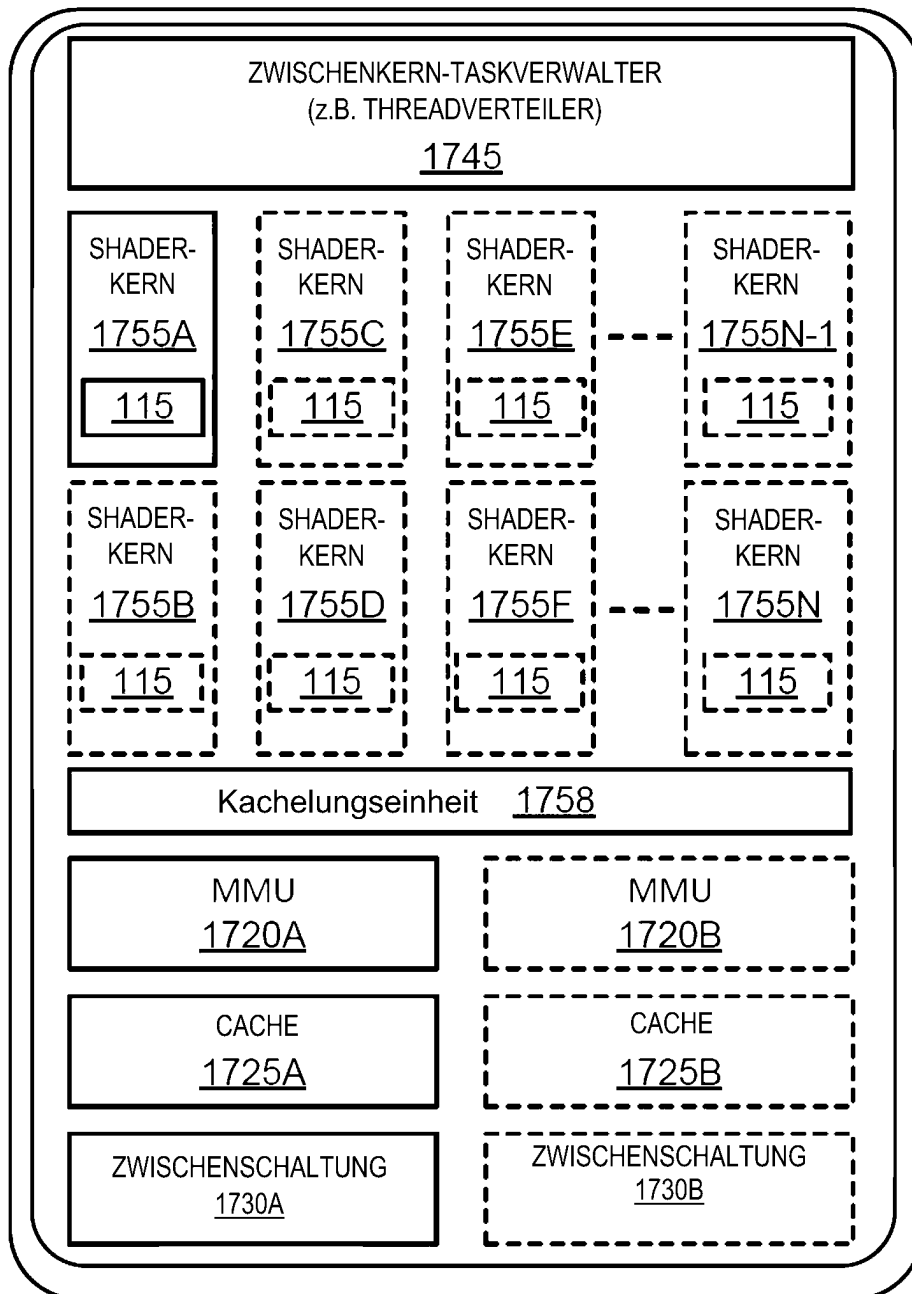


FIG. 17A

GRAFIKEN-PROZESSOR

1740

**FIG. 17B**

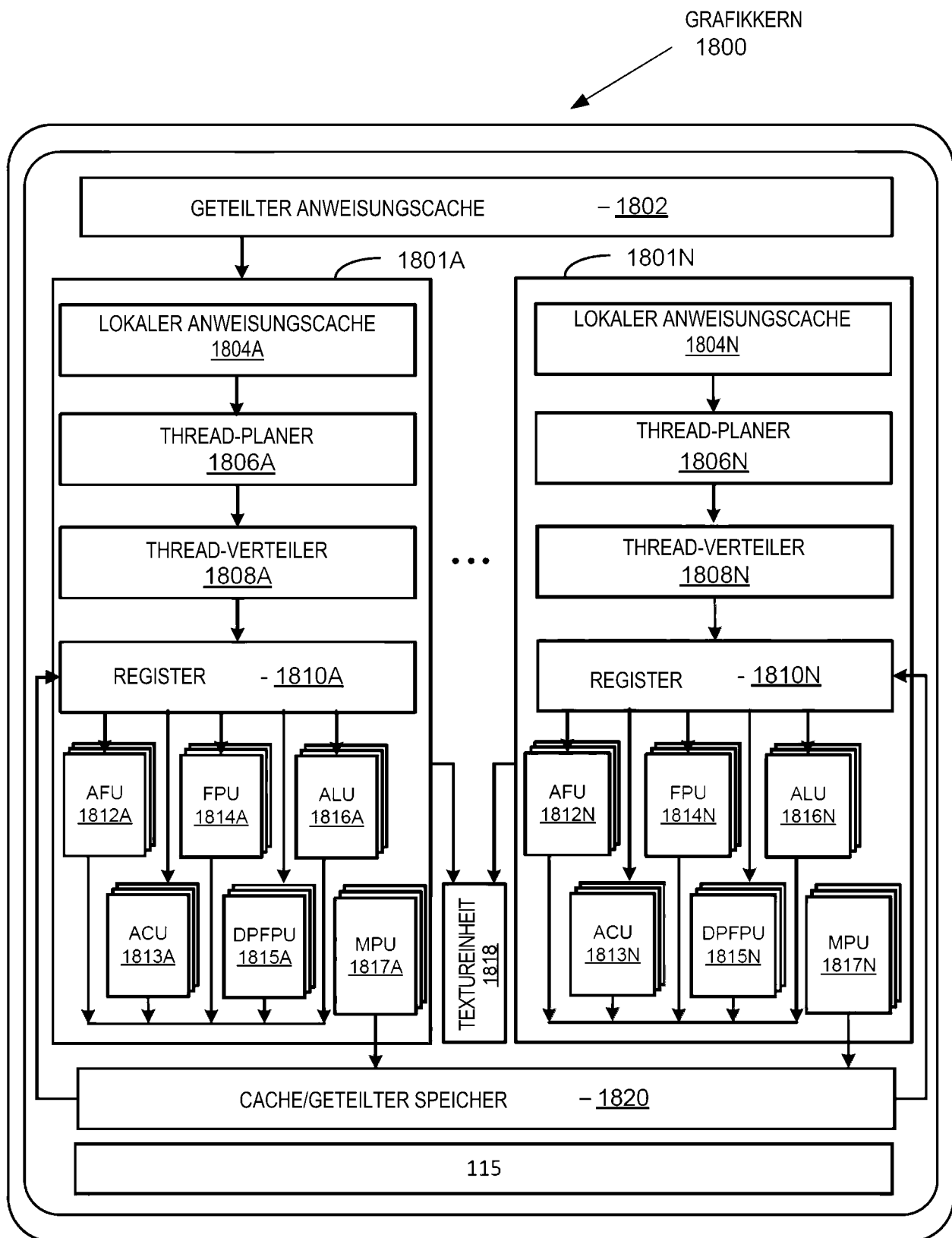


FIG. 18A

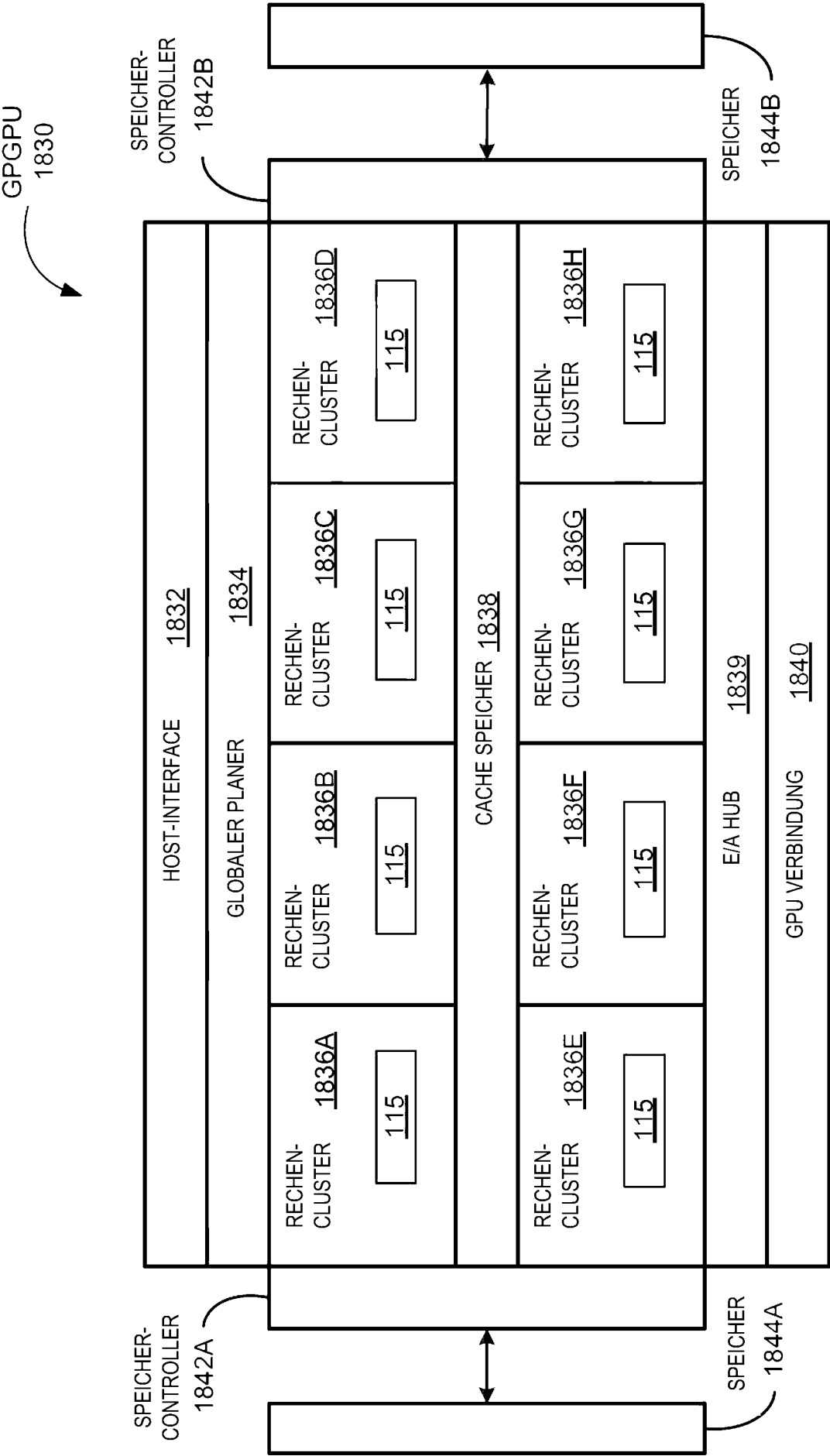


FIG. 18B

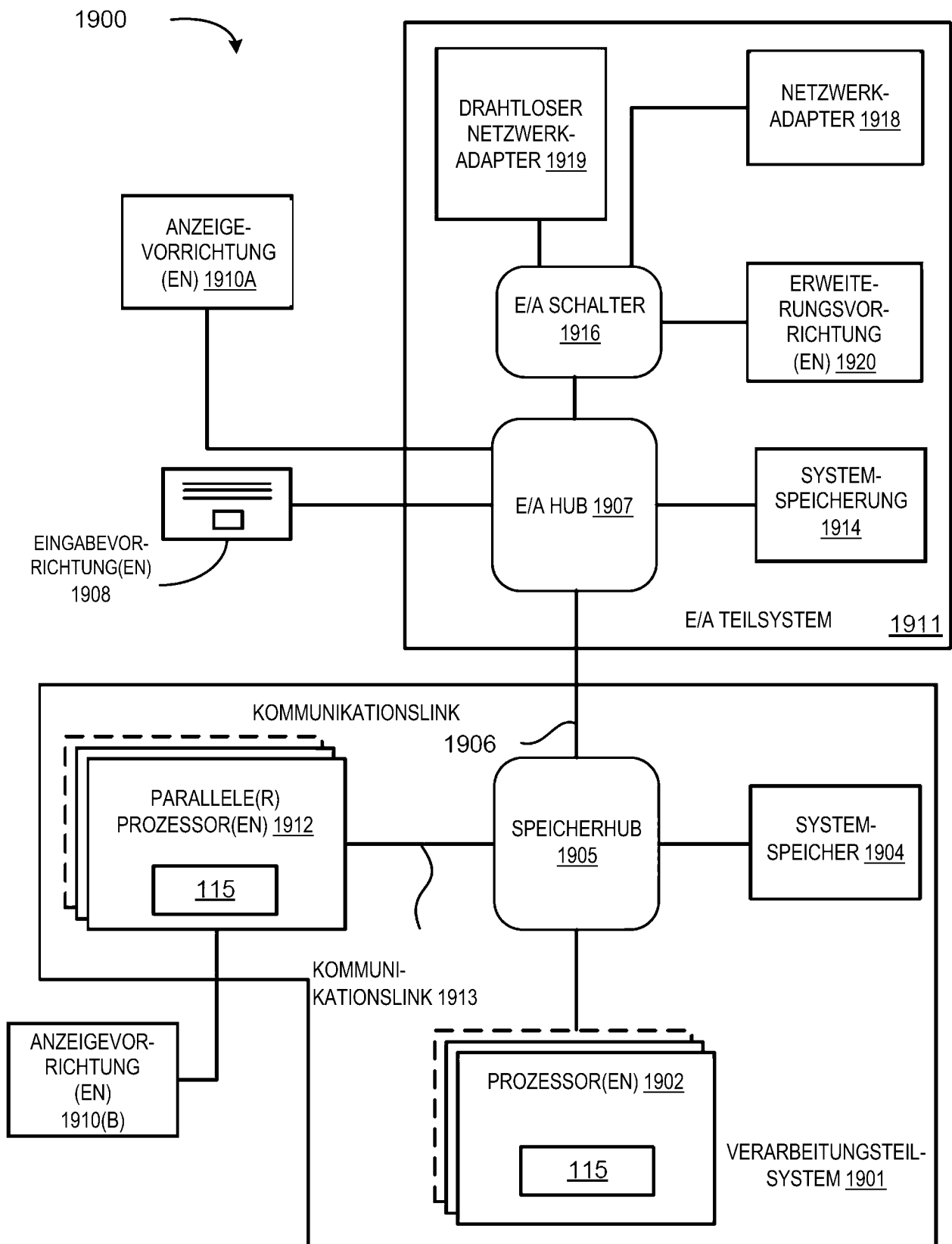
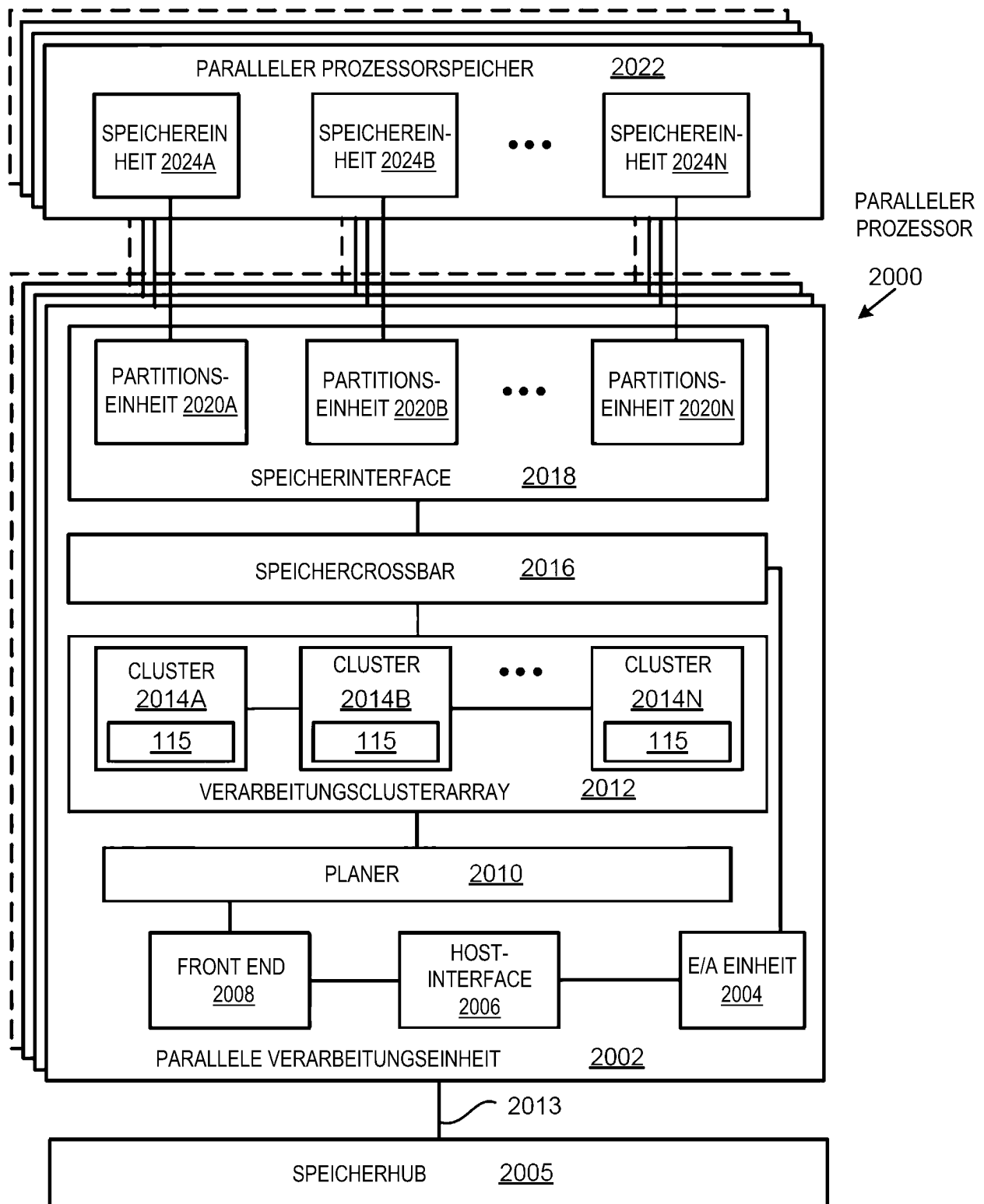


FIG. 19

**FIG. 20 A**

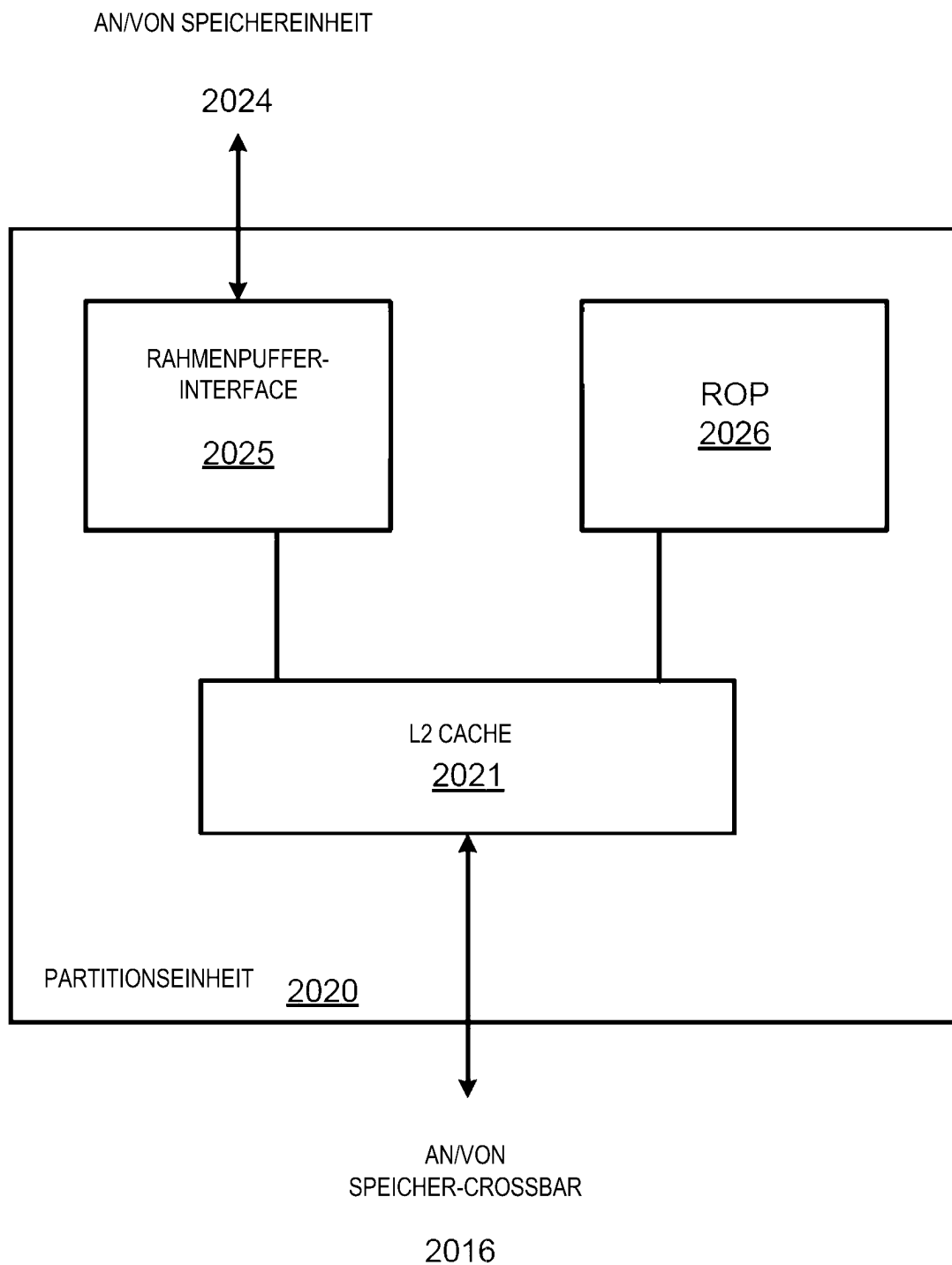


FIG. 20 B

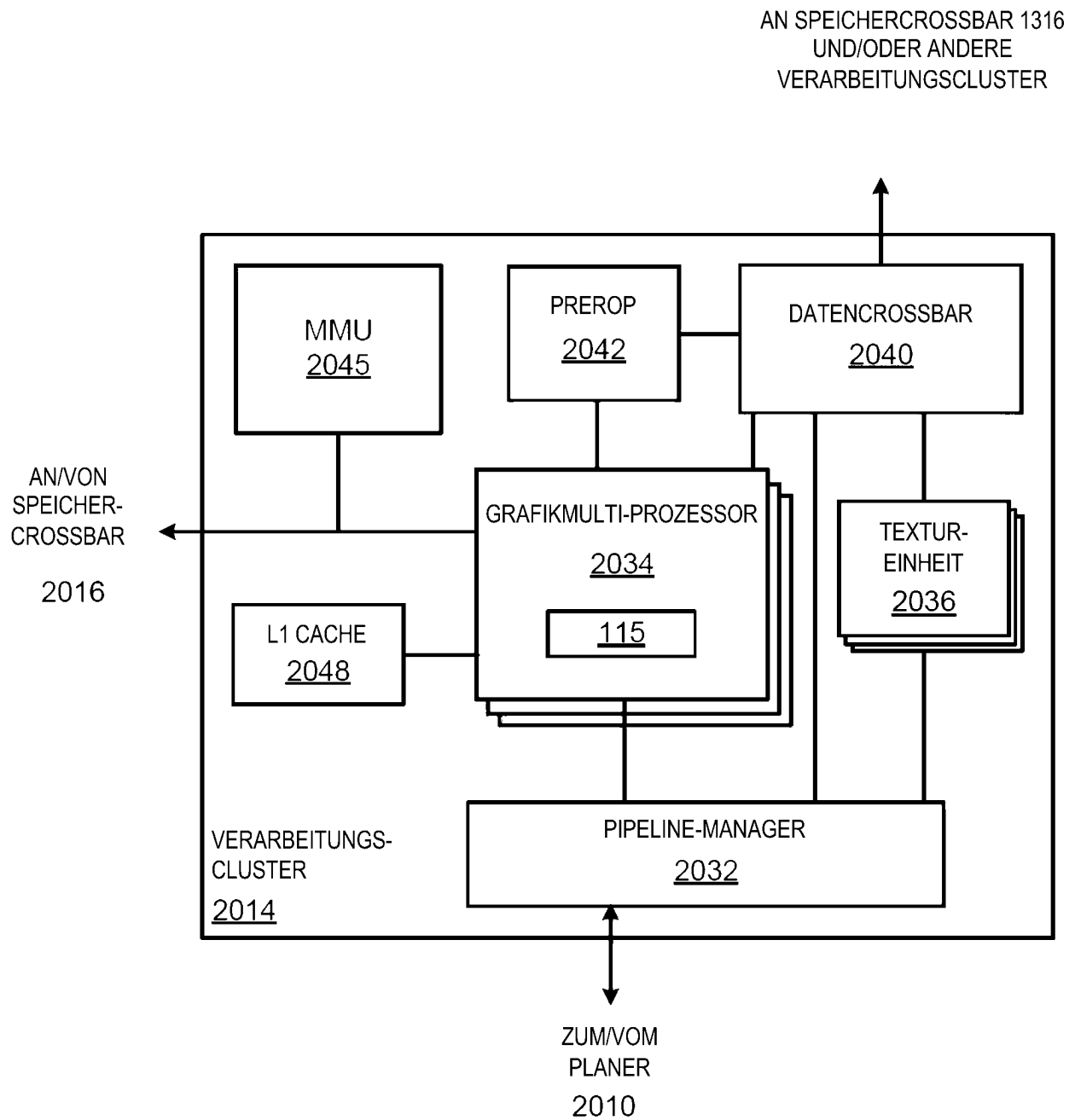
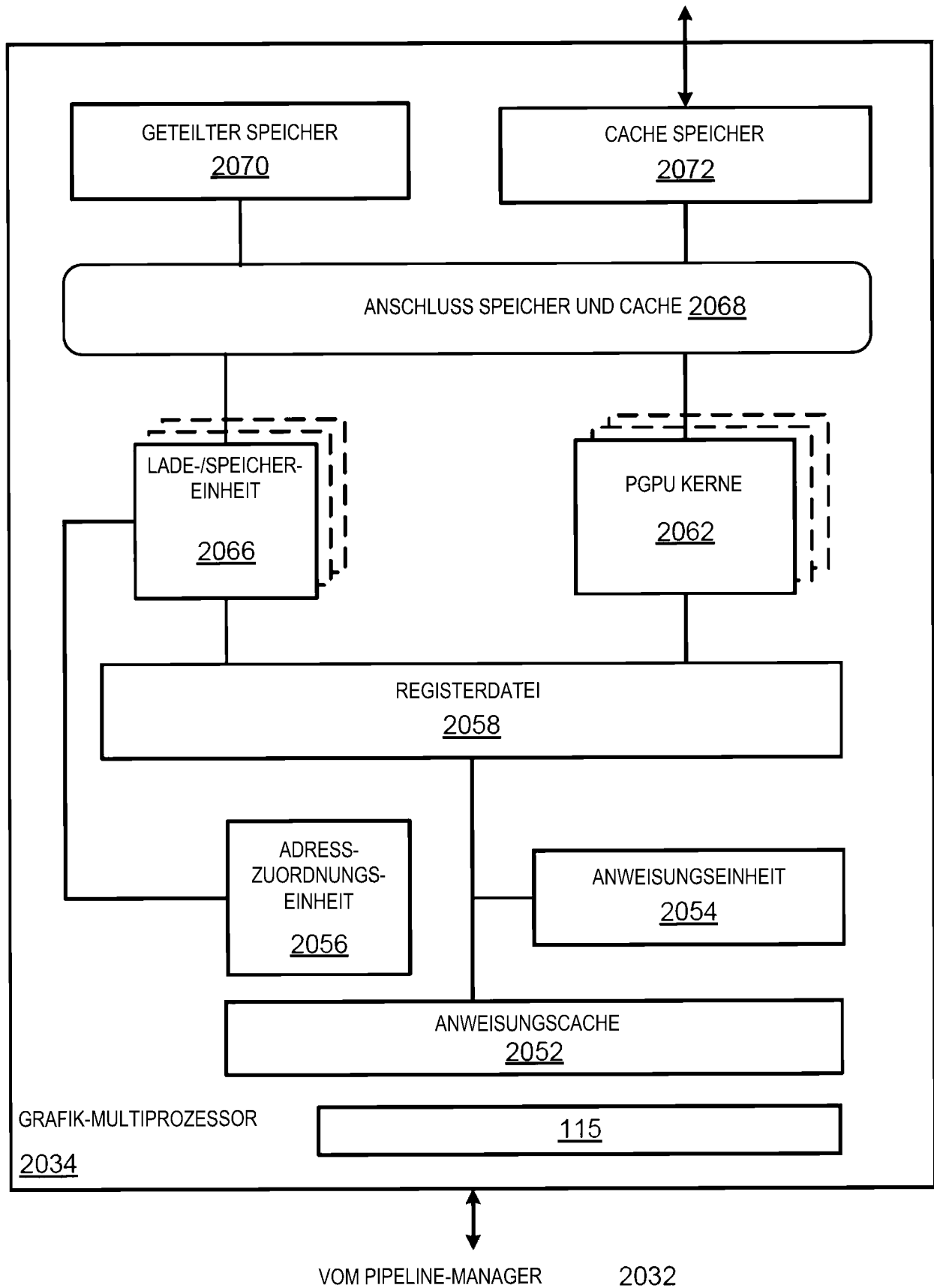


FIG. 20 C

**FIG. 20 D**

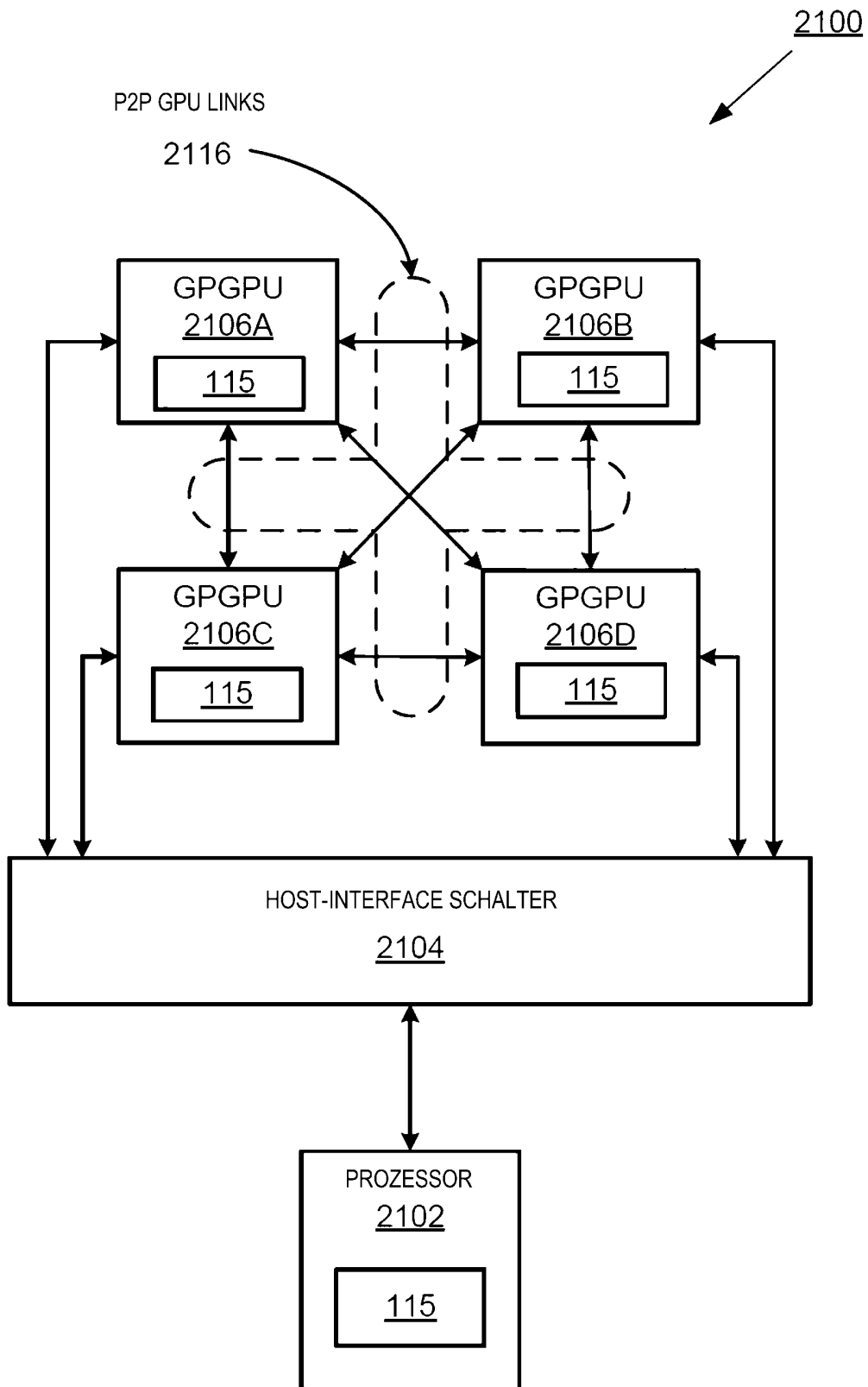


FIG. 21

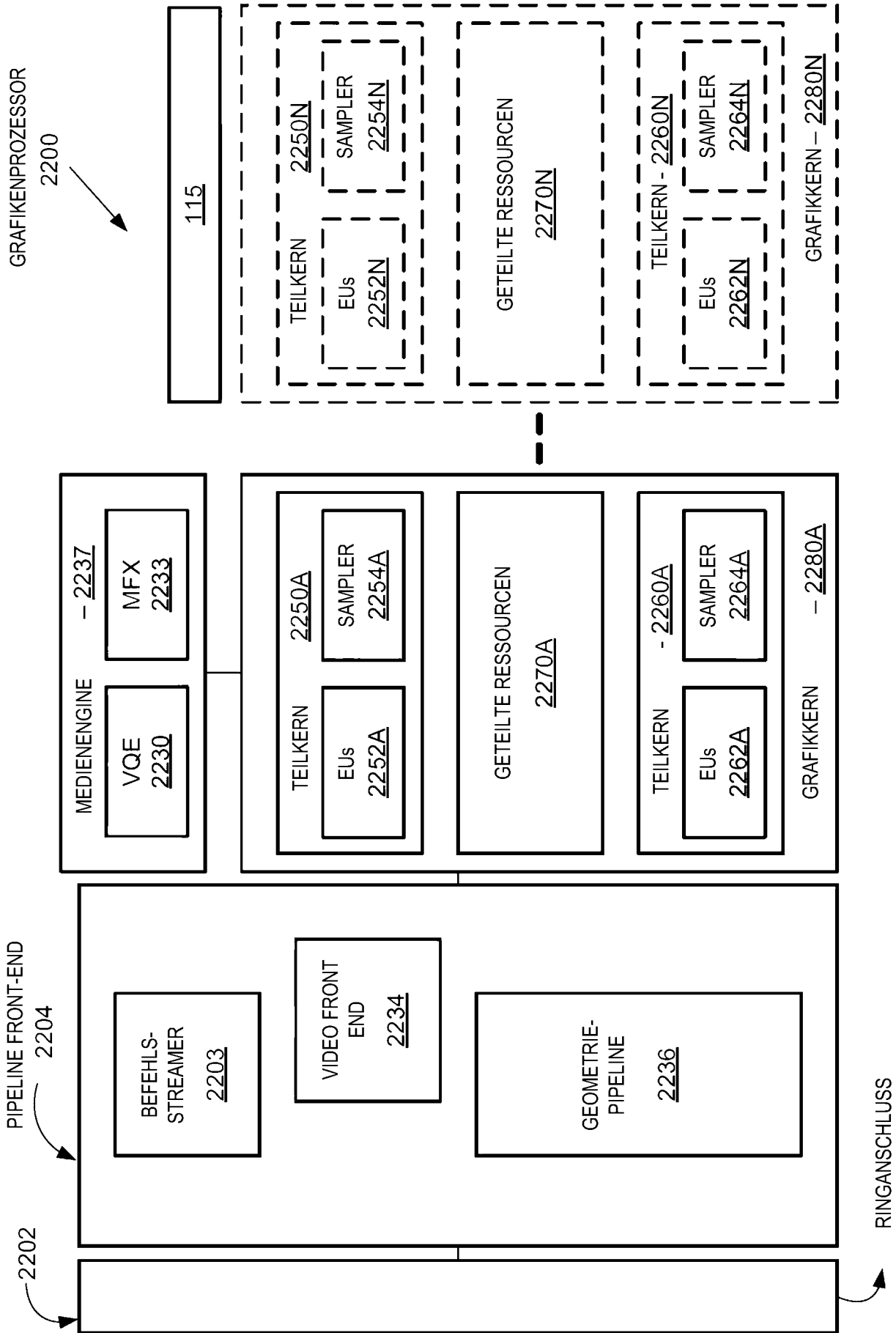
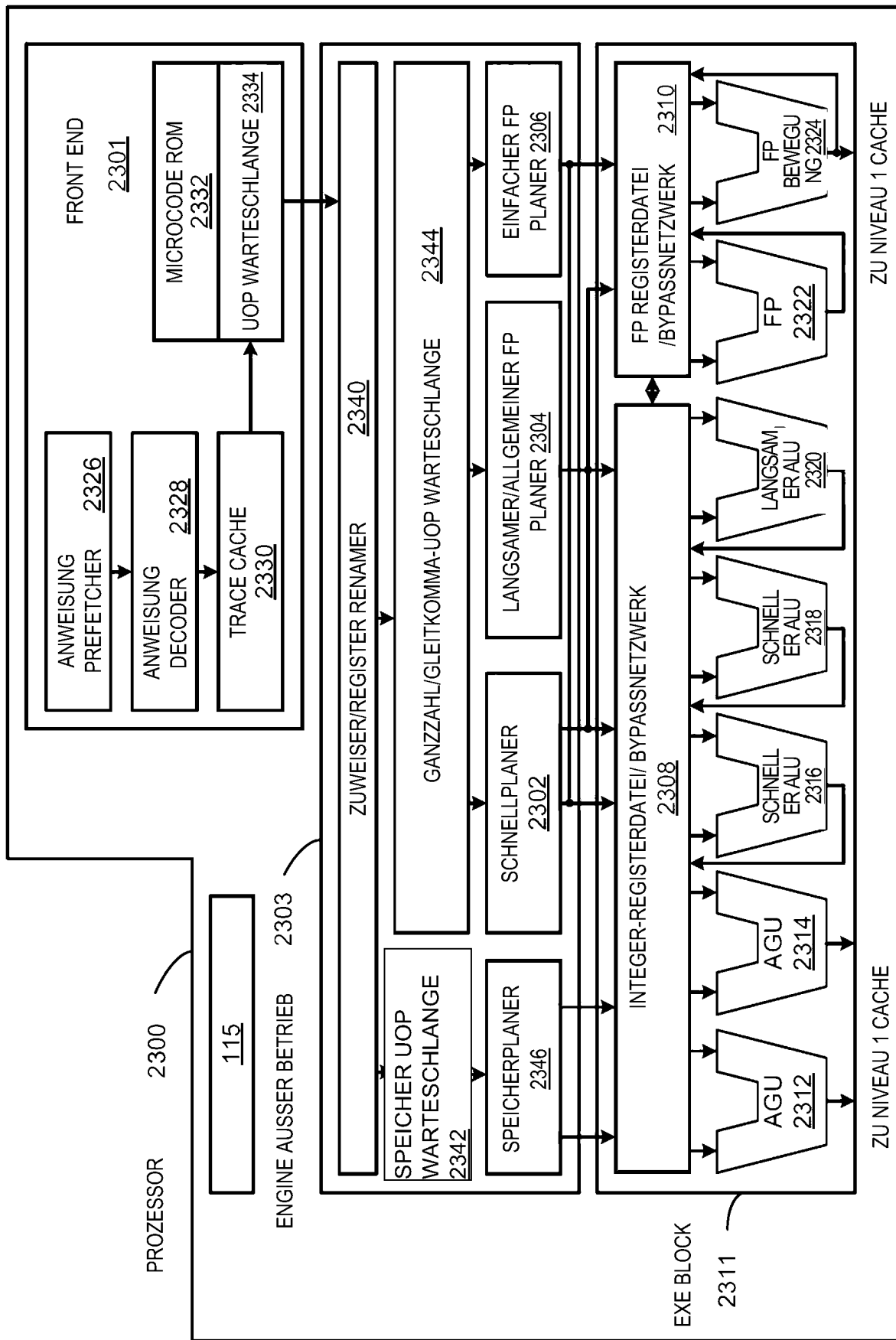


FIG. 22

**FIG. 23**

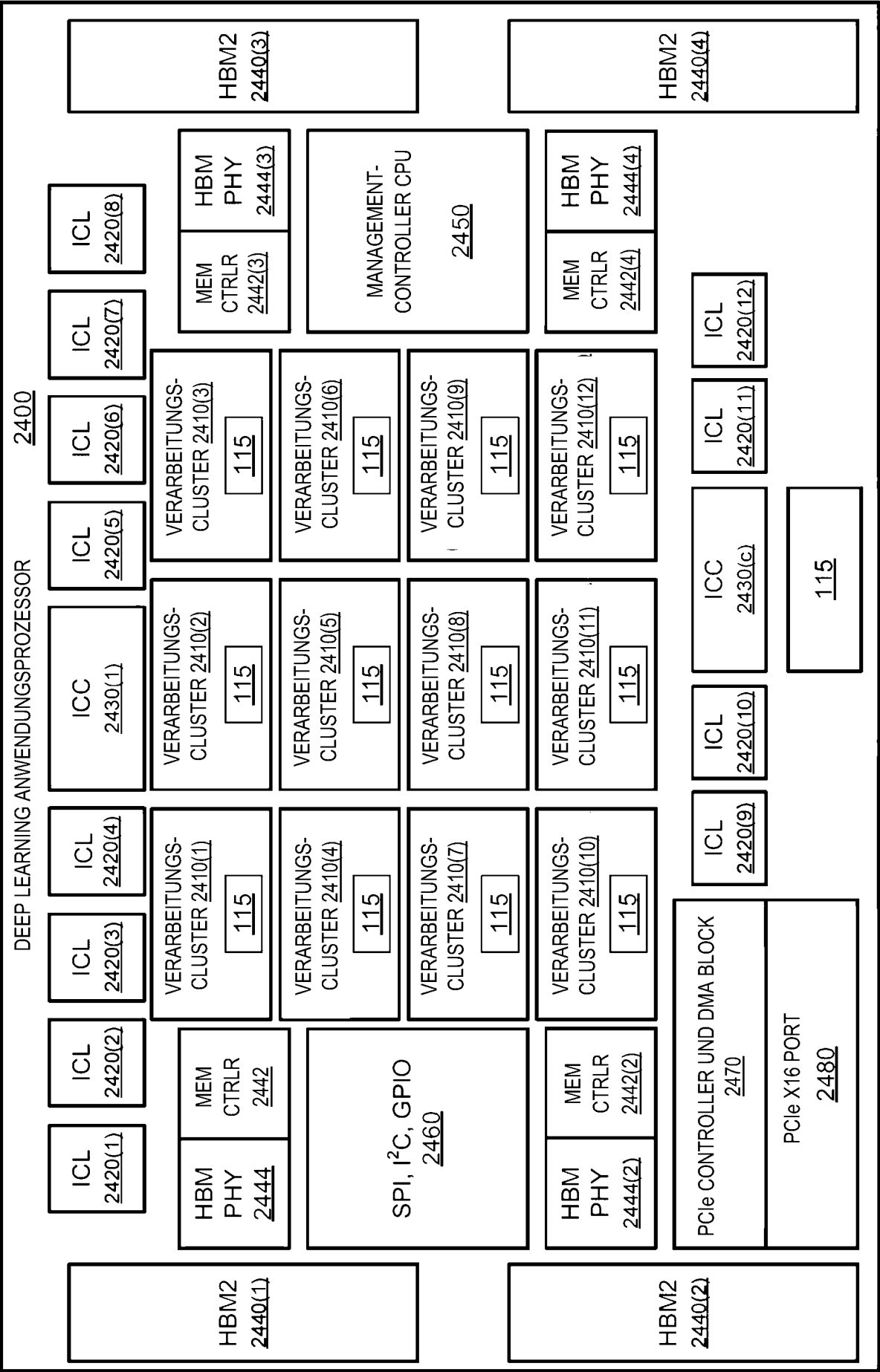


FIG. 24

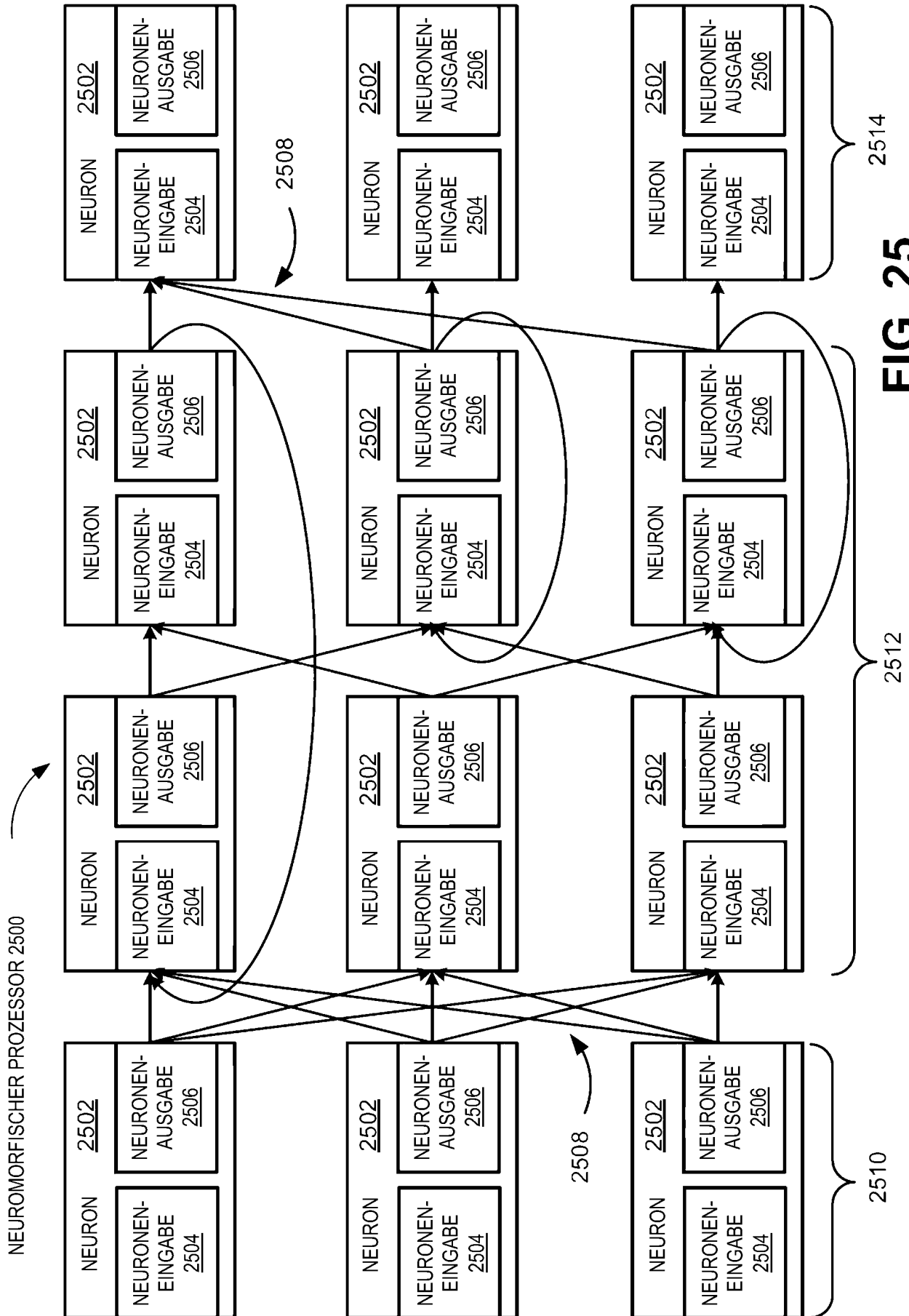


FIG. 25

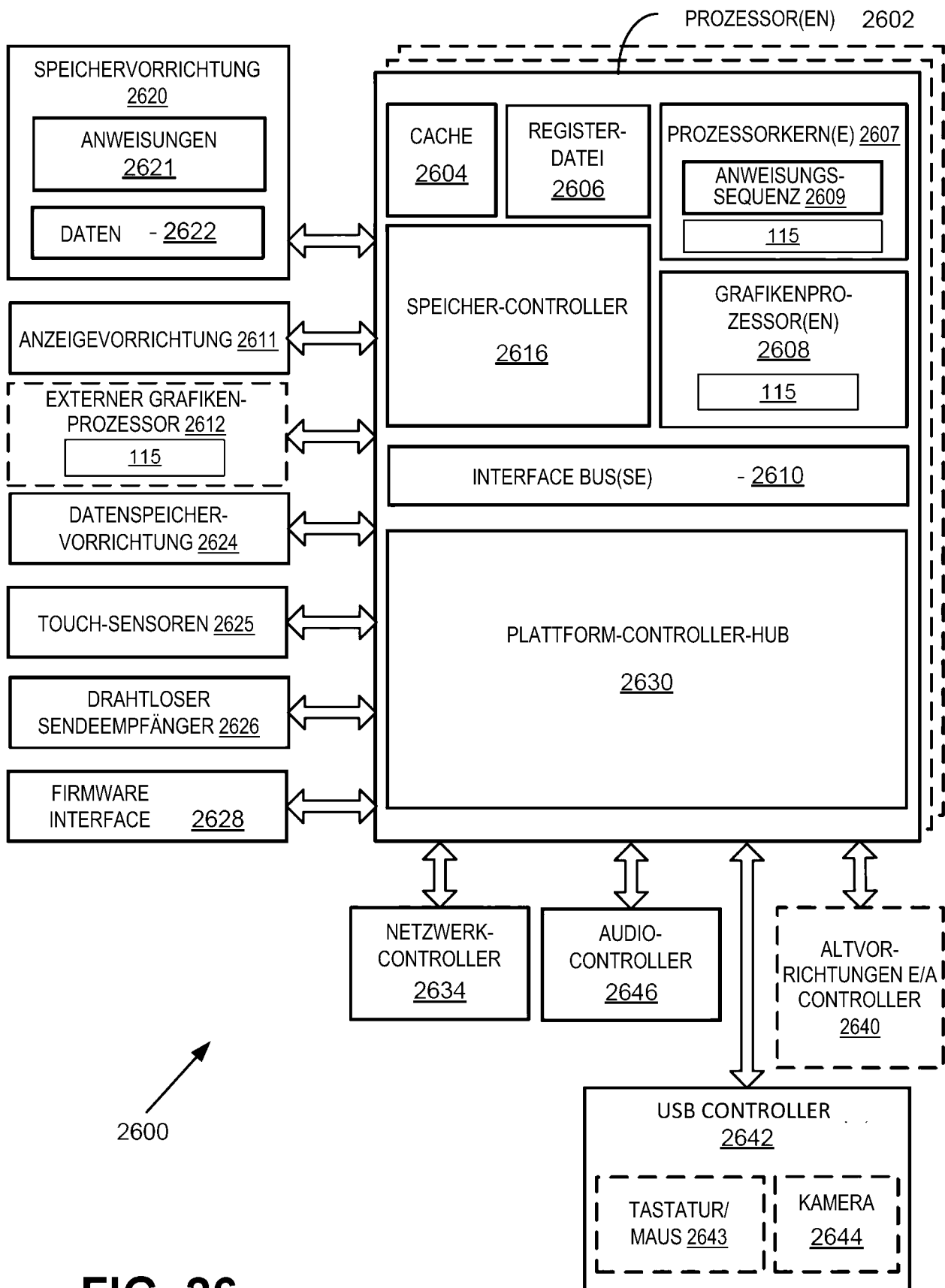


FIG. 26

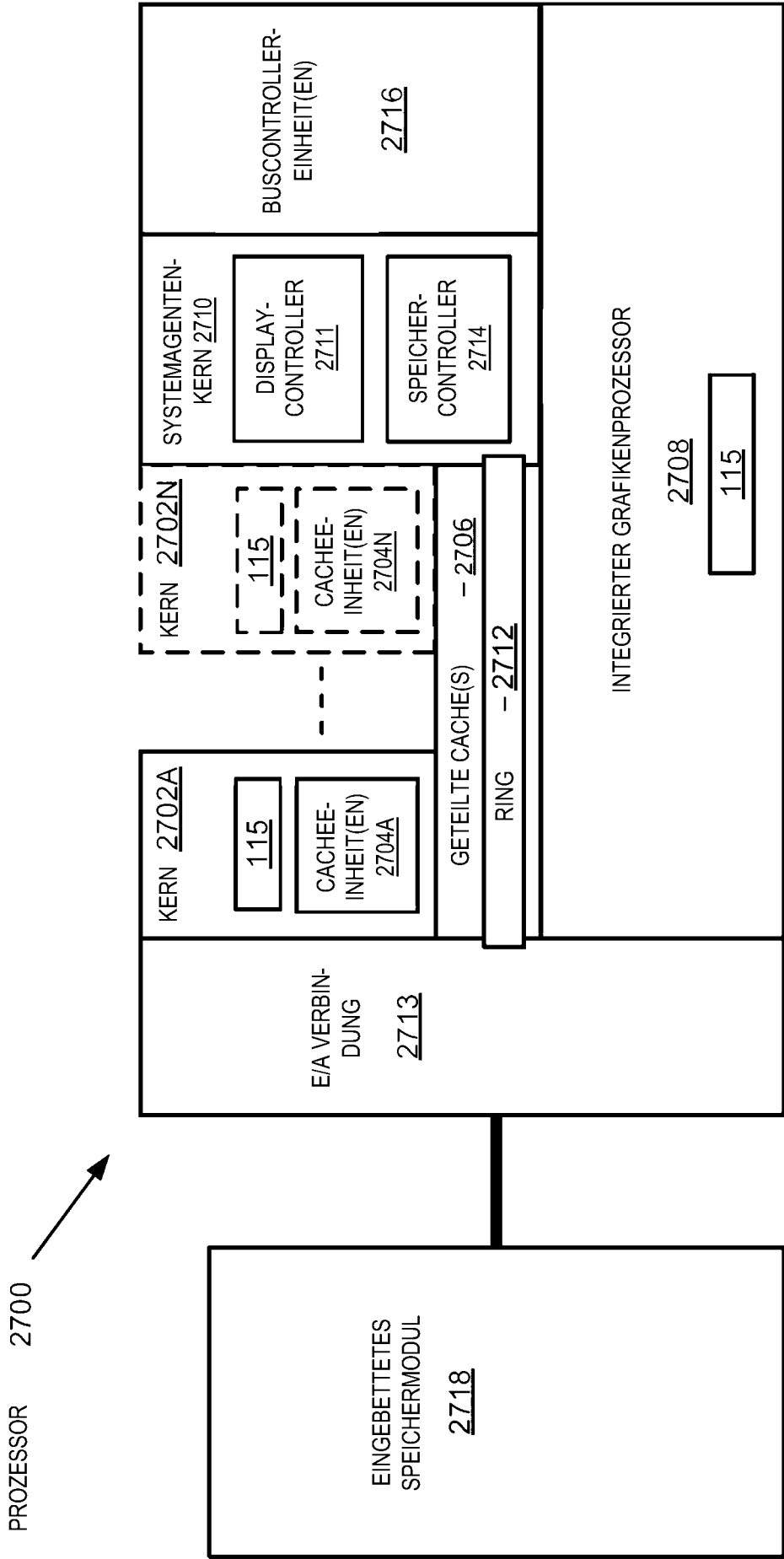


FIG. 27

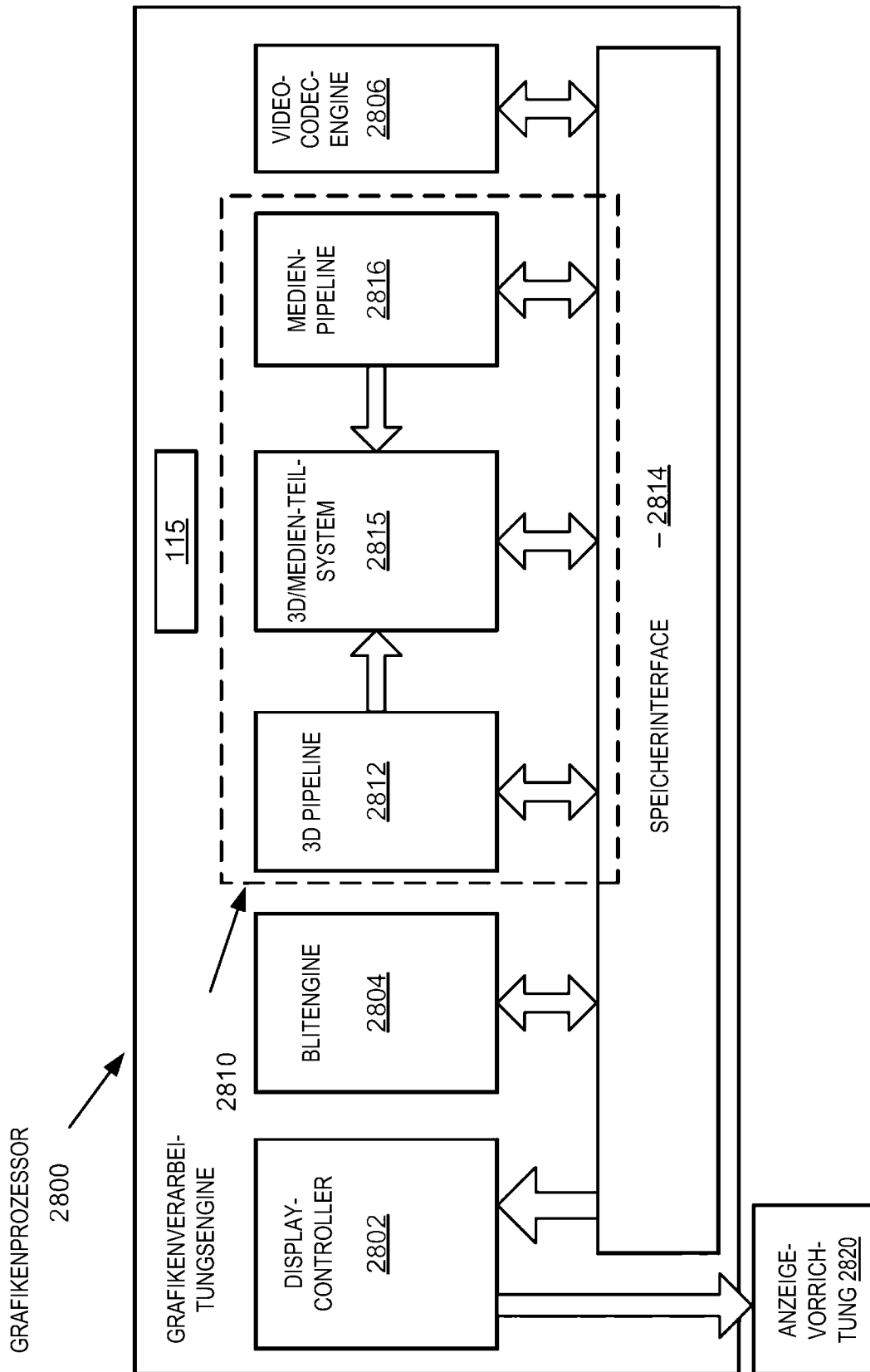


FIG. 28

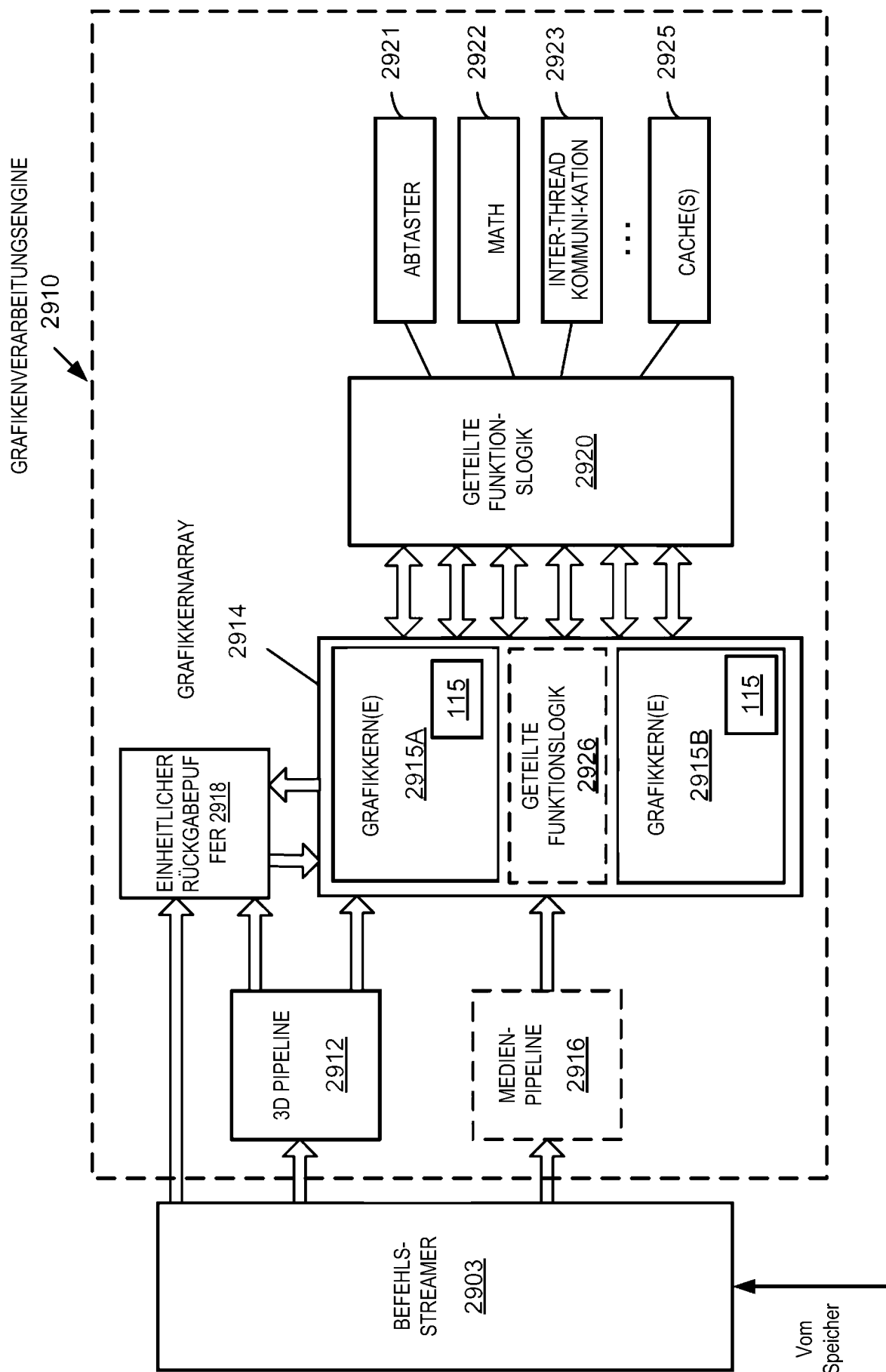
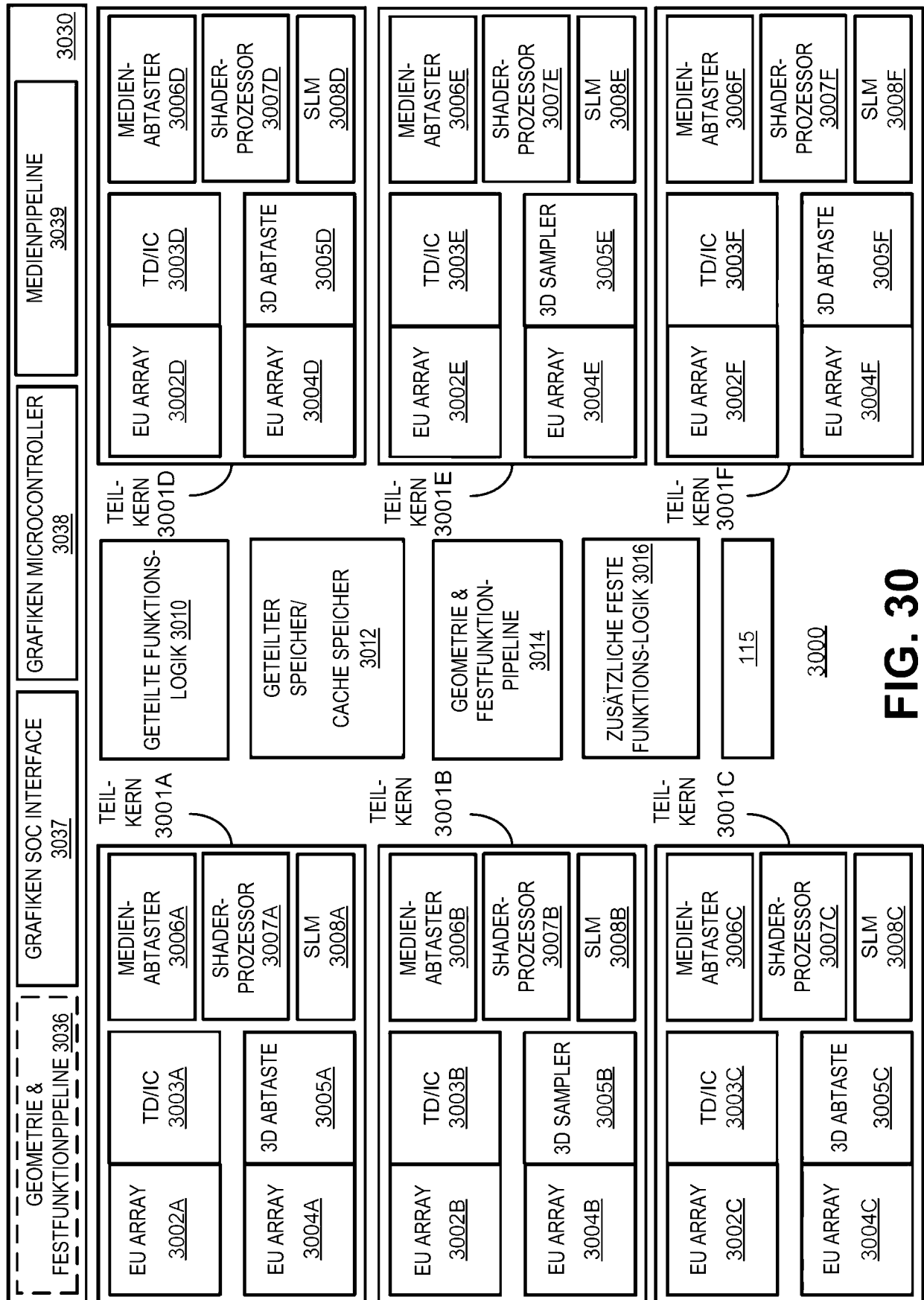


FIG. 29

**FIG. 30**

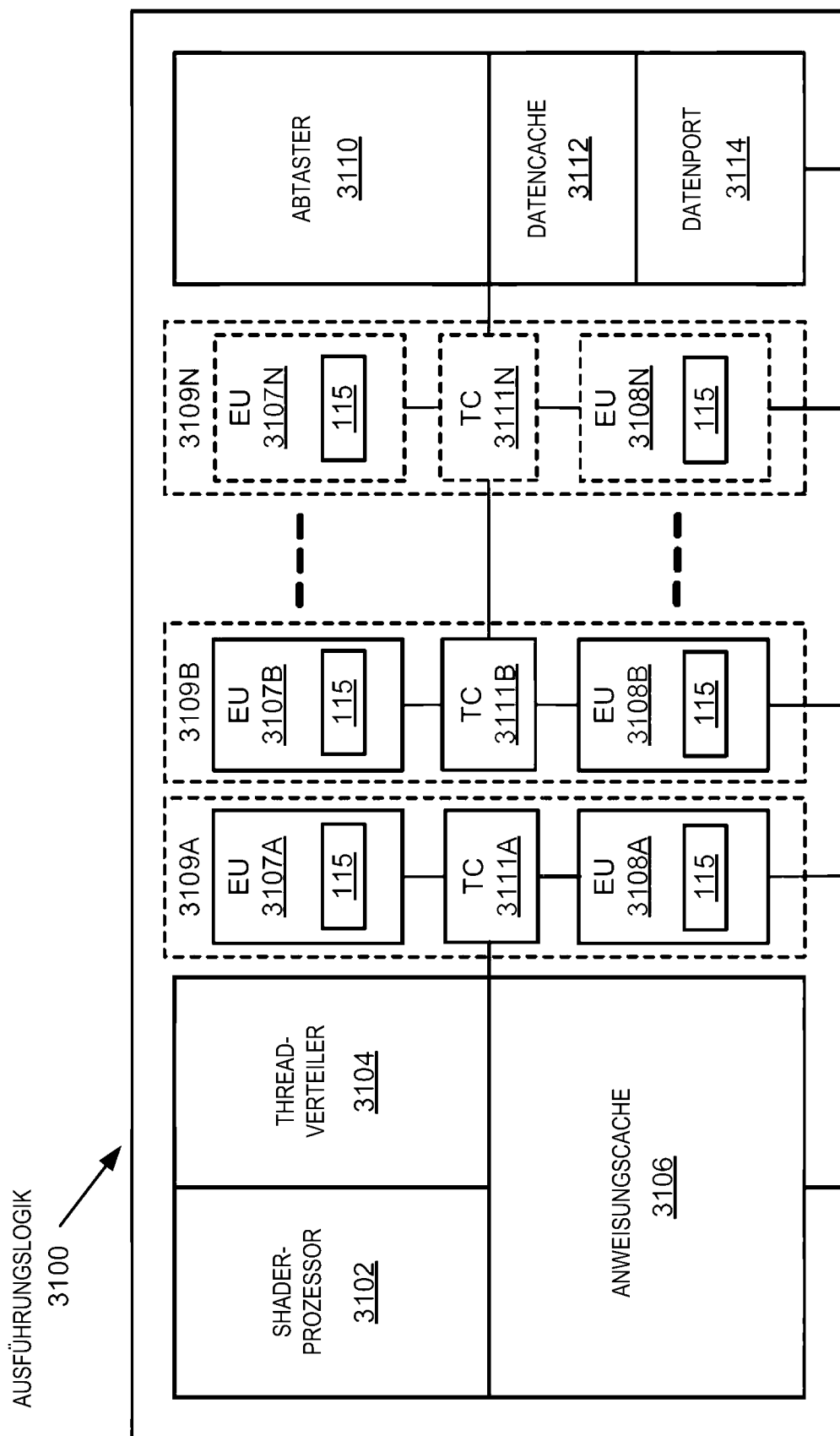


FIG. 31 A

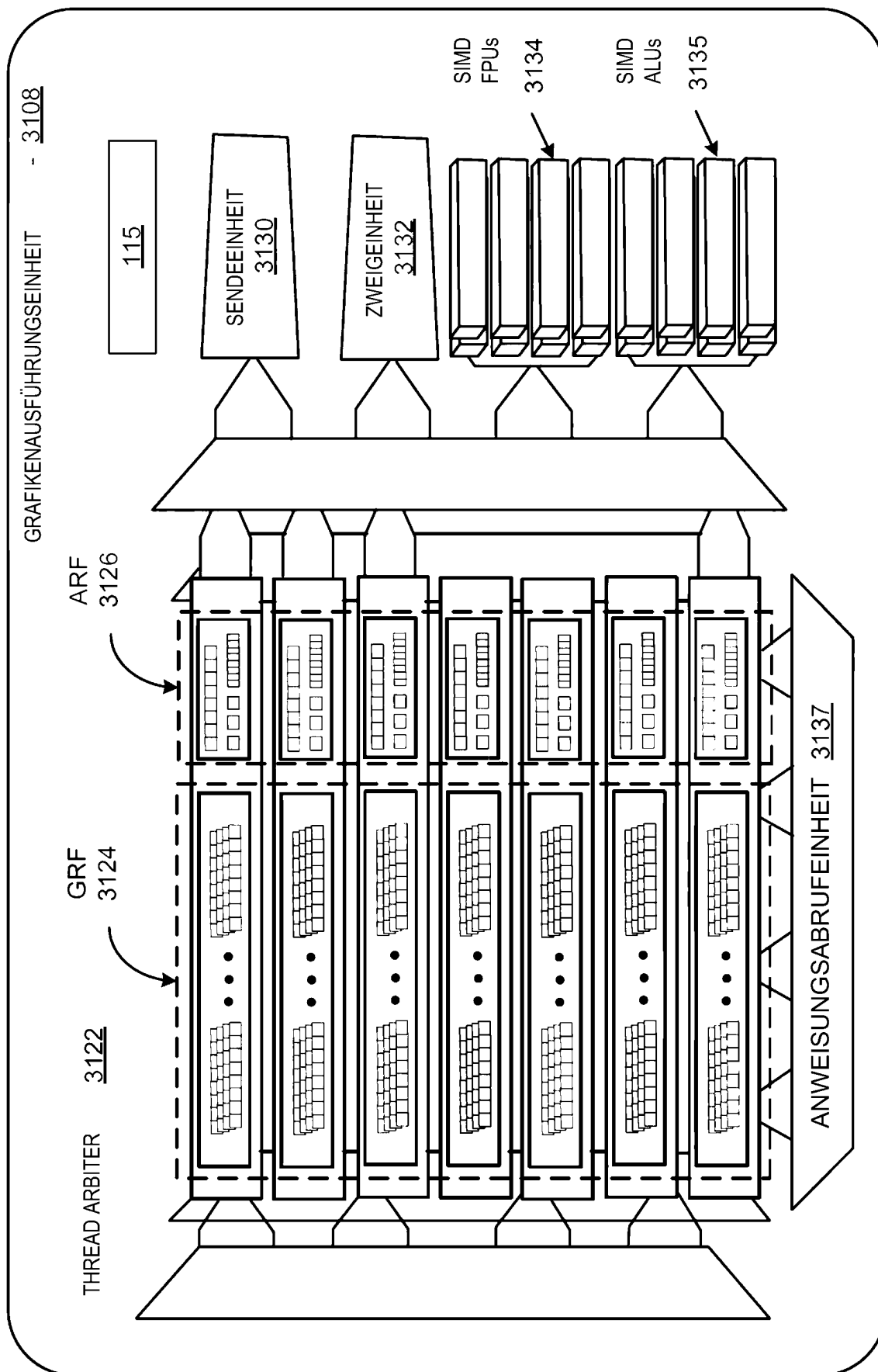
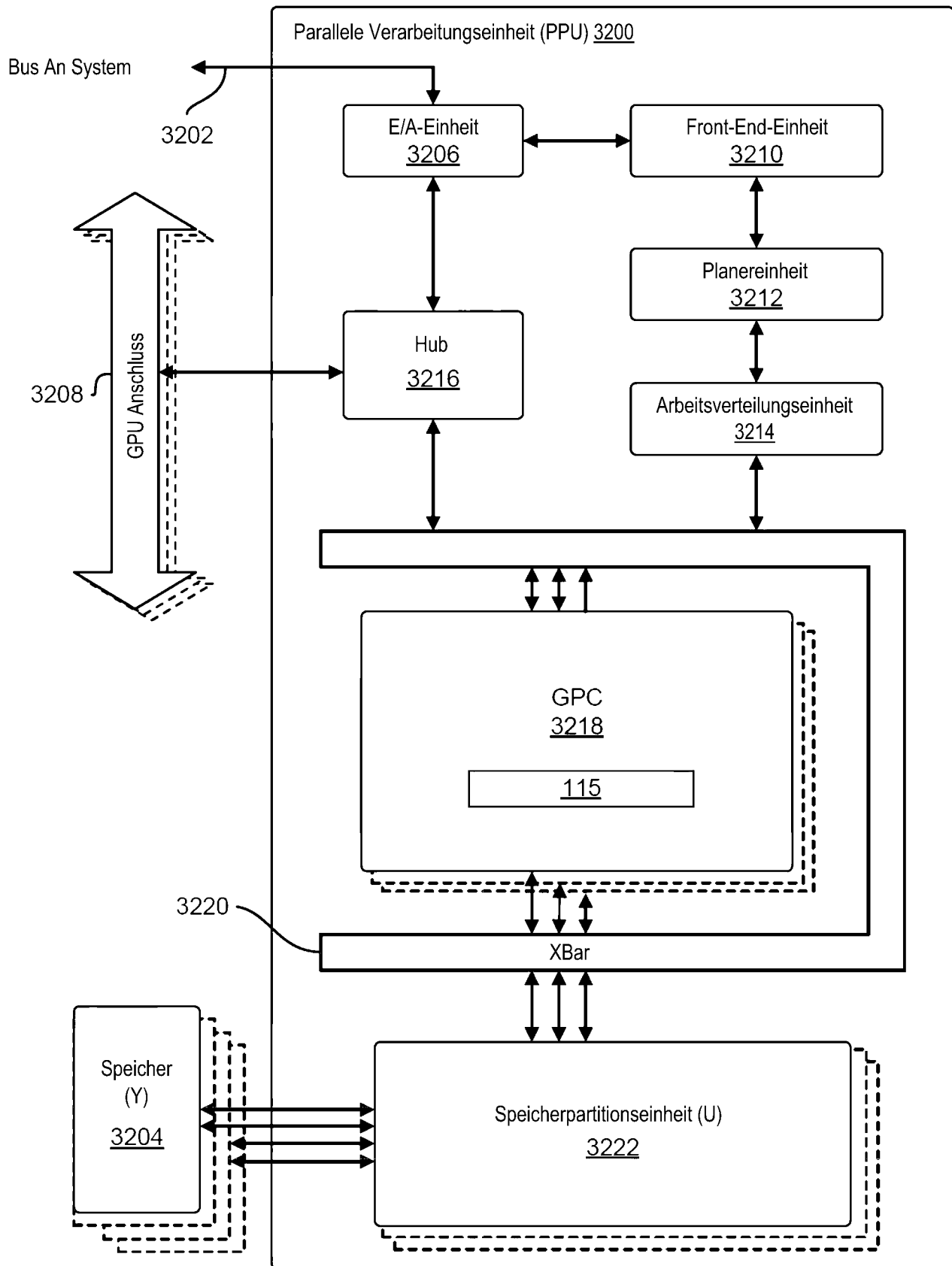
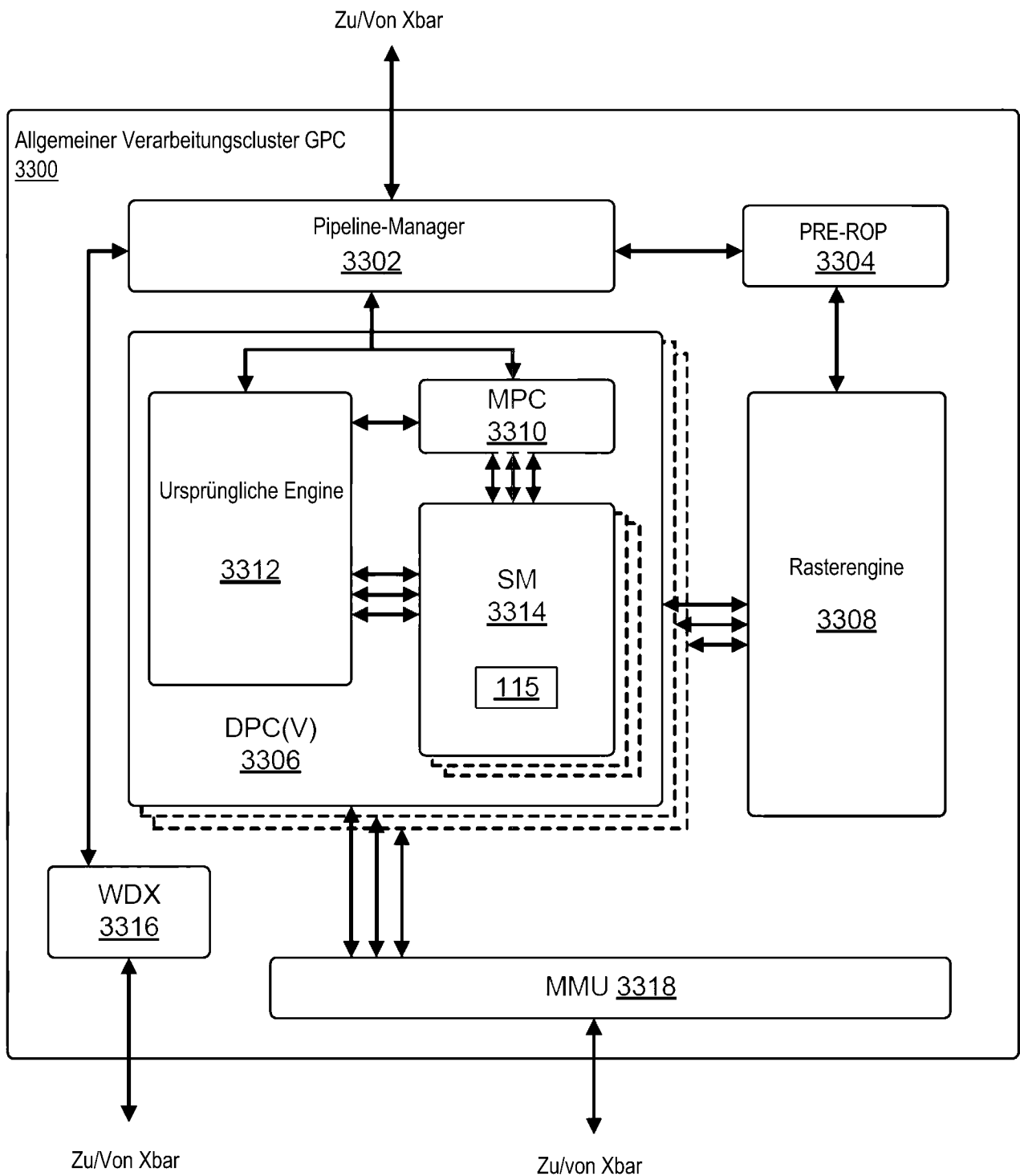


FIG. 31 B

**FIG. 32**

**FIG. 33**

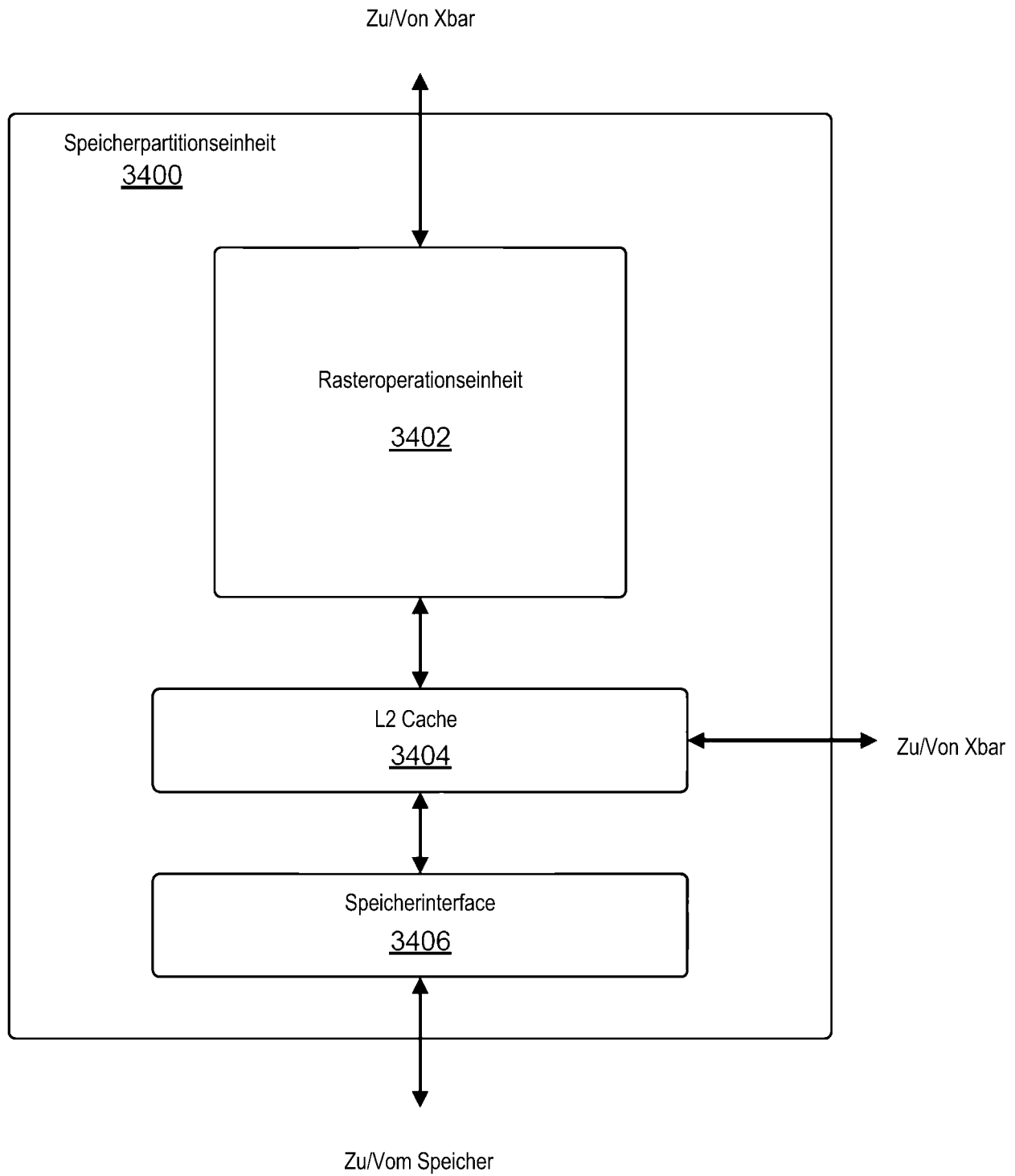
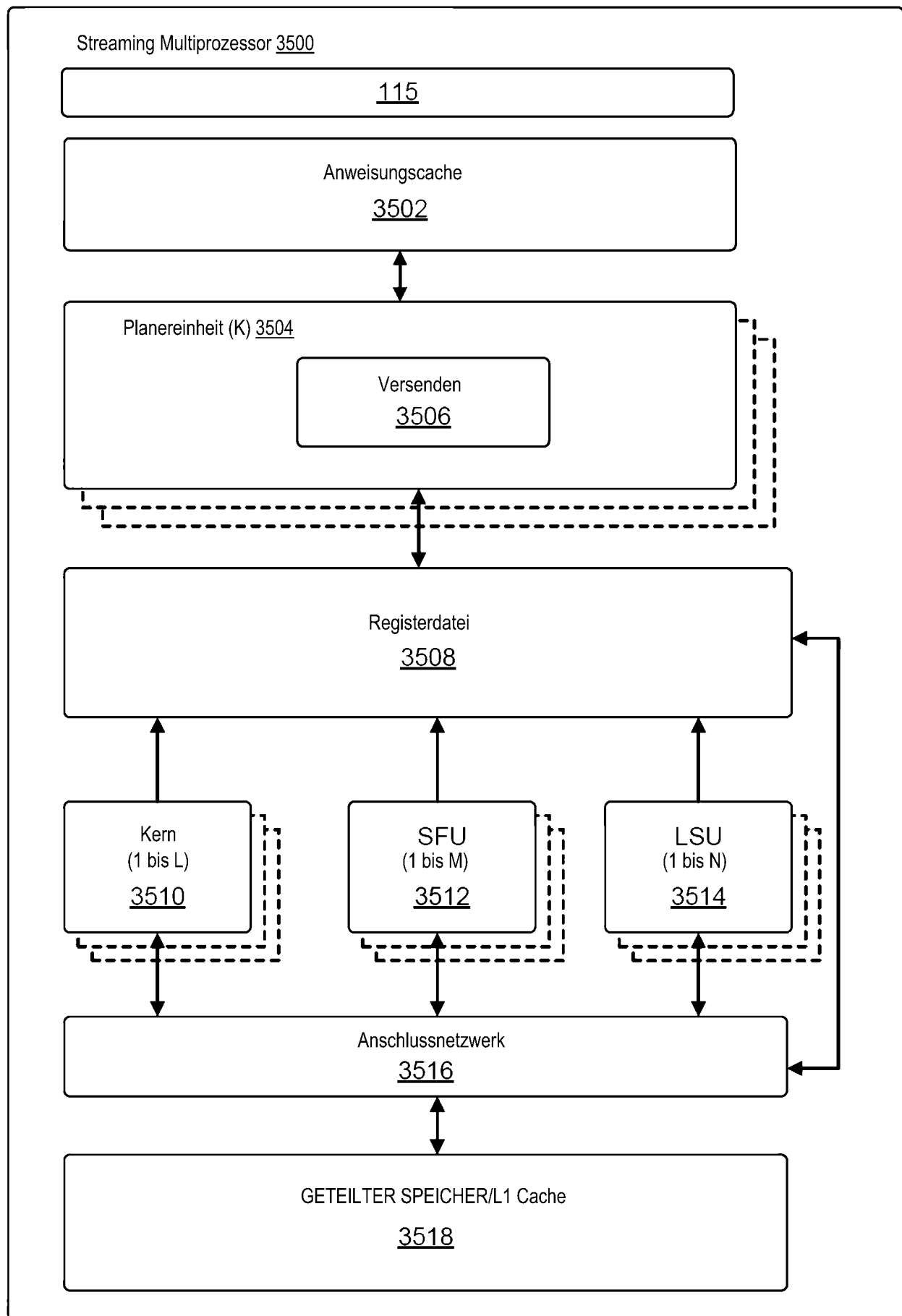


FIG. 34

**FIG. 35**

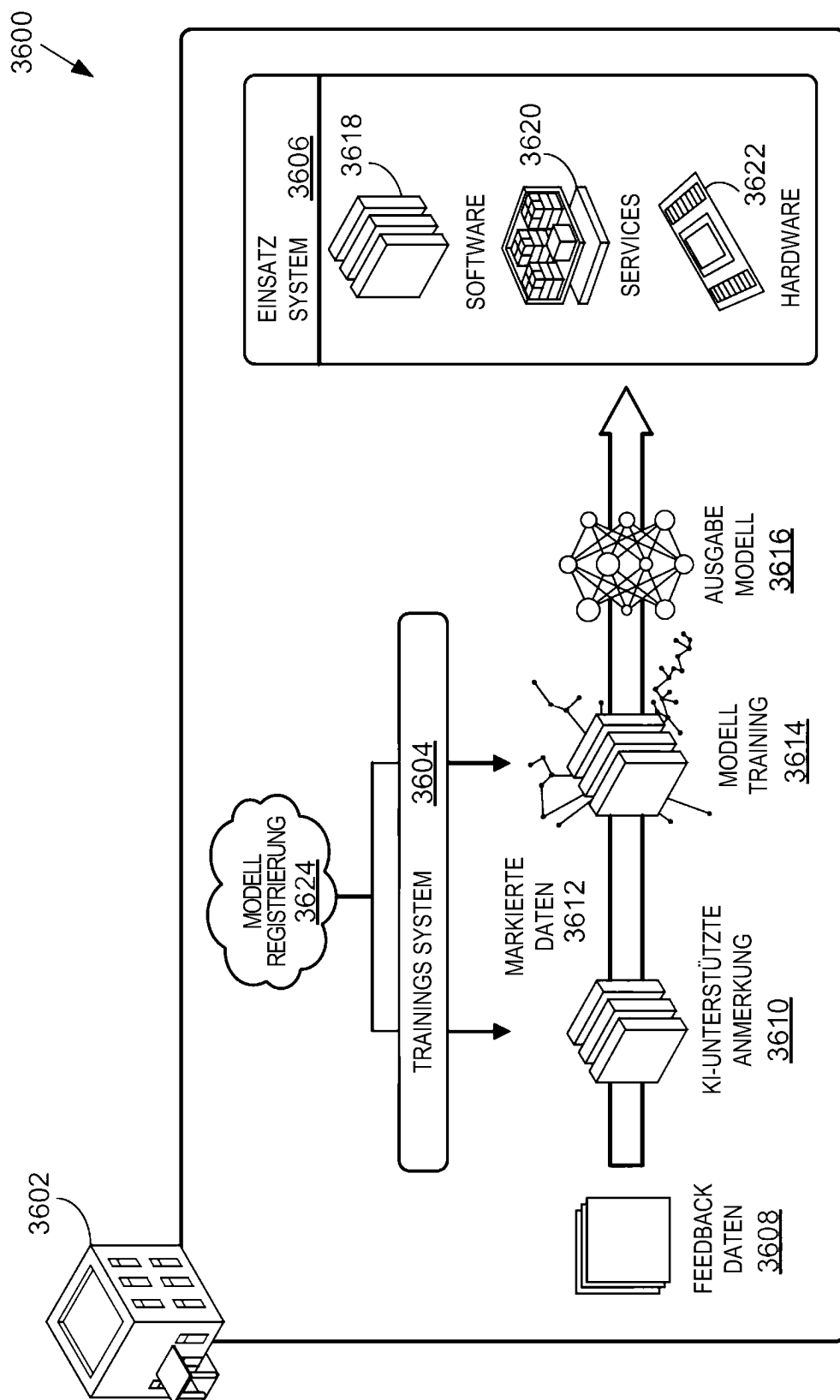


FIG. 36

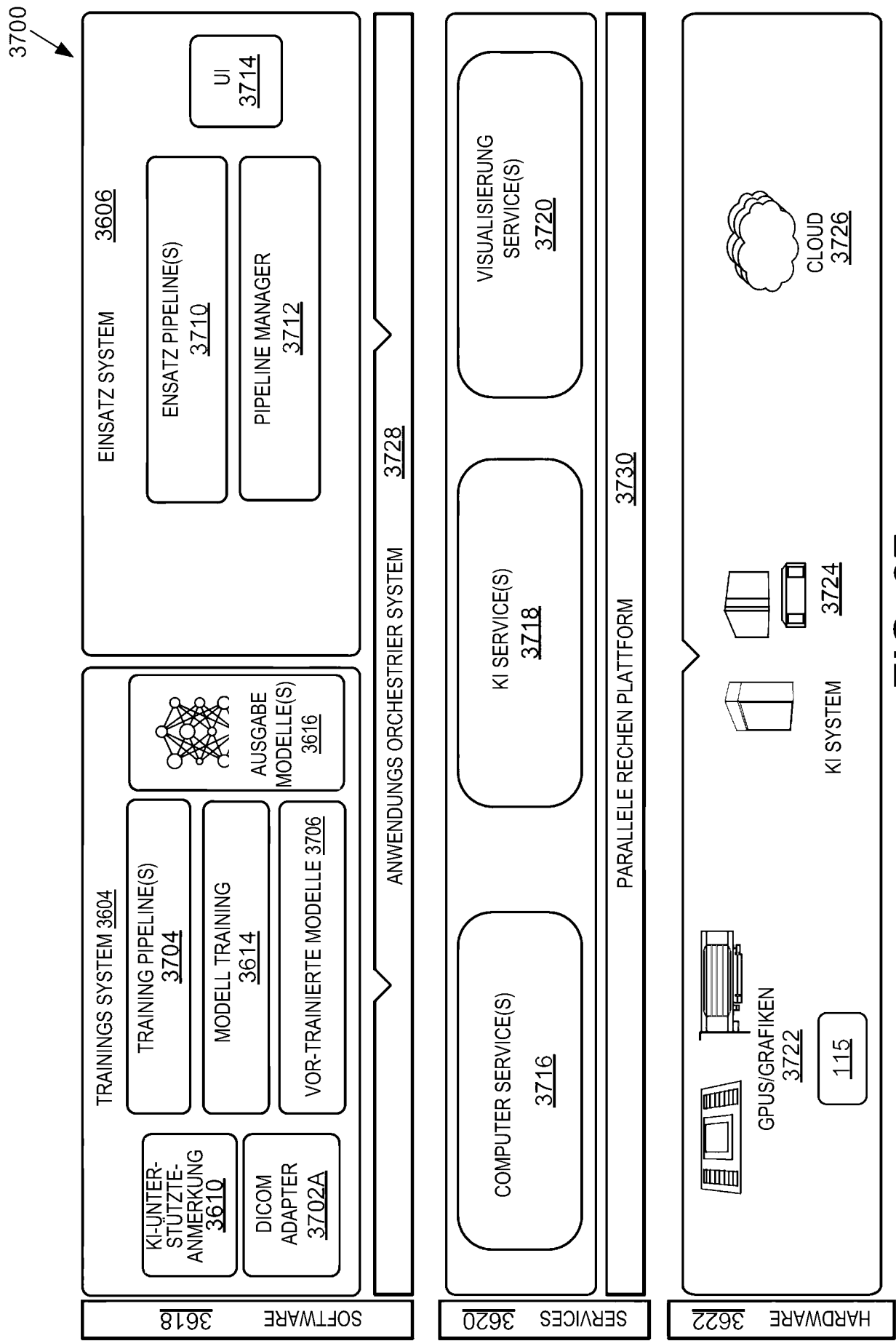


FIG. 37