(19) Deutsches Patent- und Markenamt

(10) **DE 11 2020 004 167 T5** 2022.07.14

(12) # Veröffentlichung

der internationalen Anmeldung mit der
(87) Veröffentlichungs-Nr.: **WO 2021/045905**
in der deutschen Übersetzung (Art. III § 8 Abs. 2 IntPatÜbkG)
(21) Deutsches Aktenzeichen: **11 2020 004 167.0**
(86) PCT-Aktenzeichen: **PCT/US2020/046982**
(86) PCT-Anmeldetag: **19.08.2020**
(87) PCT-Veröffentlichungstag: **11.03.2021**
(43) Veröffentlichungstag der PCT Anmeldung in deutscher Übersetzung: **14.07.2022**

(51) Int Cl.: ***G06T 1/40*** (2006.01)
G06N 3/04 (2006.01)

(30) Unionspriorität:
16/558,620 **03.09.2019** **US**

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

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

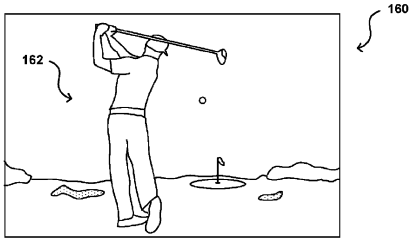
(72) Erfinder:
Shih, Kevin, Santa Clara, CA, US; Dundar, Aysegul, Santa Clara, CA, US; Garg, Animesh, Santa Clara, CA, US; Pottorff, Robert, Santa Clara, CA, US; Tao, Andrew, Santa Clara, CA, US; Catanzaro, Bryan, Santa Clara, CA, US

Prüfungsantrag gemäß § 44 PatG ist gestellt.

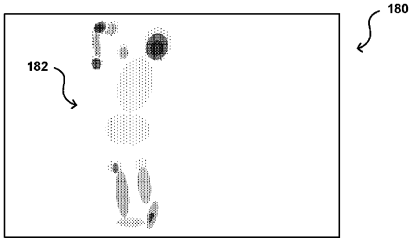
Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

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

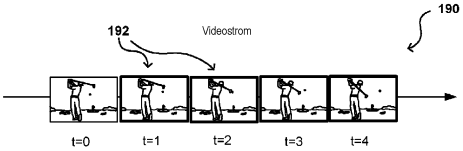
(57) Zusammenfassung: Es werden Einrichtungen, Systeme und Techniken zum Verbessern eines Videos offenbart. In mindestens einer Ausführungsform werden ein oder mehrere neuronale Netze verwendet, um aus einem ersten Video ein zweites Video zu erstellen, das ein oder mehrere zusätzliche Videobilder aufweist.



1D



1E



1F

Beschreibung

GEBIET

[0001] Mindestens eine Ausführungsform bezieht sich auf Verarbeitungsressourcen, die zur Durchführung und Ermöglichung künstlicher Intelligenz verwendet werden. Zum Beispiel bezieht sich mindestens eine Ausführungsform auf Prozessoren oder Rechensysteme, die zum Trainieren von neuronalen Netzen gemäß verschiedenen in dieser Schrift beschriebenen neuartigen Techniken verwendet werden.

ALLGEMEINER STAND DER TECHNIK

[0002] Da Videoinhalte auf immer vielfältigere Art und Weise, auf unterschiedlichen Geräten und aus unterschiedlichen Quellen konsumiert werden, kann es Fälle von Videoinhalten geben, die für eine bestimmte Verwendung oder Anwendung weniger als optimal sein können. Zum Beispiel kann ein Video eine niedrigere Bildrate oder Auflösung haben, als für eine bestimmte Anzeigevorrichtung gewünscht wäre. Es kann auch eine oder mehrere Bildauslassungen geben oder es kann ein Fehler in einem oder mehreren Datenrahmen vorliegen. Es kann auch vorkommen, dass eine Datei weniger Videoinhalte enthält als gewünscht.

Figurenliste

[0003] Verschiedene Ausführungsformen gemäß der vorliegenden Offenbarung werden unter Bezugnahme auf die Zeichnungen beschrieben, in denen Folgendes gilt:

Die **Fig. 1A**, **Fig. 1B**, **Fig. 1C**, **Fig. 1D**, **Fig. 1E** und **Fig. 1F** veranschaulichen Bilddaten, die gemäß mindestens einer Ausführungsform analysiert werden können;

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

Die **Fig. 3A** und **Fig. 3B** veranschaulichen Videobildvorhersagen gemäß mindestens einer Ausführungsform;

Fig. 4 veranschaulicht ein Videobildinferenzsystem gemäß mindestens einer Ausführungsform;

Die **Fig. 5A** und **Fig. 5B** veranschaulichen Abschnitte eines Prozesses zum Ableiten eines oder mehrerer Videobilder gemäß mindestens einer Ausführungsform;

Fig. 6 veranschaulicht ein System zum Trainieren und Inferenzieren unter Verwendung eines oder mehrerer neuronaler Netze gemäß mindestens einer Ausführungsform;

Fig. 7 veranschaulicht ein System zum Trainieren eines oder mehrerer neuronaler Netze gemäß mindestens einer Ausführungsform;

Fig. 8 veranschaulicht eine Struktur eines neuronalen Netzes gemäß mindestens einer Ausführungsform;

Fig. 9A veranschaulicht Inferenz- und/oder Trainingslogik gemäß mindestens einer Ausführungsform;

Fig. 9B veranschaulicht Inferenz- und/oder Trainingslogik gemäß mindestens einer Ausführungsform;

Fig. 10 veranschaulicht ein beispielhaftes Rechenzentrumssystem gemäß mindestens einer Ausführungsform;

Fig. 11 veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;

Fig. 12 veranschaulicht ein Computersystem 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;

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

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

Die **Fig. 17A-17B** veranschaulichen beispielhafte integrierte Schaltungen und zugeordnete Grafikprozessoren gemäß mindestens einer Ausführungsform;

Die **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 einen Verarbeitungscluster gemäß mindestens einer Ausführungsform;

Fig. 20D veranschaulicht einen Grafik-Multiprozessor gemäß mindestens einer Ausführungsform;

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

Fig. 22 veranschaulicht einen Grafikprozessor gemäß mindestens einer Ausführungsform;

Fig. 23 veranschaulicht die Mikroarchitektur eines Prozessors gemäß mindestens einer Ausführungsform;

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

Fig. 25 veranschaulicht einen beispielhaften neuromorphen Prozessor gemäß mindestens einer Ausführungsform;

Die **Fig. 26** und **Fig. 27** veranschaulichen zumindest Abschnitte eines Grafikprozessors gemäß mindestens einer Ausführungsform;

Fig. 28 veranschaulicht zumindest Abschnitte eines Grafikprozessorkerns gemäß mindestens einer Ausführungsform;

Die **Fig. 29A-29B** veranschaulicht zumindest Abschnitte eines Grafikprozessorkems gemäß mindestens einer Ausführungsform;

Fig. 30 veranschaulicht eine Parallelverarbeitungseinheit (parallel processing unit - „PPU“) gemäß mindestens einer Ausführungsform;

Fig. 31 veranschaulicht einen Universalverarbeitungscluster (general processing cluster - „GPC“) gemäß mindestens einer Ausführungsform;

Fig. 32 veranschaulicht eine Speicherpartitionseinheit einer Parallelverarbeitungseinheit („PPU“) gemäß mindestens einer Ausführungsform; und

Fig. 33 veranschaulicht einen Streaming-Multiprozessor gemäß mindestens einer Ausführungsform.

DETAILLIERTE BESCHREIBUNG

[0004] In mindestens einer Ausführungsform kann eine Sequenz 100 von Videobildern an einem Videostrom empfangen werden, wie in **Fig. 1A** veranschaulicht. In mindestens einer Ausführungsform können aufeinanderfolgende Videobilder Variationen gegenüber früheren Videobildern beinhalten. In mindestens einer Ausführungsform kann eine Bildrate, mit der ein Video aufgenommen oder erstellt wird, nicht nur bestimmen, wie oft ein Bild in dem Video aktualisiert wird, sondern auch, wie stark die Variation zwischen den Bildern ist. In mindestens einer Ausführungsform kann es wünschenswert sein, die Bildrate oder die Anzahl der Bilder pro Sekunde in diesem Videostrom oder in einer verwandten Videodatei oder einem verwandten Videoclip zu verbessern. In mindestens einer Ausführungsform kann das Erhöhen der Bildrate das Erlebnis eines Betrachters dieses Videostroms oder eine wahrgenommene Qualität dieses Videostroms verbessern. In mindestens einer Ausführungsform können eines oder mehrere dieser Videobilder aufgrund eines Fehlers aus diesem Videostrom ausgelassen werden, wie in einer Videostromsequenz 120 aus **Fig. 1B** veranschaulicht. In mindestens einer Ausführungsform können beide Situationen aus den **Fig. 1A** und **Fig. 1B** von einer Hinzufü-

gung von Videobildern profitieren, um die Bildrate dieses Videos zu verbessern oder ein oder mehrere ausgelassene Bilder zu ersetzen. In mindestens einer Ausführungsform kann ein derartiger Ansatz zu einer Videostromsequenz 140 wie etwa der in **Fig. 1C** veranschaulichten führen, bei der beliebige ausgelassene Bilder ersetzt werden und eine Gesamtbildrate erhöht wird. In mindestens einer Ausführungsform kann es wünschenswert sein, eine, beide oder keine dieser Korrekturen vorzunehmen, aber die Auflösung dieses Videostroms zu erhöhen.

[0005] In mindestens einer Ausführungsform kann eine Videobildvorhersage unter Verwendung eines aktuellen Videobilds 160, wie etwa des in **Fig. 1D** veranschaulichten, durchgeführt werden. In mindestens einer Ausführungsform kann eine Sequenz derartiger Bilder analysiert werden, um langfristige Videosequenzen zu modellieren: In mindestens einer Ausführungsform kann ein auf maschinellem Lernen basierender Ansatz verwendet werden, der die Vorteile einer latenten Darstellung nutzt, um zu versuchen, Videobilder in einem niederdimensionalen, kompakten Raum abzubilden. In mindestens einer Ausführungsform können ein oder mehrere neuronale Netze trainiert werden, um zu modellieren, wie sich diese latente Darstellung im Laufe der Zeit verändert, um diese Abbildung auf ein ursprüngliches Videobild zurück zu finden. In mindestens einer Ausführungsform werden diesem latenten Raum eine oder mehrere Beschränkungen auferlegt, um zu ermöglichen, dass dieser latente Raum interpretiert werden kann. In mindestens einer Ausführungsform beinhaltet eine Beschränkung das Modellieren eines Satzes von Koordinaten, die sich durch die Zeit bewegen, wobei diese sich bewegenden Koordinaten verwendet werden können, um das Erscheinungsbild eines oder mehrerer Objekte in einer entsprechenden Videodatei anzupassen.

[0006] In mindestens einer Ausführungsform können ein oder mehrere neuronale Netze trainiert werden, um anhand eines oder mehrerer Videobilder von vergangenen Zeitpunkten Videobilder für einen oder mehrere zukünftige Zeitpunkte, zumindest in Bezug auf die Abspielzeit eines Videos, vorherzusagen oder abzuleiten. In mindestens einer Ausführungsform können Merkmale wie etwa nicht überwachte Orientierungspunkte verwendet werden, um verschiedene Merkmale eines Objekts für Zwecke der Modellierung von Bewegungen oder anderen Veränderungen im Laufe der Zeit und zwischen benachbarten Bildern in Videodaten darzustellen. In mindestens einer Ausführungsform können Merkmale wie etwa diese Orientierungspunkte verwendet werden, um die Bewegung eines Objekts, wie etwa einer Person, durch eine Sequenz von Videobildern zu modellieren. In mindestens einer Ausführungsform kann ein Posendetektor verwendet werden, um Merkmalspunkte zu bestimmen, die den Gliedmaßen und Gelenken einer in einem Video dargestellten Person entsprechen können, und eine Bewegung der Person durch Bewegen dieser Merkmalspunkte zwischen Bildern abzuleiten. In mindestens einer Ausführungsform kann ein trainiertes Modell Merkmalspunkte mit ihrer abgeleiteten Bewegung auf Bilddaten für dieses Videobild zurück abbilden, um ein nachfolgendes Videobild abzuleiten, das eine Darstellung dieser Person beinhaltet, zu einem zukünftigen Zeitpunkt auf Grundlage einer früheren beobachteten Bewegung oder eines früheren beobachteten Verhaltens abzuleiten. In mindestens einer Ausführungsform können Merkmale wie etwa nicht überwachte Orientierungspunkte verwendet werden, um Schlüsselpunkte zu approximieren, die ein Objekt darstellen, ohne dass eine erhebliche Menge an Trainingsdaten manuell annotiert werden muss. In mindestens einer Ausführungsform kann ein Posendetektor verwendet werden, um ein Videobild, wie etwa das Bild 160 aus **Fig. 1D**, das eine Darstellung einer Person 162 beinhaltet, zu analysieren und Poseninformationen 182 zu erzeugen, wie in Bild 180 aus **Fig. 1E** veranschaulicht. In mindestens einer Ausführungsform beinhaltet das Bild 180 Bereiche, die sich bewegenden Teilen dieser Person 182 entsprechen, mit Intensitäts- oder anderen Werten, die die Bestimmung von Merkmalspunkten oder nicht überwachten Orientierungspunkten in diesen Bilddaten ermöglichen. In mindestens einer Ausführungsform muss ein Intensitätswert mindestens einen minimalen Merkmalsschwellenwert erfüllen, bevor er als Bildmerkmal berücksichtigt oder ausgewählt wird. In mindestens einer Ausführungsform kann eine Schwerpunktberechnung oder ein Ansatz mit einem absoluten Maximum verwendet werden, um eine Koordinate für einen bestimmten Merkmalspunkt zu bestimmen, die aus einem Satz von Koordinatenauswahlalgorithmen ausgewählt werden kann. In mindestens einer Ausführungsform können diese Merkmalspunkte als nicht überwachte Orientierungspunkte betrachtet werden, da es für keinen dieser Merkmalspunkte Annotationen gibt, sodass ein Posendetektor eine Koordinate für einen Ellbogen bestimmen kann, aber nicht weiß, dass dieser Punkt einem Ellbogen entspricht. In mindestens einer Ausführungsform kann eine gewisse Annotation unter Verwendung von Bildrekonstruktion oder eines anderen derartigen Prozesses durchgeführt werden, wobei die Kenntnis eines Punkttyps bei der Modellierung der Bewegung eines Objekts im Laufe der Zeit berücksichtigt werden kann. In mindestens einer Ausführungsform kann ein derartiger Prozess für eine langfristige Videovorhersage, wie etwa für eine Sequenz 190 von Bildern in einem Videostrom in **Fig. 1F**, verwendet werden, wobei eine Anzahl von Bildern 192 unter Verwendung von Informationen von einem oder mehreren früheren Bildern vorhergesagt wird.

[0007] In mindestens einer Ausführungsform kann ein System 200 zum Analysieren von Eingangsvideobildern Komponenten wie in **Fig. 2** veranschaulicht beinhalten. In mindestens einer Ausführungsform beinhaltet das System 200 drei primäre Komponenten, einschließlich eines Bildcodierers 222, eines Bilddecodierers 216 und eines zeitlichen Dynamikcodierers 214. In mindestens einer Ausführungsform wird ein Videobild 202 als Eingabe in einen Bildcodierer 222 empfangen. In mindestens einer Ausführungsform ist der Bildcodierer 222 dazu konfiguriert, ein Videobild 202 in einen faktorisierten Posenerscheinungsbildraum zu projizieren. In mindestens einer Ausführungsform beinhaltet der Bildcodierer 222 einen Posencodierer 204 zum Bestimmen von Poseninformationen für das Eingangsbild 202 und einen Erscheinungsbildcodierer 206 zum Bestimmen von Erscheinungsbildinformationen für das Eingangsbild 202. In mindestens einer Ausführungsform können die Posendaten 208 durch den Posencodierer 204 bestimmt werden, der mit einem Posenanalysator 212 verwendet werden kann, um Poseninformationen für einen Zeitpunkt t des Eingangsvideobildes 202 zu erzeugen, die an den zeitlichen Codierer 214 übergeben werden können. In mindestens einer Ausführungsform können die von dem Erscheinungsbildcodierer 206 bestimmten Erscheinungsbildinformationen direkt an den Bilddecodierer 216 übergeben werden. In mindestens einer Ausführungsform kann der Bildcodierer 216 diese Erscheinungsbildinformationen nehmen und mit vorhergesagten Posendaten (als Daten in dem Posenerscheinungsbildraum) verwenden, um ein Videobild 220 zu (re)konstruieren, das dem Eingangsbild 202 entsprechen sollte, um Orientierungspunkte zum Modellieren von Bewegung zu lernen.

[0008] In mindestens einer Ausführungsform werden die Posendaten von dem Posencodierer 204 an den zeitlichen Codierer 214 übergeben, um zu bestimmen, wie sich diese Posendarstellung für einen oder mehrere zukünftige Zeitpunkte (z. B. t+2) ändern oder unterscheiden würde. In mindestens einer Ausführungsform bezieht sich die Zeit auf die Wiedergabezeit eines Videos, da eine gespeicherte Videozeit zu verschiedenen Zeitpunkten betrachtet werden kann. In mindestens einer Ausführungsform kann bei der Videovorhersage und -interpolation zumindest für einen bestimmten Zeitraum angenommen werden, dass die Erscheinungsbildinformationen in diesem Video konstant sind und Änderungen können durch die Posendarstellung erfasst werden. In mindestens einer Ausführungsform kann der zeitliche Dynamikcodierer 214 bei gegebener Faktorisierung eines Bildes in Pose und Erscheinungsbild vorhersagen, wie sich die Posendarstellung im Laufe der Zeit ändert. In mindestens einer Ausführungsform kann der Bilddecodierer 216 dann unter Verwendung konstanter Erscheinungsbildinformationen und einer aktualisierten Posendarstellung in jedem Zeitschritt ein neues Bild 220 konstruieren.

[0009] In mindestens einer Ausführungsform kann ein derartiger Ansatz verwendet werden, um Videobilder auf Grundlage eines oder mehrerer Eingangsbilder vorherzusagen. In mindestens einer Ausführungsform können die Videoströme 300, 350 unter Verwendung eines derartigen Ansatzes, aber ohne Ground-Truth-Daten, erzeugt werden, wie in den **Fig. 3A** und **Fig. 3B** veranschaulicht. In mindestens einer Ausführungsform stimmen die abgeleiteten Bilder im Vergleich zu den entsprechenden Ground-Truth-Daten gut mit Ground-Truth-Daten für kürzere Zeiträume überein, aber wenn eine Zeitmenge ohne ein weiteres Referenzbild zunimmt, können deutliche Unterschiede auftreten und an Bedeutung gewinnen. In mindestens einer Ausführungsform können Videoströme selbst mit einer Feinabstimmung nur um einen bestimmten Betrag hinsichtlich der Auflösung oder Bildrate hochskaliert werden, ohne dass dabei potenziell auffällige Artefakte entstehen.

[0010] In mindestens einer Ausführungsform kann ein derartiger Ansatz für die Vorhersage und Interpolation von langfristigen Videodaten verwendet werden. In mindestens einer Ausführungsform werden die Videovorhersage und -interpolation als nicht überwachte latente Strukturinferenz modelliert, gefolgt von einer zeitlichen Vorhersage im latenten Raum. In mindestens einer Ausführungsform können latente Darstellungen die Vordergrundsemantik ohne explizite Überwachung erfassen, wie etwa unter Verwendung von Schlüsselpunkten oder annotierten Posendaten. In mindestens einer Ausführungsform kann jeder Orientierungspunkt auf eine Koordinate abgebildet werden, die angibt, wo ein semantisches Teil positioniert ist, was eine zuverlässige Interpolation innerhalb einer Koordinatendomäne ermöglicht, um eine vorhersagbare Bewegungsinterpolation zu erreichen. In mindestens einer Ausführungsform kann ein Bilddecodierer diese Orientierungspunkte auf eine Bilddomäne zurück abbilden, wodurch eine langfristige Videointerpolation und -extrapolation mit hoher Qualität erreicht wird, durch Arbeiten auf einem Orientierungspunktdarstellungsraum. In mindestens einer Ausführungsform wird eine selbstüberwachte Pipeline verwendet, um die langfristige Videodynamik in einer latenten expliziten Posendarstellung zu modellieren. In mindestens einer Ausführungsform werden grobe Begrenzungsfelder auf Objektebene zu Trainingszwecken verwendet.

[0011] In mindestens einer Ausführungsform kann eine latente Darstellung von K Orientierungspunkt-„Schlüsselpunkten“ gelernt werden, die jeweils ein räumlich lokales Erscheinungsbild und 2D-Koordinaten einer einzelnen starren Komponente beschreiben. In mindestens einer Ausführungsform ist diese Darstellung

interpretierbar und ergibt sich auf natürliche Weise bei der Modellierung der Dynamik verbundener oder stark korrelierter starrer Objekte in einer Szene. In mindestens einer Ausführungsform können nicht überwachte Prozesse zur Identifizierung von Orientierungspunkten verwendet werden, die lokalisierte Orientierungspunkte lernen können, die unabhängig von einer Klasse typischer Transformationen immer an denselben semantischen Stellen aktiviert werden. In mindestens einer Ausführungsform kann dies erweitert werden, um nicht überwachte Orientierungspunkte für eine langfristige Videobildvorhersage und -interpolation zu verwenden. In mindestens einer Ausführungsform kann eine beschränkte und parametrisierte Darstellung sowohl ausreichend aussagekräftig als auch stabil sein, um langfristige qualitative Schätzungen für die Videovorhersage zu erreichen, während außerdem intuitive Ergebnisse bei Interpolationsaufgaben bereitgestellt werden. In mindestens einer Ausführungsform können Posendarstellungen zumindest teilweise auf Grundlage von zweidimensionalen (2D-)Gauß-Wärmekarten sowohl in prädiktiven als auch in interpolativen Umfeldern manipuliert werden.

[0012] In mindestens einer Ausführungsform wird die Posen- und Erscheinungsbild-Faktorisierung unter Verwendung von nicht überwachten Orientierungspunkten gelernt, indem ein Eingangsbild in eine erscheinungsbildinvariante Posendarstellung und eine poseninvariante Erscheinungsbilddarstellung getrennt wird und dann versucht wird, ein ursprüngliches Eingangsbild zu rekonstruieren, um Orientierungspunkte zu lernen. In mindestens einer Ausführungsform kann ein Netz unter Verwendung von drei Modellen aufgebaut werden: einem Bild-zu-Pose-Codierer, einem Bild-zu-Erscheinungsbild-Codierer und einem Bilddecodierer, der versucht, ein Eingangsbild aus faktorisierten Pose- und Erscheinungsbilddarstellungen zu rekonstruieren.

[0013] In mindestens einer Ausführungsform kann der Posencodierer $\Phi^{pose} = Enc^{pose}$ (x)ein Eingangsbild x nehmen und einen Satz von konsistenten Teilaktivierungskarten ausgeben. In mindestens einer Ausführungsform sind diese Teilaktivierungskarten invariant gegenüber Änderungen des lokalen Erscheinungsbilds und sind über Verformungen konsistent. In mindestens einer Ausführungsform sollte eine Wärmekarte, die auf der rechten Hand einer Person aktiviert wird, über variierende Hauttöne und Beleuchtungsbedingungen invariant sein sowie die Position dieser rechten Hand über variierende Verformungen und Übersetzungen verfolgen. In mindestens einer Ausführungsform extrahiert der Erscheinungsbildcodierer $\Phi^{app} = Enc^{app}$ (x; Enc^{pose} (x))lokale Erscheinungsbildinformationen, die auf Aktivierungskarten eines Posencodierers konditioniert sind. In mindestens einer Ausführungsform stellt dieser Posencodierer bei einem gegebenen Eingangsbild x, zuerst K x Hx W Teilaktivierungskarten Φ^{pose} bereit. In mindestens einer Ausführungsform projiziert ein Erscheinungsbildcodierer zur Extraktion lokaler Erscheinungsbildvektoren dieses Bild auf eine C x H x W Erscheinungsbildmerkmalskarte M^{app}. In mindestens einer Ausführungsform kann ein Erscheinungsbildvektor für eine k. Posenaktivierungskarte wie durch Folgendes gegeben berechnet werden:

$$\Phi_{k,c}^{app} = \sum_{i}^{H} \sum_{j}^{W} \Phi_{k,i,j}^{pose} M_{c,i,j}^{app} \text{ für } c = 1 \ldots C,$$

wobei K C-dimensionale Erscheinungsbildverktoren erhalten werden. In mindestens einer Ausführungsform ist jede Aktivierungskarte in Φ^{pose} softmax-normalisiert.

[0014] In mindestens einer Ausführungsform versucht ein Bilddecodierer, ein ursprüngliches Eingangsbild zu rekonstruieren, indem die Poseninformationen aus K Aktivierungskarten mit gepoolten Erscheinungsbildvektoren für jedes der K Teile kombiniert werden. In mindestens einer Ausführungsform kann anstelle der Konditionierung auf rohen Softmax-normierten Aktivierungsposenkarten eine 2D-Gauß-Verteilung an jede Aktivierung dieser K Aktivierungsabbildungen angepasst werden, indem ihre jeweiligen Mittelwerte und Kovarianzmatrizen geschätzt werden. In mindestens einer Ausführungsform ist jedes Teil durch $\widetilde{\Phi}_{k}^{pose} = (\mu_k, \Sigma_k)$ dargestellt, wobei $\mu_k \in \mathbb{R}^2$ und $\Sigma_k \in \mathbb{R}^{2\times 2}$. In mindestens einer Ausführungsform ermöglicht eine 2D-Gauß-Approximation die Manipulation einer Posendarstellung, da sie jede Teilaktivierungskarte in eine unimodale Darstellung mit einer einfachen Parametrisierung zwingt. In mindestens einer Ausführungsform werden ein Posencodierer, ein Formcodierer und ein Bilddecodierer gemeinsam in einer vollständig selbstüberwachten Weise trainiert, wobei eine abschließende Bildrekonstruktionsaufgabe als einzige Überwachungsquelle verwendet wird.

[0015] In mindestens einer Ausführungsform werden Farb-Jittering und Thin-Plate-Spline-(TPS-)Verkrümmung in eine Trainings-Pipeline integriert, um die Erscheinungsinvarianz bzw. die Lokalisierungseigenschaften eines Posencodierers zu erzwingen. In mindestens einer Ausführungsform stellen $T_{cj}(x)$ und $T_{tps}(x)$ Farb-Jittering- bzw. Thin-Plate-Spline-Verkrümmungs-Operationen auf dem Eingangsbild x dar. In mindestens einer Ausführungsform kann ein Trainingsvorgang wie folgt ausgedrückt werden:

$$\Phi^{pose} = Enc^{pose}\left(T_{c_j}(x)\right)$$

$$\Phi^{app} = Enc^{app}\left(T_{tps}(x);\ Enc^{pose}\left(T_{tps}(x)\right)\right)$$

$$\tilde{x} = Dec\left(\widetilde{\Phi}^{pose}, \Phi^{app}\right.$$

wobei der Rekonstruktionsverlust zwischen der ursprünglichen Eingabe x und der Rekonstruktion x̃ minimiert werden kann.

[0016] In mindestens einer Ausführungsform erhalten weder ein Form- noch ein Erscheinungsbildcodierer jemals Zugang zu dem ungestörten Ziel x. In mindestens einer Ausführungsform basieren die dem BilddecodiererDec(·, ·) zugeführten Poseninformationen auf einem Farb-Jittering unterzogenen Eingangsbild, bei dem nur lokale Erscheinungsbildinformationen gestört sind. In mindestens einer Ausführungsform werden die Erscheinungsbildinformationen aus einer Thin-Plate-Spline-verkrümmten Version eines Eingangsbilds berechnet, bei der die Poseninformationen gestört sind. In mindestens einer Ausführungsform kann ein Formkodierer, der ebenfalls auf diesem TPS-verkrümmten Eingangsbild ausgeführt wird, entsprechende Positionen in neuen verkrümmten Koordinaten lokalisieren, damit korrekte Erscheinungsbildinformationen erfasst werden können. In mindestens einer Ausführungsform wird außerdem die vorhandene Bewegung in den Eingangsvideodaten als Quelle von Posenstörung verwendet. In mindestens einer Ausführungsform kann zusätzlich zu einer T_{c_j} (x) - T_{tps} (x)-Paarung eine weitere Paarung eingeschlossen werden, bei der T_{tps} durch Abtasten eines zufälligen Videobilds zwischen 3 und 60 Bildern von dem aktuellen Bild x entfernt ersetzt wird.

[0017] In mindestens einer Ausführungsform werden die 2D-Gauß-Parameter von $\tilde{\Phi}^{pose}$ auf eine Wärmekarte mit Zielausgabebreite und -höhe projiziert , um als semantische Abbildungen verwendet zu werden. In mindestens einer Ausführungsform ist eine zu verwendende Formel wie folgt gegeben:

$$s(k,l) = \frac{1}{1+(l-\mu_k)^{\in}\Sigma_k^{-1}(l-\mu_k)}$$

wobei s(k,l) ein Wärmekartenwert für die Teilabbildung k an der Koordinatenposition / ist.

[0018] In mindestens einer Ausführungsform kann für die Verluste ein VGG-Wahrnehmungsverlust mit vortrainierten ImageNet-Gewichtungen verwendet werden, kombiniert mit einem kontradiktorischen Verlust zur Verbesserung des Realismus und einem mittleren quadratischen Standardverlust auf Pixelebene. In mindestens einer Ausführungsform kann der Rekonstruktionsverlust wie folgt ausgedrückt werden:

$$\mathcal{L}_{recon}(x,\tilde{x}) = \lambda_{vgg}\mathcal{L}_{vgg}(x,\tilde{x}) + \lambda_{MSE}MSE(x,\tilde{x}) + \mathcal{L}_{adv}(\tilde{x})$$

[0019] In mindestens einer Ausführungsform kann, da eine auf nicht überwachten Orientierungspunkten basierende Posendarstellung explizit als lokalisierte Teile-Labels interpretiert werden kann, es natürlich sein, innerhalb dieses Raums zu manipulieren und zu interpolieren. In mindestens einer Ausführungsform kann jedoch darauf geachtet werden, sicherzustellen, dass eine Kovarianzmatrix für jeden Orientierungspunkt während jeglicher Manipulation positiv definit bleibt. In mindestens einer Ausführungsform kann dies in vielen Parametrisierungen erfolgen, wie etwa direkt in Kovarianzparametern, indirekt in Parametern einer Cholesky-Zerlegung oder einem Winkel und einer Größe von Hauptkomponenten oder sogar entlang eines optimalen Transports als Wasserstein-Barycenter. In mindestens einer Ausführungsform dominiert aufgrund der breiten Vielfalt möglicher projizierter Verformungen keines dieser Verfahren unter allen Bedingungen.

[0020] In mindestens einer Ausführungsform werden die Parameter einer Cholesky-Zerlegung verwendet, da sie einfach zu berechnen sein können und die Gültigkeit der Kovarianz sowohl im Interpolations- als auch im Extrapolationsparadigma gewährleisten. In mindestens einer Ausführungsform ist die Cholesky-Zerlegung jeder Kovarianzmatrix eine eindeutige Matrix $\mathcal{L}_k$ mit positiver Diagonale, sodass $\Sigma_k = L_k L_k^{\mathsf{T}}$, wobei L_k eine untere Dreiecksmatrix ist. In mindestens einer Ausführungsform kann dies für 2-dimensionale Schlüsselpunkte, wie in dieser Schrift erörtert, zu drei Skalaren führen (z. B. zwei positiven, einem unbeschränkten), die eine Kovarianz jedes Schlüsselpunkts eindeutig parametrisieren. In mindestens einer Ausführungsform bilden diese fünf Parameter, wenn sie mit einem zweidimensionalen Mittelwert μ_k gekoppelt werden, einen

Zustandsvektor für den Schlüsselpunkt k zum Zeitschritt t. In mindestens einer Ausführungsform wird die Interpolation linear durchgeführt, was wie folgt gegeben sein kann:

$$\begin{bmatrix} \hat{\mu}_k \\ \hat{L}_k \end{bmatrix}_{t=\alpha} = (1-\alpha)\begin{bmatrix} \mu_k \\ L_k \end{bmatrix}_{t=0} + \alpha\begin{bmatrix} \mu_k \\ L_k \end{bmatrix}_{t=1}$$

[0021] In mindestens einer Ausführungsform kann ein langes Kurzzeitgedächtnis (long shortterm memory - LSTM) verwendet werden, um nicht lineare zeitliche Dynamik einer Posendarstellung zu modellieren. In mindestens einer Ausführungsform muss wie bei der Interpolation darauf geachtet werden, dass die positiv definiten Kovarianzmatrizen während der Vorhersage erhalten bleiben. In mindestens einer Ausführungsform werden die Parameter einer Cholesky-Zerlegung von Kovarianzmatrizen als Vorhersageziele verwendet. In mindestens einer Ausführungsform kann ein LSTM eine Schätzung für L erzeugen, die kein gültiger Cholesky-Faktor ist, aber selbst in diesem Fall wird immer noch eine gültige Kovarianzmatrix erzeugt, wenn LL^T berechnet wird.

[0022] In mindestens einer Ausführungsform kann ein Paar von LSTMs verwendet werden, das Störungen der Gauß-Mittelwerte und Cholesky-Parameter der Kovarianzen vorhersagen kann. In mindestens einer Ausführungsform hat zusätzlich zur Cholesky-Parametrisierung die Vorhersage eines Residuums während der Extrapolation anstelle der direkten Vorhersage des Zustands dazu beigetragen, robuste langfristige Vorhersagen zu gewährleisten. In mindestens einer Ausführungsform kann dieses Zustandsresiduum oder dieser Fehlerzustand verwendet werden, um die Schätzleistung bei einer Vielfalt von Aufgaben mit nicht linearer Dynamik zu verbessern. In mindestens einer Ausführungsform kann ein Fehlerzustand zusätzlich zur Aufrechterhaltung eines sich weit in die Zeit zurück erstreckenden nicht beschränkten Gradientenpfads außerdem ein Modell veranlassen, lokale Zustandsverformungen zu lernen, die leichter zu lernen sein können, da sie klein und nahezu linear sind. In mindestens einer Ausführungsform kann dies dazu beitragen, sowohl die Trainingsleistung als auch die Generalisierung zu verbessern. In mindestens einer Ausführungsform nimmt ein LSTM bei jedem Zeitschritt eine Verkettung des vorherigem Zustands und des Zustandsresiduums, um ein nächstes Zustandsresiduum vorherzusagen.

[0023] In mindestens einer Ausführungsform ist ein selbstüberwachtes Paradigma für langfristige Videovorhersage und -interpolation in der Lage, die Bildstruktur über lange Zeiträume zu erhalten. In mindestens einer Ausführungsform wird dies durch Faktorisierung der Bilddaten in Poseninformationen und zeitlich invariante Erscheinungsbildinformationen erreicht. In mindestens einer Ausführungsform können Sammlungen von nicht überwachten 2D-Gauß-Orientierungspunkten als Posenzustand verwendet werden, da die Poseninformationen während der langfristigen Videovorhersage zahlreichen Störungen unterworfen sind. In mindestens einer Ausführungsform kann jede 2D-Gauß-Verteilung im Gauß-Parameterraum (μ,Σ) sicher gestört werden, bevor sie in 2D-Wärmekarten projiziert wird, um einem Bilddecodierer zugeführt zu werden, wodurch gewährleistet wird, dass ein Bilddecodierer eine Sammlung von unimodalen Gauß-Wärmebildern sieht. In mindestens einer Ausführungsform ist eine Posendarstellung eine Sammlung von anisotropen 2D-Gauß-Wärmekarten. In mindestens einer Ausführungsform ermöglicht eine strenge Gauß-Darstellung die Manipulation im Gauß-Parameterraum, während trotzdem bei jedem Zeitschritt eine Gauß-Verteilung gewährleistet wird. In mindestens einer Ausführungsform kann jede Gauß-Verteilung als Lokalisierung eines „Teils“ eines Objekts, wie etwa eines Handgelenks oder Beins, interpretiert werden, wobei die Kovarianz die räumliche Unterstützung des Teils grob erfasst. In mindestens einer Ausführungsform werden die Gauß-Orientierungspunkte gelernt, ohne dass eine manuelle Teileannotation erforderlich ist. In mindestens einer Ausführungsform werden die Teile, die gelernt werden, vollständig während des Trainings bestimmt.

[0024] In mindestens einer Ausführungsform können die Gauß-Verteilungen zeitlich in dem Gauß-Parameterraum manipuliert werden (z. B. Mittelwerte, Kovarianz), wodurch gewährleistet wird, dass eine Posendarstellung streng eine Sammlung von Gauß-Wärmekarten bei jedem Zeitschritt ist. In mindestens einer Ausführungsform kann dann eine bestimmte Übersetzung auf diese Gauß-Verteilungen angewendet werden. In mindestens einer Ausführungsform kann die Kovarianz schwieriger sein, da sie unter Umständen nach jeder vorhergesagten Störung invertierbar und positiv definit sein muss. In mindestens einer Ausführungsform kann eine Cholesky-Zerlegung in eine untere Dreiecksmatrix L durchgeführt werden. In mindestens einer Ausführungsform kann die Matrix L gestört und eine neue Kovarianz mit LL^T rekonstruiert werden, wodurch bei jedem Schritt eine positiv definite Kovarianzmatrix gewährleistet wird.

[0025] In mindestens einer Ausführungsform kann dann auf nicht überwachte Orientierungspunkte oder andere Merkmale, die für ein Videobild erhalten wurden, ein Glättungsprozess für den freien Raum angewen-

det werden. In mindestens einer Ausführungsform wird jeder nicht überwachte Orientierungspunkt durch eine Gauß-Wärmekarte dargestellt, die als Mittelwert und Kovarianz parametrisiert werden kann, wobei der Mittelwert eine Mittelkoordinate für ein jeweiliges Merkmal ist. In mindestens einer Ausführungsform kann ein zeitliches Dynamikmodell (z. B. ein für die Vorhersage verwendetes LSTM) trainiert werden, um anhand einer Sequenz dieser Koordinaten zu bestimmen, wie sich diese Koordinaten im Laufe der Zeit verschieben. In mindestens einer Ausführungsform kann ein neues Layout dieser Farbverteilungen im Laufe der Zeit vorhergesagt werden. In mindestens einer Ausführungsform kann dieses Layout einem anderen Netz zugeführt werden, entweder als Teil des Bilddecodierers 216 oder als Teil eines separaten Videogenerators 402, wie in **Fig. 4** veranschaulicht, das es zurück in den Bildraum abbildet, um ein abgeleitetes Video zu erstellen. In mindestens einer Ausführungsform kann lineare Interpolation verwendet werden, um eine bestimmte Bewegung in einer Sequenz von abgeleiteten Videobildern zu erfassen. In mindestens einer Ausführungsform können ein oder mehrere physische Simulatoren verwendet werden, um abzuleiten, wie sich diese Gauß-Koordinaten im Laufe der Zeit bewegen könnten. In mindestens einer Ausführungsform beinhaltet ein abgeleitetes Video ein oder mehrere zusätzliche Videobilder, entweder um neue Inhalte hinzuzufügen, fehlende Inhalte zu ergänzen oder die Qualität vorhandener Inhalte zu verbessern, wie etwa um die Bildrate oder Auflösung einer vorhandenen Videodatei oder eines vorhandenen Videostroms zu verbessern.

[0026] In mindestens einer Ausführungsform kann ein in **Fig. 5A** veranschaulichter Prozess 500 verwendet werden, um ein Modell zu trainieren, Videobilder für Zwecke wie etwa das Verbessern von Videodaten abzuleiten. In mindestens einer Ausführungsform werden ein oder mehrere Videobilder empfangen 502 oder anderweitig erhalten, die Teil einer Videodatei oder eines Videostroms sein können. In mindestens einer Ausführungsform werden Bilder aus diesem Video als Eingangstrainingsdaten für ein Trainingssystem zum Trainieren eines oder mehrerer neuronalen Netzes bereitgestellt. In mindestens einer Ausführungsform wird ein aktuelles Videobild analysiert 504, um Posen- und Erscheinungsbilddarstellungen für ein aktuelles Videobild zu bestimmen. In mindestens einer Ausführungsform kann ein Ansatz mit nicht überwachten Orientierungspunkten verwendet werden, bei dem Merkmale in dieser Posendarstellung als Gauß-Wärmekarten dargestellt werden, die jeweils als Mittelwert und Kovarianz parametrisiert werden können. In mindestens einer Ausführungsform können diese Wärmekarten verwendet werden, um eine oder mehrere Posendarstellungen vorherzusagen 506, die einem oder mehreren zukünftigen Zeitpunkten oder Zeiträumen entsprechen können. In mindestens einer Ausführungsform können die Posendarstellungsinformationen mit Erscheinungsbilddarstellungsinformationen verwendet werden 508, um ein aktuelles Videobild, das analysiert wird, zu rekonstruieren. In mindestens einer Ausführungsform werden die Eingangsversion und die rekonstruierte Version eines aktuellen Videobilds verglichen 510, um unter Verwendung einer geeigneten Verlustfunktion einen aktuellen Verlustwert zu bestimmen. In mindestens einer Ausführungsform können ein oder mehrere Netzparameter angepasst werden 512, um zu versuchen, diesen Verlust zu minimieren. Es kann eine Bestimmung getroffen werden 514, ob mehr Bilder analysiert werden sollten. In mindestens einer Ausführungsform beinhalten die Bestimmungsfaktoren, ob zusätzliche Bilder in dieser Videodatei oder diesem Videostrom vorhanden sind und ob ein Modellkonvergenz- oder Trainingsabschlusskriterium erfüllt wurde. In mindestens einer Ausführungsform kann dieser Prozess mit einem weiteren Eingangsvideobild fortgesetzt werden, wenn mehr Bilder zu analysieren sind. In mindestens einer Ausführungsform kann, wenn keine Bilder mehr zu analysieren sind, ein verbesserter Videostrom oder eine verbesserte Videodatei bereitgestellt werden 516, der bzw. die zumindest einige abgeleitete Bilder beinhaltet, wobei dieses Video eine höhere Auflösung, eine höhere Bildrate, eine Sequenz neuer Bilder oder eine geringere Anzahl von ausgelassenen oder fehlenden Bildern als ein ursprünglich eingegebenes Video oder ein ursprünglich eingegebener Satz von Videobildern aufweisen kann.

[0027] In mindestens einer Ausführungsform kann ein in **Fig. 5B** veranschaulichter Prozess 550 zum Zeitpunkt der Inferenz verwendet werden, um Videobilder für Zwecke wie etwa das Verbessern von Videodaten abzuleiten. In mindestens einer Ausführungsform können ein oder mehrere Videobilder einer Videodatei oder eines Videostroms an einem trainierten Modell empfangen werden 552. In mindestens einer Ausführungsform können Eingangsbilder aus diesem Video verwendet werden, um unter Verwendung dieses trainierten Modells zusätzliche Videobilder abzuleiten 554. In mindestens einer Ausführungsform können diese abgeleiteten Videobilder verwendet werden 566, um einen verbesserten Videostrom oder eine verbesserte Videodatei zu erzeugen.

TRAINING UND ENTWICKLUNG EINES NEURONALEN NETZES

[0028] Eine zunehmende Vielfalt von Branchen und Anwendungen nutzt die Vorteile von maschinellem Lernen. In mindestens einer Ausführungsform wurden auf Prozessoren entwickelte tiefe neuronale Netze (deep neural networks - DNNs) für verschiedene Anwendungsfälle verwendet, von selbstfahrenden Autos bis zur

schnelleren Entwicklung von Medikamenten, von der automatischen Bildanalyse für Sicherheitssysteme bis zur intelligenten Echtzeit-Sprachübersetzung in Video-Chat-Anwendungen. In mindestens einer Ausführungsform ist Deep Learning eine Technik, die den neuronalen Lernprozess des menschlichen Gehirns modelliert und dabei kontinuierlich lernt, immer intelligenter wird und mit der Zeit schneller genauere Ergebnisse liefert. Ein Kind lernt zunächst von einem Erwachsenen, verschiedene Formen richtig zu erkennen und zu klassifizieren und ist schließlich in der Lage, Formen ohne Nachhilfe zu erkennen. In ähnlicher Weise müsste in mindestens einer Ausführungsform ein Deep-Learning- oder neuronales Lemsystem, das dazu ausgelegt ist, eine ähnliche Aufgabe zu erfüllen, trainiert werden, damit es intelligenter und effizienter bei der Identifizierung von grundlegenden Objekten, verdeckten Objekten usw. wird, während es diesen Objekten außerdem einen Kontext zuweist.

[0029] In mindestens einer Ausführungsform betrachten die Neuronen in einem menschlichen Gehirn verschiedene Eingaben, die empfangen werden, werden jeder dieser Eingaben Bedeutungsstufen zugewiesen und wird die Ausgabe an andere Neuronen weitergegeben, um darauf zu reagieren. Ein künstliches Neuron oder Perzeptron ist das grundlegendste Modell eines neuronalen Netzes. In mindestens einer Ausführungsform kann ein Perzeptron eine oder mehrere Eingaben empfangen, die verschiedene Merkmale eines Objekts darstellen, für dessen Erkennung und Klassifizierung ein Perzeptron trainiert ist, und wird jedem dieser Merkmale auf Grundlage der Bedeutung dieses Merkmals beim Definieren einer Form eines Objekts eine bestimmte Gewichtung zugewiesen.

[0030] Ein tiefes neuronales Netz (DNN) beinhaltet mehrere Schichten mit vielen verbundenen Perzeptronen (z. B. Knoten), die mit enormen Mengen von Eingabedaten trainiert werden können, um komplexe Probleme schnell und mit hoher Genauigkeit zu lösen. In einem Beispiel zerlegt eine erste Schicht eines DNN-Modells ein Eingangsbild eines Autos in verschiedene Abschnitte und sucht nach grundlegenden Mustern, wie etwa Linien und Winkeln. Die zweite Schicht setzt die Linien zusammen, um nach übergeordneten Mustern, wie etwa Rädern, Windschutzscheiben und Spiegeln, zu suchen. Eine nächste Schicht identifiziert einen Fahrzeugtyp und einige letzte Schichten erzeugen ein Label für ein Eingangsbild, das ein Modell einer bestimmten Automarke identifiziert. Sobald ein DNN trainiert ist, kann dieses DNN eingesetzt und zur Identifizierung und Klassifizierung von Objekten oder Mustern in einem als Inferenz bekannten Prozess verwendet werden. Beispiele für Inferenz (einen Prozess, bei dem ein DNN nützliche Informationen aus einer gegebenen Eingabe extrahiert) beinhalten die Identifizierung handgeschriebener Zahlen auf Schecks, die in Geldautomaten eingezahlt werden, die Identifizierung von Bildern von Freunden auf Fotos, die Bereitstellung von Filmempfehlungen, die Identifizierung und Klassifizierung verschiedener Arten von Automobilen, Fußgängern und Gefahren auf der Straße in fahrerlosen Autos oder die Übersetzung menschlicher Sprache in nahezu Echtzeit.

[0031] Während des Trainings fließen Daten in einer Vorwärtspropagationsphase durch ein DNN, bis eine Vorhersage erzeugt wird, die ein der Eingabe entsprechendes Label anzeigt. Wenn ein neuronales Netz die Eingaben nicht korrekt kennzeichnet, werden Fehler zwischen einem korrekten Label und einem vorhergesagten Label analysiert und die Gewichtungen werden für jedes Merkmal während einer Rückwärtspropagationsphase angepasst, bis ein DNN die Eingabe und andere Eingaben in einem Trainingsdatensatz korrekt kennzeichnet. Das Trainieren komplexer neuronaler Netze erfordert enorme Mengen von paralleler Rechenleistung, einschließlich Gleitkommamultiplikationen und -additionen, die unterstützt werden. Inferenzieren ist weniger rechenintensiv als Training, da es sich um einen latenzabhängigen Prozess handelt, bei dem ein trainiertes neuronales Netz auf neue Eingaben angewendet wird, die es nicht zuvor gesehen hat, um Bilder zu klassifizieren, Sprache zu übersetzen und neue Informationen abzuleiten.

[0032] Neuronale Netze stützen sich in hohem Maße auf mathematische Matrixoperationen und komplexe mehrschichtige Netze erfordern enorme Mengen an Gleitkommarechenleistung und Bandbreite sowohl für Effizienz als auch für Geschwindigkeit. Mit Tausenden von Verarbeitungskernen, die für mathematische Matrixoperationen optimiert sind und eine Rechenleistung von mehreren Dutzend bis Hunderten von TFLOPS liefern, kann eine Computerplattform die Rechenleistung liefern, die für Anwendungen für künstliche Intelligenz und maschinelles Lernens auf der Grundlage tiefer neuronaler Netze erforderlich ist.

[0033] Fig. 6 veranschaulicht Komponenten eines Systems 600, das in mindestens einer Ausführungsform verwendet werden kann, um maschinelles Lernen zu trainieren und zu verwenden. Wie erörtert wird, können verschiedene Komponenten durch verschiedene Kombinationen von Rechenvorrichtungen und Ressourcen oder durch ein einziges Rechnersystem bereitgestellt werden, die unter der Kontrolle einer einzigen Entität oder mehrerer Entitäten stehen kann. Darüber hinaus können Aspekte durch verschiedene Entitäten ausgelöst, initiiert oder angefordert werden. In mindestens einer Ausführungsform kann das Trainieren eines neuro-

nalen Netzes durch einen Anbieter angewiesen werden, der einer Anbieterumgebung 606 zugeordnet ist, während in mindestens einer Ausführungsform das Trainieren von einem Kunden oder einem anderen Benutzer angefordert werden kann, der über eine Client-Vorrichtung 602 oder eine andere derartige Ressource Zugang zu einer Anbieterumgebung hat. In mindestens einer Ausführungsform können Trainingsdaten (oder durch ein trainiertes neuronales Netz zu analysierende Daten) von einem Anbieter, einem Benutzer oder einem Drittanbieter von Inhalten 624 bereitgestellt werden. In mindestens einer Ausführungsform kann die Client-Vorrichtung 602 zum Beispiel ein Fahrzeug oder ein Objekt sein, das im Auftrag eines Benutzers navigiert werden soll und das Anforderungen übermitteln und/oder Anweisungen empfangen kann, die die Navigation einer Vorrichtung unterstützen.

[0034] In mindestens einer Ausführungsform können Anforderungen über mindestens ein Netzwerk 604 übermittelt werden, um in einer Anbieterumgebung 606 empfangen zu werden. In mindestens einer Ausführungsform kann es sich bei einer Client-Vorrichtung um beliebige geeignete elektronische Vorrichtungen und/oder Rechenvorrichtungen handeln, die es einem Benutzer ermöglichen, derartige Anforderungen zu erzeugen und zu senden und die Desktop-Computer, Notebook-Computer, Computer-Server, Smartphones, Tablet-Computer, Spielkonsolen (tragbar oder nicht), Computer-Prozessoren, Computer-Logik und Set-Top-Boxen beinhalten können. Das/die Netzwerk(e) 604 kann/können ein beliebiges geeignetes Netzwerk zum Übertragen einer Anforderung oder anderer derartiger Daten beinhalten, das das Internet, ein Intranet, ein Ethernet, ein Mobilfunknetz, ein lokales Netzwerk (LAN), ein Netzwerk direkter drahtloser Verbindungen zwischen Peers usw. beinhalten kann.

[0035] In mindestens einer Ausführungsform können Anforderungen an einer Schnittstellenschicht 608 empfangen werden, die in diesem Beispiel Daten an einen Trainings- und Inferenzmanager 610 weiterleiten kann. Bei diesem Manager kann es sich um ein System oder einen Dienst handeln, das bzw. der Hardware und Software für die Verwaltung von Anforderungen und Diensten, die Daten oder Inhalten entsprechen, beinhaltet. In mindestens einer Ausführungsform kann dieser Manager eine Anforderung zum Trainieren eines neuronalen Netzes empfangen und Daten für eine Anforderung an einen Trainingsmanager 612 bereitstellen. In mindestens einer Ausführungsform kann der Trainingsmanager 612 ein geeignetes zu verwendendes Modell oder Netz auswählen, wenn es nicht durch eine Anforderung vorgegeben ist, und ein Modell unter Verwendung relevanter Trainingsdaten trainieren. In mindestens einer Ausführungsform können Trainingsdaten ein Stapel von Daten sein, die in einem Trainingsdatenspeicher 614 gespeichert sind, von der Client-Vorrichtung 602 empfangen oder von einem Drittanbieter 624 erhalten werden. In mindestens einer Ausführungsform kann der Trainingsmanager 612 für die Trainingsdaten verantwortlich sein, wie etwa durch Verwendung eines LARC-basierten Ansatzes, wie in dieser Schrift erörtert. Ein Netz kann ein beliebiges geeignetes Netz sein, wie etwa ein rekurrentes neuronales Netz (RNN) oder ein neuronales Faltungsnetz (CNN). Sobald ein Netz trainiert und erfolgreich bewertet wurde, kann ein trainierte Netz zum Beispiel in einem Modellspeicher 616 gespeichert werden, der verschiedene Modelle oder Netze für Benutzer, Anwendungen oder Dienste usw. speichern kann. In mindestens einer Ausführungsform kann es mehrere Modelle für eine einzige Anwendung oder Entität geben, die auf der Grundlage einer Reihe verschiedener Faktoren verwendet werden können.

[0036] In mindestens einer Ausführungsform kann zu einem nachfolgenden Zeitpunkt eine Anforderung von Inhalten (z. B. Pfadbestimmungen) oder Daten von der Client-Vorrichtung 602 (oder einer anderen derartigen Vorrichtung) empfangen werden, die zumindest teilweise durch ein trainiertes neuronales Netz bestimmt oder beeinflusst werden. Diese Anforderung kann zum Beispiel Eingabedaten beinhalten, die unter Verwendung eines neuronalen Netzes verarbeitet werden sollen, um eine oder mehrere Inferenzen oder andere Ausgabewerte, Klassifizierungen oder Vorhersagen zu erhalten. In mindestens einer Ausführungsform können Eingabedaten an der Schnittstellenschicht 608 empfangen und an das Inferenzmodul 618 geleitet werden, obwohl auch ein anderes System oder ein anderer Dienst verwendet werden kann. In mindestens einer Ausführungsform kann das Inferenzmodul 618 ein geeignetes trainiertes Netz, wie etwa ein trainiertes tiefes neuronales Netz (DNN), wie in dieser Schrift beschrieben, aus dem Modellspeicher 616 erhalten, wenn es nicht bereits lokal in dem Inferenzmodul 618 gespeichert ist. Das Inferenzmodul 618 kann Daten als Eingabe für ein trainiertes Netz bereitstellen, das dann eine oder mehrere Inferenzen als Ausgabe erzeugen kann. Dies kann zum Beispiel eine Klassifizierung einer Instanz von Eingabedaten beinhalten. In mindestens einer Ausführungsform können die Inferenzen dann an die Client-Vorrichtung 602 zur Anzeige oder anderen Kommunikation an einen Benutzer übertragen werden. In mindestens einer Ausführungsform können Kontextdaten für einen Benutzer auch in einem Benutzerkontextdatenspeicher 622 gespeichert werden, der Daten über einen Benutzer beinhalten kann, die als Eingabe für ein Netz beim Erzeugen von Inferenzen oder Bestimmen von Daten, die nach dem Erhalten von Instanzen an einen Benutzer zurückgegeben werden. In mindestens einer Ausführungsform können relevante Daten, die zumindest einige von Eingabe- oder Inferenzdaten beinhalten

können, auch in einer lokalen Datenbank 620 zur Verarbeitung zukünftiger Anforderungen gespeichert werden. In mindestens einer Ausführungsform kann ein Benutzer Konto- oder andere Informationen verwenden, um auf Ressourcen oder Funktionalität einer Anbieterumgebung zuzugreifen. In mindestens einer Ausführungsform, können, sofern zulässig und verfügbar, auch Benutzerdaten gesammelt und zum weiteren Trainieren von Modellen verwendet werden, um genauere Inferenzen für zukünftige Anforderungen bereitzustellen. In mindestens einer Ausführungsform können Anforderungen über eine Benutzerschnittstelle in einer Anwendung für maschinelles Lernen 626, die auf der Client-Vorrichtung 602 ausgeführt wird, empfangen und die Ergebnisse über dieselbe Schnittstelle angezeigt werden. Eine Client-Vorrichtung kann Ressourcen, wie etwa einen Prozessor 628 und einen Speicher 630, zum Erzeugen einer Anforderung und Verarbeiten von Ergebnissen oder einer Antwort sowie mindestens ein Datenspeicherelement 632 zum Speichern von Daten für die Anwendung für maschinelles Lernen 626 beinhalten.

[0037] In mindestens einer Ausführungsform ist ein Prozessor 628 (oder ein Prozessor des Trainingsmanagers 612 oder des Inferenzmoduls 618) eine zentrale Verarbeitungseinheit (central processing unit - CPU). Wie bereits erwähnt, können Ressourcen in derartigen Umgebungen jedoch GPUs verwenden, um Daten zumindest für bestimmte Arten von Anforderungen zu verarbeiten. Mit Tausenden von Kernen sind GPUs dazu ausgelegt, umfangreiche parallele Arbeitslasten zu bewältigen und sind daher beim Deep Learning zum Trainieren neuronaler Netze und Erzeugen von Vorhersagen beliebt geworden. Während die Verwendung von GPUs für Offline-Erstellungen ein schnelleres Training größerer und komplexerer Modelle ermöglichte, impliziert die Offline-Erzeugung von Vorhersagen, dass entweder Eingabemerkmale zur Anforderungszeit nicht verwendet werden können oder dass Vorhersagen für alle Permutationen von Merkmalen erzeugt und in einer Lookup-Tabelle gespeichert werden müssen, um Echtzeitanforderungen zu bedienen. Wenn ein Deep-Learning-Rahmen einen CPU-Modus unterstützt und ein Modell klein und einfach genug ist, um eine Vorwärtskopplung auf einer CPU mit einer angemessenen Latenz durchzuführen, könnte ein Dienst auf einer CPU-Instanz ein Modell hosten. In diesem Fall kann das Trainieren offline auf einer GPU und die Inferenz in Echtzeit auf einer CPU durchgeführt werden. Wenn ein CPU-Ansatz nicht praktikabel ist, kann ein Dienst auf einer GPU-Instanz laufen. Da GPUs jedoch andere Eigenschaften hinsichtlich Rechenleistung und Kosten als CPUs aufweisen, kann das Ausführen eines Dienstes, der einen Laufzeitalgorithmus auf eine GPU auslagert, erfordern, dass er anders als ein CPU-basierter Dienst ausgelegt ist.

[0038] In mindestens einer Ausführungsform können Videodaten von der Client-Vorrichtung 602 zur Verbesserung in der Anbieterumgebung 606 bereitgestellt werden. In mindestens einer Ausführungsform können Videodaten zur Verbesserung an der Client-Vorrichtung 602 verarbeitet werden. In mindestens einer Ausführungsform können Videodaten von einem Drittanbieter von Inhalten 624 gestreamt und durch den Drittanbieter 624, die Anbieterumgebung 606 oder die Client-Vorrichtung 602 verbessert werden.

[0039] **Fig. 7** veranschaulicht ein System 700, das in mindestens einer Ausführungsform verwendet werden kann, um Daten zu klassifizieren oder Inferenzen zu erzeugen. In mindestens einer Ausführungsform kann sowohl überwachtes als auch selbstüberwachtes Training in mindestens einer in dieser Schrift erörterten Ausführungsform verwendet werden. In mindestens einer Ausführungsform wird ein Satz von Trainingsdaten 702 (z. B. klassifizierten oder gekennzeichneten Daten) als Eingabe bereitgestellt, um als Trainingsdaten zu fungieren. In mindestens einer Ausführungsform können Trainingsdaten Instanzen von mindestens einem Objekttyp, für den ein neuronales Netz trainiert werden soll, sowie Informationen, die diesen Objekttyp identifizieren, beinhalten. In mindestens einer Ausführungsform können Trainingsdaten einen Satz von Bildern beinhalten, die jeweils eine Darstellung eines Objekttyps beinhalten, wobei jedes Bild zudem ein Label, Metadaten, eine Klassifizierung oder eine andere Information, die einen in einem jeweiligen Bild dargestellten Objekttyp identifizieren, beinhaltet oder diesen zugeordnet ist. Als Trainingsdaten können auch verschiedene andere Datentypen verwendet werden, die Textdaten, Audiodaten, Videodaten usw. beinhalten können. In mindestens einer Ausführungsform werden die Trainingsdaten 702 als Trainingseingabe einem Trainingsmanager 704 bereitgestellt. In mindestens einer Ausführungsform kann der Trainingsmanager 704 ein System oder Dienst sein, das/der Hardware und Software beinhaltet, wie etwa eine oder mehrere Rechenvorrichtungen, die eine Trainingsanwendung zum Trainieren eines neuronalen Netzes (oder eines anderen Modells oder Algorithmus usw.) ausführen. In mindestens einer Ausführungsform empfängt der Trainingsmanager 704 eine Anweisung oder Anforderung, die eine Art des Modells angibt, das für das Training verwendet werden soll. In mindestens einer Ausführungsform kann ein Modell ein beliebiges geeignetes statistisches Modell, ein beliebiges geeignetes Netz oder ein beliebiger geeigneter Algorithmus sein, das/der für derartige Zwecke nützlich ist, und ein künstliches neuronales Netz, einen Deep-Learning-Algorithmus, einen Learning Classificator, ein Bayes'sches Netz usw. beinhalten. In mindestens einer Ausführungsform kann der Trainingsmanager 704 ein Ausgangsmodell oder ein anderes untrainiertes Modell aus einem geeigneten Speicher 706 auswählen und Trainingsdaten 702 verwenden, um ein Modell zu trainieren und ein trainiertes

Modell 708 (z. B. ein trainiertes tiefes neuronales Netz) erzeugen, das verwendet werden kann, um ähnliche Datentypen zu klassifizieren oder andere derartige Inferenzen zu erzeugen. In mindestens einer Ausführungsform, in der keine Trainingsdaten verwendet werden, kann dennoch ein geeignetes Ausgangsmodell für das Trainieren an Eingabedaten per Trainingsmanager 704 ausgewählt werden.

[0040] In mindestens einer Ausführungsform kann ein Modell auf eine Reihe von verschiedenen Arten trainiert werden, die teilweise von einer Art des ausgewählten Modells abhängig sind. In mindestens einer Ausführungsform kann einem Algorithmus für maschinelles Lernen ein Satz von Trainingsdaten bereitgestellt werden, wobei ein Modell ein durch einen Trainingsprozess erstelltes Modellartefakt ist. In mindestens einer Ausführungsform enthält jede Instanz von Trainingsdaten enthält eine richtige Antwort (z. B. eine Klassifizierung), die als Ziel oder Zielattribut bezeichnet werden kann. In mindestens einer Ausführungsform findet ein Lernalgorithmus in den Trainingsdaten Muster, die die Attribute der Eingabedaten auf ein Ziel, eine vorherzusagende Antwort, abbilden, und wird ein Modell für maschinelles Lernen ausgegeben, das diese Muster erfasst. In mindestens einer Ausführungsform kann dann ein Modell für maschinelles Lernens verwendet werden, um Vorhersagen für neue Daten zu erhalten, für die kein Ziel vorgegeben ist.

[0041] In mindestens einer Ausführungsform kann ein Trainingsmanager 704 aus einem Satz von Modellen für maschinelles Lernen auswählen, der binäre Klassifizierungs-, Multiklassen-Klassifizierungs- und Regressionsmodelle beinhaltet. In mindestens einer Ausführungsform kann die Art des zu verwendenden Modells zumindest teilweise von der Art des vorherzusagenden Ziels abhängig sein. In mindestens einer Ausführungsform sagen Modelle für maschinelles Lernen für binäre Klassifizierungsprobleme ein binäres Ergebnis vorher, wie etwa eine von zwei möglichen Klassen. In mindestens einer Ausführungsform kann ein Lemalgorithmus, wie etwa logistische Regression, verwendet werden, um binäre Klassifizierungsmodelle zu trainieren. In mindestens einer Ausführungsform ermöglichen Modelle für maschinelles Lernen für Multiklassen-Klassifizierungsprobleme, dass Vorhersagen für mehrere Klassen erzeugt werden, wie etwa um eines von mehr als zwei Ergebnissen vorherzusagen. Multinomiale logistische Regression kann für das Trainieren von Multiklassenmodellen nützlich sein. Modelle für maschinelles Lernen für Regressionsprobleme sagen einen numerischen Wert vorher. Lineare Regression kann für das Trainieren von Regressionsmodellen nützlich sein.

[0042] In mindestens einer Ausführungsform muss ein Trainingsmanager, um ein Modell für maschinelles Lernen gemäß einer Ausführungsform zu trainieren, eine Eingangstrainingsdatenquelle sowie andere Informationen, wie etwa den Namen eines Datenattributs, das ein vorherzusagendes Ziel enthält, erforderliche Datenumwandlungsanweisungen und Trainingsparameter zur Steuerung eines Lernalgorithmus, bestimmen. In mindestens einer Ausführungsform kann ein Trainingsmanager 704 während eines Trainingsprozesses automatisch einen geeigneten Lemalgorithmus auf Grundlage einer in einer Trainingsdatenquelle vorgegebenen Zielart auswählen. In mindestens einer Ausführungsform können Algorithmen für maschinelles Lernen Parameter annehmen, die verwendet werden, um bestimmte Eigenschaften eines Trainingsprozesses und eines daraus resultierenden Modells für maschinelles Lernen zu steuern. Diese werden in dieser Schrift als Trainingsparameter bezeichnet. In mindestens einer Ausführungsform kann ein Trainingsmanager, wenn keine Trainingsparameter vorgegeben sind, Standardwerte verwenden, die bekanntermaßen für einen breiten Bereich von Aufgaben für maschinelles Lernen gut funktionieren. Beispiele für Trainingsparameter, für die Werte vorgegeben werden können, beinhalten eine maximale Modellgröße, eine maximale Anzahl von Durchläufen über Trainingsdaten, einen Mischtyp, einen Regularisierungstyp, eine Lernrate und eine Regularisierungsmenge. Es können Standardeinstellungen vorgegeben werden, mit Optionen zur Anpassung der Werte zur Feinabstimmung der Rechenleistung.

[0043] In mindestens einer Ausführungsform ist eine maximale Modellgröße eine Gesamtgröße, in Byte-Einheiten, der Muster, die während des Trainierens eines Modells erstellt werden. In mindestens einer Ausführungsform kann ein Modell standardmäßig mit einer bestimmten Größe erstellt werden, wie etwa ein Modell mit 100 MB. Wenn ein Trainingsmanager nicht in der Lage ist, genügend Muster zu bestimmen, um eine Modellgröße zu füllen, kann ein kleineres Modell erstellt werden. Wenn ein Trainingsmanager mehr Muster findet als in eine vorgegebene Größe passen, kann eine maximale Obergrenze erzwungen werden, indem die Muster abgeschnitten werden, die die Qualität eines gelernten Modells am wenigsten beeinträchtigen. Die Wahl einer Modellgröße stellt Kontrolle über einen Kompromiss zwischen der Vorhersagequalität eines Modells und den Nutzungskosten bereit. In mindestens einer Ausführungsform können kleinere Modelle dazu führen, dass ein Trainingsmanager viele Muster entfernt, um in eine maximale Größenbegrenzung zu passen, was die Qualität von Vorhersagen beeinträchtigt. In mindestens einer Ausführungsform können bei größeren Modellen die Kosten für die Abfrage von Echtzeit-Vorhersagen höher sein. In mindestens einer Ausführungsform führen größere Eingabedatensätze nicht unbedingt zu größeren Modellen, da Modelle Mus-

ter und keine Eingabedaten speichern. In mindestens einer Ausführungsform ist bei wenigen und einfachen Mustern ein resultierendes Modell klein. Bei Eingabedaten, die eine große Anzahl von Rohattributen (Eingabespalten) oder abgeleiteten Merkmalen (Ausgaben von Datentransformationen) aufweisen, ist es wahrscheinlich, dass mehr Muster gefunden und während eines Trainingsprozesses gespeichert werden.

[0044] In mindestens einer Ausführungsform kann der Trainingsmanager 704 mehrere Durchläufe oder Iterationen über Trainingsdaten durchführen, um zu versuchen, Muster zu entdecken. In mindestens einer Ausführungsform kann es eine Standardanzahl von Durchläufen, wie etwa zehn Durchläufe, geben, während in mindestens einer Ausführungsform eine maximale Anzahl von Durchläufen festgelegt werden kann, wie etwa bis zu hundert Durchläufe. In mindestens einer Ausführungsform kann kein Maximum festgelegt sein oder es kann ein Konvergenzkriterium oder ein anderer Faktor festgelegt sein, der ein Ende eines Trainingsprozesses auslöst. In mindestens einer Ausführungsform kann der Trainingsmanager 704 eine Qualität von Mustern (wie etwa für die Modellkonvergenz) während des Trainings überwachen und das Training automatisch beenden, wenn es keine weiteren Datenpunkte oder Muster mehr zu entdecken gibt. In mindestens einer Ausführungsform können Datensätze mit nur wenigen Beobachtungen mehrere Durchläufe über die Daten erfordern, um eine ausreichend hohe Modellqualität zu erreichen. Größere Datensätze können viele ähnliche Datenpunkte enthalten, was eine Notwendigkeit einer großen Anzahl von Durchläufen reduzieren kann. Eine mögliche Auswirkung der Wahl von mehreren Datendurchläufen über Daten ist, dass das Modelltraining länger dauern und im Hinblick auf Ressourcen und Systemauslastung mehr kosten kann.

[0045] In mindestens einer Ausführungsform werden die Trainingsdaten vor dem Training oder zwischen Durchläufen von Training gemischt. In mindestens einer Ausführungsform ist das Mischen ein zufälliges oder pseudozufälliges Mischen, um eine wirklich zufällige Reihenfolge zu erzeugen, obwohl es einige Beschränkungen geben kann, um sicherzustellen, dass es keine Gruppierung bestimmter Datentypen gibt, oder gemischte Daten können neu gemischt werden, wenn eine derartige Gruppierung besteht, usw. In mindestens einer Ausführungsform wird durch das Mischen eine Reihenfolge oder Anordnung, in der Daten zum Trainieren verwendet werden, geändert, sodass ein Trainingsalgorithmus nicht auf Gruppierungen ähnlicher Datentypen oder auf einen einzigen Datentyp für zu viele Beobachtungen in Folge trifft. In mindestens einer Ausführungsform könnte ein Modell trainiert werden, um ein Objekt vorherzusagen. In mindestens einer Ausführungsform könnten Daten vor dem Hochladen nach Objekttyp sortiert werden. In mindestens einer Ausführungsform kann dann ein Algorithmus die Daten alphabetisch nach Objekttyp verarbeiten, wobei er zuerst nur auf Daten für einen bestimmten Objekttyp trifft. In mindestens einer Ausführungsform beginnt ein Modell, Muster für diesen Objekttyp zu lernen. In mindestens einer Ausführungsform trifft ein Modell dann nur auf Daten für einen zweiten Objekttyp und versucht ein Modell so anzupassen, dass es zu diesem Objekttyp passt, was zu einer Verschlechterung von Mustern führen kann, die zu einem ersten Objekttyp passen. Dieser plötzliche Wechsel zwischen Objekttypen kann ein Modell erzeugen, das nicht lernt, Objekttypen genau vorherzusagen. In mindestens einer Ausführungsform kann das Mischen in mindestens einer Ausführungsform durchgeführt werden, bevor ein Trainingsdatensatz in Trainings- und Bewertungsteilsätze aufgeteilt wird, sodass eine relativ gleichmäßige Verteilung von Datentypen für beide Phasen verwendet wird. In mindestens einer Ausführungsform kann der Trainingsmanager 704 die Daten automatisch mischen, zum Beispiel unter Verwendung einer pseudozufälligen Mischtechnik.

[0046] In mindestens einer Ausführungsform kann der Trainingsmanager 704 beim Erstellen eines Modells für maschinelles Lernen in mindestens einer Ausführungsform einem Benutzer ermöglichen, Einstellungen vorzugeben oder benutzerdefinierte Optionen anzuwenden. In mindestens einer Ausführungsform kann ein Benutzer eine oder mehrere Bewertungseinstellungen vorgeben, die einen Abschnitt der Eingabedaten angeben, der für die Bewertung einer Vorhersagequalität eines Modells für maschinelles Lernen reserviert werden soll. In mindestens einer Ausführungsform kann ein Benutzer eine Richtlinie vorgeben, die angibt, welche Attribute und Attributtransformationen für das Modelltraining verfügbar sind. In mindestens einer Ausführungsform kann der Benutzer außerdem verschiedene Trainingsparameter vorgeben, die bestimmte Eigenschaften eines Trainingsprozesses und eines daraus resultierenden Modells steuern.

[0047] In mindestens einer Ausführungsform kann, sobald ein Trainingsmanager bestimmt hat, dass das Training eines Modells abgeschlossen ist, wie etwa unter Verwendung mindestens eines der in dieser Schrift erörterten Endkriterien, das trainierte Modell 708 zur Verwendung durch einen Klassifikator 714 beim Klassifizieren von (oder anderweitigen Erzeugung von Inferenzen für) Validierungsdaten 712 bereitgestellt werden. In mindestens einer Ausführungsform beinhaltet dies einen logischen Übergang zwischen einem Trainingsmodus für ein Modell und einem Inferenzmodus für ein Modell. In mindestens einer Ausführungsform wird das trainierte Modell 708 jedoch zuerst an einen Bewerter 710 weitergegeben, der eine Anwendung, einen Prozess oder einen Dienst beinhalten kann, die/der auf mindestens einer Rechenressource (z. B. einer CPU

oder GPU mindestens eines Servers) zum Bewerten einer Qualität (oder eines anderen derartigen Aspekts) eines trainierten Modells ausgeführt wird. In mindestens einer Ausführungsform wird ein Modell bewertet, um zu bestimmen, ob dieses Modell bei der Vorhersage eines Ziels an neuen und zukünftigen Daten zumindest ein akzeptables Mindest- oder Schwellenniveau an Rechenleistung bereitstellt. Ist dies nicht der Fall, kann der Trainingsmanager 704 damit fortfahren, dieses Modell zu trainieren. In mindestens einer Ausführungsform kann, da zukünftige Dateninstanzen oft unbekannte Zielwerte aufweisen, es wünschenswert sein, eine Genauigkeitsmetrik des maschinellen Lernens an Daten zu überprüfen, für die eine Zielantwort bekannt ist, und diese Bewertung als Ersatz für die Vorhersagegenauigkeit an zukünftigen Daten zu verwenden.

[0048] In mindestens einer Ausführungsform wird ein Modell unter Verwendung eines Teilsatzes von Trainingsdaten 702 bewertet, die für das Training bereitgestellt wurden. Dieser Teilsatz kann unter Verwendung eines Misch- und Aufteilansatzes, wie vorstehend erörtert, bestimmt werden. In mindestens einer Ausführungsform wird dieser Teilsatz von Bewertungsdaten mit einem Ziel gekennzeichnet und kann somit als Ground-Truth-Quelle für die Bewertung dienen. Das Bewerten einer Vorhersagegenauigkeit eines Modells für maschinelles Lernen mit denselben Daten, die zum Trainieren verwendet wurden, ist nicht sinnvoll, da positive Bewertungen für Modelle erzeugt werden könnten, die sich an die Trainingsdaten erinnern anstatt aus ihnen zu verallgemeinern. In mindestens einer Ausführungsform wird, sobald das Training abgeschlossen ist, der Teilsatz der Bewertungsdaten unter Verwendung des trainierten Modells 708 verarbeitet und der Bewerter 710 kann die Genauigkeit dieses Modells bestimmen, indem er die Ground-Truth-Daten mit entsprechenden Ausgaben (oder Vorhersagen/Beobachtungen) dieses Modells vergleicht. In mindestens einer Ausführungsform kann der Bewerter 710 in mindestens einer Ausführungsform eine Zusammenfassung oder Rechenleistungsmetrik bereitstellen, die angibt, wie gut vorhergesagte und wahre Werte übereinstimmen. In mindestens einer Ausführungsform kann, wenn ein trainiertes Modell nicht zumindest ein Kriterium für die minimale Rechenleistung oder einen anderen derartigen Genauigkeitsschwellenwert erfüllt, dann der Trainingsmanager 704 angewiesen werden, weiteres Training durchzuführen oder in einigen Fällen versuchen, ein neues oder anderes Modell zu trainieren. In mindestens einer Ausführungsform kann, wenn das trainierte Modell 708 die relevanten Kriterien erfüllt, dann ein trainiertes Modell zur Verwendung durch den Klassifikator 714 bereitgestellt werden.

[0049] In mindestens einer Ausführungsform kann es beim Erstellen und Trainieren eines Modells für maschinelles Lernen in mindestens einer Ausführungsform wünschenswert sein, Modelleinstellungen oder Trainingsparameter vorzugeben, die zu einem Modell führen, dass in der Lage ist, genaue Vorhersagen zu treffen. In mindestens einer Ausführungsform beinhalten die Parameter die Anzahl von durchzuführenden Durchläufen (vorwärts und/oder rückwärts), eine Regularisierung oder Verfeinerung, eine Modellgröße und einen Mischtyp. In mindestens einer Ausführungsform könnte das Auswählen von Modellparametereinstellungen, die eine beste Vorhersageleistung an Bewertungsdaten erzeugen, zu einer Überanpassung eines Modells führen. In mindestens einer Ausführungsform kommt es zu einer Überanpassung, wenn sich ein Modell Muster gemerkt hat, die in Trainings- und Bewertungsdatenquellen vorkommen, aber Muster in Daten nicht verallgemeinert hat. Eine Überanpassung tritt häufig auf, wenn die Trainingsdaten alle in einer Bewertung verwendeten Daten beinhalten. In mindestens einer Ausführungsform kann ein Modell, das überangepasst wurde, bei der Bewertung eine gute Leistung erbringen, aber an neuen oder anderweitigen Validierungsdaten keine genauen Vorhersagen treffen. In mindestens einer Ausführungsform kann ein Trainingsmanager, um zu vermeiden, dass ein überangepasstes Modell als bestes Modell ausgewählt wird, zusätzliche Daten reservieren, um die Rechenleistung eines Modells zu validieren. Zum Beispiel könnte der Trainingsdatensatz in 60 Prozent für das Training und 40 Prozent für die Bewertung oder Validierung aufgeteilt werden, die wiederum in zwei oder mehr Phasen unterteilt werden können. In mindestens einer Ausführungsform kann nach dem Auswählen von Modellparametern, die für die Evaluierungsdaten gut funktionieren und zu einer Konvergenz bei einer Teilmenge von Validierungsdaten, wie etwa der Hälfte dieser Validierungsdaten, führen, eine zweite Validierung mit einem Rest dieser Validierungsdaten durchgeführt werden, um die Rechenleistung dieses Modells sicherzustellen. Wenn dieses Modell die Erwartungen an die Validierungsdaten erfüllt, dann erfolgt keine Überanpassung der Daten durch dieses Modell. In mindestens einer Ausführungsform kann ein Testsatz oder ein zurückgehaltener Satz zum Testen der Parameter verwendet werden. In mindestens einer Ausführungsform hilft das Verwenden eines zweiten Validierungs- oder Testschritts bei der Auswahl geeigneter Modellparameter, um eine Überanpassung zu verhindern. Durch das Zurückhalten mehrere Daten aus einem Trainingsprozess zur Validierung sind jedoch weniger Daten für das Training verfügbar. Dies kann bei kleineren Datensätzen problematisch sein, da nicht genügend Daten für das Training verfügbar sein können. In mindestens einer Ausführungsform besteht ein Ansatz in einer derartigen Situation darin, eine Kreuzvalidierung durchzuführen, wie an anderer Stelle in dieser Schrift erörtert.

[0050] In mindestens einer Ausführungsform gibt es viele Metriken oder Einblicke, die verwendet werden können, um die Vorhersagegenauigkeit eines bestimmten Modells zu überprüfen und zu bewerten. In mindestens einer Ausführungsform enthält ein Bewertungsergebnis eine Metrik für die Vorhersagegenauigkeit, um einen Gesamterfolg eines Modells anzugeben, sowie Visualisierungen, um dabei zu helfen, die Genauigkeit eines Modells über eine Metrik für die Vorhersagegenauigkeit hinaus zu untersuchen. Ein Ergebnis kann auch eine Möglichkeit bereitstellen, die Auswirkung des Festlegens einer Bewertungsschwelle, wie etwa für eine binäre Klassifizierung, zu überprüfen, und kann Warnmeldungen zu Kriterien zur Überprüfung der Gültigkeit einer Bewertung erzeugen. Eine Wahl einer Metrik und einer Visualisierung kann zumindest teilweise von einer Art des zu bewertenden Modells abhängig sein.

[0051] In mindestens einer Ausführungsform kann ein trainiertes Modell für maschinelles Lernen, sobald es trainiert und zufriedenstellend bewertet wurde, verwendet werden, um eine Anwendung für maschinelles Lernen zu erstellen oder zu unterstützen. In einer Ausführungsform ist das Erstellen einer Anwendung für maschinelles Lernen ein iterativer Prozess, der eine Abfolge von Schritten beinhaltet. In mindestens einer Ausführungsform kann/können (ein) Kernproblem(e) des maschinellen Lernens in Bezug darauf, was beobachtet wird und welche Antwort ein Modell vorhersagen soll, formuliert werden. In mindestens einer Ausführungsform können dann Daten gesammelt, bereinigt und aufbereitet werden, um die Daten für die Verwendung durch Trainingsalgorithmen für Modelle für maschinelles Lernen geeignet zu machen. Diese Daten können visualisiert und analysiert werden, um Integritätsprüfungen durchzuführen, um eine Qualität von Daten zu validieren und die Daten zu verstehen. Es kann sein, dass Rohdaten (z. B. Eingabevariablen) und Antwortdaten (z. B. ein Ziel) nicht in einer Weise dargestellt sind, die zum Trainieren eines Modells mit hoher Vorhersageleistung verwendet werden kann. Daher kann es wünschenswert sein, Eingabedarstellungen oder -merkmale mit besserer Vorhersageleistung aus Rohvariablen zu konstruieren. Die sich daraus ergebenden Merkmale können einem Lemalgorithmus zugeführt werden, um Modelle zu erstellen und eine Qualität von Modellen an Daten zu bewerten, die aus der Modellbildung zurückgehalten wurden. Ein Modell kann dann verwendet werden, um Vorhersagen einer Zielantwort für neue Dateninstanzen zu erzeugen.

[0052] In mindestens einer Ausführungsform wird in dem System 700 aus **Fig. 7** ein trainiertes Modell 710 nach der Auswertung einem Klassifikator 714 bereitgestellt oder zur Verfügung gestellt, der in der Lage ist, ein trainiertes Modell zu verwenden, um Validierungsdaten zu verarbeiten. In mindestens einer Ausführungsform kann dies zum Beispiel Daten beinhalten, die von Nutzern oder Dritten empfangen werden und nicht klassifiziert sind, wie etwa Abfragebilder, die nach Informationen darüber suchen, was in diesen Bildern dargestellt ist. In mindestens einer Ausführungsform können die Validierungsdaten durch einen Klassifikator unter Verwendung eines trainierten Modells verarbeitet werden, und die Ergebnisse 716 (wie etwa Klassifizierungen oder Vorhersagen), die erzeugt werden, können an jeweilige Quellen zurückgesendet oder anderweitig verarbeitet oder gespeichert werden. In mindestens einer Ausführungsform und sofern eine derartige Verwendung zulässig ist, können diese nun klassifizierten Dateninstanzen in einem Trainingsdatenspeicher gespeichert werden, der für zukünftiges Trainieren des trainierten Modells 708 durch einen Trainingsmanager verwendet werden kann. In mindestens einer Ausführungsform wird ein Modell kontinuierlich trainiert, wenn neue Daten verfügbar sind, aber in mindestens einer Ausführungsform werden diese Modelle periodisch neu trainiert, wie etwa einmal pro Tag oder Woche, in Abhängigkeit von Faktoren, wie etwa einer Größe eines Datensatzes oder einer Komplexität eines Modells.

[0053] In mindestens einer Ausführungsform kann der Klassifikator 714 geeignete Hardware und Software zur Verarbeitung von Validierungsdaten 712 unter Verwendung eines trainierten Modells beinhalten. In mindestens einer Ausführungsform beinhaltet ein Klassifikator einen oder mehrere Computerserver, die jeweils eine oder mehrere Grafikverarbeitungseinheiten (GPUs) aufweisen, die in der Lage sind, Daten zu verarbeiten. In mindestens einer Ausführungsform können die Konfiguration und das Design von GPUs sie für die Verwendung bei der Verarbeitung von Daten für maschinelles Lernens wünschenswerter machen als CPUs oder andere derartige Komponenten. In mindestens einer Ausführungsform kann ein trainiertes Modell in mindestens einer Ausführungsform in den GPU-Speicher geladen und eine empfangene Dateninstanz einer GPU zur Verarbeitung bereitgestellt werden. GPUs können eine viel größere Anzahl von Kernen aufweisen als CPUs, und GPU-Kerne können auch viel weniger komplex sein. In mindestens einer Ausführungsform kann eine bestimmte GPU in der Lage sein, Tausende von Dateninstanzen gleichzeitig über verschiedene Hardware-Threads zu verarbeiten. In mindestens einer Ausführungsform kann eine GPU zudem dazu konfiguriert sein, den Gleitkommadurchsatz zu maximieren, was bedeutende zusätzliche Verarbeitungsvorteile für einen großen Datensatz bereitstellen kann.

[0054] In mindestens einer Ausführungsform können derartige Aufgaben selbst bei Verwendung von GPUs, Beschleunigern und anderer derartiger Hardware zur Beschleunigung von Aufgaben wie etwa dem Trainieren

eines Modells oder der Klassifizierung von Daten unter Verwendung eines derartigen Modells immer noch bedeutende Zeit, Ressourcenzuweisung und Kosten erfordern. In mindestens einer Ausführungsform müssten, wenn ein Modell für maschinelles Lernen in 700 Durchläufen trainiert werden soll und ein Datensatz 1.000.000 Dateninstanzen beinhaltet, die für das Trainieren verwendet werden sollen, alle Millionen Instanzen für jeden Durchlauf verarbeitet werden. Verschiedene Abschnitte einer Architektur können zudem von verschiedenen Arten von Vorrichtungen unterstützt werden. In mindestens einer Ausführungsform kann das Training unter Verwendung einer Reihe von Servern an einem logisch zentralisierten Ort durchgeführt werden, die als Dienst angeboten werden können, während die Klassifizierung der Rohdaten unter anderem durch einen derartigen Dienst oder auf einer Client-Vorrichtung durchgeführt werden kann. Diese Vorrichtungen können auch im Besitz einer gleichen Entität oder mehrerer Entitäten sein und von diesen betrieben oder gesteuert werden.

[0055] In mindestens einer Ausführungsform kann ein in **Fig. 8** veranschaulichtes beispielhaftes neuronales Netz 800 in mindestens einer Ausführungsform trainiert oder anderweitig verwendet werden. In mindestens einer Ausführungsform ist ein statistisches Modell ein künstliches neuronales Netz (artificial neural network - ANN), das mehrere Schichten von Knoten enthält, einschließlich einer Eingabeschicht 802, einer Ausgabeschicht 806 und mehrerer Schichten 804 von Zwischenknoten, die oft als „verborgene“ Schichten bezeichnet werden, da die internen Schichten und Knoten in neuronalen Netzen normalerweise nicht sichtbar oder zugänglich sind. Obwohl in mindestens einer Ausführungsform zu Erklärungszwecken nur einige wenige Zwischenschichten dargestellt sind, versteht es sich, dass es keine Begrenzung für die Anzahl der Zwischenschichten gibt, die verwendet werden können, und dass eine beliebige Begrenzung der Schichten oft ein Faktor der Ressourcen oder der Zeit ist, die für die Verarbeitung unter Verwendung eines Modell erforderlich sind. In mindestens einer Ausführungsform können zusätzliche Arten von Modellen, Netzen, Algorithmen oder Prozessen ebenfalls verwendet werden, die unter anderem andere Anzahlen oder Auswahlen von Knoten und Schichten beinhalten können. In mindestens einer Ausführungsform können die Validierungsdaten durch Schichten eines Netzes verarbeitet werden, um einen Satz von Inferenzen oder Inferenzwerten zu erzeugen, die dann einer Verlustfunktion 808 zugeführt werden können.

[0056] In mindestens einer Ausführungsform sind alle Knoten einer bestimmten Schicht mit allen Knoten einer benachbarten Schicht verbunden. In mindestens einer Ausführungsform sind die Knoten einer Zwischenschicht dann jeweils mit den Knoten zweier benachbarter Schichten verbunden. In mindestens einer Ausführungsform werden die Knoten in einigen Modellen auch als Neuronen oder verbundene Einheiten bezeichnet und die Verbindungen zwischen den Knoten werden als Kanten bezeichnet. Jeder Knoten kann eine Funktion für empfangene Eingaben ausführen, wie etwa durch Verwenden einer vorgegebenen Funktion. In mindestens einer Ausführungsform können Knoten und Kanten während des Trainings unterschiedliche Gewichtungen erhalten und einzelne Schichten von Knoten können bestimmte Arten von Transformationen an empfangenen Eingaben durchführen, wobei diese Transformationen auch während des Trainings gelernt oder angepasst werden können. In mindestens einer Ausführungsform kann das Lernen überwachtes oder selbstüberwachtes Lernen sein, was zumindest teilweise von einer Art der in einem Trainingsdatensatz enthaltenen Informationen abhängig sein kann. In mindestens einer Ausführungsform können verschiedene Arten von neuronalen Netzen verwendet werden, die ein neuronales Faltungsnetz (convolutional neural network - CNN), das eine Reihe von Faltungsschichten und einen Satz von Pooling-Schichten beinhaltet, beinhalten können und sich für Anwendungen wie etwa Bilderkennung als vorteilhaft erwiesen haben. CNNs können aufgrund einer relativ kleinen Anzahl von zu bestimmenden Parametern auch einfacher zu trainieren sein als andere Netze.

[0057] In mindestens einer Ausführungsform kann ein derartiges komplexes Modell für maschinelles Lernen unter Verwendung verschiedener Abstimmungsparametern trainiert werden. Das Auswählen der Parameter, das Anpassen eines Modells und das Bewerten eines Modells sind Teile eines Modellabstimmungsprozesses, der oft als Hyperparameteroptimierung bezeichnet wird. Eine derartige Abstimmung kann in mindestens einer Ausführungsform Introspektion eines zugrundeliegenden Modells oder zugrundeliegender Daten beinhalten. In einer Trainings- oder Produktionsumgebung kann ein robuster Arbeitsablauf wichtig sein, um eine Überanpassung von Hyperparametern zu vermeiden, wie an anderer Stelle in dieser Schrift erörtert. Die Kreuzvalidierung und das Hinzufügen von Gaußschem Rauschen zu einem Trainingsdatensatz sind Techniken, die nützlich sein können, um eine Überanpassung an einen beliebigen Datensatz zu vermeiden. Für die Optimierung der Hyperparameter kann es wünschenswert sein, die Trainings- und Validierungssätze fest zu halten. In mindestens einer Ausführungsform können die Hyperparameter in bestimmten Kategorien abgestimmt werden, die unter anderem Datenvorverarbeitung (wie etwa Übersetzung von Wörtern in Vektoren), Definition der CNN-Architektur (wie etwa Filtergrößen, Anzahl der Filter), Parameter des stochastischen Gra-

dientenabstiegs (stochastic gradient descent - SGD) (zum Beispiel Lernrate) und Regularisierung oder Verfeinerung (zum Beispiel Dropout-Wahrscheinlichkeit) beinhalten können.

[0058] In mindestens einer Ausführungsform können die Instanzen eines Datensatzes während der Vorverarbeitung in einen niedrigeren dimensionalen Raum mit einer bestimmten Größe eingebettet werden. In mindestens einer Ausführungsform ist die Größe dieses Raums ein abzustimmender Parameter. In mindestens einer Ausführungsform enthält eine Architektur eines CNN viele abstimmbare Parameter. Ein Parameter für Filtergrößen kann eine Interpretation von Informationen darstellen, die einer Größe einer zu analysierenden Instanz entspricht. In der Computerlinguistik wird dies als n-Gramm-Größe bezeichnet. Ein beispielhaftes CNN verwendet drei verschiedene Filtergrößen, die potenziell unterschiedliche n-Gramm-Größen darstellen. Eine Anzahl von Filtern pro Filtergröße kann einer Tiefe eines Filters entsprechen. Jeder Filter versucht, etwas anderes aus einer Struktur einer Instanz zu lernen, wie etwa eine Satzstruktur für Textdaten. In einer Faltungsschicht kann eine Aktivierungsfunktion eine gleichgerichtete lineare Einheit und ein Pooling-Typ sein, der als Max-Pooling festgelegt ist. Die Ergebnisse können dann zu einem eindimensionalen Vektor verkettet werden und eine letzte Schicht ist vollständig mit einer zweidimensionalen Ausgabe verbunden. Dies entspricht einer binären Klassifizierung, auf die eine Optimierungsfunktion angewendet werden kann. Eine derartige Funktion ist eine Implementierung eines quadratischen Mittelwert-(Root Mean Square - RMS-)Propagationsverfahrens des Gradientenabstiegs, wobei beispielhafte Hyperparameter die Lemrate, die Batchgröße, die maximale Gradientennormale und Epochen beinhalten können. Bei neuronalen Netzen kann die Regularisierung ein äußerst wichtiger Faktor sein. In mindestens einer Ausführungsform können die Eingabedaten relativ spärlich sein. Ein wichtiger Hyperparameter in einer derartigen Situation kann ein Dropout auf einer vorletzten Schicht sein, der einen Anteil der Knoten darstellt, die nicht in jedem Trainingszyklus „feuern“. Ein beispielhafter Trainingsprozess kann verschiedene Hyperparameterkonfigurationen auf Grundlage einer Rückmeldung über die Rechenleistung vorheriger Konfigurationen vorschlagen. Dieses Modell kann mit einer vorgeschlagenen Konfiguration trainiert werden, an einem designierten Validierungssatz bewertet werden und die Rechenleistung kann berichtet werden. Dieser Prozess kann wiederholt werden, um zum Beispiel einen Kompromiss zwischen Exploration (mehr über verschiedene Konfigurationen lernen) und Exploitation (vorheriges Wissen nutzen, um bessere Ergebnisse zu erreichen) zu finden.

[0059] Da das Training von CNNs parallelisiert werden kann und GPU-fähige Rechenressourcen genutzt werden können, können mehrere Optimierungsstrategien für verschiedene Szenarien versucht werden. Ein komplexes Szenario ermöglicht das Abstimmen der Modellarchitektur und der Parameter für die Vorverarbeitung und den stochastischen Gradientenabstieg. Dadurch wird ein Modellkonfigurationsraum erweitert. In einem Basisszenario werden nur die Parameter für die Vorverarbeitung und den stochastischen Gradientenabstieg abgestimmt. In einem komplexen Szenario kann es eine größere Anzahl von Konfigurationsparametern geben als in einem Basisszenario. Das Abstimmen in einem gemeinsamen Raum kann unter Verwendung einer linearen oder exponentiellen Anzahl von Schritten durchgeführt werden, wobei eine Optimierungsschleife für Modelle durchlaufen wird. Die Kosten für einen derartigen Abstimmungsprozess können bedeutend geringer sein als für Abstimmungsprozesse, wie etwa Zufallssuche und Gittersuche, ohne einen bedeutenden Verlust der Rechenleistung.

[0060] In mindestens einer Ausführungsform kann Rückpropagierung verwendet werden, um einen Gradienten zu berechnen, der zum Bestimmen von Gewichtungen für ein neuronales Netz verwendet wird. Rückpropagierung ist eine Form der Differenzierung und kann von einem Gradientenabstiegs-Optimierungsalgorithmus verwendet werden, um die auf verschiedene Knoten oder Neuronen angewendeten Gewichtungen anzupassen, wie vorstehend erörtert. Die Gewichtungen können unter Verwendung eines Gradienten einer relevanten Verlustfunktion bestimmt werden. Die Rückpropagierung kann eine Ableitung einer Verlustfunktion in Bezug auf die durch ein statistisches Modell erzeugte Ausgabe verwenden. Wie bereits erwähnt, können verschiedene Knoten zugeordnete Aktivierungsfunktionen aufweisen, die die Ausgabe jeweiliger Knoten definieren. Verschiedene Aktivierungsfunktionen können nach Bedarf verwendet werden und radiale Basisfunktionen (RBFs) und Sigmoide beinhalten, die von verschiedenen Stützvektormaschinen (support vector machines - SVM) zur Transformation von Daten verwendet werden können. Eine Aktivierungsfunktion einer Zwischenschicht von Knoten wird in dieser Schrift als Kernel des inneren Produkts bezeichnet. Diese Funktionen können zum Beispiel Identitätsfunktionen, Stufenfunktionen, Sigmoidalfunktionen, Rampenfunktionen usw. beinhalten. Aktivierungsfunktionen können unter anderem auch linear oder nicht linear sein.

[0061] In mindestens einer Ausführungsform wird ein untrainiertes neuronales Netz unter Verwendung eines Trainingsdatensatzes trainiert. In mindestens einer Ausführungsform ist der Trainingsrahmen ein PyTorch-Rahmen, ein TensorFlow-, Boost-, Caffe-, Microsoft-Cognitive-Toolkit/CNTK-, MXNet-, Chainer-, Keras-, Deepleaming4j- oder ein anderer Trainingsrahmen. In mindestens einer Ausführungsform trainiert der Trai-

ningsrahmen ein untrainiertes neuronales Netz und ermöglicht, dass es unter Verwendung von in dieser Schrift beschriebenen Verarbeitungsressourcen trainiert wird, um ein trainiertes neuronales Netz zu erzeugen. In mindestens einer Ausführungsform können die Gewichtungen zufällig oder durch Vortraining unter Verwendung eines Deep-Belief-Netzes gewählt werden. In mindestens einer Ausführungsform kann das Training entweder auf überwachte, teilweise überwachte, selbstüberwachte oder nicht überwachte Weise durchgeführt werden.

[0062] In mindestens einer Ausführungsform wird das untrainierte neuronale Netz unter Verwendung von überwachtem Lernen trainiert, wobei der Trainingsdatensatz eine Eingabe beinhaltet, die mit einer gewünschten Ausgabe für eine Eingabe gepaart ist, oder wobei der Trainingsdatensatz eine Eingabe beinhaltet, die eine bekannte Ausgabe aufweist, und eine Ausgabe des neuronalen Netzes manuell bewertet wird. In mindestens einer Ausführungsform wird das untrainierte neuronale Netz auf überwachte Weise trainiert, verarbeitet es Eingaben aus dem Trainingsdatensatz und vergleicht es die resultierenden Ausgaben mit einem Satz von erwarteten oder gewünschten Ausgaben. In mindestens einer Ausführungsform werden Fehler dann durch das untrainierte neuronale Netz rückpropagiert. In mindestens einer Ausführungsform passt der Trainingsrahmen Gewichtungen an, die das untrainierte neuronale Netz steuern. In mindestens einer Ausführungsform beinhaltet der Trainingsrahmen Werkzeuge, um zu überwachen, wie gut das untrainierte neuronale Netz zu einem Modell konvergiert, wie etwa dem trainierten neuronalen Netz, das dazu geeignet ist, korrekte Antworten zu erzeugen, wie etwa in dem Ergebnis, die auf bekannten Eingabedaten, wie etwa neuen Daten, basieren. In mindestens einer Ausführungsform trainiert der Trainingsrahmen das untrainierte neuronale Netz wiederholt, während die Gewichtungen angepasst werden, um eine Ausgabe des untrainierten neuronalen Netzes unter Verwendung einer Verlustfunktion und eines Anpassungsalgorithmus, wie etwa des stochastischen Gradientenabstiegs, zu verfeinern. In mindestens einer Ausführungsform trainiert der Trainingsrahmen das untrainierte neuronale Netz, bis das untrainierte neuronale Netz eine gewünschte Genauigkeit erreicht. In mindestens einer Ausführungsform kann das trainierte neuronale Netz dann eingesetzt werden, um eine beliebige Anzahl von Operationen für maschinelles Lernen zu implementieren.

[0063] In mindestens einer Ausführungsform wird das untrainierte neuronale Netz unter Verwendung von selbstüberwachtem Lernen trainiert, wobei das untrainierte neuronale Netz versucht, sich selbst unter Verwendung von nicht gekennzeichneten Daten zu trainieren. In mindestens einer Ausführungsform beinhaltet der Trainingsdatensatz für selbstüberwachtes Lernen Eingabedaten ohne beliebige zugeordnete Ausgabedaten oder „Ground-Truth“-Daten. In mindestens einer Ausführungsform kann das untrainierte neuronale Netz Gruppierungen innerhalb des Trainingsdatensatzes lernen und bestimmen, wie einzelne Eingaben mit dem untrainierten Datensatz in Beziehung stehen. In mindestens einer Ausführungsform kann selbstüberwachtes Training verwendet werden, um eine selbstorganisierende Karte zu erzeugen, die eine Art von trainiertem neuronalen Netz ist, das in der Lage ist, Operationen durchzuführen, die beim Reduzieren der Dimensionalität neuer Daten nützlich sind. In mindestens einer Ausführungsform kann selbstüberwachtes Training auch verwendet werden, um Anomaliedetektion durchzuführen, was die Identifizierung von Datenpunkten in einem neuen Datensatz ermöglicht, die von normalen Mustern des neuen Datensatzes abweichen.

[0064] In mindestens einer Ausführungsform kann halbüberwachtes Lernen verwendet werden, wobei es sich um eine Technik handelt, bei welcher der Trainingsdatensatz eine Mischung aus gekennzeichneten und nicht gekennzeichneten Daten beinhaltet. In mindestens einer Ausführungsform kann der Trainingsrahmen verwendet werden, um inkrementelles Lernen durchzuführen, wie etwa durch Transferlerntechniken. In mindestens einer Ausführungsform ermöglicht das inkrementelle Lernen es dem trainierten neuronalen Netz, sich an neue Daten anzupassen, ohne das Wissen zu vergessen, das dem trainierten neuronalen Netz während des anfänglichen Trainings beigebracht wurde.

INFERENZ- UND TRAININGSLOGIK

[0065] Fig. 9A veranschaulicht Inferenz- und/oder Trainingslogik 915, die verwendet wird, um Inferenz- und/oder Trainingsoperationen durchzuführen, die mit einer oder mehreren Ausführungsformen assoziiert sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt.

[0066] In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 ohne Einschränkung Code- und/oder Datenspeicher 901 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 915 einen Code- und/oder Datenspeicher 901 beinhalten oder an diesen gekoppelt sein, um Graphencode oder andere Software zum Steuern der Zeitsteuerung und/oder Reihenfolge zu speichern, in der Gewichtungs- und/oder andere Parameterinformationen geladen werden sollen, um Logik zu konfigurieren, einschließlich Integer- und/oder Gleitkommaeinheiten (zusammen als arithmetisch-logische Einheiten (arithmetic logic units - ALUs) bezeichnet). In mindestens einer Ausführungsform lädt Code, wie etwa Graphencode, Gewichtungs- oder andere Parameterinformationen in Prozessor-ALUs auf Grundlage einer Architektur eines neuronalen Netzes, dem der Code entspricht. In mindestens einer Ausführungsform speichert der Code- und/oder Datenspeicher 901 Gewichtungsparameter und/oder Eingabe-/Ausgabedaten jeder Schicht eines neuronalen Netzes, die in Verbindung mit einer oder mehreren Ausführungsformen trainiert oder verwendet wird, während der Vorwärtspropagierung von Eingabe-/Ausgabedaten und/oder Gewichtungsparametern während des Trainierens und/oder Inferenzierens unter Verwendung von Aspekten einer oder mehrerer Ausführungsformen. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Code- und/oder Datenspeichers 901 in einem anderen chipinternen oder chipextemen Datenspeicher, einschließlich des L1-, L2- oder L3-Caches oder Systemspeichers eines Prozessors, enthalten sein.

[0067] In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Code- und/oder Datenspeichers 901 zu einem oder mehreren Prozessoren oder anderen Hardware-Logikvorrichtungen oder -Schaltungen intern oder extern sein. In mindestens einer Ausführungsform kann der Code und/oder Code- und/oder Datenspeicher 901 Cache-Speicher, dynamischer zufällig adressierbarer Speicher (dynamic randomly addressable memory - „DRAM“), statischer zufällig adressierbarer Speicher (static randomly addressable memory - „SRAM“), nicht flüchtiger Speicher (z. B. Flash-Speicher) oder anderer Speicher sein. In mindestens einer Ausführungsform kann die Wahl, ob der Code und/oder Code- und/oder Datenspeicher 901 zu einem Prozessor, der zum Beispiel aus DRAM, SRAM, Flash oder einer anderen Speicherart besteht, intern oder extern ist, von dem verfügbaren Speicher auf dem Chip oder außerhalb des Chips, den Latenzanforderungen der durchzuführenden Trainings- und/oder Ableitungsfunktionen, der Batchgröße der Daten, die bei der Ableitung und/oder dem Training eines neuronalen Netzes verwendet werden, oder einer Kombination dieser Faktoren abhängen.

[0068] In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 ohne Einschränkung einen Code- und/oder Datenspeicher 905 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 905 Gewichtungsparameter und/oder Eingabe-/Ausgabedaten jeder Schicht eines neuronalen Netzes, die in Verbindung mit einer oder mehreren Ausführungsformen trainiert oder verwendet wird, während der Rückwärtspropagierung von Eingabe-/Ausgabedaten und/oder Gewichtungsparametern während des Trainierens und/oder Inferenzierens unter Verwendung von Aspekten einer oder mehrerer Ausführungsformen. In mindestens einer Ausführungsform kann die Trainingslogik 915 einen Code- und/oder Datenspeicher 905 beinhalten oder an diesen gekoppelt sein, um Graphencode oder andere Software zum Steuern der Zeitsteuerung und/oder Reihenfolge zu speichern, in der Gewichtungs- und/oder andere Parameterinformationen geladen werden sollen, um Logik zu konfigurieren, einschließlich Integer- und/oder Gleitkommaeinheiten (zusammen als arithmetisch-logische Einheiten (ALUs) bezeichnet). In mindestens einer Ausführungsform lädt Code, wie etwa Graphencode, Gewichtungs- oder andere Parameterinformationen in Prozessor-ALUs auf Grundlage einer Architektur eines neuronalen Netzes, dem der Code entspricht. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Code- und/oder Datenspeichers 905 in einem anderen chipinternen oder chip externen Datenspeicher, einschließlich des L1-, L2- oder L3-Caches oder Systemspeichers eines Prozessors, enthalten sein. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Code- und/oder Datenspeichers 905 zu einem oder mehreren Prozessoren oder anderen Hardware-Logikvorrichtungen oder -Schaltungen intern oder extern sein. In mindestens einer Ausführungsform kann der Code- und/oder Datenspeicher 905 Cache-Speicher, DRAM, SRAM, nicht flüchtiger Speicher (z. B. Flash-Speicher) oder anderer Speicher sein. In mindestens einer Ausführungsform kann die Wahl, ob der Code- und/oder Datenspeicher 905 zu einem Prozessor, der zum Beispiel aus DRAM, SRAM, Flash oder einer anderen Speicherart besteht, intern oder extern ist, von dem verfügbaren Speicher auf dem Chip oder außerhalb des Chips, den Latenzanforderungen der durchzuführenden Trainings- und/oder Ableitungsfunktionen, der Batchgröße der Daten, die bei der Ableitung und/oder dem Training eines neuronalen Netzes verwendet werden, oder einer Kombination dieser Faktoren abhängen.

[0069] In mindestens einer Ausführungsform können der Code- und/oder Datenspeicher 901 und der Code- und/oder Datenspeicher 905 separate Speicherstrukturen sein. In mindestens einer Ausführungsform können

der Code- und/oder Datenspeicher 901 und der Code- und/oder Datenspeicher 905 dieselbe Speicherstruktur sein. In mindestens einer Ausführungsform können der Code- und/oder Datenspeicher 901 und der Code- und/oder Datenspeicher 905 teilweise dieselbe Speicherstruktur und teilweise separate Speicherstrukturen sein. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Code- und/oder Datenspeichers 901 und/oder des Code- und/oder Datenspeichers 905 in einem anderen chipinternen oder chipexternen Datenspeicher, einschließlich des L1-, L2- oder L3-Caches oder Systemspeichers eines Prozessors, enthalten sein.

[0070] In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 ohne Einschränkung eine oder mehrere arithmetisch-logische Einheit(en) („ALU(s)“) 910 beinhalten, einschließlich Integer- und/oder Gleitkommaeinheiten, um logische und/oder mathematische Operationen durchzuführen, die zumindest teilweise auf Trainings- und/oder Inferenzcode (z. B. Graphencode) basieren oder dadurch angegeben werden, wobei ein Ergebnis davon Aktivierungen (z. B. Ausgabewerte von Schichten oder Neuronen innerhalb eines neuronalen Netzes) produzieren kann, die in einem Aktivierungsspeicher 920 gespeichert sind und die Funktionen von Eingabe/Ausgabe- und/oder Gewichtungsparameterdaten sind, die in dem Code- und/oder Datenspeicher 901 und/oder dem Code- und/oder Datenspeicher 905 gespeichert sind. In mindestens einer Ausführungsform werden in dem Aktivierungsspeicher 920 gespeicherte Aktivierungen gemäß linearer algebraischer und/oder matrixbasierter Mathematik erzeugt, die durch die ALU(s) 910 als Reaktion auf das Durchführen von Anweisungen oder anderem Code durchgeführt wird, wobei in dem Code- und/oder Datenspeicher 905 und/oder dem Code- und/oder Datenspeicher 901 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 905 oder dem Code- und/oder Datenspeicher 901 oder einem anderen chipinternen oder -externen Speicher gespeichert sein können.

[0071] In mindestens einer Ausführungsform sind die ALU(s) 910 innerhalb eines oder mehrerer Prozessoren oder anderer Hardware-Logikvorrichtungen oder -Schaltungen enthalten, während in einer anderen Ausführungsform die ALU(s) 910 zu einem Prozessor oder einer anderen Hardware-Logikvorrichtung oder -Schaltung extern sein können, der/die sie verwendet (z. B. ein Koprozessor). In mindestens einer Ausführungsform können die ALUs 910 innerhalb der Ausführungseinheiten eines Prozessors oder anderweitig innerhalb einer Bank von ALUs enthalten sein, worauf die Ausführungseinheiten eines Prozessors zugreifen können, entweder innerhalb des gleichen Prozessors oder verteilt auf unterschiedliche Prozessoren unterschiedlichen Typs (z. B. zentrale Verarbeitungseinheiten, Grafikverarbeitungseinheiten, Festfunktionseinheiten usw.). In mindestens einer Ausführungsform können sich der Code- und/oder Datenspeicher 901, der Code- und/oder Datenspeicher 905 und der Aktivierungsspeicher 920 auf demselben Prozessor oder einer anderen Hardware-Logikvorrichtung oder -schaltung befinden, während sie sich in einer anderen Ausführungsform in verschiedenen Prozessoren oder anderen Hardware-Logikvorrichtungen oder -schaltungen oder in einer Kombination aus gleichen und verschiedenen Prozessoren oder anderen Hardware-Logikvorrichtungen oder -schaltungen befinden können. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Aktivierungsspeichers 920 in einem anderen chipinternen oder chip externen Datenspeicher, einschließlich des L1-, L2- oder L3-Caches oder Systemspeichers eines Prozessors, enthalten sein. Des Weiteren kann der Inferenz- und/oder Trainingscode mit anderem Code gespeichert sein, auf den ein Prozessor oder eine andere Hardware-Logik oder -Schaltung zugreifen kann und der unter Verwendung der Abruf-, Decodier-, Planungs-, Ausführungs-, Ausscheidungs- und/oder anderen Logikschaltungen eines Prozessors abgerufen und/oder verarbeitet wird.

[0072] In mindestens einer Ausführungsform kann der Aktivierungsspeicher 920 Cache-Speicher, DRAM, SRAM, nicht flüchtiger Speicher (z. B. Flash-Speicher) oder anderer Speicher sein. In mindestens einer Ausführungsform kann sich der Aktivierungsspeicher 920 ganz oder teilweise innerhalb oder außerhalb eines oder mehrerer Prozessoren oder anderer logischer Schaltungen befinden. In mindestens einer Ausführungsform kann die Wahl, ob der Code- und/oder Aktivierungsspeicher 920 zu einem Prozessor, der zum Beispiel aus DRAM, SRAM, Flash oder einem anderen Speichertyp besteht, intern oder extern ist, von dem verfügbaren Speicher auf dem Chip oder außerhalb des Chips, den Latenzanforderungen der durchzuführenden Trainings- und/oder Ableitungsfunktionen, der Batchgröße der Daten, die bei der Ableitung und/oder dem Training eines neuronalen Netzes verwendet werden, oder einer Kombination dieser Faktoren abhängig sein. In mindestens einer Ausführungsform kann die in **Fig. 9A** veranschaulichte Inferenz- und/oder Trainingslogik 915 in Verbindung mit einer anwendungsspezifischen integrierten Schaltung (application-specific integrated circuit - ASIC) verwendet werden, wie etwa der TensorFlow® Processing Unit von Google, einer Inferenzverarbeitungseinheit (inference processing unit - IPU) von Graphcore™ oder einem Prozessor vom Typ Nervana® (z. B. „Lake Crest“) der Intel Corp. In mindestens einer Ausführungsform kann die in **Fig. 9A** veran-

schaulichte Inferenz- und/oder Trainingslogik 915 in Verbindung mit Hardware einer zentralen Verarbeitungseinheit („CPU“), Hardware einer Grafikverarbeitungseinheit („GPU“) oder anderer Hardware, wie etwa feldprogrammierbaren Gate-Arrays („FPGAs“) verwendet werden.

[0073] Fig. 9B veranschaulicht eine Inferenz- und/oder Trainingslogik 915 gemäß mindestens einer oder mehreren Ausführungsformen. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 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. 9B** veranschaulichte Inferenz- und/oder Trainingslogik 915 in Verbindung mit einer anwendungsspezifischen integrierten Schaltung (ASIC) verwendet werden, wie etwa der TensorFlow® Processing Unit von Google, einer Inferenzverarbeitungseinheit (IPU) von Graphcore™ oder einem Prozessor vom Typ Nervana® (z. B. „Lake Crest“) der Intel Corp. In mindestens einer Ausführungsform kann die in **Fig. 9B** veranschaulichte Inferenz- und/oder Trainingslogik 915 in Verbindung mit Hardware einer zentralen Verarbeitungseinheit (CPU), Hardware einer Grafikverarbeitungseinheit (GPU) oder anderer Hardware, wie etwa feldprogrammierbaren Gate-Arrays (FPGAs) verwendet werden. In mindestens einer Ausführungsform beinhaltet die Inferenz- und/oder Trainingslogik 915 ohne Einschränkung den Code- und/oder Datenspeicher 901 und den Code- und/oder Datenspeicher 905, die zum Speichern von Code (z. B. Graphencode), Gewichtungswerten und/oder anderen Informationen, einschließlich Verzerrungswerten, Gradienteninformationen, Momentwerten und/oder anderer Parameter- oder Hyperparameterinformationen, verwendet werden können. In mindestens einer Ausführungsform, die in **Fig. 9B** veranschaulicht ist, ist jeder des Code- und/oder Datenspeichers 901 und des Code- und/oder Datenspeichers 905 mit einer dedizierten Rechenressource, wie etwa der Rechen-Hardware 902 bzw. der Rechen-Hardware 906, assoziiert. In mindestens einer Ausführungsform umfasst jede der Rechen-Hardware 902 und der Rechen-Hardware 906 eine oder mehrere ALUs, die mathematische Funktionen, wie etwa lineare algebraische Funktionen, nur an Informationen durchführen, die in dem Code- und/oder Datenspeicher 901 bzw. dem Code- und/oder Datenspeicher 905 gespeichert sind, wobei das Ergebnis davon in dem Aktivierungsspeicher 920 gespeichert wird.

[0074] In mindestens einer Ausführungsform entspricht jeder der Code- und/oder Datenspeicher 901 und 905 und die entsprechende Rechen-Hardware 902 bzw. 906 verschiedenen Schichten eines neuronalen Netzes, sodass die resultierende Aktivierung von einem „Speicher-/Rechenpaar 901/902“ des Code- und/oder Datenspeichers 901 und der Rechenhardware 902 als Eingabe für das „Speicher-/Rechenpaar 905/906“ des Code- und/oder Datenspeichers 905 und der Rechen-Hardware 906 bereitgestellt wird, um die konzeptionelle Organisation eines neuronalen Netzes zu spiegeln. In mindestens einer Ausführungsform kann jedes der Speicher-/Rechenpaare 901/902 und 905/906 mehr als einer Schicht eines neuronalen Netzes entsprechen. In mindestens einer Ausführungsform können zusätzliche Speicher-/Rechenpaare (nicht gezeigt) im Anschluss an oder parallel zu den Speicher-/Rechenpaaren 901/902 und 905/906 in die Inferenz- und/oder Trainingslogik 915 integriert werden.

RECHENZENTRUM

[0075] Fig. 10 veranschaulicht ein beispielhaftes Rechenzentrum 1000, in dem mindestens eine Ausführungsform verwendet werden kann. In mindestens einer Ausführungsform beinhaltet das Rechenzentrum 1000 eine Rechenzentrumsinfrastrukturschicht 1010, eine Rahmenschicht 1020, eine Softwareschicht 1030 und eine Anwendungsschicht 1040.

[0076] In mindestens einer Ausführungsform, wie in **Fig. 10** gezeigt, kann die Rechenzentrumsinfrastrukturschicht 1010 einen Ressourcen-Orchestrator 1012, gruppierte Rechenressourcen 1014 und Knoten-Rechenressourcen (node computing resources - „Knoten-C.R.s“) 1016(1)-1016(N) beinhalten, wobei „N“ eine beliebige positive ganze Zahl darstellt. In mindestens einer Ausführungsform können die Knoten-C.R.s 1016(1)-1016(N) eine beliebige Anzahl von zentralen Verarbeitungseinheiten („CPUs“) oder anderen Prozessoren (einschließlich Beschleunigern, feldprogrammierbarer Gate-Arrays (FPGAs), Grafikprozessoren usw.), Arbeitsspeichervorrichtungen (z. B. dynamischer Festwertspeicher), Datenspeichervorrichtungen (z. B. Solid-State- oder Festplattenlaufwerke), Netzwerk-Eingabe-/Ausgabe-(„NW-E/A“-)Vorrichtungen, Netzwerk-Switches, virtuellen Maschinen („VMs“), Leistungsmodulen und Kühlmodulen usw. beinhalten, sind aber nicht darauf beschränkt. In mindestens einer Ausführungsform kann es sich bei einer oder mehreren Knoten-C.R.s unter den Knoten-C.R.s 1016(1)-1016(N) um einen Server handeln, der eine oder mehrere der vorstehend erwähnten Rechenressourcen aufweist.

[0077] In mindestens einer Ausführungsform können die gruppierten Rechenressourcen 1014 separate Gruppierungen von Knoten-C.R.s beinhalten, die innerhalb eines oder mehrerer Racks (nicht gezeigt) oder vieler Racks untergebracht sind, die in Rechenzentren an verschiedenen geografischen Standorten untergebracht sind (ebenfalls nicht gezeigt). Separate Gruppierungen von Knoten-C.R.s innerhalb der gruppierten Rechenressourcen 1014 können gruppierte Rechen-, Netzwerk-, Arbeitsspeicher- oder Datenspeicherressourcen beinhalten, die zur Unterstützung einer oder mehrerer Arbeitslasten konfiguriert oder zugewiesen sein können. In mindestens einer Ausführungsform können mehrere Knoten-C.R.s, die CPUs oder Prozessoren beinhalten, in einem oder mehreren Racks gruppiert sein, um Rechenressourcen bereitzustellen, um eine oder mehrere Arbeitslasten zu unterstützen. In mindestens einer Ausführungsform können ein oder mehrere Racks auch eine beliebige Anzahl von Leistungsmodulen, Kühlmodulen und Netzwerk-Switches in beliebiger Kombination beinhalten.

[0078] In mindestens einer Ausführungsform kann der Ressourcen-Orchestrator 1012 eine oder mehrere Knoten-C.R.s 1016(1)-1016(N) und/oder gruppierte Rechenressourcen 1014 konfigurieren oder anderweitig steuern. In mindestens einer Ausführungsform kann der Ressourcen-Orchestrator 1012 eine Verwaltungsentität für Softwaregestaltungsinfrastruktur (software design infrastructure - „SDI“) für das Rechenzentrum 1000 beinhalten. In mindestens einer Ausführungsform kann der Ressourcen-Orchestrator Hardware, Software oder eine Kombination davon beinhalten.

[0079] In mindestens einer Ausführungsform, wie in **Fig. 10** gezeigt, beinhaltet die Rahmenschicht 1020 einen Aufgabenplaner 1022, einen Konfigurations-Manager 1024, einen Ressourcen-Manager 1026 und ein verteiltes Dateisystem 1028. In mindestens einer Ausführungsform kann die Rahmenschicht 1020 einen Rahmen zum Unterstützen von Software 1032 der Software-Schicht 1030 und/oder einer oder mehreren Anwendung(en) 1042 der Anwendungsschicht 1040 beinhalten. In mindestens einer Ausführungsform kann/können die Software 1032 oder die Anwendung(en) 1042 jeweils webbasierte Dienstsoftware oder - anwendungen beinhalten, wie etwa diejenigen, die von Amazon Web Services, Google Cloud und Microsoft Azure bereitgestellt werden. In mindestens einer Ausführungsform kann es sich bei der Rahmenschicht 1020 um eine Art von freiem und Open-Source-Software-Webanwendungsrahmen, wie etwa Apache Spark™ (im Folgenden „Spark“) handeln, der das verteilte Dateisystem 1028 für die Verarbeitung großer Datenmengen (z. B. „Big Data“) verwenden kann, ohne darauf beschränkt zu sein. In mindestens einer Ausführungsform kann der Aufgabenplaner 1022 einen Spark-Treiber beinhalten, um die Planung von Arbeitslasten zu erleichtern, die durch verschiedene Schichten des Rechenzentrums 1000 unterstützt werden. In mindestens einer Ausführungsform kann der Konfigurations-Manager 1024 in der Lage sein, unterschiedliche Schichten, wie etwa die Software-Schicht 1030 und die Rahmenschicht 1020, einschließlich Spark und des verteilten Dateisystems 1028, zu konfigurieren, um die Verarbeitung großer Datenmengen zu unterstützen. In mindestens einer Ausführungsform kann der Ressourcen-Manager 1026 in der Lage sein, geclusterte oder gruppierte Rechenressourcen zu verwalten, die zur Unterstützung des verteilten Dateisystems 1028 und des Aufgaben-Planers 1022 abgebildet oder zugewiesen sind. In mindestens einer Ausführungsform können geclusterte oder gruppierte Rechenressourcen die gruppierte Rechenressource 1014 auf der Rechenzentrumsinfrastrukturschicht 1010 beinhalten. In mindestens einer Ausführungsform kann sich der Ressourcen-Manager 1026 mit dem Ressourcen-Orchestrator 1012 koordinieren, um diese abgebildeten oder zugewiesenen Rechenressourcen zu verwalten.

[0080] In mindestens einer Ausführungsform kann die Software 1032, die in der Software-Schicht 1030 enthalten ist, Software beinhalten, die zumindest von Abschnitten der Knoten-C.R.s 1016(1)-1016(N), den gruppierten Rechenressourcen 1014 und/oder dem verteilten Dateisystem 1028 der Rahmenschicht 1020 verwendet wird. Eine oder mehrere Arten von Software können Software zum Durchsuchen von Internet-Webseiten, Software zum Scannen von E-Mails auf Viren, Datenbank-Software und Software zum Streaming von Videoinhalten beinhalten, ohne darauf beschränkt zu sein.

[0081] In mindestens einer Ausführungsform können die in der Anwendungsschicht 1040 enthaltenen Anwendung(en) 1042 eine oder mehrere Arten von Anwendungen beinhalten, die zumindest von Abschnitten der Knoten-C.R.s 1016(1)-1016(N), den gruppierten Rechenressourcen 1014 und/oder dem verteilten Dateisystem 1028 der Rahmenschicht 1020 verwendet werden. Zu einer oder mehreren Arten von Anwendungen können eine beliebige Anzahl von genomischen Anwendungen, eine kognitive Berechnung und eine Anwendung für maschinelles Lernen gehören, einschließlich Trainings- oder Inferenz-Software, Rahmen-Software für maschinelles Lernen (z. B. PyTorch, TensorFlow, Caffe usw.) oder andere Anwendungen für maschinelles Lernen, die in Verbindung mit einer oder mehreren Ausführungsformen verwendet werden, ohne darauf beschränkt zu sein.

[0082] In mindestens einer Ausführungsform können beliebige des Konfigurations-Managers 1024, des Ressourcen-Managers 1026 und des Ressourcen-Orchestrators 1012 eine beliebige Anzahl und Art von selbstmodifizierenden Handlungen implementieren, die auf einer beliebigen Menge und Art von Daten basieren, die auf eine beliebige technisch machbare Weise erfasst wurden. In mindestens einer Ausführungsform können selbstmodifizierende Handlungen einen Rechenzentrumsbetreiber des Rechenzentrums 1000 dahingehend entlasten, möglicherweise schlechte Konfigurationsentscheidungen zu treffen und möglicherweise nicht ausgelastete und/oder schlecht funktionierende Abschnitte eines Rechenzentrums zu vermeiden.

[0083] In mindestens einer Ausführungsform kann das Rechenzentrum 1000 Werkzeuge, Dienste, Software oder andere Ressourcen beinhalten, um ein oder mehrere Modelle für maschinelles Lernen zu trainieren oder Informationen unter Verwendung eines oder mehrerer Modelle für maschinelles Lernen gemäß einer oder mehrerer in dieser Schrift beschriebener Ausführungsformen vorherzusagen oder abzuleiten. Zum Beispiel kann in mindestens einer Ausführungsform ein Modell für maschinelles Lernen trainiert werden, indem Gewichtungsparameter gemäß einer Architektur eines neuronalen Netzes unter Verwendung von Software- und Rechenressourcen berechnet werden, die vorstehend in Bezug auf das Rechenzentrum 1000 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 1000 beschriebenen Ressourcen abzuleiten oder vorherzusagen, indem Gewichtungsparameter verwendet werden, die durch eine oder mehrere in dieser Schrift beschriebene Trainingstechniken berechnet werden.

[0084] 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.

[0085] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 in dem System aus **Fig. 10** für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0086] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

COMPUTERSYSTEME

[0087] **Fig. 11** ist ein Blockdiagramm, das ein beispielhaftes Computersystem veranschaulicht, das ein System mit miteinander verbundenen Vorrichtungen und Komponenten, ein System auf einem Chip (system-on-a-Chip - SOC) oder eine Kombination davon 1100 sein kann, gebildet mit einem Prozessor, der Ausführungseinheiten zur Ausführung einer Anweisung gemäß mindestens einer Ausführungsform beinhalten kann. In mindestens einer Ausführungsform kann das Computersystem 1100 ohne Einschränkung eine Komponente, wie etwa einen Prozessor 1102, beinhalten, um Ausführungseinheiten einschließlich Logik zur Durchführung von Algorithmen für Prozessdaten gemäß der vorliegenden Offenbarung zu verwenden, wie etwa in der in dieser Schrift beschriebenen Ausführungsform. In mindestens einer Ausführungsform kann das Computersystem 1100 Prozessoren beinhalten, wie etwa PENTIUM®-Prozessorfamilie, Mikroprozessoren von Xeon™, Itanium® , XScale™ und/oder StrongARM™, Intel® Core™ oder Intel® Nervana™, die von Intel Corporation aus Santa Clara, Kalifornien, erhältlich sind, obwohl auch andere Systeme (die PCs mit anderen Mikroprozessoren, Engineering-Workstations, Set-Top-Boxen und dergleichen beinhalten) verwendet werden können. In mindestens einer Ausführungsform kann das Computersystem 1100 eine Version des Betriebssystems WINDOWS ausführen, das von der Microsoft Corporation in Redmond, Washington, erhältlich ist, obwohl auch andere Betriebssysteme (z. B. UNIX und Linux), eingebettete Software und/oder grafische Benutzeroberflächen verwendet werden können.

[0088] Ausführungsformen können in anderen Vorrichtungen wie etwa tragbaren Vorrichtungen und eingebetteten Anwendungen verwendet werden. Einige Beispiele für tragbare Vorrichtungen beinhalten Mobiltelefone, Internetprotokoll-Vorrichtungen, Digitalkameras, persönliche digitale Assistenten („PDAs“) und tragbare PCs. In mindestens einer Ausführungsform können eingebettete Anwendungen einen Mikrocontroller, einen digitalen Signalprozessor (digital signal processor - „DSP“), ein System auf einem Chip, Netzcomputer („NetPCs“), Set-Top-Boxen, Netz-Hubs, Switches für ein Weitverkehrsnetz (wide area network - „WAN“) oder ein beliebiges anderes System beinhalten, das eine oder mehrere Anweisungen gemäß mindestens einer Ausführungsform durchführen kann.

[0089] In mindestens einer Ausführungsform kann das Computersystem 1100 ohne Einschränkung den Prozessor 1102 beinhalten, der ohne Einschränkung eine oder mehrere Ausführungseinheiten 1108 beinhalten kann, um Trainieren und/oder Inferenzieren eines Modells für maschinelles Lernens gemäß den in dieser Schrift beschriebenen Techniken 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 an einen Prozessorbus 1110 gekoppelt sein, der Datensignale zwischen dem Prozessor 1102 und anderen Komponenten in dem Computersystem 1100 übertragen kann.

[0090] In mindestens einer Ausführungsform kann der Prozessor 1102 ohne Einschränkung einen internen Cache-Speicher („Cache“) 1104 der Ebene 1 (Level 1 - „L1“) 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 Cache-Speicher extern zu dem Prozessor 1102 befinden. Andere Ausführungsformen können in Abhängigkeit von der konkreten Implementierung und den Anforderungen auch eine Kombination aus sowohl internen als auch externen Caches beinhalten. In mindestens einer Ausführungsform kann die Registerdatei 1106 unterschiedliche Arten von Daten in verschiedenen Registern speichern, einschließlich ohne Einschränkung Integerregistem, Gleitkommaregistem, Statusregistern und Anweisungszeigerregistern.

[0091] In mindestens einer Ausführungsform befindet sich die Ausführungseinheit 1108, einschließlich ohne Einschränkung der Logik zum Durchführen von Ganzzahl- 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 zum Handhaben eines gepackten Anweisungssatzes 1109 beinhalten. In mindestens einer Ausführungsform können durch die Aufnahme des gepackten Anweisungssatzes 1109 in einen Anweisungssatz eines Universalprozessors 1102 zusammen mit zugeordneten Schaltungen zum Ausführen von Anweisungen Operationen, die durch viele Multimedia-Anwendungen verwendet werden, unter Verwendung von gepackten Daten in einem Universalprozessor 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 für die Durchführung von Operationen an Paketdaten genutzt wird, wodurch die Notwendigkeit entfällt, kleinere Dateneinheiten über den Datenbus des Prozessors zu übertragen, um einen oder mehrere Operationen an einem Datenelement nach dem anderen durchzuführen.

[0092] In mindestens einer Ausführungsform kann die Ausführungseinheit 1108 auch in Mikrocontrollern, eingebetteten Prozessoren, Grafikvorrichtungen, DSPs und anderen Arten 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 als dynamische Direktzugriffsspeicher-(„DRAM“-)Vorrichtung, statische Direktzugriffsspeicher-(„SRAM“-)Vorrichtung, Flash-Speichervorrichtung oder eine andere Speichervorrichtung implementiert sein. In mindestens einer Ausführungsform kann der Speicher 1120 Anweisung(en) 1119 und/oder Daten 1121 speichern, die durch Datensignale dargestellt werden, die durch den Prozessor 1102 ausgeführt werden können.

[0093] In mindestens einer Ausführungsform kann ein Systemlogikchip mit dem Prozessorbus 1110 und dem Speicher 1120 gekoppelt sein. In mindestens einer Ausführungsform kann der Systemlogikchip ohne Einschränkung einen Speichersteuerungs-Hub (memory controller hub - „MCH“) 1116 beinhalten und kann

der Prozessor 1102 mit dem MCH 1116 über den Prozessorbus 1110 kommunizieren. In mindestens einer Ausführungsform kann der MCH 1116 dem Speicher 1120 einen Speicherpfad 1118 mit hoher Bandbreite für die Anweisungs- und Datenspeicherung sowie für die Speicherung von Grafikbefehlen, Daten und Texturen bereitstellen. In mindestens einer Ausführungsform kann der MCH 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 1122 überbrücken. In mindestens einer Ausführungsform kann der Systemlogikchip einen Grafikport zur Kopplung mit einer Grafiksteuerung bereitstellen. In mindestens einer Ausführungsform kann der MCH 1116 durch einen Speicherpfad 1118 mit hoher Bandbreite mit dem Speicher 1120 gekoppelt sein und kann die Grafik-/Videokarte 1112 durch eine Accelerated-Graphics-Port-(„AGP“-)Zusammenschaltung 1114 mit dem MCH 1116 gekoppelt sein.

[0094] In mindestens einer Ausführungsform kann das Computersystem 1100 die System-E/A 1122 verwenden, die ein proprietärer Hub-Schnittstellenbus ist, um den MCH 1116 mit dem 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 der lokale E/A-Bus ohne Einschränkung einen Hochgeschwindigkeits-E/A-Bus zur Verbindung von Peripheriegeräten mit dem Speicher 1120, dem Chipsatz und dem Prozessor 1102 beinhalten. Beispiele können ohne Einschränkung eine Audiosteuerung 1129, einen Firmware-Hub („Flash-BIOS“) 1128, einen drahtlosen Transceiver 1126, einen Datenspeicher 1124, eine Legacy-E/A-Steuerung 1123 mit Benutzereingabe- und Tastaturschnittstellen 1125, einen seriellen Erweiterungsport 1127, wie etwa Universal Serial Bus („USB“), und eine Netzwerksteuerung 1134 beinhalten. Der Datenspeicher 1124 kann ein Festplattenlaufwerk, ein Diskettenlaufwerk, eine CD-ROM-Vorrichtung, eine Flash-Speichervorrichtung oder eine andere Massenspeichervorrichtung umfassen.

[0095] In mindestens einer Ausführungsform veranschaulicht **Fig. 11** ein System, das zusammengeschaltete Hardware-Vorrichtungen oder „Chips“ beinhaltet, wohingegen in anderen Ausführungsformen **Fig. 11** ein beispielhaftes System auf einem Chip („SoC“) veranschaulichen kann. In mindestens einer Ausführungsform können die Vorrichtungen mit proprietären Zusammenschaltungen, standardisierten Zusammenschaltungen (z. B. PCIe) oder einer Kombination davon zusammengeschaltet sein. In mindestens einer Ausführungsform sind eine oder mehrere Komponenten des Computersystems 1100 unter Verwendung von Compute-Express-Link-(CXL-)Zusammenschaltungen zusammengeschaltet.

[0096] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 in dem System aus **Fig. 11** 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.

[0097] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0098] **Fig. 12** ist ein Blockdiagramm, das eine elektronische Vorrichtung 1200 zum Verwenden eines Prozessors 1210 gemäß mindestens einer Ausführungsform veranschaulicht. 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.

[0099] In mindestens einer Ausführungsform kann das System 1200 ohne Einschränkung einen Prozessor 1210 beinhalten, der mit einer beliebigen geeigneten Anzahl oder Art von Komponenten, Peripheriegeräten, Modulen oder Vorrichtungen kommunikativ gekoppelt ist. In mindestens einer Ausführungsform ist der Prozessor 1210 unter Verwendung eines Busses oder einer Schnittstelle gekoppelt, wie etwa eines 1°C-Busses, eines Systemverwaltungsbusses (System Management Bus - „SMBus“), eines Low-Pin-Count-(LPC-)Busses, einer seriellen peripheren Schnittstelle (Serial Peripheral Interface - „SPI“), eines High-Definition-Audio-(„HDA“-)Busses, eines Serial-Advance-Technology-Attachment-(„SATA“-)Busses, eines Universal Serial Bus

(„USB“) (Versionen 1, 2, 3) oder eines Universal-Asynchronous-Receiver/Transmitter-(„UART“-)Busses. In mindestens einer Ausführungsform veranschaulicht **Fig. 12** ein System, das zusammengeschaltete Hardware-Vorrichtungen oder „Chips“ beinhaltet, wohingegen in anderen Ausführungsformen **Fig. 12** ein beispielhaftes System auf einem Chip „SoC“ veranschaulichen kann. In mindestens einer Ausführungsform können die in **Fig. 12** veranschaulichten Vorrichtungen mit proprietären Zusammenschaltungen, standardisierten Zusammenschaltungen (z. B. PCIe) oder einer Kombination davon zusammengeschaltet sein. In mindestens einer Ausführungsform sind eine oder mehrere Komponenten aus **Fig. 12** unter Verwendung von Compute-Express-Link-(CXL-)Zusammenschaltungen zusammengeschaltet.

[0100] In mindestens einer Ausführungsform kann **Fig. 12** eine Anzeige 1224, einen Touchscreen 1225, ein Touchpad 1230, eine Einheit zur Nahfeldkommunikation (Near Field Communications - „NFC“) 1245, einen Sensor-Hub 1240, einen Wärmesensor 1246, einen Express-Chipsatz („EC“) 1235, ein Trusted-Platform-Modul („TPM“) 1238, BIOS-/Firmware-/Flash-Speicher („BIOS, FW Flash“) 1222, einen DSP 1260, ein Laufwerk 1220, wie etwa ein Solid-State-Platten- (Solid State Disk - „SSD“) oder ein Festplattenlaufwerk (Hard Disk Drive - „HDD“), eine Einheit für ein drahtloses lokales Netzwerk (wireless local area network - „WLAN“) 1250, eine Bluetooth-Einheit 1252, eine Einheit für ein drahtloses Weitverkehrsnetzwerk (Wireless Wide Area Network - „WWAN“) 1256, ein globales Positionsbestimmungssystem (GPS) 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, die zum Beispiel im LPDDR3-Standard implementiert ist, beinhalten. Diese Komponenten können jeweils auf eine beliebige geeignete Weise implementiert sein.

[0101] In mindestens einer Ausführungsform können andere Komponenten durch die vorstehend erörterten Komponenten kommunikativ mit dem 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 mit dem Sensor-Hub 1240 gekoppelt sein. In mindestens einer Ausführungsform können ein Wärmesensor 1239, ein Lüfter 1237, eine Tastatur 1246 und ein Touchpad 1230 kommunikativ mit dem EC 1235 gekoppelt sein. In mindestens einer Ausführungsform können ein Lautsprecher 1263, Kopfhörer 1264 und ein Mikrofon („Mic“) 1265 kommunikativ mit einer Audioeinheit („Audio-Codec und Klasse-D-Verst“) 1262 gekoppelt sein, die wiederum kommunikativ mit dem DSP 1260 gekoppelt sein kann. In mindestens einer Ausführungsform kann die Audioeinheit 1264 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 mit der WWAN-Einheit 1256 gekoppelt sein. In mindestens einer Ausführungsform können Komponenten, wie etwa die WLAN-Einheit 1250 und die Bluetooth-Einheit 1252 sowie die WWAN-Einheit 1256, in einem Next Generation Form Factor („NGFF“) implementiert sein.

[0102] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 in dem System aus **Fig. 12** für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0103] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0104] **Fig. 13** veranschaulicht ein Computersystem 1300 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist das Computersystem 1300 dazu konfiguriert, verschiedene Prozesse und Verfahren zu implementieren, die in dieser Offenbarung beschrieben sind.

[0105] 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 eine Steuerlogik (z. B. implementiert als Hardware, Software

oder eine Kombination davon) und Daten werden im Hauptspeicher 1304 gespeichert, der die Form eines Direktzugriffsspeichers (random access memory - „RAM“) annehmen kann. In mindestens einer Ausführungsform stellt ein Netzwerkschnittstellen-Teilsystem („Netzwerkschnittstelle“) 1322 eine Schnittstelle zu anderen Rechenvorrichtung und Netzwerken bereit, um Daten von anderen Systemen zu empfangen und von dem Computersystem 1300 an diese zu übertragen.

[0106] In mindestens einer Ausführungsform beinhaltet das Computersystem 1300 in mindestens einer Ausführungsform ohne Einschränkung Eingabevorrichtungen 1308, ein Parallelverarbeitungssystem 1312 und Anzeigevorrichtungen 1306, die unter Verwendung einer herkömmlichen Kathodenstrahlröhre (cathode ray tube „CRT“), Flüssigkristallanzeige (liquid crystal display - „LCD“), Leuchtdiode (light emitting diode - „LED“), Plasmaanzeige oder anderen geeigneten Anzeigetechnologien implementiert sein können. In mindestens einer Ausführungsform werden Benutzereingaben von Eingabevorrichtungen 1308, wie etwa Tastatur, Maus, Touchpad, Mikrofon und anderen, empfangen. In mindestens einer Ausführungsform kann sich jedes der vorgenannten Module auf einer einzelnen Halbleiterplattform befinden, um ein Verarbeitungssystem zu bilden.

[0107] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 in dem System aus **Fig. 13** 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.

[0108] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0109] 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 eine beliebige Art 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.

[0110] 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(e) beliebige(s) Anweisungsausführungssystem, -einrichtung oder - vorrichtung sein, das/die in der Lage ist, Anweisungen auszuführen. In mindestens einer Ausführungsform kann die Verarbeitungseinheit 1430 ohne Einschränkung eine beliebige Anzahl und eine beliebige Art von Verarbeitungskernen (nicht gezeigt) beinhalten. In mindestens einer Ausführungsform umfasst der Verarbeitungskern 1430 eine anwendungsspezifische integrierte Schaltung („ASIC“), die für die Durchführung beliebiger Mengen und Arten von Operationen im Zusammenhang mit maschinellem Lernen optimiert ist. Zum Beispiel ist in mindestens einer Ausführungsform der Verarbeitungskern 1430 eine Tensor-Verarbeitungseinheit (tensor processing unit - „TPU“), die für die Durchführung von Inferenzoperationen für maschinelles Lernen optimiert ist. In mindestens einer Ausführungsform ist der Verarbeitungskern 1430 eine Sichtverarbeitungseinheit (vision processing unit - „VPU“), die für die Durchführung von Inferenzoperationen für maschinelles Sehen und maschinelles Lernen optimiert ist.

[0111] In mindestens einer Ausführungsform kann die USB-Schnittstelle 1440 eine beliebige Art 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-Stecker. In mindestens einer Ausführungsform kann die USB-Schnittstellenlogik 1450 eine beliebige Menge und Art 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.

[0112] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitge-

stellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 in dem System aus **Fig. 14** 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.

[0113] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0114] **Fig. 15A** veranschaulicht eine beispielhafte Architektur, in der eine Vielzahl von GPUs 1510-1513 mit einer Vielzahl von Mehrkern-Prozessoren 1505-1506 über Hochgeschwindigkeitsverknüpfungen 1540-1543 (z. B. Busse, Punkt-zu-Punkt-Zusammenschaltungen usw.) kommunikativ gekoppelt ist. In einer Ausführungsform unterstützen die Hochgeschwindigkeitsverbindungen 1540-1543 einen Kommunikationsdurchsatz von 4 GB/s, 30 GB/s, 80 GB/s oder höher. Verschiedene Zusammenschaltungsprotokolle können verwendet werden, einschließlich, aber nicht beschränkt auf, PCIe 4.0 oder 5.0 und NVLink 2.0.

[0115] Zusätzlich und in einer Ausführungsform sind zwei oder mehr der GPUs 1510-1513 über Hochgeschwindigkeitsverbindungen 1529-1530 zusammengeschaltet, die unter Verwendung von Protokollen/Verbindungen implementiert sein können, die gleich wie oder anders als diejenigen sind, die für die Hochgeschwindigkeitsverbindungen 1540-1543 verwendet werden. Auf ähnliche Weise können zwei oder mehr der Mehrkern-Prozessoren 1505-1506 über eine Hochgeschwindigkeitsverbindung 1528 verbunden sein, bei der es sich um symmetrische Multiprozessor-(SMP-)Busse handeln kann, die mit 20 GB/s, 30 GB/s, 120 GB/s oder höher arbeiten. Alternativ kann die gesamte Kommunikation zwischen den verschiedenen in **Fig. 15A** gezeigten Systemkomponenten über dieselben Protokolle/Verknüpfungen erfolgen (z. B. über eine gemeinsame Zusammenschaltungsstruktur).

[0116] In einer Ausführungsform ist jeder Mehrkern-Prozessor 1505-1506 jeweils über Speicherzusammenschaltungen 1526-1527 kommunikativ mit einem Prozessorspeicher 1501-1502 gekoppelt und ist jede GPU 1510-1513 jeweils über GPU-Speicherzusammenschaltungen 1550-1553 kommunikativ mit dem GPU-Speicher 1520-1523 gekoppelt. Die Speicherzusammenschaltungen 1526-1527 und 1550-1553 können gleiche oder unterschiedliche Speicherzugriffstechnologien verwenden. Bei den Prozessorspeichern 1501-1502 und den GPU-Speichern 1520-1523 kann es sich beispielsweise um flüchtige Speicher, wie etwa dynamische Direktzugriffsspeicher (DRAMs) (einschließlich gestapelter DRAMs), Grafik-DDR-SDRAM (GDDR) (z. B. GDDR5, GDDR6) oder Speicher mit hoher Bandbreite (High Bandwidth Memory - HBM), und/oder um nicht flüchtige Speicher, wie etwa 3D XPoint oder Nano-Ram, handeln. In einer Ausführungsform kann ein Abschnitt der Prozessorspeicher 1501-1502 ein flüchtiger Speicher sein und kann ein anderer Abschnitt ein nicht flüchtiger Speicher sein (z. B. unter Verwendung einer Speicherhierarchie mit zwei Ebenen (two-level memory - 2LM)).

[0117] Wie nachstehend beschrieben, können verschiedene Prozessoren 1505-1506 und GPUs 1510-1513 zwar physisch mit einem konkreten Speicher 1501-1502 bzw. 1520-1523 gekoppelt sein, kann jedoch eine vereinheitlichte Speicherarchitektur implementiert werden, bei der ein und derselbe virtuelle Systemadressraum (auch als „effektiver Adressraum“ bezeichnet) auf verschiedene physische Speicher verteilt ist. Zum Beispiel können die Prozessorspeicher 1501-1502 jeweils 64 GB Systemspeicheradressraum umfassen und können die GPU-Speicher 1520-1523 jeweils 32 GB Systemspeicheradressraum umfassen (was in diesem Beispiel zu einem adressierbaren Speicher von insgesamt 256 GB führt).

[0118] **Fig. 15B** veranschaulicht zusätzliche Details für eine Zusammenschaltung zwischen einem Mehrkernprozessor 1507 und einem Grafikbeschleunigungsmodul 1546 gemäß einer beispielhaften Ausführungsform. Das Grafikbeschleunigungsmodul 1546 kann einen oder mehrere GPU-Chips beinhalten, die auf einer Leitungskarte integriert sind, die über eine Hochgeschwindigkeitsverbindung 1540 mit dem Prozessor 1507 gekoppelt ist. Alternativ kann das Grafikbeschleunigungsmodul 1546 in demselben Gehäuse oder Chip wie der Prozessor 1507 integriert sein.

[0119] In mindestens einer Ausführungsform beinhaltet der veranschaulichte Prozessor 1507 eine Vielzahl von Kernen 1560A-1560D, jeder mit einem Adressübersetzungspuffer 1561A-1561D und einem oder mehreren Caches 1562A-1562D. In mindestens einer Ausführungsform können die Kerne 1560A-1560D verschiedene andere Komponenten zur Ausführung von Anweisungen und Verarbeitung von Daten beinhalten, die nicht veranschaulicht sind. Die Caches 1562A-1562D können Caches der Ebene 1 (L1) und der Ebene 2

(L2) umfassen. Außerdem können ein oder mehrere gemeinsam genutzte Caches 1556 in den Caches 1562A-1562D enthalten sein und von 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. Der Prozessor 1507 und das Grafikbeschleunigungsmodul 1546 sind mit dem Systemspeicher 1514 verbunden, der die Prozessorspeicher 1501-1502 aus **Fig. 15A** beinhalten kann.

[0120] Die Kohärenz wird für Daten und Anweisungen, die in verschiedenen Caches 1562A-1562D, 1556 und im Systemspeicher 1514 gespeichert sind, über eine Zwischenkemkommunikation über einen Kohärenzbus 1564 aufrechterhalten. Zum Beispiel kann in mindestens einer Ausführungsform jeder Cache eine Cache-Kohärenzlogik/-schaltung aufweisen, die ihm zugeordnet ist, um als Reaktion auf erfasste Lese- oder Schreibvorgänge in konkreten Cache-Zeilen über den Kohärenzbus 1564 zu kommunizieren. In einer Implementierung wird ein Cache-Snooping-Protokoll über den Kohärenzbus 1564 implementiert, um Cache-Zugriffe zu kontrollieren.

[0121] In einer Ausführungsform koppelt eine Proxy-Schaltung 1525 das Grafikbeschleunigungsmodul 1546 kommunikativ an den Kohärenzbus 1564, sodass das Grafikbeschleunigungsmodul 1546 an einem Cache-Kohärenzprotokoll als Peer der Kerne 1560A-1560D teilnehmen kann. Insbesondere stellt eine Schnittstelle 1535 Konnektivität zu der Proxy-Schaltung 1525 über eine Hochgeschwindigkeitsverbindung 1540 (z. B. einen PCIe-Bus, NVLink usw.) bereit und eine Schnittstelle 1537 verbindet das Grafikbeschleunigungsmodul 1546 mit der Verbindung 1540.

[0122] In einer Implementierung stellt eine Beschleuniger-Integrationsschaltung 1536 Cache-Verwaltungs-, Speicherzugriffs-, Kontextverwaltungs- und Unterbrechungsverwaltungsdienste im Auftrag einer Vielzahl von Grafikverarbeitungs-Engines 1531, 1532, N des Grafikbeschleunigungsmoduls 1546 bereit. Die Grafikverarbeitungs-Engines 1531, 1532, N können jeweils eine separate Grafikverarbeitungseinheit (GPU) umfassen. Alternativ können die Grafikverarbeitungs-Engines 1531, 1532, N unterschiedliche Arten von Grafikverarbeitungs-Engines innerhalb einer GPU umfassen, wie etwa Grafikausführungseinheiten, Medienverarbeitungs-Engines (z. B. Videocodierer/-decodierer), Abtaster und Blit-Engines. In mindestens einer Ausführungsform kann das Grafikbeschleunigungsmodul 1546 eine GPU mit einer Vielzahl von Grafikverarbeitungs-Engines 1531-1532, N sein, oder können die Grafikverarbeitungs-Engines 1531-1532, N einzelne GPUs sein, die in einem gemeinsamen Gehäuse, auf einer gemeinsamen Leitungskarte oder auf einem gemeinsamen Chip integriert sind.

[0123] In einer Ausführungsform beinhaltet die Beschleuniger-Integrationsschaltung 1536 eine Speicherverwaltungseinheit (memory management unit - MMU) 1539 zur Durchführung verschiedener Speicherverwaltungsfunktionen, wie etwa Übersetzungen von virtuellem in physischen Speicher (auch als Übersetzungen von effektivem in realen Speicher bezeichnet) und Speicherzugriffsprotokolle für den Zugriff auf den Systemspeicher 1514. Die MMU 1539 kann auch einen Adressübersetzungspuffer (translation lookaside buffer - TLB) (nicht gezeigt) zum Zwischenspeichern von Übersetzungen von virtuellen/effektiven in physische/reale Adressen beinhalten. In einer Implementierung speichert ein Cache 1538 Befehle und Daten für den effizienten Zugriff durch die Grafikverarbeitungs-Engines 1531-1532, N. In einer Ausführungsform werden die in dem Cache 1538 und in den Grafikspeichern 1533-1534, M gespeicherten Daten mit den Kern-Caches 1562A-1562D, 1556 und dem Systemspeicher 1514 kohärent gehalten. Wie vorstehend erwähnt, kann dies über die Proxy-Schaltung 1525 im Auftrag des Caches 1538 und der Speicher 1533-1534, M erfolgen (z. B. Senden von Aktualisierungen an den Cache 1538 in Bezug auf Modifikationen/Zugriffe auf Cache-Leitungen in den Prozessor-Caches 1562A-1562D, 1556 und Empfangen von Aktualisierungen von dem Cache 1538).

[0124] Ein Satz von Registern 1545 speichert Kontextdaten für Threads, die von den Grafikverarbeitungs-Engines 1531-1532, 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 von einer Grafikverarbeitungs-Engine ausgeführt werden kann). Zum Beispiel kann die Kontextverwaltungsschaltung 1548 bei einer Kontextumschaltung aktuelle Registerwerte in einer bezeichneten Region in dem Speicher speichern (z. B. identifiziert durch einen Kontextzeiger). Sie kann dann bei der Rückkehr zu einem Kontext die Registerwerte wiederherstellen. In einer Ausführungsform empfängt und verarbeitet eine Unterbrechungsverwaltungsschaltung 1547 von Systemvorrichtungen empfangene Unterbrechungen.

[0125] In einer Implementierung werden virtuelle/effektive Adressen von einer Grafikverarbeitungs-Engine 1531 durch die MMU 1539 in reale/physische Adressen in dem Systemspeicher 1514 übersetzt. Eine Ausführungsform der Beschleuniger-Integrationsschaltung 1536 unterstützt mehrere (z. B. 4, 8, 16) Grafikbeschleunigermodule 1546 und/oder andere Beschleunigervorrichtungen. Das Grafikbeschleunigermodul 1546 kann für eine einzelne Anwendung dediziert sein, die auf dem Prozessor 1507 ausgeführt wird, oder von mehreren Anwendungen gemeinsam genutzt werden. In einer Ausführungsform ist eine virtualisierte Grafikausführungsumgebung dargestellt, in der die Ressourcen der Grafikverarbeitungs-Engines 1531-1532, N mit mehreren Anwendungen oder virtuellen Maschinen (VMs) gemeinsam genutzt werden. In mindestens einer Ausführungsform können die Ressourcen in „Scheiben“ unterteilt werden, die unterschiedlichen VMs und/oder Anwendungen auf Grundlage von Verarbeitungsanforderungen und Prioritäten, die VMs und/oder Anwendungen zugeordnet sind, zugewiesen werden.

[0126] In mindestens einer Ausführungsform fungiert die Beschleuniger-Integrationsschaltung 1536 als Brücke zu einem System für das Grafikbeschleunigungsmodul 1546 und sie stellt Adressübersetzungs- und Systemspeicher-Cache-Dienste bereit. Darüber hinaus kann die Beschleuniger-Integrationsschaltung 1536 Virtualisierungseinrichtungen für einen Host-Prozessor bereitstellen, um die Virtualisierung der Grafikverarbeitungs-Engines 1531-1532, N, Unterbrechungen und Speicherverwaltung zu verwalten.

[0127] Da die Hardware-Ressourcen der Grafikverarbeitungs-Engines 1531-1532, N explizit auf einen realen Adressraum abgebildet sind, den der Host-Prozessor 1507 sieht, kann ein beliebiger Host-Prozessor diese Ressourcen unter Verwendung eines effektiven Adresswerts direkt adressieren. Eine Funktion der Beschleuniger-Integrationsschaltung 1536 ist in einer Ausführungsform die physische Trennung der Grafikverarbeitungs-Engines 1531-1532, N, sodass sie einem System als unabhängige Einheiten erscheinen.

[0128] In mindestens einer Ausführungsform sind ein oder mehrere Grafikspeicher 1533-1534, M mit jeder der Grafikverarbeitungs-Engines 1531-1532, N gekoppelt. Die Grafikspeicher 1533-1534, M speichern Anweisungen und Daten, die von jeder der Grafikverarbeitungs-Engines 1531-1532, N verarbeitet werden. Bei den Grafikspeichern 1533-1534, M kann es sich um flüchtige Speicher, wie zum Beispiel DRAMs (einschließlich gestapelter DRAMs), GDDR-Speicher (z. B. GDDR5, GDDR6) oder HBM, und/oder um nicht flüchtigen Speicher, wie zum Beispiel 3D XPoint oder Nano-Ram, handeln.

[0129] In einer Ausführungsform werden zum Reduzieren des Datenverkehrs über die Verbindung 1540 Verzerrungstechniken verwendet, um sicherzustellen, dass es sich bei den in den Grafikspeichern 1533-1534, M gespeicherten Daten um Daten handelt, die am häufigsten durch die Grafikverarbeitungs-Engines 1531-1532, N verwendet werden und vorzugsweise nicht durch die Kerne 1560A-1560D verwendet werden (zumindest nicht häufig). Auf ähnliche Weise versucht ein Verzerrungsmechanismus, Daten, die von den Kernen (und vorzugsweise nicht von den Grafikverarbeitungs-Engines 1531-1532, N) benötigt werden, innerhalb der Caches 1562A-1562D, 1556 der Kerne und des Systemspeichers 1514 zu behalten.

[0130] Fig. 15C veranschaulicht eine andere beispielhafte Ausführungsform, bei der die Beschleuniger-Integrationsschaltung 1536 in den Prozessor 1507 integriert ist. Zumindest in dieser Ausführungsform kommunizieren die Grafikverarbeitungs-Engines 1531-1532, N direkt über die Hochgeschwindigkeitsverbindung 1540 mit der Beschleuniger-Integrationsschaltung 1536 über die Schnittstelle 1537 und die Schnittstelle 1535 (die wiederum eine beliebige Form von Bus- oder Schnittstellenprotokoll verwenden kann). Die Beschleuniger-Integrationsschaltung 1536 kann dieselben Operationen durchführen wie diejenigen, die in Bezug auf **Fig. 15B** beschrieben sind, aber möglicherweise mit einem höheren Durchsatz, da sie sich in unmittelbarer Nähe zu dem Kohärenzbus 1564 und den Caches 1562A-1562D, 1556 befindet. Mindestens eine Ausführungsform unterstützt unterschiedliche Programmiermodelle, einschließlich eines Programmiermodells mit dediziertem Prozess (ohne Virtualisierung des Grafikbeschleunigungsmoduls) und gemeinsam genutzter Programmiermodelle (mit Virtualisierung), die Programmiermodelle beinhalten können, die durch die Beschleuniger-Integrationsschaltung 1536 gesteuert werden, und Programmiermodelle, die durch das Grafikbeschleunigungsmodul 1546 gesteuert werden.

[0131] In mindestens einer Ausführungsform sind die Grafikverarbeitungs-Engines 1531-1532, N für eine einzelne Anwendung oder einen einzelnen Prozess unter einem einzelnen Betriebssystem dediziert. In mindestens einer Ausführungsform kann eine einzelne Anwendung andere Anwendungsanforderungen zu den Grafikverarbeitungs-Engines 1531-1532, N lenken und so eine Virtualisierung innerhalb einer VM/Partition bereitstellen.

[0132] In mindestens einer Ausführungsform können die Grafikverarbeitungs-Engines 1531-1532, N von mehreren VM-/Anwendungspartitionen gemeinsam genutzt werden. In mindestens einer Ausführungsform können gemeinsam genutzte Modelle einen System-Hypervisor verwenden, um die Grafikverarbeitungs-Engines 1531-1532, N zu virtualisieren, um Zugriff durch jedes Betriebssystem zu ermöglichen. Bei Systemen mit einzelner Partition ohne einen Hypervisor befinden sich die Grafikverarbeitungs-Engines 1531-1532, N im Besitz eines Betriebssystems. In mindestens einer Ausführungsform kann ein Betriebssystem die Grafikverarbeitungs-Engines 1531-1532, N virtualisieren, um Zugriff auf jeden Prozess oder jede Anwendung bereitzustellen.

[0133] In mindestens einer Ausführungsform wählt das Grafikbeschleunigungsmodul 1546 oder eine einzelne Grafikverarbeitungs-Engine 1531-1532, N ein Prozesselement unter Verwendung eines Prozessidentifikators aus. In mindestens einer Ausführungsform sind die Prozesselemente im Systemspeicher 1514 gespeichert und können unter Verwendung der in dieser Schrift beschriebenen Techniken zur Übersetzung von effektiven Adressen in reale Adressen adressiert werden. In mindestens einer Ausführungsform kann ein Prozessidentifikator ein implementierungsspezifischer Wert sein, der einem Host-Prozess bereitgestellt wird, wenn er seinen Kontext bei der Grafikverarbeitungs-Engine 1531-1532, 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.

[0134] Fig. 15D veranschaulicht eine beispielhafte Beschleuniger-Integrationsscheibe 1590. Im vorliegenden Zusammenhang umfasst eine „Scheibe“ einen vorgegebenen Abschnitt der Verarbeitungsressourcen der Beschleuniger-Integrationsschaltung 1536. Der effektive Adressraum 1582 der Anwendung innerhalb des Systemspeichers 1514 speichert Prozesselemente 1583. In einer Ausführungsform werden Prozesselemente 1583 als Reaktion auf GPU-Aufrufe 1581 von Anwendungen 1580, die auf dem Prozessor 1507 ausgeführt werden, gespeichert. Ein Prozesselement 1583 enthält den Prozesszustand für die entsprechende Anwendung 1580. Ein in dem Prozesselement 1583 enthaltener Arbeitsdeskriptor (work descriptor - WD) 1584 kann eine einzelne von einer Anwendung angeforderte Aufgabe sein oder einen Zeiger auf eine Warteschlange von Aufgaben enthalten. In mindestens einer Ausführungsform ist der WD 1584 ein Zeiger auf eine Aufgabeanforderungswarteschlange im Adressraum 1582 einer Anwendung.

[0135] Das Grafikbeschleunigungsmodul 1546 und/oder die einzelnen Grafikverarbeitungs-Engines 1531-1532, N können von allen oder einer Teilmenge der Prozesse in einem System gemeinsam genutzt werden. In mindestens einer Ausführungsform kann eine Infrastruktur zum Einrichten des Prozesszustands und zum Senden eines WD 1584 an ein Grafikbeschleunigungsmodul 1546 zum Starten einer Aufgabe in einer virtualisierten Umgebung enthalten sein.

[0136] In mindestens einer Ausführungsform ist ein Programmiermodell mit dediziertem Prozess implementierungsspezifisch. In diesem Modell besitzt ein einzelner Prozess das Grafikbeschleunigungsmodul 1546 oder eine einzelne Grafikverarbeitungs-Engine 1531. Da sich das Grafikbeschleunigungsmodul 1546 im Besitz eines einzelnen Prozesses befindet, initialisiert ein Hypervisor die Beschleuniger-Integrationsschaltung 1536 für eine besitzende Partition und initialisiert ein Betriebssystem die Beschleuniger-Integrationsschaltung 1536 für einen besitzenden Prozess, wenn das Grafikbeschleunigungsmodul 1546 zugewiesen ist.

[0137] Im Betrieb ruft eine WD-Abrufeinheit 1591 in der Beschleunigerintegrations-Scheibe 1590 den nächsten WD 1584 ab, der eine Angabe der Arbeit beinhaltet, die durch eine oder mehrere Grafikverarbeitungs-Engines des Grafikbeschleunigungsmoduls 1546 zu erledigen ist. Daten vom WD 1584 können in Registern 1545 gespeichert und von der MMU 1539, der Unterbrechungsverwaltungsschaltung 1547 und/oder der Kontextverwaltungsschaltung 1548 verwendet werden, wie veranschaulicht. Eine Ausführungsform der MMU 1539 beinhaltet zum Beispiel eine Segment-/Seitenlaufschaltung für den Zugriff auf Segment-/Seitentabellen 1586 innerhalb des virtuellen Adressraums 1585 des OS. Die Unterbrechungsverwaltungsschaltung 1547 kann von dem Grafikbeschleunigungsmodul 1546 empfangene Unterbrechungsereignisse 1592 verarbeiten. Beim Durchführen von Grafikoperationen wird eine effektive Adresse 1593, die durch eine Grafikverarbeitungs-Engine 1531-1532, N erzeugt wird, durch die MMU 1539 in eine reale Adresse übersetzt.

[0138] In einer Ausführungsform wird derselbe Satz von Registern 1545 für jede Grafikverarbeitungs-Engine 1531-1532, N und/oder jedes Grafikbeschleunigungsmodul 1546 dupliziert und kann von einem Hypervisor oder Betriebssystem initialisiert werden. Jedes dieser duplizierten Register kann in einer Beschleuniger-Integrationsscheibe 1590 enthalten sein. Beispielhafte Register, die durch einen Hypervisor initialisiert werden können, sind in Tabelle 1 gezeigt.

Tabelle 1 - Durch Hypervisor initialisierte Register

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

[0139] Beispielhafte Register, die von einem Betriebssystem initialisiert werden können, sind in Tabelle 2 gezeigt.

Tabelle 2 - Durch Betriebssystem initialisierte Register

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

[0140] In einer Ausführungsform ist jeder WD 1584 spezifisch für ein konkretes Grafikbeschleunigungsmodul 1546 und/oder die Grafikverarbeitungs-Engines 1531-1532, N. Er enthält alle Informationen, die von einer Grafikverarbeitungs-Engine 1531-1532, N benötigt werden, um Arbeit zu verrichten, oder er kann ein Zeiger auf einen Speicherort sein, an dem eine Anwendung eine Befehlswarteschlange von zu verrichtender Arbeit eingerichtet hat.

[0141] Fig. 15E veranschaulicht zusätzliche Details für eine beispielhafte Ausführungsform eines gemeinsam genutzten Modells. Diese Ausführungsform beinhaltet einen realen Hypervisor-Adressraum 1598, in dem eine Prozesselementliste 1599 gespeichert ist. Auf den realen Hypervisor-Adressraum 1598 kann über einen Hypervisor 1596 zugegriffen werden, der die Grafikbeschleunigungsmodul-Engines für das Betriebssystem 1595 virtualisiert.

[0142] 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. Es gibt zwei Programmiermodelle, bei denen das Grafikbeschleunigungsmodul 1546 von mehreren Prozessen und Partitionen gemeinsam genutzt wird: über Zeitscheiben gemeinsam genutzt (time-sliced shared) und über gerichtete Grafik gemeinsam genutzt (graphics-directed shared).

[0143] In diesem Modell besitzt der System-Hypervisor 1596 das Grafikbeschleunigungsmodul 1546 und stellt seine Funktion allen Betriebssystemen 1595 zur Verfügung. Damit ein Grafikbeschleunigungsmodul 1546 die Virtualisierung durch den System-Hypervisor 1596 unterstützen kann, muss das Grafikbeschleunigungsmodul 1546 Folgendes einhalten: 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 spezifizierten Zeitspanne abgeschlossen wird, einschließlich beliebiger Übersetzungsfehler, oder das Grafikbeschleunigungsmodul 1546 stellt eine Fähigkeit bereit, die Verarbeitung einer Aufgabe vorwegzu-

nehmen. 3) Dem Grafikbeschleunigungsmodul 1546 muss Fairness zwischen den Prozessen garantiert werden, wenn es in einem gerichteten gemeinsam genutzten Programmiermodell arbeitet.

[0144] In mindestens einer Ausführungsform muss die Anwendung 1580 einen Systemaufruf des Betriebssystems 1595 mit einer Art des Grafikbeschleunigungsmoduls 1546, einem Arbeitsdeskriptor (WD), einem Autoritätsmaskenregister-(Authority Mask Register - AMR-)Wert und einem Kontextsicherungs-/-Wiederherstellungsbereichszeiger (Context Save/Restore Area Pointer - CSRP) ausführen. In mindestens einer Ausführungsform beschreibt die Art des Grafikbeschleunigungsmoduls 1546 eine gezielte Beschleunigungsfunktion für einen Systemaufruf. In mindestens einer Ausführungsform kann die Art des Grafikbeschleunigungsmoduls 1546 ein systemspezifischer Wert sein. In mindestens einer Ausführungsform ist der WD spezifisch für das Grafikbeschleunigungsmodul 1546 formatiert und er 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, um durch das Grafikbeschleunigungsmodul 1546 zu verrichtende Arbeit zu beschreiben. In einer Ausführungsform ist ein AMR-Wert ein AMR-Zustand, der für einen aktuellen Prozess zu verwenden ist. In mindestens einer Ausführungsform ist ein Wert, der an ein Betriebssystem übergeben wird, ähnlich einer Anwendung, die einen AMR festlegt. Wenn Implementierungen der Beschleuniger-Integrationsschaltung 1536 und des Grafikbeschleunigungsmoduls 1546 kein Benutzer-Autoritätsmasken-Überschreibungsregister (User Authority Mask Override Register - UAMOR) unterstützen, kann ein Betriebssystem einen aktuellen UAMOR-Wert auf einen AMR-Wert anwenden, bevor ein AMR in einem Hypervisor-Aufruf übergeben wird. Der Hypervisor 1596 kann optional einen aktuellen Autoritätsmasken-Überschreibungsregister-(Authority Mask Override Register - AMOR-) Wert anwenden, bevor ein AMR in dem Prozesselement 1583 platziert wird. In mindestens einer Ausführungsform ist CSRP eines der Register 1545, die eine effektive Adresse eines Bereichs in dem effektiven Adressraum 1582 einer Anwendung für das Grafikbeschleunigungsmodul 1546 zum Sichern und Wiederherstellen des Kontextzustands enthalten. Dieser Zeiger ist optional, wenn zwischen den Aufgaben oder beim Vorwegnehmen einer Aufgabe kein Zustand gespeichert werden muss. In mindestens einer Ausführungsform kann der Kontextsicherungs-/-Wiederherstellungsbereich gepinnter Systemspeicher sein.

[0145] Beim Empfangen eines Systemaufrufs kann das Betriebssystem 1595 verifizieren, ob die Anwendung 1580 registriert ist und die Autorität zum Verwenden des Grafikbeschleunigungsmoduls 1546 erhalten hat. Das Betriebssystem 1595 ruft dann den Hypervisor 1596 mit den in Tabelle 3 gezeigten Informationen auf.

Tabelle 3 - OS-zu-Hypervisor-Aufrufparameter

1	Ein Arbeitsdeskriptor (WD)
2	Ein Wert eines Autoritätsmaskenregisters (AMR) (möglicherweise maskiert)
3	Ein Kontextsicherungs-/-Wiederherstellungsbereichszeiger (CSRP) für effektive Adressen (EA)
4	Eine Prozess-ID (PID) und optional eine Thread-ID (TID)
5	Ein Beschleunigemutzungsaufzeichnungszeiger (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)

[0146] Beim Empfang eines Hypervisor-Aufrufs verifiziert der Hypervisor 1596, dass das Betriebssystem 1595 registriert ist und die Autorität zur Verwendung des Grafikbeschleunigungsmoduls 1546 erhalten hat. Der Hypervisor 1596 setzt dann das Prozesselement 1583 in eine mit dem Prozesselement verknüpfte Liste für eine entsprechende Art des Grafikbeschleunigungsmodul 1546 ein. Ein Prozesselement kann die in Tabelle 4 gezeigten Informationen enthalten.

Tabelle 4 - Prozesselementinformationen

1	Ein Arbeitsdeskriptor (WD)
2	Ein Wert eines Autoritätsmaskenregisters (AMR) (möglicherweise maskiert).
3	Ein Kontextsicherungs-/-Wiederherstellungsbereichszeiger (CSRP) für effektive Adressen (EA)

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

[0147] In mindestens einer Ausführungsform initialisiert der Hypervisor eine Vielzahl von Registern 1545 der Beschleuniger-Integrationsscheibe 1590.

[0148] Wie in **Fig. 15F** veranschaulicht, wird in mindestens einer Ausführungsform ein vereinheitlichter Speicher verwendet, der über einen gemeinsamen virtuellen Speicheradressraum adressierbar ist, der für den Zugriff auf die physischen Prozessorspeicher 1501-1502 und die GPU-Speicher 1520-1523 verwendet wird. In dieser Implementierung verwenden Operationen, die auf den GPUs 1510-1513 ausgeführt werden, denselben virtuellen/effektiven Speicheradressraum für den Zugriff auf die Prozessorspeicher 1501-1502 und umgekehrt, was die Programmierbarkeit vereinfacht. In einer Ausführungsform wird ein erster Abschnitt eines virtuellen/effektiven Adressraums dem Prozessorspeicher 1501 zugewiesen, ein zweiter Abschnitt dem zweiten Prozessorspeicher 1502, ein dritter Abschnitt dem GPU-Speicher 1520 usw. In mindestens einer Ausführungsform wird dadurch ein gesamter virtueller/effektiver Speicherraum (mitunter als effektiver Adressraum bezeichnet) über jeden der Prozessorspeicher 1501-1502 und GPU-Speicher 1520-1523 verteilt, sodass ein beliebiger Prozessor oder eine beliebige GPU auf einen beliebigen physischen Speicher mit einer virtuellen Adresse zugreifen kann, die auf diesen Speicher abgebildet ist.

[0149] In einer Ausführungsform stellt die Verzerrungs-/Kohärenzverwaltungsschaltung 1594A-1594E innerhalb einer oder mehrerer MMUs 1539A-1539E die Cache-Kohärenz zwischen Caches eines oder mehrerer Host-Prozessoren (z. B. 1505) und GPUs 1510-1513 sicher und implementiert Verzerrungstechniken, die physische Speicher angeben, in denen bestimmte Arten von Daten gespeichert werden sollten. Wenngleich in **Fig. 15F** mehrere Instanzen der Verzerrungs-/Kohärenzverwaltungsschaltung 1594A-1594E veranschaulicht sind, kann die Verzerrungs-/Kohärenzschaltung innerhalb einer MMU eines oder mehrerer Host-Prozessoren 1505 und/oder innerhalb der Beschleuniger-Integrationsschaltung 1536 implementiert sein.

[0150] Eine Ausführungsform ermöglicht es, den GPU-zugewiesenen Speicher 1520-1523 als Teil des Systemspeichers zuzuordnen und unter Verwendung von gemeinsam genutzter virtueller Speicher-(Shared Virtual Memory SVM-)Technologie darauf zuzugreifen, ohne jedoch Rechenleistungsnachteile zu erleiden, die der vollständigen System-Cache-Kohärenz zugeordnet sind. In mindestens einer Ausführungsform bietet die Fähigkeit des GPU-gebundenen Speichers 1520-1523, auf den als Systemspeicher ohne lästigen Cache-Kohärenz-Overhead zugegriffen wird, eine vorteilhafte Betriebsumgebung für GPU-Auslagerung. Diese Anordnung ermöglicht es der Software des Host-Prozessors 1505, Operanden einzurichten und auf Berechnungsergebnisse zuzugreifen, und zwar ohne den Overhead der traditionellen E/A-DMA-Datenkopien. Derartige traditionelle Kopien beinhalten Treiberaufrufe, Unterbrechungen und speicherzugeordnete E/A-(memory mapped I/O - MMIO-)Zugriffe, die alle im Vergleich zu einfachen Speicherzugriffen ineffizient sind. In mindestens einer Ausführungsform kann eine Fähigkeit, ohne Cache-Kohärenz-Overheads auf den GPU-gebundenen Speicher 1520-1523 zuzugreifen, für die Ausführungszeit einer ausgelagerten Berechnung entscheidend sein. In Fällen mit erheblichem Streaming-Schreibspeicherverkehr kann zum Beispiel der Cache-Kohärenz-Overhead die effektive Schreibbandbreite, die von einer GPU 1510-1513 gesehen wird, erheblich reduzieren. In mindestens einer Ausführungsform können die Effizienz der Operandeneinrichtung, die Effizienz des Ergebniszugriffs und die Effizienz der GPU-Berechnung eine Rolle beim Bestimmen der Effektivität einer GPU-Auslagerung spielen.

[0151] In mindestens einer Ausführungsform wird die Auswahl von GPU-Verzerrung und Host-Prozessorverzerrung durch eine Verzerrungs-Tracker-Datenstruktur angetrieben. Es kann zum Beispiel eine Verzerrungstabelle verwendet werden, die eine seitengranulare Struktur sein kann (d. h. mit einer Granularität einer Speicherseite gesteuert), die 1 oder 2 Bit pro GPU-gebundener Speicherseite beinhaltet. In mindestens einer Ausführungsform kann eine Verzerrungstabelle in einem gestohlenen Speicherbereich eines oder mehrerer GPU-gebundener Speicher 1520-1523 implementiert werden, mit oder ohne Verzerrungs-Cache in einer GPU 1510-1513 (um z. B. häufig/kürzlich verwendete Einträge einer Verzerrungstabelle zwischenzuspeichern). Alternativ kann eine gesamte Verzerrungstabelle innerhalb einer GPU gepflegt werden.

[0152] In mindestens einer Ausführungsform wird vor dem tatsächlichen Zugriff auf einen GPU-Speicher auf einen Verzerrungstabelleneintrag zugegriffen, der jedem Zugriff auf den GPU-gebundenen Speicher 1520-1523 zugeordnet ist, wodurch die folgenden Operationen verursacht werden. Zunächst werden lokale Anforderungen von der GPU 1510-1513, die ihre Seite in der GPU-Verzerrung finden, direkt an einen entsprechenden GPU-Speicher 1520-1523 weitergeleitet. Lokale Anforderungen von einer GPU, die ihre Seite in der Host-Verzerrung findet, werden an den Prozessor 1505 weitergeleitet (z. B. über eine Hochgeschwindigkeitsverbindung, wie vorstehend erörtert). In einer Ausführungsform schließen Anforderungen von dem Prozessor 1505, die eine angeforderte Seite in der Host-Prozessor-Verzerrung finden, eine Anforderung wie ein normales Lesen des Speichers ab. Alternativ können Anforderungen, die an eine GPU-Verzerrungsseite gerichtet sind, an die GPU 1510-1513 weitergeleitet werden. In mindestens einer Ausführungsform kann eine GPU dann eine Seite in eine Host-Prozessor-Verzerrung umwandeln, falls sie derzeitig keine Seite verwendet. In mindestens einer Ausführungsform kann der Verzerrungszustand einer Seite entweder durch einen softwarebasierten Mechanismus, einen hardwareunterstützten softwarebasierten Mechanismus oder, für einen begrenzten Satz von Fällen, einen rein hardwarebasierten Mechanismus geändert werden.

[0153] Ein Mechanismus zum Ändern des Verzerrungszustands setzt in mindestens einer Ausführungsform einen API-Aufruf (z. B. OpenCL) ein, der wiederum den Vorrichtungstreiber einer GPU aufruft, der wiederum eine Nachricht an eine GPU sendet (oder einen Befehlsdeskriptor in die Warteschlange stellt), die sie anweist, einen Verzerrungszustand zu ändern und bei einigen Übergängen eine Cache-Leerungsoperation in einem Host durchzuführen. In mindestens einer Ausführungsform wird eine Cache-Leerungsoperation für einen Übergang von der Verzerrung des Host-Prozessors 1505 zu der Verzerrung der GPU verwendet, jedoch nicht für einen entgegengesetzten Übergang.

[0154] In einer Ausführungsform wird die Cache-Kohärenz aufrechterhalten, indem bewirkt wird, dass GPU-verzerrte Seiten durch den Host-Prozessor 1505 vorübergehend nicht zwischengespeichert werden können. Um auf diese Seiten zuzugreifen, kann 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 vorteilhaft, sicherzustellen, dass GPU-verzerrte Seiten solche sind, die von einer GPU, aber nicht von dem Host-Prozessor 1505, benötigt werden und umgekehrt.

[0155] Inferenz- und/oder Trainingslogik 915 wird verwendet, um eine oder mehrere Ausführungsformen durchzuführen. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt.

[0156] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0157] Fig. 16 veranschaulicht beispielhafte integrierte Schaltungen und zugeordnete Grafikprozessoren, die unter Verwendung eines oder mehrerer IP-Kerne hergestellt werden können, gemäß verschiedenen in dieser Schrift beschriebenen Ausführungsformen. Zusätzlich zu dem, was veranschaulicht ist, können andere Logik und Schaltungen in mindestens einer Ausführungsform enthalten sein, einschließlich zusätzlicher Grafikprozessoren/-kerne, Peripherieschnittstellensteuerungen oder Universalprozessorkeme.

[0158] Fig. 16 ist ein Blockdiagramm, das eine beispielhafte integrierte Schaltung 1600 als System auf einem Chip, die unter Verwendung eines oder mehrerer IP-Kerne hergestellt werden kann, gemäß mindestens einer Ausführungsform veranschaulicht. In mindestens einer Ausführungsform beinhaltet die integrierte Schaltung 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, von denen jeder beliebige ein modularer IP-Kern sein kann. In mindestens einer Ausführungsform beinhaltet die integrierte Schaltung 1600 Peripherie- oder Buslogik, die 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 mit einer oder mehreren von einer High-Definition-Multimedia-Interface-(HDMI-)Steuerung 1650 und einer Mobile-Industry-Processor-Interface-(MIPI-)Anzeigeschnittstelle 1655 gekoppelt ist. In mindestens einer Ausführungsform kann der Speicher durch ein Flash-Speicherteilsystem 1660 bereitgestellt sein, das Flash-Speicher und eine Flash-Speichersteuerung beinhaltet. In mindestens einer Ausführungsform kann die 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.

[0159] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 in der integrierten Schaltung 1600 für Inferenz- oder Vorhersageoperationen verwendet werden, die zumindest teilweise auf Gewichtungsparametern basieren, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0160] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0161] Die **Fig. 17A-17B** veranschaulichen beispielhafte integrierte Schaltungen und zugeordnete Grafikprozessoren, die unter Verwendung eines oder mehrerer IP-Kerne hergestellt werden können, gemäß verschiedenen in dieser Schrift beschriebenen Ausführungsformen. Zusätzlich zu dem, was veranschaulicht ist, können andere Logik und Schaltungen in mindestens einer Ausführungsform enthalten sein, einschließlich zusätzlicher Grafikprozessoren/-kerne, Peripherieschnittstellensteuerungen oder Universalprozessorkeme.

[0162] Die **Fig. 17A-17sind** Blockdiagramme, die beispielhafte Grafikprozessoren zur Verwendung innerhalb eines SoC gemäß in dieser Schrift beschriebenen Ausführungsformen veranschaulichen. **Fig. 17A** veranschaulicht einen beispielhaften Grafikprozessor 1710 einer integrierte Schaltung als System auf einem Chip, die unter Verwendung eines oder mehrerer IP-Kerne hergestellt werden kann, gemäß mindestens einer Ausführungsform. **Fig. 17B** veranschaulicht einen zusätzlichen beispielhaften Grafikprozessor 1740 einer integrierte Schaltung als System auf einem Chip, die unter Verwendung eines oder mehrerer IP-Kerne hergestellt werden kann, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist der Grafikprozessor 1710 aus **Fig. 17A** ein Grafikprozessorkern mit niedriger Leistung. In mindestens einer Ausführungsform ist der Grafikprozessor 1740 aus **Fig. 17B** ein Grafikprozessorkern mit höherer Rechenleistung. In mindestens einer Ausführungsform kann jeder der Grafikprozessoren 1710, 1740 eine Variante des Grafikprozessors 1610 aus **Fig. 16** sein.

[0163] In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 1710 einen Vertex-Prozessor 1705 und einen oder mehrere Fragmentprozessor(en) 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 separate Logik ausführen, sodass der Vertex-Prozessor 1705 optimiert ist, um Operationen für Vertex-Shader-Programme auszuführen, während ein oder mehrere Fragmentprozessor(en) 1715A-1715N Shading-Operationen für Fragmente (z. B. Pixel) für Fragment- oder Pixel-Shader-Programme ausführen. In mindestens einer Ausführungsform führt der Vertex-Prozessor 1705 eine Vertex-Verarbeitungsstufe einer 3D-Grafikpipeline durch und erzeugt Primitiv- und Vertexdaten. In mindestens einer Ausführungsform verwenden ein oder mehrere Fragmentprozessor(en) 1715A-1715N Primitiv- und Vertexdaten, die durch den Vertex-Prozessor 1705 erzeugt werden, um einen Bildpuffer zu erzeugen, der auf einer Anzeigevorrichtung angezeigt wird. In mindestens einer Ausführungsform sind ein oder mehrere Fragmentprozessor(en) 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 bereitgestellt ist.

[0164] In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 1710 zusätzlich eine(n) oder mehrere Speicherverwaltungseinheiten (MMUs) 1720A-1720B, Cache(s) 1725A-1725B und Schaltungszusammenschaltung(en) 1730A-1730B. In mindestens einer Ausführungsform stellen eine oder mehrere MMU(s) 1720A-1720B eine Abbildung von virtuellen auf physische Adressen für den Grafikprozessor 1710 bereit,

einschließlich für den Vertexprozessor 1705 und/oder den/die Fragmentprozessor(en) 1715A-1715N, die auf in Speicher gespeicherte Vertex- oder Bild-/Texturdaten verweisen kann, zusätzlich zu den in einem oder mehreren Cache(s) 1725A-1725B gespeicherten Vertex- oder Bild-/Texturdaten. In mindestens einer Ausführungsform können eine oder mehrere MMU(s) 1720A-1720B mit anderen MMUs innerhalb des Systems synchronisiert werden, einschließlich einer oder mehrerer MMUs, die einem oder mehreren Anwendungsprozessor(en) 1605, Bildprozessoren 1615 und/oder Videoprozessoren 1620 aus **Fig. 16** zugeordnet sind, sodass jeder Prozessor 1605-1620 an einem gemeinsam genutzten oder vereinheitlichten virtuellen Speichersystem teilnehmen kann. In mindestens einer Ausführungsform ermöglichen es eine oder mehrere Schaltungszusammenschaltung(en) 1730A-1730B dem Grafikprozessor 1710, entweder über einen internen Bus des SoC oder über eine direkte Verbindung eine Schnittstelle mit anderen IP-Kernen innerhalb des SoC zu bilden.

[0165] In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 1740 eine(n) oder mehrere MMU(s) 1720A-1720B, Cache(s) 1725A-1725B und Schaltungszusammenschaltung(en) 1730A-1730B des Grafikprozessors 1710 aus **Fig. 17A**. In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 1740 einen oder mehrere Shader-Kern(e) 1755A-1755N (z. B. 1755A, 1755B, 1755C, 1755D, 1755E, 1755F bis 1755N-1 und 1755N), was eine vereinheitlichte Shader-Kernarchitektur bereitstellt, bei der ein einzelner Kern oder eine einzelne Art von Kern alle Arten von programmierbarem Shader-Code ausführen kann, einschließlich Shader-Programmcode zur Implementierung von Vertex-Shadem, Fragment-Shadern und/oder Rechen-Shadern. In mindestens einer Ausführungsform kann eine Anzahl der Shader-Kerne variieren. In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 1740 einen Zwischenkern-Aufgaben-Manager 1745, der als Thread-Zuteiler fungiert, um Ausführungs-Threads einem oder mehreren Shader-Kernen 1755A-1755N zuzuteilen, sowie eine Kachelungseinheit 1758 zum Beschleunigen von Kachelungsoperationen für kachelbasiertes Rendering, bei dem Rendering-Operationen für eine Szene im Bildraum unterteilt werden, um zum Beispiel die lokale räumliche Kohärenz innerhalb einer Szene auszunutzen oder die Nutzung interner Caches zu optimieren.

[0166] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 in der integrierten Schaltung 17A und/oder 17B für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0167] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0168] Die **Fig. 18A-18B** veranschaulichen zusätzliche beispielhafte Grafikprozessorlogik gemäß in dieser Schrift beschriebenen Ausführungsformen. **Fig. 18A** veranschaulicht einen Grafikkern 1800, der in mindestens einer Ausführungsform in dem Grafikprozessor 1610 aus **Fig. 16** enthalten sein kann und in mindestens einer Ausführungsform ein vereinheitlichter 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 Mehrchipmodul geeignet ist.

[0169] 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 Scheiben 1801A-1801N oder eine Partition für jeden Kern beinhalten und ein Grafikprozessor kann mehrere Instanzen des Grafikkerns 1800 beinhalten. Die Scheiben 1801A-1801N können eine 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 Scheiben 1801A-1801N einen Satz zusätzlicher Funktionseinheiten (additional function units - AFUs 1812A-1812N), Gleitkommaeinheiten (floating-point units - FPU 1814A-1814N), arithmetisch-logischer Einheiten (ALUs 1816-1816N) für Integer, 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.

[0170] In mindestens einer Ausführungsform können die FPUs 1814A-1814N Gleitkommaoperationen mit einfacher Genauigkeit (32 Bit) und mit 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 ganzzahlige Operationen mit variabler Genauigkeit mit 8-Bit-, 16-Bit- und 32-Bit-Genauigkeit durchführen und für Operationen mit gemischter Genauigkeit konfiguriert sein. In mindestens einer Ausführungsform können die MPUs 1817A-1817N auch für Matrixoperationen mit gemischter Genauigkeit konfiguriert sein, einschließlich Gleitkomma- und ganzzahligen 8-Bit-Operationen mit halber Genauigkeit. In mindestens einer Ausführungsform können die MPUs 1817A-1817N eine Vielfalt von Matrixoperationen durchführen, um Anwendungsrahmen für maschinelles Lernen zu beschleunigen, einschließlich des Ermöglichens der Unterstützung für eine beschleunigte allgemeine Matrix-zu-Matrix-Multiplikation (general matrix to matrix multiplication - GEMM). In mindestens einer Ausführungsform können die AFUs 1812A-1812N zusätzliche logische Operationen ausführen, die nicht durch Gleitkomma- oder Ganzzahleinheiten unterstützt werden, einschließlich trigonometrischer Operationen (z. B. Sinus, Cosinus usw.).

[0171] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 in dem Grafikkern 1800 für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0172] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0173] **Fig. 18B** veranschaulicht eine Universalverarbeitungseinheit (GPGPU) 1830, die in mindestens einer Ausführungsform dazu konfiguriert sein kann, zu ermöglichen, dass hochparallele Rechenoperationen durch ein Array von Grafikverarbeitungseinheiten durchgeführt werden. In mindestens einer Ausführungsform kann die GPGPU 1830 direkt mit anderen Instanzen der GPGPU 1830 verknüpft sein, um ein Multi-GPU-Cluster zu erzeugen, um die Trainingsgeschwindigkeit für tiefe neuronale Netze zu verbessern. In mindestens einer Ausführungsform beinhaltet die GPGPU 1830 eine Host-Schnittstelle 1832, um eine Verbindung mit einem Host-Prozessor zu ermöglichen. In mindestens einer Ausführungsform ist die Host-Schnittstelle 1832 eine PCI-Express-Schnittstelle. In mindestens einer Ausführungsform kann die Host-Schnittstelle 1832 eine anbieterspezifische Kommunikationsschnittstelle oder Kommunikationsstruktur sein. In mindestens einer Ausführungsform empfängt die GPGPU 1830 Befehle von einem Host-Prozessor und sie verwendet einen globalen Planer 1834, um diesen Befehlen zugeordnete Ausführungs-Threads an einen Satz von Rechenclustern 1836A-1836H zu verteilen. In mindestens einer Ausführungsform nutzen die Rechencluster 1836A-1836H einen Cache-Speicher 1838 gemeinsam. In mindestens einer Ausführungsform kann der Cache-Speicher 1838 als Cache höherer Ebene für Cache-Speicher innerhalb der Rechencluster 1836A-1836H dienen.

[0174] In mindestens einer Ausführungsform beinhaltet die GPGPU 1830 Speicher 1844A-1844B, der über einen Satz von Speichersteuerungen 1842A-1842B mit den Rechenclustern 1836A-1836H gekoppelt ist. In mindestens einer Ausführungsform kann der Speicher 1844A-1844B verschiedene Arten von Speichervorrichtungen beinhalten, einschließlich dynamischen Direktzugriffsspeichers (DRAM) oder Grafik-Direktzugriffsspeichers, wie etwa synchronen Grafik-Direktzugriffsspeichers (SGRAM), einschließlich Grafik-Double-Data-Rate-(GDDR-)Speichers.

[0175] In mindestens einer Ausführungsform beinhalten die Rechencluster 1836A-1836H jeweils einen Satz von Grafikkernen, wie etwa den Grafikkern 1800 aus **Fig. 18A**, der mehrere Arten von logischen Einheiten für Integer und Gleitkommazahlen beinhalten kann, die Rechenoperationen mit einer Reihe von Genauigkeiten durchführen können, einschließlich solcher, die für Berechnungen für maschinelles Lernen geeignet sind. Zum Beispiel kann in mindestens einer Ausführungsform mindestens eine Teilmenge der Gleitkommaeinheiten in jedem der Rechencluster 1836A-1836H dazu konfiguriert sein, 16-Bit- oder 32-Bit-Gleitkommaoperationen durchzuführen, während eine andere Teilmenge der Gleitkommaeinheiten dazu konfiguriert sein kann, 64-Bit-Gleitkommaoperationen durchzuführen.

[0176] In mindestens einer Ausführungsform können mehrere Instanzen der GPGPU 1830 dazu konfiguriert sein, als Rechencluster zu 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 Host-Schnittstelle 1832. In mindestens einer Ausführungsform beinhaltet die GPGPU 1830 einen E/A-Hub 1839, der die GPGPU 1830 mit einer GPU-Verbindung 1840 koppelt, die eine direkte Verbindung zu anderen Instanzen der GPGPU 1830 ermöglicht. In mindestens einer Ausführungsform ist die GPU-Verbindung 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-Verbindung 1840 mit einer Hochgeschwindigkeitszusammenschaltung gekoppelt, um Daten an andere GPGPUs oder Parallelprozessoren zu senden und davon zu empfangen. In mindestens einer Ausführungsform befinden sich mehrere Instanzen der GPGPU 1830 in separaten Datenverarbeitungssystemen und kommunizieren über eine Netzwerkvorrichtung, auf die über die Host-Schnittstelle 1832 zugegriffen werden kann. In mindestens einer Ausführungsform kann die GPU-Verbindung 1840 dazu konfiguriert sein, eine Verbindung zu einem Host-Prozessor zusätzlich zu oder alternativ zu der Host-Schnittstelle 1832 zu ermöglichen.

[0177] 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 weniger Rechencluster 1836A-1836H beinhalten, als wenn die GPGPU zum Trainieren eines neuronalen Netzes verwendet wird. In mindestens einer Ausführungsform kann sich die dem Speicher 1844A-1844B zugeordnete Speichertechnologie zwischen Inferenz- und Trainingskonfigurationen unterscheiden, wobei den Trainingskonfigurationen Speichertechnologien mit höherer Bandbreite gewidmet sind. In mindestens einer Ausführungsform kann die Inferenzkonfiguration der GPGPU 1830 inferenzspezifische Anweisungen unterstützen. In mindestens einer Ausführungsform kann eine Inferenzkonfiguration zum Beispiel Unterstützung für eine oder mehrere 8-Bit-Integer-Skalarprodukt-Anweisungen bereitstellen, die während der Inferenzoperationen für eingesetzte neuronale Netze verwendet werden können.

[0178] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 in der GPGPU 1830 für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0179] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0180] **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 kann in einen oder mehrere Prozessor(en) 1902 integriert sein. In mindestens einer Ausführungsform ist der Speicher-Hub 1905 über eine Kommunikationsverbindung 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 einer Anzeigesteuerung, die in einem oder mehreren Prozessor(en) 1902 enthalten sein kann, ermöglichen, einer oder mehreren Anzeigevorrichtung(en) 1910A Ausgaben bereitzustellen. In mindestens einer Ausführungsform können eine oder mehrere mit dem E/A-Hub 1907 gekoppelte Anzeigevorrichtung(en) 1910A eine lokale, interne oder eingebettete Anzeigevorrichtung beinhalten.

[0181] In mindestens einer Ausführungsform beinhaltet das Verarbeitungsteilsystem 1901 einen oder mehrere Parallelprozessor(en) 1912, die über einen Bus oder eine andere Kommunikationsverbindung 1913 an den Speicher-Hub 1905 gekoppelt sind. In mindestens einer Ausführungsform kann die Kommunikationsverbindung 1913 eine von einer beliebigen Anzahl von standardbasierten Kommunikationsverbindungstechnolo-

gien oder -protokollen sein, wie etwa PCI Express, ohne darauf beschränkt zu sein, oder eine anbieterspezifische Kommunikationsschnittstelle oder Kommunikationsstruktur. In mindestens einer Ausführungsform bilden ein oder mehrere Parallelprozessor(en) 1912 ein rechnerisch fokussiertes Parallel- oder Vektorverarbeitungssystem, das eine große Anzahl von Verarbeitungskernen und/oder Verarbeitungsclustern beinhalten kann, wie etwa einen Prozessor mit vielen integrierten Kernen (Many Integrated Core - MIC). In mindestens einer Ausführungsform bilden einige oder alle 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 ein oder mehrere Parallelprozessor(en) 1912 auch eine Anzeigesteuerung und eine Anzeigeschnittstelle (nicht gezeigt) beinhalten, um eine direkte Verbindung zu einer oder mehreren Anzeigevorrichtung(en) 1910B zu ermöglichen.

[0182] 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 ein Netzwerkadapter 1918 und/oder ein drahtloser Netzwerkadapter 1919, die in (eine) Plattform(en) integriert werden können, sowie verschiedenen anderen Vorrichtungen, die über eine oder mehrere Erweiterungsvorrichtung(en) 1920 hinzugefügt werden können. In mindestens einer Ausführungsform kann der Netzwerkadapter 1918 ein Ethemet-Adapter oder ein anderer drahtgebundener Netzwerkadapter sein. In mindestens einer Ausführungsform kann der drahtlose Netzwerkadapter 1919 eine oder mehrere Wi-Fi-, Bluetooth-, Nahfeldkommunikations- (NFC-) oder eine andere Netzwerkvorrichtung beinhalten, die ein oder mehrere drahtlose Funkgeräte beinhaltet.

[0183] In mindestens einer Ausführungsform kann das Rechensystem 1900 andere, nicht explizit gezeigte Komponenten beinhalten, z. B. USB- oder andere Portverbindungen, optische Speicherlaufwerke, Videoaufnahmevorrichtungen usw., die ebenfalls mit dem E/A-Hub 1907 verbunden sein können. In mindestens einer Ausführungsform können die Kommunikationspfade, die verschiedene Komponenten in **Fig. 19** zusammenschalten, unter Verwendung beliebiger geeigneter Protokolle implementiert werden, wie etwa auf PCI (Peripheral Component Interconnect) basierender Protokolle (z. B. PCI-Express) oder anderer Bus- oder Punkt-zu-Punkt-Kommunikationsschnittstellen und/oder -protokoll(e), wie etwa NV-Link-Hochgeschwindigkeitszusammenschaltung, oder Zusammenschaltungsprotokolle.

[0184] In mindestens einer Ausführungsform beinhalten ein oder mehrere Parallelprozessor(en) 1912 Schaltungen, die für die Grafik- und Videoverarbeitung optimiert sind, einschließlich zum Beispiel Videoausgabeschaltungen, und stellen eine Grafikverarbeitungseinheit (GPU) dar. In mindestens einer Ausführungsform beinhalten ein oder mehrere Parallelprozessor(en) 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 ein oder mehrere Parallelprozessor(en) 1912, der Speicher-Hub 1905, der/die Prozessor(en) 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 Mehrchipmodul (multi-chip module - MCM) integriert sein, das mit anderen Mehrchipmodulen Modulen zu einem modularen Rechensystem zusammengeschaltet sein kann.

[0185] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 in dem System aus FIG. 1900 für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0186] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

PROZESSOREN

[0187] **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 Parallelprozessor(en) 1912, die in **Fig. 19** gemäß einer beispielhaften Ausführungsform gezeigt sind.

[0188] 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 Kommunikation mit anderen Vorrichtungen ermöglicht, einschließlich anderer Instanzen 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 des Speicher-Hubs 1905, mit anderen Vorrichtungen verbunden. In mindestens einer Ausführungsform bilden Verbindungen zwischen dem Speicher-Hub 1905 und der E/A-Einheit 2004 eine Kommunikationsverbindung 1913. In mindestens einer Ausführungsform ist die E/A-Einheit 2004 mit einer Host-Schnittstelle 2006 und einer Speicherkreuzschiene 2016 verbunden, wobei die Host-Schnittstelle 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.

[0189] 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 dazu konfiguriert ist, Befehle oder andere Arbeitselemente an ein Verarbeitungsclusterarray 2012 zu verteilen. In mindestens einer Ausführungsform stellt der Planer 2010 sicher, dass das Verarbeitungsclusterarray 2012 ordnungsgemäß konfiguriert ist und sich in einem gültigen Zustand befindet, bevor Aufgaben an das Verarbeitungsclusterarray 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 dazu konfigurierbar, komplexe Planungs- und Arbeitsverteilungsoperationen mit grober und feiner Granularität durchzuführen, was eine schnelle Präemption und Kontextumschaltung von Threads ermöglicht, die auf dem Verarbeitungsarray 2012 ausgeführt werden. In mindestens einer Ausführungsform kann die Host-Software Arbeitslasten für die Planung auf dem Verarbeitungsarray 2012 über eine von mehreren Grafikverarbeitungs-Doorbells prüfen. In mindestens einer Ausführungsform können die Arbeitslasten dann automatisch auf das Verarbeitungsarray 2012 durch die Logik des Planers 2010 innerhalb eines Mikrocontrollers, einschließlich des Planers 2010, verteilt werden.

[0190] In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2012 bis zu „N“ Verarbeitungscluster beinhalten (z. B. Cluster 2014A, Cluster 2014B bis Cluster 2014N). In mindestens einer Ausführungsform kann jedes Cluster 2014A-2014N des Verarbeitungsclusterarrays 2012 eine große Anzahl von gleichzeitigen 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 von der Arbeitslast variieren können, die für jede Art von Programm oder Berechnung entsteht. In mindestens einer Ausführungsform kann die Planung dynamisch durch den Planer 2010 gehandhabt werden oder teilweise durch Compiler-Logik während der Kompilierung der Programmlogik unterstützt werden, die für die Ausführung durch das Verarbeitungsclusterarray 2012 konfiguriert ist. In mindestens einer Ausführungsform können unterschiedliche Cluster 2014A-2014N des Verarbeitungsclusterarrays 2012 zum Verarbeiten unterschiedlicher Arten von Programmen oder zum Durchführen unterschiedlicher Arten von Berechnungen zugewiesen sein.

[0191] In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2012 dazu konfiguriert sein, verschiedene Arten von Parallelverarbeitungsoperationen durchzuführen. In mindestens einer Ausführungsform ist das Verarbeitungsclusterarray 2012 dazu konfiguriert, Universal-Parallelberechnungsoperationen durchzuführen. In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2012 zum Beispiel Logik zum Ausführen von Verarbeitungsaufgaben beinhalten, einschließlich des Filterns von Video- und/oder Audiodaten, des Durchführens von Modellierungsoperationen, einschließlich Physikoperationen, und des Durchführens von Datentransformationen.

[0192] In mindestens einer Ausführungsform ist das Verarbeitungsclusterarray 2012 dazu konfiguriert, Parallelgrafikverarbeitungsoperationen durchzuführen. In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2012 zusätzliche Logik beinhalten, um die Ausführung solcher Grafikverarbeitungsoperationen zu unterstützen, einschließlich, aber nicht beschränkt auf, Texturabtastlogik, um Texturoperationen durchzuführen, sowie Tesselationslogik und andere Vertex-Verarbeitungslogik. In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2012 dazu konfiguriert sein, grafikverarbeitungsbezogene Shader-Programme auszuführen, wie etwa ohne Einschränkung Vertex-Shader, Tesselations-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 übertragen. In mindestens einer Ausführungsform können die übertragenen Daten während der Verarbeitung in einem chipinternen Speicher (z. B. dem Parallelprozessorspeicher 2022) während der Verarbeitung gespeichert und dann wieder in den Systemspeicher geschrieben werden.

[0193] Wenn die Parallelverarbeitungseinheit 2002 zum Durchführen der Grafikverarbeitung verwendet wird, kann der Planer 2010 in mindestens einer Ausführungsform dazu konfiguriert sein, eine Verarbeitungsarbeitslast in annähernd gleich große Aufgaben aufzuteilen, 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 dazu konfiguriert sein, verschiedene Arten von Verarbeitung durchführen. Zum Beispiel kann in mindestens einer Ausführungsform ein erster Abschnitt dazu konfiguriert sein, Vertex-Shading und Topologieerzeugung durchzuführen, ein zweiter Abschnitt dazu konfiguriert sein, Tesselations- und Geometrie-Shading durchzuführen und ein dritter Abschnitt dazu konfiguriert sein, Pixel-Shading oder andere Bildschirmraumoperationen durchzuführen, um ein gerendertes Bild zur Anzeige zu produzieren. In mindestens einer Ausführungsform können Zwischendaten, die durch eines oder mehrere der Cluster 2014A-2014N erzeugt werden, in Puffern gespeichert werden, um zu ermöglichen, dass Zwischendaten zwischen den Clustern 2014A-2014N zur weiteren Verarbeitung übertragen werden.

[0194] In mindestens einer Ausführungsform kann das Verarbeitungsclusterarray 2012 auszuführende Verarbeitungsaufgaben über den Planer 2010 empfangen, der von dem Frontend 2008 Befehle empfängt, die Verarbeitungsaufgaben definieren. In mindestens einer Ausführungsform können Verarbeitungsaufgaben Indizes von zu verarbeitenden Daten beinhalten, z. B. Oberflächen-(Patch-)Daten, Primitivdaten, Vertexdaten und/oder Pixeldaten sowie Zustandsparameter und Befehle, die definieren, wie Daten verarbeitet werden sollen (z. B. welches Programm ausgeführt werden soll). In mindestens einer Ausführungsform kann der Planer 2010 dazu konfiguriert sein, Indizes abzurufen, die Aufgaben entsprechen, oder kann Indizes von dem Frontend 2008 empfangen. In mindestens einer Ausführungsform kann das Frontend 2008 dazu konfiguriert sein, sicherzustellen, dass das Verarbeitungsclusterarray 2012 in einen gültigen Zustand konfiguriert wird, bevor eine durch eingehende Befehlspuffer (z. B. Batch-Puffer, Push-Puffer usw.) vorgegebene Arbeitslast initiiert wird.

[0195] In mindestens einer Ausführungsform kann jede von einer oder mehreren Instanzen der Parallelverarbeitungseinheit 2002 mit dem 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 2020A, Partitionseinheit 2020B bis Partitionseinheit 2020N) 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 dazu konfiguriert, gleich einer Anzahl von Speichereinheiten zu sein, 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 darf eine Anzahl der Partitionseinheiten 2020A-2020N nicht gleich einer Anzahl der Speichervorrichtungen sein.

[0196] In mindestens einer Ausführungsform können die Speichereinheiten 2024A-2024N verschiedene Arten von Speichervorrichtungen beinhalten, einschließlich dynamischen Direktzugriffsspeichers (DRAM) oder Grafik-Direktzugriffsspeichers, wie etwa synchronen Grafik-Direktzugriffsspeichers (SGRAM), einschließlich Grafik-Double-Data-Rate-(GDDR-)Speichers. In mindestens einer Ausführungsform können die Speichereinheiten 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 Bildpuffer 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 vereinheitlichten Speicherausgestaltung ausgeschlossen werden, die Systemspeicher in Verbindung mit lokalem Cache-Speicher verwendet.

[0197] In mindestens einer Ausführungsform kann ein beliebiges 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 dazu konfiguriert sein, eine Ausgabe jedes Clusters 2014A-2014N an eine beliebige Partitionseinheit 2020A-2020N oder an ein anderes Cluster 2014A-2014N zu übertragen, das zusätzliche Verarbeitungsoperationen an einer Ausgabe durchführen kann. In mindestens einer Ausführungsform kann jedes 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.

[0198] 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 dazu konfiguriert sein, zusammenzuarbeiten, auch wenn die unterschiedlichen Instanzen unterschiedliche Anzahlen von Verarbeitungskernen, unterschiedliche Mengen von lokalem Parallelprozessorspeicher und/oder andere Konfigurationsunterschiede aufweisen. Zum Beispiel können in mindestens einer Ausführungsform einige Instanzen der Parallelverarbeitungseinheit 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 beinhalten, in einer Vielfalt von Konfigurationen und Formfaktoren implementiert sein, einschließlich, aber nicht beschränkt auf, Desktop-, Laptop- oder tragbarer Personalcomputer, Server, Arbeitsstationen, Spielekonsolen und/oder eingebetteter Systeme.

[0199] **Fig. 20B** 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. 20A**. In mindestens einer Ausführungsform beinhaltet die Partitionseinheit 2020 einen L2-Cache 2021, eine Bildpufferschnittstelle 2025 und eine Rasteroperationseinheit (raster operations unit-„ROP“) 2026. Der L2-Cache 2021 ist ein Lese-/Schreib-Cache, der dazu konfiguriert ist, Lade- und Sicherungsoperationen durchzuführen, 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 Bildpufferschnittstelle 2025 zur Verarbeitung ausgegeben. In mindestens einer Ausführungsform können Aktualisierungen auch über die Bildpufferschnittstelle 2025 zur Verarbeitung an einen Bildpuffer gesendet werden. In mindestens einer Ausführungsform bildet die Bildpufferschnittstelle 2025 eine Schnittstelle mit einer der Speichereinheiten in dem Parallelprozessorspeicher, wie etwa mit den Speichereinheiten 2024A-2024N aus **Fig. 20** (z. B. in dem Parallelprozessorspeicher 2022).

[0200] In mindestens einer Ausführungsform ist die ROP 2026 eine Verarbeitungseinheit, die Rasteroperationen, wie etwa Stencil, Z-Test, Blending usw., durchführt. In mindestens einer Ausführungsform gibt die ROP 2026 dann verarbeitete Grafikdaten aus, die im 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 verlustfreie Komprimierungslogik sein, die einen oder mehrere von mehreren Komprimierungsalgorithmen verwendet. Die Komprimierungslogik, die von der ROP 2026 ausgeführt wird, kann auf Grundlage von statistischen Eigenschaften der zu komprimierenden Daten variieren. Zum Beispiel wird in mindestens einer Ausführungsform die Delta-Farbkomprimierung an Tiefen- und Farbdaten auf Kachelbasis ausgeführt.

[0201] In mindestens einer Ausführungsform ist die ROP 2026 innerhalb jedes Verarbeitungsclusters (z.B. Cluster 2014A-2014N aus **Fig. 20A)** anstatt 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 einer Anzeigevorrichtung angezeigt werden, wie etwa einer der einen oder der mehreren Anzeigevorrichtung(en) 1910 aus **Fig. 19** zur weiteren Verarbeitung durch den/die Prozessor(en) 1902 geroutet werden oder zur weiteren Verarbeitung durch eine der Verarbeitungsentitäten innerhalb des Parallelprozessors 2000 aus **Fig. 20A** geroutet werden.

[0202] 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. 20A**. In mindestens einer Ausführungsform können einer oder mehrere des/der Verarbeitungscluster(s) 2014 dazu konfiguriert sein, viele Threads parallel auszuführen, wobei sich „Thread“ auf eine Instanz eines konkreten Programms bezieht, das an einem konkreten Satz von Eingabedaten ausgeführt wird. In mindestens einer Ausführungsform werden Ausgabetechniken für Single-Instruction-Multiple-Data-(SIMD-)Anweisungen verwendet, um die parallele Ausführung einer großen Anzahl von Threads zu unterstützen, ohne mehrere unabhängige Anweisungseinheiten bereitzustellen. In mindestens einer Ausführungsform werden Single-Instruction-Multiple-Thread-(SIMT-)Techniken verwendet, um die parallele Ausführung einer großen Anzahl von im Allgemeinen synchronisierten Threads zu unterstützen, wobei eine gemeinsame Anweisungseinheit verwendet wird, die dazu konfiguriert ist, Anweisungen an einen Satz von Verarbeitungs-Engines innerhalb jedes der Verarbeitungscluster auszugeben.

[0203] In mindestens einer Ausführungsform kann der Betrieb des Verarbeitungsclusters 2014 über einen Pipeline-Manager 2032 gesteuert werden, der die Verarbeitungsaufgaben auf die SIMT-Parallelprozessoren verteilt. In mindestens einer Ausführungsform empfängt der Pipeline-Manager 2032 Anweisungen von dem Planer 2010 aus **Fig. 20A** 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 Arten 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 kann eine Datenkreuzschiene 2040 verwendet werden, um verarbeitete Daten an eines von mehreren möglichen Zielen, einschließlich anderer Shader-Einheiten, zu verteilen. In mindestens einer Ausführungsform kann der Pipeline-Manager 2032 die Verteilung der verarbeiteten Daten ermöglichen, indem er Ziele für die zu verteilenden verarbeiteten Daten gegenüber der Datenkreuzschiene 2040 angibt.

[0204] 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 funktionelle Ausführungslogik pipelineartig konfiguriert sein, wobei neue Anweisungen ausgegeben werden können, bevor vorherige Anweisungen abgeschlossen sind. In mindestens einer Ausführungsform unterstützt die funktionelle Ausführungslogik eine Vielfalt von Operationen, einschließlich Ganzzahl- und Gleitkommaarithmetik, Vergleichsoperationen, Boolescher Operationen, Bitverschiebung und Berechnung verschiedener algebraischer Funktionen. In mindestens einer Ausführungsform kann dieselbe Hardware einer funktionellen Einheit genutzt werden, um unterschiedliche Operationen auszuführen, und eine beliebige Kombination von funktionellen Einheiten kann vorhanden sein.

[0205] In mindestens einer Ausführungsform stellen die an das Verarbeitungscluster 2014 übertragenen Anweisungen einen Thread dar. In mindestens einer Ausführungsform ist ein Satz von Threads, der über einen Satz von Parallelverarbeitungs-Engines ausgeführt wird, eine Thread-Gruppe. In mindestens einer Ausführungsform führt die Thread-Gruppe ein Programm an unterschiedlichen Eingabedaten aus. In mindestens einer Ausführungsform kann jeder Thread innerhalb einer Thread-Gruppe einer anderen Verarbeitungs-Engine innerhalb eines Grafik-Multiprozessors 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. In mindestens einer Ausführungsform, wenn eine Thread-Gruppe weniger Threads beinhaltet als eine Anzahl von Verarbeitungs-Engines, können eine oder mehrere Verarbeitungs-Engines während der Zyklen, in denen diese Thread-Gruppe verarbeitet wird, inaktiv sein. In mindestens einer Ausführungsform kann eine Thread-Gruppe auch mehr Threads beinhalten als eine Anzahl von Verar-

beitungs-Engines innerhalb des Grafik-Multiprozessors 2034. In mindestens einer Ausführungsform, wenn eine Thread-Gruppe mehr Threads als Verarbeitungs-Engines innerhalb des Grafik-Multiprozessors 2034 beinhaltet, kann die Verarbeitung über aufeinanderfolgende Taktzyklen durchgeführt werden. In mindestens einer Ausführungsform können mehrere Thread-Gruppen gleichzeitig auf einem Grafik-Multiprozessor 2034 ausgeführt werden.

[0206] In mindestens einer Ausführungsform beinhaltet der Grafik-Multiprozessor 2034 einen internen Cache-Speicher zum Durchführen von Lade- und Speicheroperationen. In mindestens einer Ausführungsform kann der Grafik-Multiprozessor 2034 auf einen internen Cache verzichten und einen Cache-Speicher (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 aus **Fig. 20A)**, die von allen Verarbeitungsclustern 2014 gemeinsam genutzt werden und zum Übertragen von Daten zwischen Threads verwendet werden können. In mindestens einer Ausführungsform kann der Grafik-Multiprozessor 2034 auch auf den globalen chipexternen Speicher zugreifen, der einen oder mehrere von lokalem Parallelprozessorspeicher und/oder Systemspeicher beinhalten kann. In mindestens einer Ausführungsform kann ein beliebiger Speicher, der zu der Parallelverarbeitungseinheit 2002 extern ist, als globaler Speicher verwendet werden. In mindestens einer Ausführungsform beinhaltet das Verarbeitungscluster 2014 mehrere Instanzen des Grafik-Multiprozessors 2034, die gemeinsame Anweisungen und Daten gemeinsam nutzen können, die im L1-Cache 2048 gespeichert sein können.

[0207] In mindestens einer Ausführungsform kann jedes Verarbeitungscluster 2014 eine Speicherverwaltungseinheit („MMU“) 2045 beinhalten, die dazu konfiguriert ist, virtuelle Adressen auf physische Adressen abzubilden. In mindestens einer Ausführungsform können sich eine oder mehrere Instanzen der MMU 2045 innerhalb der Speicherschnittstelle 2018 aus **Fig. 20A** befinden. 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 Cache-Zeilenindex. In mindestens einer Ausführungsform kann die MMU 2045 Adressübersetzungspuffer (TLB) oder Caches beinhalten, die sich in dem Grafik-Multiprozessor 2034 oder in dem L1-Cache oder Verarbeitungscluster 2014 befinden können. In mindestens einer Ausführungsform wird die physische Adresse verarbeitet, um die Oberflächendaten-Zugriffslokalität zu verteilen, um eine effiziente Anforderungsverschachtelung zwischen den Partitionseinheiten zu ermöglichen. In mindestens einer Ausführungsform kann der Cache-Zeilenindex verwendet werden, um zu bestimmen, ob eine Anforderung für eine Cache-Zeile ein Treffer oder eine Auslassung ist.

[0208] 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 Texturabtastpositionen, Lesen von Texturdaten und Filtern von Texturdaten. In mindestens einer Ausführungsform werden die Texturdaten aus einem internen Textur-L1-Cache (nicht gezeigt) oder aus einem L1-Cache innerhalb des Grafik-Multiprozessors 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 Aufgaben an die Datenkreuzschiene 2040 aus, um (eine) verarbeitete Aufgabe(n) einem anderen Verarbeitungscluster 2014 zur weiteren Verarbeitung bereitzustellen oder um (eine) verarbeitete Aufgabe(n) über die Speicherkreuzschiene 2016 in einem L2-Cache, lokalen Parallelprozessorspeicher oder Systemspeicher zu speichern. In mindestens einer Ausführungsform ist eine preROP 2042 (pre-raster operations unit - Vor-Rasteroperationseinheit) dazu konfiguriert, Daten von dem Grafik-Multiprozessor 2034 zu empfangen und Daten an ROP-Einheiten zu leiten, die sich in Partitionseinheiten, wie in dieser Schrift beschrieben, befinden können (z. B. Partitionseinheiten 2020A-2020N aus **Fig. 20A)**. In mindestens einer Ausführungsform kann die PreROP-Einheit 2042 Optimierungen für die Farbmischung durchführen, Pixelfarbdaten organisieren und Adressübersetzungen durchführen.

[0209] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 in dem Grafikverarbeitungscluster 2014 für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0210] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0211] Fig. 20D zeigt einen Grafik-Multiprozessor 2034 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist der Grafik-Multiprozessor 2034 mit dem Pipelinemanager 2032 des Verarbeitungsclusters 2014 gekoppelt. In mindestens einer Ausführungsform weist der Grafik-Multiprozessor 2034 eine Ausführungspipeline auf, die einen Anweisungs-Cache 2052, eine Anweisungseinheit 2054, eine Adressabbildungseinheit 2056, eine Registerdatei 2058, einen oder mehrere Kerne 2062 einer Universal-Grafikverarbeitungseinheit (GPGPU) und eine oder mehrere Lade-/Speichereinheiten 2066 beinhaltet, ohne darauf beschränkt zu sein. Der/die GPGPU-Kern(e) 2062 und die Lade-/Speichereinheit(en) 2066 sind über eine Speicher- und Cache-Zusammenschaltung 2068 mit dem Cache-Speicher 2072 und dem gemeinsam genutzten Speicher 2070 gekoppelt.

[0212] In mindestens einer Ausführungsform empfängt der Anweisungs-Cache 2052 einen Strom aus auszuführenden Anweisungen von dem Pipeline-Manager 2032. In mindestens einer Ausführungsform werden die Anweisungen im Anweisungs-Cache 2052 zwischengespeichert und von der Anweisungseinheit 2054 zur Ausführung zugeteilt. In mindestens einer Ausführungsform kann die Anweisungseinheit 2054 Anweisungen als Thread-Gruppen (z. B. Warps) zuteilen, wobei jede Thread-Gruppe einer anderen Ausführungseinheit innerhalb des/der GPGPU-Kerns/- Kerne 2062 zugewiesen ist. In mindestens einer Ausführungsform kann eine Anweisung auf einen beliebigen von einem lokalen, gemeinsam genutzten oder globalen Adressraum zugreifen, indem sie eine Adresse innerhalb eines vereinheitlichten Adressraums vorgibt. In mindestens einer Ausführungsform kann die Adressabbildungseinheit 2056 verwendet werden, um Adressen in einem vereinheitlichten Adressraum in eine eindeutige Speicheradresse zu übersetzen, auf die durch die Lade-/Speichereinheit(en) 2066 zugegriffen werden kann.

[0213] In mindestens einer Ausführungsform stellt die Registerdatei 2058 einen Satz von Registern für funktionelle Einheiten des Grafik-Multiprozessors 2034 bereit. In mindestens einer Ausführungsform stellt die Registerdatei 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 Registerdatei 2058 derart zwischen den einzelnen funktionellen Einheiten aufgeteilt, dass jeder funktionellen Einheit ein dedizierter Abschnitt der Registerdatei 2058 zugewiesen ist. In mindestens einer Ausführungsform ist die Registerdatei 2058 auf unterschiedliche Warps aufgeteilt, die durch den Grafik-Multiprozessor 2034 ausgeführt werden.

[0214] In mindestens einer Ausführungsform können die GPGPU-Kerne 2062 jeweils Gleitkommaeinheiten (floating point units - FPUs) und/oder arithmetisch-logische Einheiten (ALUs) für Integer beinhalten, die zum Ausführen von Anweisungen des Grafik-Multiprozessors 2034 verwendet werden. Die GPGPU-Kerne 2062 können eine ähnliche Architektur aufweisen oder sich bezüglich der Architektur unterscheiden. In mindestens einer Ausführungsform beinhaltet ein erster Abschnitt der GPGPU-Kerne 2062 eine FPU mit einfacher Genauigkeit und eine Integer-ALU, während ein zweiter Abschnitt der GPGPU-Kerne eine FPU mit doppelter Genauigkeit beinhaltet. In mindestens einer Ausführungsform können FPUs den Standard IEEE 754-2008 für Gleitkommaarithmetik implementieren oder Gleitkommaarithmetik mit variabler Genauigkeit ermöglichen. In mindestens einer Ausführungsform kann der Grafik-Multiprozessor 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 auch Fest- oder Spezialfunktionslogik beinhalten.

[0215] In mindestens einer Ausführungsform beinhalten die GPGPU-Kerne 2062 SIMD-Logik, die 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 durch einen Shader-Compiler erzeugt werden oder automatisch erzeugt werden, wenn Programme ausgeführt werden, die für Single-Program-Multiple-Data-(SPMD-) oder SIMT-Architekturen geschrieben und kompiliert wurden. In mindestens einer Ausführungsform können mehrere Threads eines für ein SIMT-Ausführungsmodell konfigurierten Programms über eine einzelne SIMD-Anweisung ausgeführt werden. Zum Beispiel können in mindestens einer Ausführungsform acht SIMT-Threads, die gleiche oder ähnliche Operationen durchführen, parallel über eine einzelne SIMD8-Logikeinheit ausgeführt werden.

[0216] In mindestens einer Ausführungsform ist die Speicher- und Cache-Zusammenschaltung 2068 ein Zusammenschaltungsnetz, das jede funktionelle Einheit des Grafik-Multiprozessors 2034 mit der Registerdatei 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 Registerdatei 2058 zu implementieren. In mindestens einer Ausführungsform kann die Registerdatei 2058 mit der gleichen Frequenz wie die GPGPU-Kerne 2062 arbeiten, sodass die Datenübertragung zwischen den GPGPU-Kernen 2062 und der Registerdatei 2058 eine sehr geringe Latenz aufweist. In mindestens einer Ausführungsform kann der gemeinsam genutzte Speicher 2070 verwendet werden, um die Kommunikation zwischen Threads zu ermöglichen, die auf funktionellen Einheiten innerhalb des Grafik-Multiprozessors 2034 ausgeführt werden. In mindestens einer Ausführungsform kann der Cache-Speicher 2072 zum Beispiel als Daten-Cache verwendet werden, um Texturdaten zwischenzuspeichern, 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 zwischengespeicherten Daten, die innerhalb des Cache-Speichers 2072 gespeichert sind, programmatisch Daten innerhalb des gemeinsam genutzten Speichers speichern.

[0217] In mindestens einer Ausführungsform ist ein Parallelprozessor oder eine GPGPU, wie in dieser Schrift beschrieben, kommunikativ an Host-/Prozessorkerne gekoppelt, um Grafikoperationen, Operationen für maschinelles Lernen, Musteranalyseoperationen und verschiedene Funktionen einer Universal-GPU (GPGPU) zu beschleunigen. In mindestens einer Ausführungsform kann die GPU über einen Bus oder eine andere Zusammenschaltung (z. B. eine Hochgeschwindigkeitszusammenschaltung wie PCIe oder NVLink) mit dem Host-Prozessor/den Kernen kommunikativ gekoppelt sein. In mindestens einer Ausführungsform kann die GPU in demselben Gehäuse oder Chip wie die Kerne integriert sein und kommunikativ über einen internen Prozessorbus/eine interne Zusammenschaltung (d. h. intern zum Gehäuse oder Chip) mit den Kernen gekoppelt sein. In mindestens einer Ausführungsform können die Prozessorkerne unabhängig von der Weise, auf welche die GPU verbunden ist, der GPU Arbeit in Form von Sequenzen von Befehlen/Anweisungen zuweisen, die in einem Arbeitsdeskriptor enthalten sind. In mindestens einer Ausführungsform verwendet die GPU dann dedizierte Schaltungen/Logik zur effizienten Verarbeitung dieser Befehle/Anweisungen.

[0218] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 in dem Grafik-Multiprozessor 2034 für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0219] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0220] **Fig. 21** veranschaulicht ein Multi-GPU-Rechensystem 11100 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann das Multi-GPU-Rechensystem 11100 einen Prozessor 11102 beinhalten, der über einen Host-Schnittstellen-Switch 11104 an mehrere Universal-Grafikverarbeitungseinheiten (GPGPUs) 11106A-D gekoppelt ist. In mindestens einer Ausführungsform ist der Host-Schnittstellen-Switch 11104 eine PCI-Express-Switch-Vorrichtung, die den Prozessor 11102 an einen PCI-Express-Bus koppelt, über den der Prozessor 11102 mit den GPGPUs 11106A-D kommunizieren kann. Die GPGPUs 11106A-D können über einen Satz von Hochgeschwindigkeits-Punkt-zu-Punkt-GPU-zu-GPU-Verbindungen 11116 zusammengeschaltet sein. In mindestens einer Ausführungsform sind die GPU-zu-GPU-Verbindungen 11116 mit jeder der GPGPUs 11106A-D über eine dedizierte GPU-Verbindung verbunden. In mindestens einer Ausführungsform ermöglichen die P2P-GPU-Verbindungen 11116 direkte Kommunikation zwischen jeder der GPGPUs 11106A-D, ohne dass Kommunikation über den Host-Schnittstellenbus 11104 erforderlich ist, mit dem der Prozessor 11102 verbunden ist. In mindestens einer Ausführungsform, bei der GPU-zu-GPU-Verkehr zu den P2P-GPU-Verbindungen 11116 geleitet wird, bleibt der Host-Schnittstellenbus 11104 für den Systemspeicherzugriff oder zum Kommunizieren mit anderen Instanzen des Multi-GPU-Rechensystems 11100 verfügbar, zum Beispiel über eine oder mehrere Netzwerkvorrichtungen. Während in mindestens einer Ausführungsform die GPGPUs 11106A-D mit dem Prozessor 11102 über den Host-Schnitt-

stellen-Switch 11104 verbunden sind, beinhaltet der Prozessor 11102 in mindestens einer Ausführungsform eine direkte Unterstützung für P2P-GPU-Verbindungen 11116 und kann direkt mit den GPGPUs 11106A-D verbunden sein.

[0221] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 in dem Multi-GPU-Rechensystem 11100 für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0222] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0223] **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 Grafikkerne 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.

[0224] 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 die Grafikkern(e) 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 einem mit einer Medien-Engine 2237 gekoppelten Video-Frontend 2234 Befehle zu. In mindestens einer Ausführungsform beinhaltet die Medien-Engine 2237 eine Videoqualitäts-Engine (Video Quality Engine - VQE) 2230 für die Video- und Bildnachbearbeitung und eine Multiformat-Codier/Decodier-(multi-format encode/decode - MFX-)Engine 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-Ausfiihrungsressourcen, die von mindestens einem Grafikkern 2280A bereitgestellt werden.

[0225] In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2200 skalierbare Thread-Ausfiihrungsressourcen mit modularen Kernen 2280A-2280N (mitunter als Kernscheiben bezeichnet), die jeweils mehrere Teilkerne 2250A-2250N, 2260A-2260N (mitunter als Kernteilscheiben bezeichnet) aufweisen. In mindestens einer Ausführungsform kann der Grafikprozessor 2200 eine beliebige Anzahl von Grafikkernen 2280A bis 2280N aufweisen. In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2200 einen Grafikkern 2280A, der mindestens einen ersten Teilkern 2250A und einen zweiten Teilkern 2260A aufweist. In mindestens einer Ausführungsform ist der Grafikprozessor 2200 ein Prozessor mit niedriger Leistung mit einem einzelnen Teilkern (z. B. 2250A). In mindestens einer Ausführungsform beinhaltet der Grafikprozessor 2200 mehrere Grafikkerne 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-/Texturabtastern 2254A-2254N. In mindestens einer Ausführungsform beinhaltet jeder Teilkern in den zweiten Teilkernen 2260A-2260N mindestens einen zweiten Satz von Ausführungseinheiten 2262A-2262N und Abtastern 2264A-2264N. In mindestens einer Ausführungsform nutzen die Teilkerne 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 Cache-Speicher und Pixeloperationslogik.

[0226] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Infe-

renz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 915 in dem Grafikprozessor 2200 für Inferenz- oder Vorhersageoperationen zumindest teilweise auf Grundlage von Gewichtungsparametern verwendet werden, die unter Verwendung von Trainingsoperationen für neuronale Netze, Funktionen und/oder Architekturen von neuronalen Netzen oder in dieser Schrift beschriebenen Anwendungsfällen für neuronale Netze berechnet wurden.

[0227] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0228] **Fig. 23** ist ein Blockdiagramm, das die Mikroarchitektur für einen Prozessor 2300, der Logikschaltungen zum Durchführen von Anweisungen beinhalten kann, gemäß mindestens einer Ausführungsform veranschaulicht. 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 Gleitkommaform verfügbar sind, mit Elementen mit gepackten Daten arbeiten, die mit Single-Instruction-Multiple-Data-(„SIMD“-) und Streaming-SIMD-Erweiterungs-(„SSE“-)Anweisungen einhergehen. In mindestens einer Ausführungsform können 128 Bit breite XMM-Register, die sich auf die SSE2-, SSE3-, SSE4-, AVX- oder darüber hinausgehende Technologie beziehen (allgemein als „SSEx“ bezeichnet), derartige Operanden mit gepackten Daten aufbewahren. In mindestens einer Ausführungsform kann der Prozessor 2300 Anweisungen zum Beschleunigen von Algorithmen für maschinelles Lernen oder Deep Learning, Training oder Inferenzieren durchführen.

[0229] 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 der Prozessorpipeline verwendet werden sollen. In mindestens einer Ausführungsform kann das Frontend 2301 mehrere Einheiten beinhalten. In mindestens einer Ausführungsform ruft ein Anweisungsvorabrufer 2326 Anweisungen aus dem Speicher ab und führt die Anweisungen einem Anweisungsdecodierer 2328 zu, der die Anweisungen wiederum decodiert oder interpretiert. Zum Beispiel decodiert in mindestens einer Ausführungsform der Anweisungsdecodierer 2328 eine empfangene Anweisung in einen oder mehrere Operationen, die als „Mikroanweisungen“ oder „Mikrooperationen“ bezeichnet werden (auch als „Mikro-Ops“ oder „µOps“ bezeichnet) und von der Maschine ausgeführt werden können. In mindestens einer Ausführungsform parst der Anweisungsdecodierer 2328 die Anweisung in einen Opcode und entsprechende Daten- und Steuerfelder, die von der Mikroarchitektur verwendet werden können, um Operationen gemäß mindestens einer Ausführungsform durchzuführen. In mindestens einer Ausführungsform kann ein Abfolge-Cache 2330 decodierte µOps in programmgeordnete Sequenzen oder Abfolgen in einer µOp-Warteschlange 2334 zur Ausführung zusammenstellen. Wenn der Abfolge-Cache 2330 auf eine komplexe Anweisung stößt, stellt in mindestens einer Ausführungsform ein Mikrocode-ROM 2332 die für den Abschluss einer Operation benötigten µOps bereit.

[0230] In mindestens einer Ausführungsform können einige Anweisungen in eine einzelne Mikro-Op umgewandelt werden, während andere mehrere Mikro-Ops benötigen, um eine vollständige Operation abzuschließen. In mindestens einer Ausführungsform kann der Anweisungsdecodierer 2328 auf den Mikrocode-ROM 2332 zugreifen, um die Anweisung durchzuführen, wenn mehr als vier Mikro-Ops benötigt werden, um eine Anweisung abzuschließen. In mindestens einer Ausführungsform kann eine Anweisung in eine kleine Anzahl von Mikro-Ops zur Verarbeitung in dem Anweisungsdecodierer 2328 decodiert werden. In mindestens einer Ausführungsform kann eine Anweisung im Mikrocode-ROM 2332 gespeichert werden, wenn eine Reihe von Mikro-Ops zur Ausführung der Operation erforderlich ist. In mindestens einer Ausführungsform bezieht sich der Abfolge-Cache 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 fertiggestellt hat, das Abrufen von Mikro-Ops aus dem Abfolge-Cache 2330 wieder aufnehmen.

[0231] In mindestens einer Ausführungsform kann eine Engine zur Ausführung in einer anderen Reihenfolge („Out-of-Order-Engine“) 2303 Anweisungen zur Ausführung vorbereiten. In mindestens einer Ausführungsform verfügt die Out-of-Order-Ausführungslogik über eine Reihe von Puffern, um den Fluss der Anweisungen zu glätten und neu zu ordnen, um die Rechenleistung zu optimieren, während sie eine Pipeline durchlaufen und zur Ausführung geplant werden. In mindestens einer Ausführungsform beinhaltet die Out-of-Order-Ausführungs-Engine 2303 ohne Einschränkung einen Zuweiser/Registerumbenenner 2340, eine Speicher-µOp-Warteschlange 2342, eine Integer-/Gleitkomma-|iOp-Wartesch lange 2344, einen Speicherplaner 2346, einen schnellen Planer 2302, einen langsamen/allgemeinen Gleitkommaplaner („langsamer/allgemeiner FP-Planer“) 2304 und einen einfachen Gleitkommaplaner („einfacher FP-Planer“) 2306. In mindestens einer Ausführungsform werden der schnelle Planer 2302, der langsame/allgemeine Gleitkommaplaner 2304 und der einfache Gleitkommaplaner 2306 in dieser Schrift auch zusammen 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 Registerdatei um. In mindestens einer Ausführungsform weist der Zuweiser/Registerumbenenner 2340 auch einen Eintrag für jede µOp in einer von zwei µOp-Warteschlangen zu, und zwar in der Speicher-uOp-Warteschlange 2342 für Speicheroperationen und der Integer-/Gleitkomma-jlOp-Warteschlange 2344 für Nicht-Speicheroperationen, vor dem Speicherplaner 2346 und den µOp-Planern 2302, 2304, 2306. 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 mindestens einer Ausführungsform auf jeder Hälfte des Haupttaktzyklus planen, während der langsame/allgemeine Gleitkomma-Planer 2304 und der einfache Gleitkomma-Planer 2306 einmal pro Hauptprozessortaktzyklus planen können. In mindestens einer Ausführungsform vermitteln die µOp-Planer 2302, 2304, 2306 Zuteilungsports, um µOps zur Ausführung zu planen.

[0232] In mindestens einer Ausführungsform beinhaltet der Ausführungsblock 2311 ohne Einschränkung eine Integerregisterdatei/ein Umgehungsnetz 2308, eine Gleitkommaregisterdatei/ein Umgehungsnetz („FP-Registerdatei/Umgehungsnetz“) 2310, Adresserzeugungseinheiten (address generation units - „AGUs“) 2312 und 2314, schnelle arithmetisch-logische Einheiten (ALUs) („schnelle ALUs“) 2316 und 2318, eine langsame arithmetisch-logische Einheit („langsame ALU“) 2320, eine Gleitkomma-ALU („FP“) 2322 und eine Gleitkomma-Bewegungseinheit („FP-Bewegung“) 2324. In mindestens einer Ausführungsform werden die Integerregisterdatei/das Umgehungsnetz 2308 und die Gleitkommaregisterdatei/das Umgehungsnetz 2310 in dieser Schrift auch als „Registerdateien 2308, 2310“ bezeichnet. In mindestens einer Ausführungsform werden die AGUs 2312 und 2314, die schnellen ALUs 2316 und 2318, die langsame ALU 2320, die Gleitkomma-ALU 2322 und die Gleitkomma-Bewegungseinheit 2324 in dieser Schrift auch als „Ausführungseinheiten 2312, 2314, 2316, 2318, 2320, 2322 und 2324“ bezeichnet. In mindestens einer Ausführungsform kann der Ausführungsblock b11 ohne Einschränkung eine beliebige Anzahl (einschließlich null) und Art von Registerdateien, Umgehungsnetzen, Adresserzeugungseinheiten und Ausführungseinheiten in beliebiger Kombination beinhalten.

[0233] In mindestens einer Ausführungsform können die Registerdateien 2308, 2310 zwischen den µOp-Planern 2302, 2304, 2306 und den Ausführungseinheiten 2312, 2314, 2316, 2318, 2320, 2322 und 2324 angeordnet sein. In mindestens einer Ausführungsform führt die Integerregisterdatei/das Umgehungsnetz 2308 Integeroperationen durch. In mindestens einer Ausführungsform führt die Gleitkommaregisterdatei/das Umgehungsnetz 2310 Gleitkommaoperationen durch. In mindestens einer Ausführungsform kann jede der Registerdateien 2308, 2310 ohne Einschränkung ein Umgehungsnetz beinhalten, das gerade abgeschlossene Ergebnisse, die noch nicht in die Registerdatei geschrieben wurden, zu neuen abhängigen µOps umgehen oder weiterleiten kann. In mindestens einer Ausführungsform können die Registerdateien 2308, 2310 miteinander Daten kommunizieren. In mindestens einer Ausführungsform kann die Integerregisterdatei/das Umgehungsnetz 2308 ohne Einschränkung zwei separate Registerdateien beinhalten, eine Registerdatei für Daten niederer Ordnung mit zweiunddreißig Bit und eine zweite Registerdatei für Daten hoher Ordnung mit zweiunddreißig Bit. In mindestens einer Ausführungsform kann die Gleitkommaregisterdatei/das Umgehungsnetz 2310 ohne Einschränkung 128 Bit breite Einträge beinhalten, da Gleitkommaanweisungen typischerweise Operanden mit einer Breite von 64 bis 128 Bit aufweisen.

[0234] In mindestens einer Ausführungsform können die Ausführungseinheiten 2312, 2314, 2316, 2318, 2320, 2322, 2324 Anweisungen ausführen. In mindestens einer Ausführungsform speichern die Registerdateien 2308, 2310 Integer- und Gleitkommadaten-Operandenwerte, die Mikroanweisungen für die Ausführung benötigen. In mindestens einer Ausführungsform kann der Prozessor 2300 ohne Einschränkung eine belie-

bige Anzahl und Kombination der Ausführungseinheiten 2312, 2314, 2316, 2318, 2320, 2322, 2324 beinhalten. In mindestens einer Ausführungsform können die Gleitkomma-ALU 2322 und die Gleitkomma-Bewegungseinheit 2324 Gleitkomma-, MMX-, SIMD-, AVX- und SSE- oder andere Operationen ausführen, einschließlich spezialisierter Anweisungen für maschinelles Lernen. In mindestens einer Ausführungsform kann die Gleitkomma-ALU 2322 ohne Einschränkung einen 64 Bit mal 64 Bit großen Gleitkommadividierer zum Ausführen von Divisions-, Quadratwurzel- und Rest-Mikro-Ops beinhalten. In mindestens einer Ausführungsform können Anweisungen, an denen ein Gleitkommawert beteiligt ist, mit Gleitkomma-Hardware gehandhabt werden. In mindestens einer Ausführungsform können ALU-Operationen an die schnellen ALUs 2316, 2318 übergeben werden. In mindestens einer Ausführungsform können die schnellen ALUs 2316, 2318 schnelle Operationen mit einer effektiven Latenz von einem halben Taktzyklus ausführen. In mindestens einer Ausführungsform gehen die meisten komplexen Integeroperationen an die langsame ALU 2320, da die langsame ALU 2320 ohne Einschränkung Integerausführungs-Hardware für Operationen vom Typ mit langer Latenz beinhalten kann, wie etwa einen Multiplikator, Verschiebungen, Flag-Logik und Verzweigungsverarbeitung. In mindestens einer Ausführungsform können die Lade-/Speicheroperationen eines Speichers durch die AGUs 2312, 2314 ausgeführt werden. In mindestens einer Ausführungsform können die schnelle ALU 2316, die schnelle ALU 2318 und die langsame ALU 2320 Integeroperationen an 64-Bit-Datenoperanden durchführen. In mindestens einer Ausführungsform können die schnelle ALU 2316, die schnelle ALU 2318 und die langsame ALU 2320 dazu implementiert sein, eine Vielfalt von Datenbitgrößen zu unterstützen, einschließlich sechzehn, zweiunddreißig, 128, 256 usw. In mindestens einer Ausführungsform können die Gleitkomma-ALU 2322 und die Gleitkomma-Bewegungseinheit 2324 dazu implementiert sein, einen Bereich von Operanden zu unterstützen, die Bits mit verschiedenen Breiten aufweisen. In mindestens einer Ausführungsform können die Gleitkomma-ALU 2322 und die Gleitkomma-Bewegungseinheit 2324 an 128 Bit breiten Paket-Datenoperanden in Verbindung mit SIMD- und Multimedia-Anweisungen arbeiten.

[0235] In mindestens einer Ausführungsform teilen die µOp-Planer 2302, 2304, 2306 abhängige Operationen zu, bevor die Ausführung einer übergeordneten Last beendet ist. In mindestens einer Ausführungsform kann der Prozessor 2300, da µOps in dem Prozessor 2300 spekulativ geplant und ausgeführt werden können, auch Logik zum Handhaben von Speicherfehlern beinhalten. In mindestens einer Ausführungsform können, falls eine Datenlast in einem Daten-Cache einen Fehler verursacht, abhängige Operationen in einer Pipeline im Gange sein, die einen Planer mit temporär inkorrekten Daten verlassen haben. In mindestens einer Ausführungsform verfolgt ein Wiederholungsmechanismus Anweisungen, die falsche Daten verwenden, und führt sie erneut aus. In mindestens einer Ausführungsform könnte es sein, dass abhängige Operationen wiederholt werden müssen, und es kann unabhängigen Operationen ermöglicht werden, abgeschlossen zu werden. In mindestens einer Ausführungsform können die Planer und ein Wiederholungsmechanismus mindestens einer Ausführungsform eines Prozessors auch dafür ausgestaltet sein, Anweisungssequenzen für Zeichenfolgenvergleichsoperationen abzufangen.

[0236] In mindestens einer Ausführungsform kann sich der Ausdruck „Register“ auf prozessorinterne Speicherorte beziehen, die als Teil von Anweisungen zur Identifizierung von Operanden verwendet werden können. In mindestens einer Ausführungsform kann es sich bei den Registern um diejenigen handeln, die von außerhalb des Prozessors (aus der Sicht eines Programmierers) nutzbar sind. In mindestens einer Ausführungsform sind die Register möglicherweise nicht auf einen konkreten Schaltungstyp beschränkt. Vielmehr kann ein Register in mindestens einer Ausführungsform Daten speichern, Daten bereitstellen und die in dieser Schrift beschriebenen Funktionen durchführen. In mindestens einer Ausführungsform können die in dieser Schrift beschriebenen Register durch Schaltungen innerhalb eines Prozessors unter Verwendung einer beliebigen Anzahl unterschiedlicher Techniken implementiert werden, wie etwa dedizierter physischer Register, dynamisch zugewiesener physischer Register unter Verwendung von Registerumbenennung, Kombinationen aus dedizierten und dynamisch zugewiesenen physischen Registern usw. In mindestens einer Ausführungsform speichern Integerregister 32-Bit-Integerdaten. Eine Registerdatei aus mindestens einer Ausführungsform beinhaltet zudem acht Multimedia-SIMD-Register für gepackte Daten.

[0237] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform können Abschnitte der oder die gesamte Inferenz- und/oder Trainingslogik 915 in den Ausführungsblock 2311 und andere gezeigte oder nicht gezeigte Speicher oder Register integriert sein. Zum Beispiel können in mindestens einer Ausführungsform die in dieser Schrift beschriebenen Trainings- und/oder Inferenztechniken eine oder mehrere der in dem Ausführungsblock 2311 veranschaulichten ALUs verwenden. Darüber hinaus können Gewichtungsparameter 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 in dieser Schrift beschriebene Algorithmen des maschinellen Lernens, neuronale Netzarchitekturen, Anwendungsfälle oder Trainingstechniken durchzuführen.

[0238] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0239] **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 den Deep-Learning-Anwendungsprozessor 2400 dazu veranlassen, einige oder alle der in dieser gesamten Offenbarung beschriebenen Prozesse und Techniken durchzuführen. 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 Verarbeitungscluster 2410(1)-2410(12), chipübergreifende Verbindungen (Inter-Chip Links - „ICLs“) 2420(1)-2420(12), chipübergreifende Steuerungen (Inter-Chip Controllers - „ICCs“) 2430(1)-2430(2), Speichersteuerungen (memory controllers - „Mem Ctrlrs“) 2442(1)-2442(4), eine physische Schicht 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“), eine Interconnect-Express-Steuerung für Peripheriekomponenten und einen Direktspeicherzugriffsblock („PCIe-Steuerung und DMA“) 2470 und einen sechzehnspurigen Interconnect-Express-Port für Peripheriekomponenten („PCI-Express x 16“) 2480.

[0240] 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 in dieser Schrift beschriebenen, berechnet wurden. In mindestens einer Ausführungsform kann jedes 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 eine beliebige Art von Verarbeitungsclustem 2400 beinhalten. In mindestens einer Ausführungsform sind die chipübergreifenden Verbindungen 2420 bidirektional. In mindestens einer Ausführungsform ermöglichen die chipübergreifenden Verbindungen 2420 und die chipübergreifenden Steuerungen 2430 mehreren Deep-Learning-Anwendungsprozessoren 2400 den Austausch von Informationen, einschließlich Aktivierungsinformationen, die aus dem Durchführen eines oder mehrerer Algorithmen des maschineller Lernens resultieren, 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 eine beliebige Art von ICLs 2420 und ICCs 2430 beinhalten.

[0241] In mindestens einer Ausführungsform stellen die HBM2s 2440 insgesamt 32 Gigabyte (GB) Speicher bereit. Die HBM2 2440(i) ist sowohl der Speichersteuerung 2442(i) als auch der HBM PHY 2444(i) zugeordnet. In mindestens einer Ausführungsform kann eine beliebige Anzahl von HBM2s 2440 eine beliebige Art und eine beliebige Gesamtmenge an Speicher mit hoher Bandbreite bereitstellen und einer beliebigen Anzahl (einschließlich null) und einer beliebigen Art von Speichersteuerungen 2442 und HBM PHYs 2444 zugeordnet sein. In mindestens einer Ausführungsform können die SPI, die I2C, der GPIO 2460, die PCIe-Steuerung und der 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 durchführbare Weise ermöglichen.

[0242] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform wird der Deep-Learning-Anwendungsprozessor 2400 verwendet, um ein Modell für maschinelles Lernen, wie etwa ein neuronales Netz, zu trainieren, die dem Deep-Learning-Anwendungsprozessor 2400 bereitgestellten Informationen vorherzusagen oder abzuleiten. In mindestens einer Ausführungsform wird der Deep-Learning-Anwendungsprozessor 2400 verwendet, um Informationen auf Grundlage eines trainierten Modells für maschinelles Lernen (z. B. eines neuronalen Netzes) abzuleiten

oder vorherzusagen, das durch einen anderen Prozessor oder ein anderes System oder durch den Deep-Learning-Anwendungsprozessor 2400 trainiert wurde. In mindestens einer Ausführungsform kann der Prozessor 2400 verwendet werden, um einen oder mehrere der in dieser Schrift beschriebenen Anwendungsfälle für neuronale Netze durchzuführen.

[0243] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0244] 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, die extern zu dem neuromorphen Prozessor 2500 sind, empfangen. In mindestens einer Ausführungsform können diese Eingaben an ein oder mehrere Neuronen 2502 innerhalb des neuromorphen Prozessors 2500 übertragen 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 übertragen werden können. Zum Beispiel können in mindestens einer Ausführungsform die Neuroneneingänge 2504 und die Neuronenausgänge 2506 über Synapsen 2508 zusammengeschaltet sein.

[0245] In mindestens einer Ausführungsform können die Neuronen 2502 und die Synapsen 2508 derart zusammengeschaltet sein, dass der neuromorphe Prozessor 2500 arbeitet, um die durch den neuromorphen Prozessor 2500 empfangenen Informationen zu verarbeiten oder zu analysieren. In mindestens einer Ausführungsform können die Neuronen 2502 einen Ausgabeimpuls (oder „Fire“ oder „Spike“) übertragen, wenn die über den Neuroneneingang 2504 empfangenen Eingaben einen Schwellenwert überschreiten. In mindestens einer Ausführungsform können die Neuronen 2502 die an den Neuroneneingängen 2504 empfangenen Signale summieren oder integrieren. Zum Beispiel können in mindestens einer Ausführungsform die Neuronen 2502 als leckende Integrate-and-Fire-Neuronen implementiert sein, wobei, falls eine Summe (als „Membranpotential“ bezeichnet) einen Schwellenwert überschreitet, das Neuron 2502 eine Ausgabe (oder „Feuern“) unter Verwendung einer Übertragungsfunktion, wie etwa einer Sigmoid- oder Schwellenwertfunktion, erzeugen kann. In mindestens einer Ausführungsform kann ein leckendes Integrate-and-Fire-Neuron Signale, die an den Neuroneneingängen 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 Komparatorschaltungen oder -logik beinhalten, die einen Ausgabe-Spike an dem 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, sobald das Membranpotential auf 0 zurückgesetzt wurde, nach einer geeigneten Zeitspanne (oder Refraktärzeit) den normalen Betrieb wiederaufnehmen.

[0246] In mindestens einer Ausführungsform können die Neuronen 2502 durch die Synapsen 2508 zusammengeschaltet sein. In mindestens einer Ausführungsform können die Synapsen 2508 arbeiten, um Signale von einer Ausgabe eines ersten Neurons 2502 an eine Eingabe eines zweiten Neurons 2502 zu übertragen. In mindestens einer Ausführungsform können die Neuronen 2502 Informationen über mehr als eine Instanz der Synapse 2508 übertragen. In mindestens einer Ausführungsform können eine oder mehrere Instanzen des Neuronenausgangs 2506 über eine Instanz der Synapse 2508 mit einer Instanz des Neuroneneingangs 2504 in dem gleichen Neuron 2502 verbunden sein. In mindestens einer Ausführungsform kann eine Instanz

des Neurons 2502, die eine über eine Instanz der Synapse 2508 zu übertragende Ausgabe erzeugt, als „präsynaptisches Neuron“ in Bezug auf diese Instanz der Synapse 2508 bezeichnet werden. In mindestens einer Ausführungsform kann eine Instanz des Neurons 2502, die eine über eine Instanz der Synapse 2508 übertragene Eingabe empfängt, als „postsynaptisches Neuron“ in Bezug auf diese Instanz der Synapse 2508 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.

[0247] In mindestens einer Ausführungsform können die Neuronen 2502 in eine oder mehrere Schichten organisiert sein. Jede Instanz des Neurons 2502 kann einen Neuronenausgang 2506 aufweisen, der sich über eine oder mehrere Synapsen 2508 zu einem oder mehreren Neuroneneingängen 2504 auffächern kann. In mindestens einer Ausführungsform können die Neuronenausgänge 2506 der Neuronen 2502 in einer ersten Schicht 2510 mit den Neuroneneingängen 2504 der 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 sich 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 sich jede Instanz des Neurons 2502 in einer Instanz der zweiten Schicht 2512 zu weniger als allen 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 Neuronen 2502 in der zweiten Schicht 2512 zu Neuronen 2502 in mehreren anderen Schichten auffächern, einschließlich zu Neuronen 2502 in (derselben) zweiten Schicht 2512. 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 vorwärtsgekoppelten Schichten beinhalten, einschließlich ohne Einschränkung sowohl spärlich verbundener vorwärtsgekoppelter Schichten als auch vollständig verbundener vorwärtsgekoppelter Schichten.

[0248] 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 Schaltungen oder Logik beinhalten, die es ermöglicht, Synapsen je nach Bedarf auf Grundlage der Topologie des neuronalen Netzes und des Neuronen-Fan-In/-Out unterschiedlichen Neuronen 2502 zuzuweisen. Zum Beispiel können in mindestens einer Ausführungsform die Synapsen 2508 mit den Neuronen 2502 unter Verwendung einer Zusammenschaltungsstruktur, wie etwa eines Netzes auf einem Chip, oder mit dedizierten Verbindungen verbunden sein. In mindestens einer Ausführungsform können die Synapsenzusammenschaltungen und Komponenten davon unter Verwendung von Schaltungen oder Logik implementiert sein.

[0249] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0250] Fig. 26 ist ein Blockdiagramm eines Verarbeitungssystems gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform beinhaltet das System 2600 einen oder mehrere Prozessoren 2602 und einen oder mehrere Grafikprozessoren 2608 und kann ein Einzelprozessor-Desktopsystem, ein Multiprozessor-Arbeitsstationssystem oder ein Serversystem sein, das eine große Anzahl von Prozessoren 2602 oder Prozessorkernen 2607 aufweist. 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.

[0251] 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, mit dieser gekoppelt oder in diese integriert sein,

wie z. B. eine tragbare Smartwatch-Vorrichtung, eine Smart-Eyewear-Vorrichtung, eine Augmented-Reality-Vorrichtung oder eine Virtual-Reality-Vorrichtung. In mindestens einer Ausführungsform ist das Verarbeitungssystem 2600 eine Femseh- oder Set-Top-Box-Vorrichtung, die einen oder mehrere Prozessoren 2602 und eine grafische Schnittstelle aufweist, die durch einen oder mehrere Grafikprozessoren 2608 erzeugt wird.

[0252] In mindestens einer Ausführungsform beinhalten ein oder mehrere Prozessoren 2602 jeweils einen oder mehrere Prozessorkerne 2607 zum Verarbeiten von Anweisungen, die bei ihrer Ausführung Operationen für System- und Benutzer-Software durchführen. In mindestens einer Ausführungsform ist jeder von einem oder mehreren Prozessorkernen 2607 dazu konfiguriert, einen spezifischen Anweisungssatz 2609 zu verarbeiten. In mindestens einer Ausführungsform kann der Anweisungssatz 2609 Complex Instruction Set Computing („CISC“), Reduced Instruction Set Computing („RISC“) oder Rechnen über Very Long Instruction Word („VLIW“) ermöglichen. In mindestens einer Ausführungsform können die Prozessorkerne 2607 jeweils einen anderen Anweisungssatz 2609 verarbeiten, der Anweisungen enthalten kann, um die Emulation anderer Anweisungssätze zu erleichtern. In mindestens einer Ausführungsform kann der Prozessorkern 2607 auch andere Verarbeitungsvorrichtungen beinhalten, wie etwa einen digitalen Signalprozessor (DSP).

[0253] 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 Cache-Speicher 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 Prozessorkemen 2607 unter Verwendung bekannter Cache-Kohärenztechniken gemeinsam genutzt werden kann. In mindestens einer Ausführungsform ist in dem Prozessor 2602 zusätzlich eine Registerdatei 2606 enthalten, die unterschiedliche Arten von Registern zum Speichern unterschiedlicher Datenarten beinhalten kann (z. B. Ganzzahlregister, Gleitkommaregister, Statusregister und ein Anweisungszeigerregister). In mindestens einer Ausführungsform kann die Registerdatei 2606 Universalregister oder andere Register beinhalten.

[0254] In mindestens einer Ausführungsform sind ein oder mehrere Prozessor(en) 2602 mit einem oder mehreren Schnittstellenbus(sen) 2610 gekoppelt, um Kommunikationssignale, wie etwa Adress-, Daten- oder Steuersignale, zwischen dem Prozessor 2602 und anderen Komponenten in dem System 2600 zu übertragen. In mindestens einer Ausführungsform kann der Schnittstellenbus 2610 in einer Ausführungsform ein Prozessorbus sein, wie etwa eine Version eines Direct-Media-Interface-(„DMI“-)Busses. In mindestens einer Ausführungsform ist die Schnittstelle 2610 nicht auf einen DMI-Bus beschränkt, sondern kann einen oder mehrere Peripheral-Component-Interconnect-Busse (z. B. PCI, PCI Express), Speicherbusse oder andere Arten von Schnittstellenbussen beinhalten. In mindestens einer Ausführungsform beinhalten der/die Prozessor(en) 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 Plattformsteuerungs-Hub (platform controller hub - PCH) 2630 Verbindungen mit E/A-Vorrichtungen über einen lokalen E/A-Bus bereitstellt.

[0255] 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 zudem 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 dem/den 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.

[0256] In mindestens einer Ausführungsform ermöglicht der Plattformsteuerungs-Hub 2630, dass Peripheriegeräte mit der Speichervorrichtung 2620 und dem Prozessor 2602 über einen Hochgeschwindigkeits-E/A-Bus verbunden werden. In mindestens einer Ausführungsform beinhalten die E/A-Peripheriegeräte eine Audiosteuerung 2646, eine Netzwerksteuerung 2634, eine Firmware-Schnittstelle 2628, einen drahtlosen Transceiver 2626, Berührungssensoren 2625 und eine Datenspeichervorrichtung 2624 (z. B. Festplattenlaufwerk, 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 Transceiver 2626 ein Wi-Fi-Transceiver, ein Bluetooth-Transceiver oder ein Mobilfunknetz-Transceiver, wie etwa ein 3G-, 4G- oder Long-Term-Evolution-(„LTE“-)Transceiver, sein. In mindestens einer Ausführungsform ermöglicht die Firmware-Schnittstelle 2628 die Kommunikation mit der System-Firmware und kann zum Beispiel eine Unified Extensible Firmware Interface (UEFI) sein. In mindestens einer Ausführungsform kann die Netzwerksteuerung 2634 eine Netzwerkverbindung zu einem drahtgebundenen Netzwerk ermöglichen. In mindestens einer Ausführungsform ist eine Hochleistungsnetzwerksteuerung (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 Legacy-E/A-Steuerung 2640 zur Kopplung von Legacy-Vorrichtungen (z. B. Personal System 2 (PS/2)) mit dem System. In mindestens einer Ausführungsform kann der Plattformsteuerungs-Hub 2630 auch mit einer oder mehreren Universal-Serial-Bus-(„USB“-)Steuerungen 2642 verbunden sein, die mit Eingabevorrichtungen, wie etwa Kombinationen aus Tastatur und Maus 2643, einer Kamera 2644 oder anderen USB-Eingabevorrichtungen, verbunden sind.

[0257] 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 der Plattformsteuerungs-Hub 2630 und/oder die Speichersteuerung 2616 extern zu einem oder mehreren Prozessor(en) 2602 sein. Zum Beispiel kann das System 2600 in mindestens einer Ausführungsform eine externe Speichersteuerung 2616 und einen Plattformsteuerungs-Hub 2630 beinhalten, der als Speichersteuerungs-Hub und Peripheriesteuerungs-Hub innerhalb eines Systemchipsatzes konfiguriert sein kann, der mit den Prozessor(en) 2602 in Kommunikation steht.

[0258] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform können Abschnitte oder die gesamte Inferenz- und/oder Trainingslogik 915 in den Grafikprozessor 2600 integriert sein. Zum Beispiel können in mindestens einer Ausführungsform die in dieser Schrift beschriebenen Trainings- und/oder Inferenztechniken eine oder mehrere der in dem Grafikprozessor 2612 verkörperten 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. 9A** oder **Fig. 9B** veranschaulichten Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtungsparameter 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 in dieser Schrift beschriebene Algorithmen für maschinelles Lernen, neuronale Netzarchitekturen, Anwendungsfälle oder Trainingstechniken durchzuführen.

[0259] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0260] Fig. 27 ist ein Blockdiagramm eines Prozessors 2700, der einen oder mehrere Prozessorkerne 2702A-2702N, eine integrierte Speichersteuerung 2714 und einen integrierten Grafikprozessor 2708 aufweist, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann der Prozessor 2700 zusätzliche Kerne bis einschließlich des zusätzlichen Kerns 2702N beinhalten, die durch Kästen mit gestrichelten Linien dargestellt sind. In mindestens einer Ausführungsform beinhaltet jeder der Prozessorkerne 2702A-2702N eine oder mehrere interne Cache-Einheiten 2704A-2704N. In mindestens einer Ausführungsform hat jeder Prozessorkern auch Zugriff auf eine oder mehrere gemeinsam genutzte Cache-Einheiten 2706.

[0261] In mindestens einer Ausführungsform stellen die internen Cache-Einheiten 2704A-2704N und die gemeinsam genutzten Cache-Einheiten 2706 eine Cache-Speicherhierarchie innerhalb des Prozessors 2700 dar. In mindestens einer Ausführungsform können die Cache-Speichereinheiten 2704A-2704N mindestens eine Ebene von Anweisungs- und Daten-Cache innerhalb jedes Prozessorkerns und eine oder mehrere Ebenen von gemeinsam genutztem Cache der mittleren Ebene, wie etwa einen Cache der Ebene 2 (L2), Ebene 3 (L3), Ebene 4 (L4) oder anderer Ebenen, beinhalten, wobei ein höchstes Cache-Ebene vor dem externen Speicher als LLC klassifiziert ist. In mindestens einer Ausführungsform hält die Cache-Kohärenzlogik die Kohärenz zwischen verschiedenen Cache-Einheiten 2706 und 2704A-2704N aufrecht.

[0262] In mindestens einer Ausführungsform kann der Prozessor 2700 auch einen Satz von einer oder mehreren Bussteuerungseinheiten 2716 und einen Systemagentenkern 2710 beinhalten. In mindestens einer Ausführungsform verwalten eine oder mehrere Bussteuereinheiten 2716 einen Satz von peripheren Bussen, wie zum Beispiel einen oder mehrere PCI- oder PCI-Express-Busse. In mindestens einer Ausführungsform stellt der Systemagentenkern 2710 eine Verwaltungsfunktionalität 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.

[0263] In mindestens einer Ausführungsform beinhalten einer oder mehrere der Prozessorkerne 2702A-2702N Unterstützung für simultanes Multi-Threading. In mindestens einer Ausführungsform beinhaltet der Systemagentenkern 2710 Komponenten zum Koordinieren und Betreiben der Kerne 2702A-2702N während der Multi-Thread-Verarbeitung. In mindestens einer Ausführungsform kann der Systemagentenkern 2710 zusätzlich eine Leistungssteuereinheit (power control unit - PCU) beinhalten, die Logik und Komponenten zum Regulieren eines oder mehrerer Leistungszustände der Prozessorkerne 2702A-2702N und des Grafikprozessors 2708 beinhaltet.

[0264] In mindestens einer Ausführungsform beinhaltet der Prozessor 2700 zusätzlich den Grafikprozessor 2708 zum Ausführen von Grafikverarbeitungsoperationen. In mindestens einer Ausführungsform ist der Grafikprozessor 2708 mit gemeinsam genutzten Cache-Einheiten 2706 und dem Systemagentenkern 2710 gekoppelt, der eine oder mehrere integrierte Speichersteuerungen 2714 beinhaltet. In mindestens einer Ausführungsform beinhaltet der Systemagentenkern 2710 zudem eine Anzeigesteuerung 2711, um die Grafikprozessorausgabe zu einer oder mehreren gekoppelten Anzeigen zu treiben. 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.

[0265] In mindestens einer Ausführungsform wird eine ringbasierte Zusammenschaltungseinheit 2712 zur Kopplung interner Komponenten des Prozessors 2700 verwendet. In mindestens einer Ausführungsform kann eine alternative Zusammenschaltungseinheit verwendet werden, wie etwa eine Punkt-zu-Punkt-Zusammenschaltung, eine geschaltete Zusammenschaltung oder andere Techniken. In mindestens einer Ausführungsform ist der Grafikprozessor 2708 über eine E/A-Verbindung 2713 mit der Ringzusammenschaltung 2712 gekoppelt.

[0266] In mindestens einer Ausführungsform stellt die E/A-Verbindung 2713 mindestens eine von mehreren Varietäten von E/A-Verbindungen dar, einschließlich einer gehäuseinternen E/A-Verbindung, die Kommunikation zwischen verschiedenen Prozessorkomponenten und einem eingebetteten Speichermodul 2718 mit hoher Rechenleistung, wie etwa einem eDRAM-Modul, ermöglicht. In mindestens einer Ausführungsform verwenden jeder der Prozessorkerne 2702A-2702N und der Grafikprozessor 2708 eingebettete Speichermodule 2718 als gemeinsam genutzten Cache der letzten Ebene.

[0267] 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 bezüglich der 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 eines anderen Anweisungssatzes ausführen. In mindestens einer Ausführungsform sind die Prozessorkerne 2702A-2702N bezüglich der Mikroarchitektur heterogen, wobei ein oder mehrere Kerne, die einen relativ betrachtet höheren Leistungsverbrauch aufweisen, mit einem oder mehreren Kernen, die einen niedrigeren Leistungsverbrauch aufweisen, gekoppelt sind. In min-

destens einer Ausführungsform kann der Prozessor 2700 auf einem oder mehreren Chips oder als SoC-integrierte Schaltung implementiert sein.

[0268] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform können Abschnitte oder die gesamte Inferenz- und/oder Trainingslogik 915 in den Prozessor 2700 integriert sein. Zum Beispiel können in mindestens einer Ausführungsform die in dieser Schrift beschriebenen Trainings- und/oder Inferenztechniken eine oder mehrere der ALUs verwenden, die in dem Grafikprozessor 2612, in dem/den Grafikkern(en) 2702A-2702N oder in anderen Komponenten 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. 9A** oder **Fig. 9B** veranschaulichten Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtungsparameter in chipinternem oder chipexternem Speicher und/oder in Registern (gezeigt oder nicht gezeigt) gespeichert werden, die ALUs des Grafikprozessors 2700 konfigurieren, um eine(n) oder mehrere in dieser Schrift beschriebene Algorithmen für maschinelles Lernen, neuronale Netzarchitekturen, Anwendungsfälle oder Trainingstechniken durchzuführen.

[0269] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0270] **Fig. 28** ist ein Blockdiagramm von Hardware-Logik eines Grafikprozessorkerns 2800 gemäß mindestens einer in dieser Schrift beschriebenen Ausführungsform. In mindestens einer Ausführungsform ist der Grafikprozessorkern 2800 in einem Grafikkernarray enthalten. In mindestens einer Ausführungsform kann es sich bei dem Grafikprozessorkern 2800, mitunter als Kernscheibe bezeichnet, um einen oder mehrere Grafikkerne innerhalb eines modularen Grafikprozessors handeln. In mindestens einer Ausführungsform ist der Grafikprozessorkern 2800 beispielhaft für eine Grafikkernscheibe und ein Grafikprozessor, wie in dieser Schrift beschrieben, kann auf Grundlage der angestrebten Leistungs- und Rechenleistungshüllkurven mehrere Grafikkernscheiben beinhalten. In mindestens einer Ausführungsform kann jeder Grafikkern 2800 einen Festfunktionsblock 2830 beinhalten, der mit mehreren Teilkernen 2801A-2801F gekoppelt ist, die auch als Teilscheiben bezeichnet werden und modulare Blöcke von Universal- und Festfunktionslogik beinhalten.

[0271] In mindestens einer Ausführungsform beinhaltet der Festfunktionsblock 2830 eine Geometrie-/Festfunktionspipeline 2836, die von allen Teilkernen im Grafikprozessor 2800 gemeinsam genutzt werden kann, zum Beispiel in Implementierungen mit Grafikprozessoren mit niedrigerer Rechenleistung und/oder niedrigerer Leistung. In mindestens einer Ausführungsform beinhaltet die Geometrie-/Festfunktionspipeline 2836 eine 3D-Festfunktionspipeline, eine Video-Frontend-Einheit, einen Thread-Spawner und Thread-Zuteiler sowie einen Manager für vereinheitlichten Rückgabepuffer, der den vereinheitlichten Rückgabepuffer verwaltet.

[0272] In mindestens einer Ausführungsform beinhaltet der Festfunktionsblock 2830 auch eine Grafik-SoC-Schnittstelle 2837, einen Grafik-Mikrocontroller 2838 und eine Medienpipeline 2839. In mindestens einer festen Ausführungsform stellt die Grafik-SoC-Schnittstelle 2837 eine Schnittstelle zwischen dem Grafikkern 2800 und anderen Prozessorkernen innerhalb einer integrierten Schaltung als System auf einem Chip bereit. In mindestens einer Ausführungsform ist der Grafik-Mikrocontroller 2838 ein programmierbarer Teilprozessor, der dazu konfiguriert werden kann, er verschiedene Funktionen des Grafikprozessors 2800 zu verwalten, einschließlich Thread-Zuteilung, -Planung und -Präemption. In mindestens einer Ausführungsform beinhaltet die Medienpipeline 2839 Logik zum Erleichtern der Decodierung, Codierung, Vorverarbeitung und/oder Nachverarbeitung von Multimedia-Daten, einschließlich Bild- und Videodaten. In mindestens einer Ausführungsform implementiert die Medienpipeline 2839 Medienoperationen über Anforderungen an Rechen- oder Abtastlogik innerhalb der Teilkerne 2801-2801F.

[0273] In mindestens einer Ausführungsform ermöglicht die SoC-Schnittstelle 2837 es dem Grafikkern 2800, mit Universal-Anwendungsprozessorkernen (z. B. CPUs) und/oder anderen Komponenten innerhalb eines SoC zu kommunizieren, einschließlich Speicherhierarchieelementen, wie etwa einem gemeinsam genutzten Cache-Speicher der letzten Ebene, System-RAM und/oder eingebettetem chipinternem oder gehäuseintemem DRAM. In mindestens einer Ausführungsform kann die SoC-Schnittstelle 2837 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 Grafikkern 2800 und den CPUs innerhalb eines SoCs gemeinsam genutzt werden kann. In mindestens einer Ausführungsform kann die SoC-Schnittstelle 2837 auch Leistungsverwaltungssteuerungen für den Grafikkern 2800 implementieren und eine Schnittstelle zwischen einer Taktdomäne des Grafikkerns 2800 und anderen Taktdomänen innerhalb eines SoCs ermöglichen. In mindestens einer Ausführungsform ermöglicht die SoC-Schnittstelle 2837 den Empfang von Befehlspuffern von einem Befehls-Streamer und einem globalen Thread-Zuteiler, die dazu konfiguriert sind, jedem von einem oder mehreren Grafikkernen innerhalb eines Grafikprozessors Befehle und Anweisungen bereitzustellen. In mindestens einer Ausführungsform können Befehle und Anweisungen an die Medienpipeline 2839 gesendet werden, wenn Medienoperationen durchgeführt werden sollen, oder an eine Geometrie- und Festfunktionspipeline (z. B. Geometrie- und Festfunktionspipeline 2836, Geometrie- und Festfunktionspipeline 2814), wenn Grafikverarbeitungsoperationen durchgeführt werden sollen.

[0274] In mindestens einer Ausführungsform kann der Grafik-Mikrocontroller 2838 dazu konfiguriert sein, verschiedene Planungs- und Verwaltungsaufgaben für den Grafikkern 2800 durchzuführen. In mindestens einer Ausführungsform kann der Grafik-Mikrocontroller 2838 die Planung der Grafik- und/oder Rechenarbeitslast auf verschiedenen Grafikparallel-Engines innerhalb der Arrays 2802A-2802F, 2804A-2804F der Ausführungseinheiten (execution unit - EU) innerhalb der Teilkerne 2801A-2801F durchführen. In mindestens einer Ausführungsform kann Host-Software, die auf einem CPU-Kern eines SoC, einschließlich des Grafikkerns 2800, ausgeführt wird, Arbeitslasten an eine von mehreren Grafikprozessor-Doorbells übermitteln, die eine Planungsoperation auf einer geeigneten Grafik-Engine aufruft. In mindestens einer Ausführungsform beinhalten die Planungsoperationen das Bestimmen, welche Arbeitslast als Nächstes laufen soll, das Absenden einer Arbeitslast an einen Befehls-Streamer, das Vorwegnehmen bestehender Arbeitslasten, die auf einer Engine laufen, das Überwachen des Fortschritts einer Arbeitslast und das Benachrichtigen der Host-Software, wenn eine Arbeitslast abgeschlossen ist. In mindestens einer Ausführungsform kann der Grafik-Mikrocontroller 2838 auch Zustände mit niedriger Leistung oder inaktive Zustände für den Grafikkern 2800 erleichtern, wobei dem Grafikkern 2800 eine Fähigkeit bereitgestellt wird, Register innerhalb des Grafikkerns 2800 über Zustandsübergänge mit niedriger Leistung unabhängig von einem Betriebssystem und/oder einer Grafiktreiber-Software auf einem System zu sichern und wiederherzustellen.

[0275] In mindestens einer Ausführungsform kann der Grafikkern 2800 mehr oder weniger als die veranschaulichten Teilkerne 2801A-2801F aufweisen, bis zu N modulare Teilkerne. Für jeden Satz von N Teilkernen kann der Grafikkern 2800 in mindestens einer Ausführungsform auch gemeinsam genutzte Funktionslogik 2810, gemeinsam genutzten und/oder Cache-Speicher 2812, eine Geometrie-/Festfunktionspipeline 2814 sowie zusätzliche Festfunktionslogik 2816 zum Beschleunigen verschiedener Grafik- und Rechenverarbeitungsoperationen beinhalten. In mindestens einer Ausführungsform kann die gemeinsam genutzte Funktionslogik 2810 Logikeinheiten (z. B. Abtaster-, Mathematik- und/oder Zwischen-Thread-Kommunikationslogik) beinhalten, die von N Teilkernen innerhalb des Grafikkerns 2800 gemeinsam genutzt werden können. In mindestens einer Ausführungsform kann der feste, gemeinsam genutzte und/oder Cache-Speicher 2812 ein Cache der letzten Ebene für N Teilkerne 2801A-2801F innerhalb des Grafikkerns 2800 sein und kann auch als gemeinsam genutzter Speicher dienen, auf den mehrere Teilkerne zugreifen können. In mindestens einer Ausführungsform kann die Geometrie-/Festfunktionspipeline 2814 anstelle der Geometrie-/Festfunktionspipeline 2836 innerhalb des Festfunktionsblocks 2830 enthalten sein und gleiche oder ähnliche Logikeinheiten beinhalten.

[0276] In mindestens einer Ausführungsform beinhaltet der Grafikkern 2800 zusätzliche Festfunktionslogik 2816, die verschiedene Festfunktionsbeschleunigungslogik zur Verwendung durch den Grafikkern 2800 beinhalten kann. In mindestens einer Ausführungsform beinhaltet die zusätzliche Festfunktionslogik 2816 eine zusätzliche Geometriepipeline für die Verwendung beim Shading von nur der Position. Beim Shading von nur der Position existieren mindestens zwei Geometriepipelines, während eine vollständige Geometriepipeline innerhalb der Geometrie-/Festfunktionspipeline 2816, 2836 und eine Culling-Pipeline, die eine zusätzliche Geometriepipeline ist, die innerhalb der zusätzlichen Festfunktionslogik 2816 enthalten sein kann. In mindestens einer Ausführungsform ist die Culling-Pipeline eine abgespeckte Version einer vollständigen Geometriepipeline. In mindestens einer Ausführungsform können eine vollständige Pipeline und eine Culling-Pipeline unterschiedliche Instanzen einer Anwendung ausführen, wobei jede Instanz einen separaten Kontext aufweist. In mindestens einer Ausführungsform kann das Shading von nur der Position lange Culling-Läufe von verworfenen Dreiecken verbergen, sodass ermöglicht wird, dass das Shading in einigen Fällen früher abgeschlossen wird. Zum Beispiel kann in mindestens einer Ausführungsform die Culling-Pipeline-Logik innerhalb der zusätzlichen Festfunktionslogik 2816 Positions-Shader parallel zu einer Hauptanwendung ausführen und erzeugt im Allgemeinen kritische Ergebnisse schneller als eine vollständige Pipeline, da die Culling-Pipeline das Positionsattribut von Vertices abruft und schattiert, ohne eine Rasterung und ein

Rendering von Pixeln in einem Bildspeicher durchzuführen. In mindestens einer Ausführungsform kann die Culling-Pipeline erzeugte kritische Ergebnisse verwenden, um Sichtbarkeitsinformationen für alle Dreiecke zu berechnen, ohne Rücksicht darauf, ob diese Dreiecke Culling unterzogen werden. In mindestens einer Ausführungsform kann die vollständige Pipeline (die in diesem Fall als Wiederholungspipeline bezeichnet werden kann) Sichtbarkeitsinformationen verbrauchen, um Culling unterzogene Dreiecke zu überspringen, um nur sichtbare Dreiecke zu schattieren, die schließlich an eine Rasterungsphase übergeben werden.

[0277] In mindestens einer Ausführungsform kann die zusätzliche Festfunktionslogik 2816 auch Logik zur Beschleunigung des maschinellen Lernens wie etwa Festfunktions-Matrixmultiplikationslogik für Implementationen beinhalten, die Optimierungen für das Training oder das Inferenzieren des maschinellen Lernens beinhalten.

[0278] In mindestens einer Ausführungsform beinhaltet jeder Grafikteilkern 2801A-2801F 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 Grafikteilkerne 2801A-2801F mehrere EU-Arrays 2802A-2802F, 2804A-2804F, Logik 2803A-2803F zur Thread-Zuteilung und Zwischen-Thread-Kommunikation (thread dispatch/inter-thread communication - TD/IC), einen 3D-(z. B. Textur-)Abtaster 2805A-2805F, einen Medienabtaster 2806A-2806F, einen Shader-Prozessor 2807A-2807F und gemeinsam genutzten lokalen Speicher (shared local memory - SLM) 2808A-2808F. Die EU-Arrays 2802A-2802F, 2804A-2804F beinhalten jeweils mehrere Ausführungseinheiten, bei denen es sich um Universal-Grafikverarbeitungseinheiten handelt, die in der Lage sind, Gleitkomma- und Integer-/Festkomma-Logikoperationen im Dienste einer Grafik-, Medien- oder Rechenoperation, einschließlich Grafik-, Medien- oder Rechen-Shader-Programmen, durchzuführen. In mindestens einer Ausführungsform führt die TD/IC-Logik 2803A-2803F lokale Thread-Zuteilungs- und Thread-Steueroperationen für Ausführungseinheiten innerhalb eines Teilkerns durch und erleichtert die Kommunikation zwischen Threads, die auf Ausführungseinheiten eines Teilkerns ausgeführt werden. In mindestens einer Ausführungsform können die 3D-Abtaster 2805A-2805F Textur- oder andere mit 3D-Grafik verwandte Daten in Speicher lesen. In mindestens einer Ausführungsform kann der 3D-Abtaster Texturdaten auf Grundlage eines konfigurierten Abtastzustands und eines Texturformats, das einer gegebenen Textur zugeordnet ist, unterschiedlich lesen. In mindestens einer Ausführungsform kann der Medien-Abtaster 2806A-2806F ähnliche Leseoperationen auf Grundlage einer und eines Formats durchführen, die den Mediendaten zugeordnet sind. In mindestens einer Ausführungsform kann jeder Grafikteilkern 2801A-2801F alternativ einen vereinheitlichten 3D- und Medienabtaster beinhalten. In mindestens einer Ausführungsform können Threads, die auf Ausführungseinheiten innerhalb jedes der Teilkerne 2801A-2801F ausgeführt werden, den gemeinsam genutzten lokalen Speicher 2808A-2808F innerhalb jedes Teilkerns verwenden, um zu ermöglichen, dass Threads, die innerhalb einer Thread-Gruppe ausgeführt werden, unter Verwendung eines gemeinsamen Pools von chipinternem Speicher ausgeführt werden.

[0279] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform können Abschnitte oder die gesamte Inferenz- und/oder Trainingslogik 915 in den Grafikprozessor 2810 integriert sein. Zum Beispiel können in mindestens einer Ausführungsform die in dieser Schrift beschriebenen Trainings- und/oder Inferenztechniken eine oder mehrere der ALUs verwenden, die in dem Grafikprozessor 2612, in dem Grafik-Mikrocontroller 2838, in der Geometrie- und Festfunktionspipeline 2814 und 2836 oder einer anderen Logik in **Fig. 27**. Darüber hinaus können in mindestens einer Ausführungsform die in dieser Schrift beschriebenen Inferenz- und/oder Trainingsoperationen unter Verwendung von anderer Logik als der in den **Fig. 9A** oder **Fig. 9B** veranschaulichten Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtungsparameter 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 in dieser Schrift beschriebene Algorithmen für maschinelles Lernen, neuronale Netzarchitekturen, Anwendungsfälle oder Trainingstechniken durchzuführen.

[0280] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0281] Die **Fig. 29A-29B** veranschaulichen die Thread-Ausführungslogik 2900, die ein Array von Verarbeitungselementen eines Grafikprozessorkerns beinhaltet, gemäß mindestens einer Ausführungsform. **Fig. 29A**

veranschaulicht mindestens eine Ausführungsform, in der die Thread-Ausführungslogik 2900 verwendet wird. **Fig. 29B** veranschaulicht beispielhafte interne Details einer Ausführungseinheit gemäß mindestens einer Ausführungsform.

[0282] Wie in **Fig. 29A** veranschaulicht, beinhaltet die Thread-Ausführungslogik 2900 in mindestens einer Ausführungsform einen Shader-Prozessor 2902, einen Thread-Zuteiler 2904, einen Anweisungs-Cache 2906, ein skalierbares Ausführungseinheitenarray einschließlich einer Vielzahl von Ausführungseinheiten 2908A-2908N, (einen) Abtaster 2910, einen Daten-Cache 2912 und einen Datenport 2914. In mindestens einer Ausführungsform kann ein skalierbares Ausführungseinheitenarray dynamisch skalieren, indem eine oder mehrere Ausführungseinheiten (z. B. beliebige der Ausführungseinheiten 2908A, 2908B, 2908C, 2908D bis 2908N-1 und 2908N) zum Beispiel auf Grundlage von Rechenanforderungen einer Arbeitslast aktiviert oder deaktiviert werden. In mindestens einer Ausführungsform sind die skalierbaren Ausführungseinheiten über eine Zusammenschaltungsstruktur miteinander verbunden, die mit allen Ausführungseinheiten verbunden ist. In mindestens einer Ausführungsform umfasst die Thread-Ausführungslogik 2900 eine oder mehrere Verbindungen zu einem Speicher, z. B. einem Systemspeicher oder einem Cache-Speicher, durch eines oder mehrere des Anweisungs-Caches 2906, des Datenports 2914, des Abtasters 2910 und der Ausführungseinheiten 2908A-2908N. In mindestens einer Ausführungsform ist jede Ausführungseinheit (z. B. 2908A) 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 2908A-2908N so skalierbar, dass es eine beliebige Anzahl einzelner Ausführungseinheiten beinhaltet.

[0283] In mindestens einer Ausführungsform werden die Ausführungseinheiten 2908A-2908N hauptsächlich zur Ausführung von Shader-Programmen verwendet. In mindestens einer Ausführungsform kann der Shader-Prozessor 2902 verschiedene Shader-Programme verarbeiten und den Shader-Programmen zugeordnete Ausführungs-Threads über einen Thread-Zuteiler 2904 zuteilen. In mindestens einer Ausführungsform beinhaltet der Thread-Zuteiler 2904 Logik zur Vermittlung von Thread-Initiierungsanforderungen von Grafik- und Medienpipelines und zur Instanziierung angeforderter Threads auf einer oder mehreren Ausführungseinheiten in den Ausführungseinheiten 2908A-2908N. Zum Beispiel kann in mindestens einer Ausführungsform eine Geometriepipeline Vertex-, Tesselations- oder Geometrie-Shader der Thread-Ausführungslogik zum Verarbeiten zuteilen. In mindestens einer Ausführungsform kann der Thread-Zuteiler 2904 auch Laufzeit-Thread-Erzeugungsanforderungen von ausführenden Shader-Programmen verarbeiten.

[0284] In mindestens einer Ausführungsform unterstützen die Ausführungseinheiten 2908A-2908N einen Anweisungssatz, der systemeigene Unterstützung für viele Standard-3D-Grafik-Shader-Anweisungen beinhaltet, sodass Shader-Programme aus Grafikbibliotheken (z. B. Direct 3D und OpenGL) mit einer minimalen Übersetzung ausgeführt werden. In mindestens einer Ausführungsform unterstützen die Ausführungseinheiten die Vertex- und Geometrieverarbeitung (z. B. Vertex-Programme, Geometrieprogramme, Vertex-Shader), die Pixelverarbeitung (z.B. Pixel-Shader, Fragment-Shader) und die Universalverarbeitung (z. B. Rechen- und Medien-Shader). In mindestens einer Ausführungsform ist jede der Ausführungseinheiten 2908A-2908N, die eine oder mehrere arithmetisch-logische Einheiten (ALUs) beinhalten, zur Multi-Issue-Single-Instruction-Multiple-Data-(SIMD-)Ausführung in der Lage und der Mehr-Thread-Betrieb ermöglicht eine effiziente Ausführungsumgebung trotz Speicherzugriffen mit höherer Latenz. In mindestens einer Ausführungsform weist jeder Hardware-Thread innerhalb jeder Ausführungseinheit eine dedizierte Registerdatei mit hoher Bandbreite und einen zugeordneten unabhängigen Thread-Zustand auf. In mindestens einer Ausführungsform erfolgt die Ausführung mit mehreren Ausgaben pro Takt an Pipelines, die zu Integer- und Gleitkommaoperationen mit einfacher oder doppelter Genauigkeit, SIMD-Verzweigungsfähigkeit, logischen Operationen, transzendentalen Operationen und anderen sonstigen Operationen in der Lage sind. In mindestens einer Ausführungsform bewirkt die Abhängigkeitslogik in den Ausführungseinheiten 2908A-2908N, dass ein wartender Thread schläft, bis die angeforderten Daten zurückgegeben wurden, während er auf Daten aus dem Speicher oder einer der gemeinsam genutzten Funktionen wartet. In mindestens einer Ausführungsform können, während ein wartender Thread schläft, Hardware-Ressourcen für die Verarbeitung anderer Threads verwendet werden. Zum Beispiel kann in mindestens einer Ausführungsform eine Ausführungseinheit während einer Verzögerung, die einer Vertex-Shader-Operation zugeordnet ist, Operationen für einen Pixel-Shader, Fragment-Shader oder eine andere Art von Shader-Programm, einschließlich eines anderen Vertex-Shaders, durchführen.

[0285] In mindestens einer Ausführungsform arbeitet jede Ausführungseinheit in den Ausführungseinheiten 2908A-2908N an Arrays von Datenelementen. In mindestens einer Ausführungsform ist die Anzahl der Datenelemente die „Ausführungsgröße“ oder die Anzahl der Kanäle für eine Anweisung. In mindestens einer

Ausführungsform ist ein Ausführungskanal eine logische Ausführungseinheit für den Zugriff auf Datenelemente, die Maskierung und die Flusssteuerung innerhalb von Anweisungen. In mindestens einer Ausführungsform kann die Anzahl der Kanäle unabhängig von einer Anzahl der physischen arithmetisch-logischen Einheiten (ALUs) oder Gleitkommaeinheiten (FPUs) für einen konkreten Grafikprozessor sein. In mindestens einer Ausführungsform unterstützen die Ausführungseinheiten 2908A-2908N Integer- und Gleitkommadatenarten.

[0286] In mindestens einer Ausführungsform beinhaltet ein Anweisungssatz einer Ausführungseinheit SIMD-Anweisungen. In mindestens einer Ausführungsform können verschiedene Datenelemente als gepackter Datentyp in einem Register gespeichert werden und eine Ausführungseinheit verarbeitet verschiedene Elemente auf Grundlage der Datengröße der Elemente. Zum Beispiel werden in mindestens einer Ausführungsform beim Betreiben an einem 256 Bit breiten Vektor 256 Bit eines Vektors in einem Register gespeichert und eine Ausführungseinheit wird an einem Vektor als vier separate 64 Bit große gepackte Datenelemente (Datenelemente der Größe Vierfachwort (Quad-Word- QW)), acht separate 32 Bit große gepackte Datenelemente (Datenelemente der Größe Doppelwort (Double Word - DW)), sechzehn separate 16 Bit große gepackte Datenelemente (Datenelemente der Größe Wort (Word - W)) oder zweiunddreißig separate 8 Bit große Datenelemente (Datenelemente der Größe Byte (B)) betrieben. In mindestens einer Ausführungsform sind jedoch andere Vektorbreiten und Registergrößen möglich.

[0287] In mindestens einer Ausführungsform können eine oder mehrere Ausführungseinheiten zu einer fusionierten Ausführungseinheit 2909A-2909N kombiniert werden, die Thread-Steuerlogik (2907A-2907N) aufweist, die fusionierten EUs gemeinsam ist. In mindestens einer Ausführungsform können mehrere EUs zu einer EU-Gruppe fusioniert werden. In mindestens einer Ausführungsform kann jede EU in der fusionierten EU-Gruppe dazu konfiguriert sein, einen separaten SIMD-Hardware-Thread auszuführen. Die Anzahl der EUs in einer fusionierten EU-Gruppe kann gemäß verschiedenen Ausführungsformen variieren. In mindestens einer Ausführungsform können verschiedene SIMD-Breiten pro EU durchgeführt werden, einschließlich, aber nicht beschränkt auf, SIMD8, SIMD16 und SIMD32. In mindestens einer Ausführungsform beinhaltet jede fusionierte Grafikausführungseinheit 2909A-2909N mindestens zwei Ausführungseinheiten. Zum Beispiel beinhaltet in mindestens einer Ausführungsform die fusionierte Ausführungseinheit 2909A eine erste EU 2908A, eine zweite EU 2908B und Thread-Steuerlogik 2907A, die der ersten EU 2908A und der zweiten EU 2908B gemeinsam ist. In mindestens einer Ausführungsform steuert die Thread-Steuerlogik 2907A Threads, die auf der fusionierten Grafikausführungseinheit 2909A ausgeführt werden, sodass jede EU innerhalb der fusionierten Ausführungseinheiten 2909A-2909N unter Verwendung eines gemeinsamen Anweisungszeigerregisters ausgeführt werden kann.

[0288] In mindestens einer Ausführungsform sind ein oder mehrere interne Anweisungs-Caches (z.B. 2906) in der Thread-Ausführungslogik 2900 enthalten, um Thread-Anweisungen für Ausführungseinheiten zwischenzuspeichern. In mindestens einer Ausführungsform sind ein oder mehrere Daten-Caches (z. B. 2912) enthalten, um Thread-Daten während der Thread-Ausführung zwischenzuspeichern. In mindestens einer Ausführungsform ist ein Abtaster 2910 enthalten, der Texturabtastung für 3D-Operationen und Medienabtastung für Medienoperationen bereitstellt. In mindestens einer Ausführungsform beinhaltet der Abtaster 2910 eine spezielle Textur- oder Medienabtastfunktionalität, um Textur- oder Mediendaten während eines Abtastprozesses zu verarbeiten, bevor die abgetasteten Daten einer Ausführungseinheit bereitgestellt werden.

[0289] Während der Ausführung senden Grafik- und Medienpipelines in mindestens einer Ausführungsform Thread-Initiierungsanforderungen über die Thread-Erzeugungs- und - Zuteilungslogik an die Thread-Ausführungslogik 2900. 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 2902 aufgerufen, um Ausgabeinformationen weiter zu berechnen und zu veranlassen, dass die Ergebnisse in Ausgabeflächen (z. B. Farbpuffer, Tiefenpuffer, Schablonenpuffer usw.) geschrieben werden. In mindestens einer Ausführungsform berechnet ein Pixel-Shader oder Fragment-Shader die Werte verschiedener Vertex-Attribute, die über ein gerastertes Objekt interpoliert werden sollen. In mindestens einer Ausführungsform führt die Pixelprozessorlogik innerhalb des Shader-Prozessors 2902 dann ein über eine Anwendungsprogrammierschnittstelle (API) zugeführtes Pixel- oder Fragment-Shader-Programm aus. In mindestens einer Ausführungsform teilt der Shader-Prozessor 2902 zum Ausführen eines Shader-Programms Threads über den Thread-Zuteiler 2904 einer Ausführungseinheit (z.B. 2908A) zu. In mindestens einer Ausführungsform verwendet der Shader-Prozessor 2902 die Texturabtastlogik in dem Abtaster 2910, um auf Texturdaten in Texturkarten zuzugreifen, die in Speicher gespeichert sind. In mindestens einer Ausführungsform werden durch arithmetische Operationen an Texturdaten und ein-

gegebenen Geometriedaten Pixelfarbdaten für jedes geometrische Fragment berechnet oder ein oder mehrere Pixel von der weiteren Verarbeitung ausgeschlossen.

[0290] In mindestens einer Ausführungsform stellt der Datenport 2914 einen Speicherzugriffsmechanismus für die Thread-Ausführungslogik 2900 bereit, um verarbeitete Daten zur weiteren Verarbeitung an einer Grafikprozessor-Ausgabepipeline an Speicher auszugeben. In mindestens einer Ausführungsform beinhaltet der Datenport 2914 einen oder mehrere Cache-Speicher (z. B. den Daten-Cache 2912) oder ist an diesen gekoppelt, um Daten für den Speicherzugriff über einen Datenport zwischenzuspeichern.

[0291] Wie in **Fig. 29B** veranschaulicht, kann eine Grafikausführungseinheit 2908 in mindestens einer Ausführungsform eine Anweisungsabrufeinheit 2937, ein Array von allgemeinen Registerdateien (general register file - GRF) 2924, eine Anordnung von architektonischen Registerdateien (architectural register file - ARF) 2926, einen Thread-Vermittler 2922, eine Sendeeinheit 2930, eine Verzweigungseinheit 2932, einen Satz SIMD-Gleitkommaeinheiten (FPUs) 2934 und in mindestens einer Ausführungsform einen Satz dedizierter Integer-SIMD-ALUs 2935 beinhalten. In mindestens einer Ausführungsform beinhalten die GRF 2924 und die ARF 2926 einen Satz allgemeiner Registerdateien und Architekturregisterdateien, die jedem simultanen Hardware-Thread zugeordnet sind, der in der Grafikausführungseinheit 2908 aktiv sein kann. In mindestens einer Ausführungsform wird der architektonische Zustand pro Thread in der ARF 2926 verwaltet, während die während der Thread-Ausführung verwendeten Daten in der GRF 2924 gespeichert werden. In mindestens einer Ausführungsform kann der Ausführungszustand jedes Threads, einschließlich der Anweisungszeiger für jeden Thread, in Thread-spezifischen Registern in der ARF 2926 aufbewahrt werden.

[0292] In mindestens einer Ausführungsform weist die Grafikausführungseinheit 2908 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 auf Grundlage einer Zielanzahl von simultanen Threads und Anzahl von Registern pro Ausführungseinheit fein abgestimmt werden kann, wobei die Ressourcen der Ausführungseinheit auf die Logik aufgeteilt werden, die zum Ausführen mehrerer simultaner Threads verwendet wird.

[0293] In mindestens einer Ausführungsform kann die Grafikausführungseinheit 2908 mehrere Anweisungen gleichzeitig ausgeben, die jeweils unterschiedliche Anweisungen sein können. In mindestens einer Ausführungsform kann der Thread-Vermittler 2922 des Grafikausführungseinheits-Threads 2908 Anweisungen einer der Sendeeinheit 2930, der Verzweigungseinheit 2942 oder der SIMD-FPU(s) 2934 zur Ausführung zuteilen. In mindestens einer Ausführungsform kann jeder Ausführungs-Thread auf 128 Universalregister innerhalb der GRF 2924 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 KByte innerhalb der GRF 2924 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 eine Anzahl der Threads pro Ausführungseinheit gemäß Ausführungsformen auch variieren kann. In mindestens einer Ausführungsform, in der sieben Threads auf 4 KByte zugreifen können, kann die GRF 2924 insgesamt 28 KByte 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.

[0294] In mindestens einer Ausführungsform werden Speicheroperationen, Abtasteroperationen und andere Systemkommunikationen mit längerer Latenz über „Sende“-Anweisungen zugeteilt, die durch Nachrichtenweitergabe an die Sendeeinheit 2930 ausgeführt werden. In mindestens einer Ausführungsform werden Verzweigungsanweisungen einer dedizierten Verzweigungseinheit 2932 zugeteilt, um SIMD-Divergenz und schließlich -Konvergenz zu ermöglichen.

[0295] In mindestens einer Ausführungsform beinhaltet die Grafikausführungseinheit 2908 eine oder mehrere SIMD-Gleitkommaeinheiten (FPU(s)) 2934 zur Durchführung von Gleitkommaoperationen. In mindestens einer Ausführungsform unterstützen die FPU(s) 2934 auch Integerberechnung. In mindestens einer Ausführungsform können die FPU(s) 2934 bis zur Anzahl von M 32-Bit-Gleitkomma(oder -Integer)operationen über SMID ausführen oder bis zu 2M 16-Bit-Integer- oder 16-Bit-Gleitkommaoperationen über SMID ausführen. In mindestens einer Ausführungsform stellt mindestens eine der FPU(s) erweiterte mathematische Fähigkeiten bereit, um transzendentale 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 2935 vorhanden, der spezifisch zum Durchführen von Operationen, die Berechnungen für maschinelles Lernen zugeordnet sind, optimiert sein kann.

[0296] In mindestens einer Ausführungsform können Arrays von mehreren Instanzen der Grafikausführungseinheit 2908 in einer Grafikteilkern-Gruppierung (z. B. einer Teilscheibe) instanziiert sein. In mindestens einer Ausführungsform kann die Ausführungseinheit 2908 Anweisungen über eine Vielzahl von Ausführungskanälen hinweg ausführen. In mindestens einer Ausführungsform wird jeder Thread, der auf der Grafikausführungseinheit 2908 ausgeführt wird, auf einem anderen Kanal ausgeführt.

[0297] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform können Abschnitte oder die gesamte Inferenz- und/oder Trainingslogik 915 in die Ausführungslogik 2900 integriert sein. Darüber hinaus können in mindestens einer Ausführungsform die in dieser Schrift beschriebenen Inferenz- und/oder Trainingsoperationen unter Verwendung von anderer Logik als der in den **Fig. 9A** oder **Fig. 9B** veranschaulichten Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtungsparameter im chipinternen oder chip externen Speicher und/oder in Registern (gezeigt oder nicht gezeigt) gespeichert werden, die ALUs der Ausführungslogik 2900 konfigurieren, um eine(n) oder mehrere in dieser Schrift beschriebene Algorithmen für maschinelles Lernen, neuronale Netzarchitekturen, Anwendungsfälle oder Trainingstechniken durchzuführen.

[0298] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0299] **Fig. 30** veranschaulicht eine Parallelverarbeitungseinheit („PPU“) 3000 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist die PPU 3000 mit maschinenlesbarem Code konfiguriert, der bei Ausführung durch die PPU 3000 die PPU 3000 dazu veranlasst, einige oder alle der in dieser Offenbarung beschriebenen Prozesse und Techniken durchzuführen. In mindestens einer Ausführungsform ist die PPU 3000 ein Multi-Thread-Prozessor, der auf einer oder mehreren Vorrichtungen mit integrierter Schaltung implementiert ist und der Multi-Threading als Technik zur Latenzverbergung nutzt, die dazu ausgestaltet ist, computerlesbare Anweisungen (auch als maschinenlesbare Anweisungen oder einfach Anweisungen bezeichnet) auf mehreren Threads parallel zu verarbeiten. In mindestens einer Ausführungsform bezieht sich ein Thread auf einen Ausführungs-Thread und er ist eine Instanziierung eines Satzes von Anweisungen, der dazu konfiguriert ist, durch die PPU 3000 ausgeführt zu werden. In mindestens einer Ausführungsform ist die PPU 3000 eine Grafikverarbeitungseinheit („GPU“), die dazu konfiguriert ist, eine Grafik-Rendering-Pipeline zum Verarbeiten dreidimensionaler („3D“) Grafikdaten zu implementieren, um zweidimensionale („2D“) Bilddaten zur Anzeige auf einer Anzeigevorrichtung, wie etwa einer Flüssigkristallanzeige-(„LCD“-)Vorrichtung, zu erzeugen. In mindestens einer Ausführungsform wird die PPU 3000 verwendet, um Berechnungen, wie etwa lineare Algebraoperationen und Operationen für maschinelles Lernen, durchzuführen. **Fig. 30** veranschaulicht ein Beispiel für einen Parallelprozessor, der nur zu Veranschaulichungszwecken dient und als nicht einschränkendes Beispiel für Prozessorarchitekturen ausgelegt werden sollte, die innerhalb des Umfangs dieser Offenbarung in Betracht gezogen werden, und dass ein beliebiger geeigneter Prozessor zur Ergänzung und/oder als Ersatz für diesen eingesetzt werden kann.

[0300] In mindestens einer Ausführungsform sind eine oder mehrere PPUs 3000 dazu konfiguriert, Anwendungen für Hochleistungsrechnen (High Performance Computing - „HPC“), für ein Rechenzentrum und für maschinelles Lernen zu beschleunigen. In mindestens einer Ausführungsform ist die PPU 3000 dazu konfiguriert, Deep-Learning-Systeme und - Anwendungen zu beschleunigen, einschließlich der folgenden nicht einschränkenden Beispiele: Plattformen für autonome Fahrzeuge, Deep Learning, Sprach-, Bild- und Texterkennungssysteme mit hoher Genauigkeit, intelligente Videoanalytik, Molekularsimulationen, Arzneimittelforschung, Krankheitsdiagnose, Wettervorhersage, Big-Data-Analytik, Astronomie, Molekulardynamiksimulation, Finanzmodellierung, Robotik, Fabrikautomatisierung, Echtzeit-Sprachübersetzung, Online-Suchoptimierungen und personalisierte Benutzerempfehlungen und andere.

[0301] In mindestens einer Ausführungsform beinhaltet die PPU 3000 ohne Einschränkung eine Eingabe/-Ausgabe-(„E/A“-)Einheit 3006, eine Frontend-Einheit 3010, eine Planer-Einheit 3012, eine Arbeitsverteilungseinheit 3014, einen Hub 3016, eine Kreuzschiene („Xbar“) 3020, ein oder mehrere Universalverarbeitungscluster („GPCs“) 3018 und eine oder mehrere Partitionseinheiten („Speicherpartitionseinheiten“) 3022. In mindestens einer Ausführungsform ist die PPU 3000 mit einem Host-Prozessor oder anderen PPUs 3000

über eine oder mehrere Hochgeschwindigkeits-GPU-Zusammenschaltungen („GPU-Zusammenschaltungen“) 3008 verbunden. In mindestens einer Ausführungsform ist die PPU 3000 über eine Zusammenschaltung 3002 mit einem Host-Prozessor oder anderen Peripherievorrichtungen verbunden. In mindestens einer Ausführungsform ist die PPU 3000 mit einem lokalen Speicher verbunden, der eine oder mehrere Speichervorrichtungen („Speicher“) 3004 umfasst. In mindestens einer Ausführungsform beinhalten die Speichervorrichtungen 3004 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.

[0302] In mindestens einer Ausführungsform kann sich die Hochgeschwindigkeits-GPU-Zusammenschaltung 3008 auf eine drahtbasierte mehrspurige Kommunikationsverbindung beziehen, die durch Systeme zum Skalieren verwendet wird und eine oder mehrere PPUs 3000 in Kombination mit einer oder mehreren zentralen Verarbeitungseinheiten („CPUs“) beinhaltet und die Cache-Kohärenz zwischen PPUs 3000 und CPUs sowie CPU-Mastering unterstützt. In mindestens einer Ausführungsform werden Daten und/oder Befehle durch die Hochgeschwindigkeits-GPU-Zusammenschaltung 3008 über den Hub 3016 zu/von anderen Einheiten der PPU 3000 übertragen, wie etwa einer/einem oder mehreren Kopier-Engines, Videocodierern, Videodecodierem, Leistungsverwaltungseinheiten und anderen Komponenten, die in **Fig. 30**.

[0303] In mindestens einer Ausführungsform ist die E/A-Einheit 3006 dazu konfiguriert, Kommunikationen (z. B. Befehle, Daten) von einem Host-Prozessor (in **Fig. 30** nicht veranschaulicht) über den Systembus 3002 zu übertragen und zu empfangen In mindestens einer Ausführungsform kommuniziert die E/A-Einheit 3006 mit dem Host-Prozessor direkt über den Systembus 3002 oder durch eine oder mehrere Zwischenvorrichtungen, wie etwa eine Speicherbrücke. In mindestens einer Ausführungsform kann die E/A-Einheit 3006 über den Systembus 3002 mit einem oder mehreren anderen Prozessoren kommunizieren, wie etwa einer oder mehreren der PPUs 3000. In mindestens einer Ausführungsform implementiert die E/A-Einheit 3006 eine Peripheral-Component-Interconnect-Express-(„PCIe“-)Schnittstelle für die Kommunikation über einen PCIe-Bus. In mindestens einer Ausführungsform implementiert die E/A-Einheit 3006 Schnittstellen zum Kommunizieren mit externen Vorrichtungen.

[0304] In mindestens einer Ausführungsform decodiert die E/A-Einheit 3006 über den Systembus 3002 empfangene Pakete. In mindestens einer Ausführungsform stellen mindestens einige Pakete Befehle dar, die dazu konfiguriert sind, die PPU 3000 dazu zu veranlassen, verschiedene Operationen durchzuführen. In mindestens einer Ausführungsform überträgt die E/A-Einheit 3006 decodierte Befehle an verschiedene andere Einheiten der PPU 3000, wie durch Befehle vorgegeben. In mindestens einer Ausführungsform werden Befehle an die Frontend-Einheit 3010 übertragen und/oder an den Hub 3016 oder andere Einheiten der PPU 3000 übertragen, wie etwa eine oder mehrere Kopier-Engines, einen Videocodierer, einen Videodecodierer, eine Leistungsverwaltungseinheit usw. (in **Fig. 30** nicht explizit veranschaulicht). In mindestens einer Ausführungsform ist die E/A-Einheit 3006 dazu konfiguriert, Kommunikation zwischen und unter verschiedenen logischen Einheiten der PPU 3000 zu leiten.

[0305] In mindestens einer Ausführungsform codiert ein durch den Host-Prozessor ausgeführtes Programm einen Befehlsstrom in einem Puffer, der der PPU 3000 Arbeitslasten zur Verarbeitung bereitstellt. In mindestens einer Ausführungsform umfasst eine Arbeitslast Anweisungen und Daten, die durch diese Anweisungen verarbeitet werden sollen. In mindestens einer Ausführungsform ist der Puffer eine Region in einem Speicher, auf die sowohl der Host-Prozessor als auch die PPU 3000 zugreifen können (z. B. Lesen/Schreiben) - eine Host-Schnittstelleneinheit kann dazu konfiguriert sein, auf den Puffer in einem Systemspeicher zuzugreifen, der mit dem Systembus 3002 verbunden ist, und zwar über Speicheranforderungen, die über den Systembus 3002 von der E/A-Einheit 3006 übertragen werden. In mindestens einer Ausführungsform schreibt der Host-Prozessor den Befehlsstrom in den Puffer und überträgt dann einen Zeiger für den Start des Befehlsstroms an die PPU 3000, sodass die Frontend-Einheit 3010 Zeiger für einen oder mehrere Befehlsströme empfängt und einen oder mehrere Befehlsströme verwaltet, indem sie Befehle aus den Befehlsströmen liest und Befehle an verschiedene Einheiten der PPU 3000 weiterleitet.

[0306] In mindestens einer Ausführungsform ist die Frontend-Einheit 3010 an die Planer-Einheit 3012 gekoppelt, die verschiedene GPCs 3018 zum Verarbeiten von Aufgaben konfiguriert, die durch einen oder mehrere Befehlsströme definiert sind. In mindestens einer Ausführungsform ist die Planer-Einheit 3012 dazu konfiguriert, Zustandsinformationen in Bezug auf verschiedene durch die Planer-Einheit 3012 verwaltete Aufgaben zu verfolgen, wobei die Zustandsinformationen angeben können, welchem der GPCs 3018 eine Aufgabe zugewiesen ist, ob die Aufgabe Task aktiv oder inaktiv ist, welche Prioritätsstufe der Aufgabe zugeord-

net ist usw. In mindestens einer Ausführungsform verwaltet die Planer-Einheit 3012 die Ausführung einer Vielzahl von Aufgaben auf einem oder mehreren GPCs 3018.

[0307] In mindestens einer Ausführungsform ist die Planer-Einheit 3012 an die Arbeitsverteilungseinheit 3014 gekoppelt, die dazu konfiguriert ist, Aufgaben zur Ausführung auf den GPCs 3018 zuzuteilen. In mindestens einer Ausführungsform verfolgt die Arbeitsverteilungseinheit 3014 eine Anzahl geplanter Aufgaben nach, die von der Planer-Einheit 3012 empfangen wurde, und die Arbeitsverteilungseinheit 3014 verwaltet einen Pool ausstehender Aufgaben und einen Pool aktiver Aufgaben für jeden der GPCs 3018. In mindestens einer Ausführungsform umfasst der Pool ausstehender Aufgaben eine Anzahl von Schlitzen (z. B. 32 Schlitze), die Aufgaben enthalten, die zur Verarbeitung durch einen konkreten GPC 3018 zugewiesen sind; der Pool aktiver Aufgaben kann eine Anzahl von Schlitzen (z. B. 4 Schlitze) für Aufgaben umfassen, die aktiv von den GPCs 3018 verarbeitet werden, sodass, wenn einer der GPCs 3018 die Ausführung einer Aufgabe abschließt, diese Aufgabe aus dem Pool aktiver Aufgaben für den GPC 3018 entfernt wird und eine der anderen Aufgaben aus dem Pool ausstehender Aufgaben ausgewählt und zur Ausführung auf dem GPC 3018 geplant wird. In mindestens einer Ausführungsform wird, wenn eine aktive Aufgabe auf dem GPC 3018 inaktiv ist, wie etwa während sie darauf wartet, dass eine Datenabhängigkeit aufgelöst wird, dann die aktive Aufgabe aus dem GPC 3018 entfernt und in den Pool ausstehender Aufgaben zurückgeführt, während eine andere Aufgabe im Pool ausstehender Aufgaben ausgewählt und zur Ausführung auf dem GPC 3018 geplant wird.

[0308] In mindestens einer Ausführungsform kommuniziert die Arbeitsverteilungseinheit 3014 mit einem oder mehreren GPCs 3018 über die XBar 3020. In mindestens einer Ausführungsform ist die XBar 3020 ein Zusammenschaltungsnetz, das viele Einheiten der PPU 3000 an andere Einheiten der PPU 3000 koppelt und dazu konfiguriert sein kann, die Arbeitsverteilungseinheit 3014 an einen konkreten GPC 3018 zu koppeln. In mindestens einer Ausführungsform können auch eine oder mehrere andere Einheiten der PPU 3000 über den Hub 3016 mit der XBar 3020 verbunden sein.

[0309] In mindestens einer Ausführungsform werden Aufgaben durch die Planer-Einheit 3012 verwaltet und durch die Arbeitsverteilungseinheit 3014 einem der GPCs 3018 zugeteilt. Der GPC 3018 ist dazu konfiguriert, die Aufgabe zu verarbeiten und Ergebnisse zu erzeugen. In mindestens einer Ausführungsform können die Ergebnisse durch andere Aufgaben innerhalb des GPC 3018 verbraucht, über die XBar 3020 an einen anderen GPC 3018 geleitet oder in dem Speicher 3004 gespeichert werden. In mindestens einer Ausführungsform können die Ergebnisse über die Partitionseinheiten 3022, die eine Speicherschnittstelle zum Lesen und Schreiben von Daten in den/aus dem Speicher 3004 implementieren, in den Speicher 3004 geschrieben werden. In mindestens einer Ausführungsform können die Ergebnisse über die Hochgeschwindigkeits-GPU-Zusammenschaltung 3008 an eine andere PPU 3004 oder CPU übertragen werden. In mindestens einer Ausführungsform beinhaltet die PPU 3000 ohne Einschränkung eine Anzahl U von Partitionseinheiten 3022, die gleich der Anzahl von separaten und unterschiedlichen Speichervorrichtungen 3004 ist, die mit der PPU 3000 verbunden sind. In mindestens einer Ausführungsform wird die Partitionseinheit 3022 nachstehend in Verbindung mit **Fig. 32** detaillierter beschrieben.

[0310] In mindestens einer Ausführungsform führt ein Host-Prozessor einen Treiberkernel aus, der eine Anwendungsprogrammierschnittstelle („API“) implementiert, die es einer oder mehreren auf dem Host-Prozessor ausgeführten Anwendungen ermöglicht, Operationen zur Ausführung auf der PPU 3000 zu planen. In mindestens einer Ausführungsform werden mehrere Rechenanwendungen simultan durch die PPU 3000 ausgeführt und die PPU 3000 stellt Isolierung, Dienstgüte (quality of service - „QoS“) und unabhängige Adressräume für mehrere Rechenanwendungen bereit. In mindestens einer Ausführungsform erzeugt eine Anwendung Anweisungen (z. B. in Form von API-Aufrufen), die den Treiberkernel dazu veranlassen, eine oder mehrere Tasks zur Ausführung durch die PPU 3000 zu erzeugen, und der Treiberkernel gibt Tasks an einen oder mehrere Datenströme aus, die von der PPU 3000 verarbeitet werden. In mindestens einer Ausführungsform umfasst jede Aufgabe eine oder mehrere Gruppen von verwandten Threads, die als Warp bezeichnet werden können. In mindestens einer Ausführungsform umfasst ein Warp eine Vielzahl von zugehörigen Threads (z. B. 32 Threads), die parallel ausgeführt werden können. In mindestens einer Ausführungsform können sich kooperierende Threads auf eine Vielzahl von Threads beziehen, die Anweisungen zum Durchführen von Aufgaben beinhalten und die Daten durch einen gemeinsam genutzten Speicher austauschen. In mindestens einer Ausführungsform werden Threads und kooperierende Threads gemäß mindestens einer Ausführungsform in Verbindung mit **Fig. 32**.

[0311] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Infe-

renz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform wird der Deep-Learning-Anwendungsprozessor verwendet, um ein Modell für maschinelles Lernen, wie etwa ein neuronales Netz, zu trainieren, der PPU 3000 bereitgestellte Informationen vorherzusagen oder abzuleiten. In mindestens einer Ausführungsform wird die PPU 3000 verwendet, um Informationen auf der Grundlage eines trainierten Models des maschinellen Lernens (z. B. eines neuronalen Netzes) abzuleiten oder vorherzusagen, das von einem anderen Prozessor oder System oder von der PPU 3000 trainiert wurde. In mindestens einer Ausführungsform kann die PPU 3000 verwendet werden, um einen oder mehrere der in dieser Schrift beschriebenen Anwendungsfälle für neuronale Netze durchzuführen.

[0312] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0313] Fig. 31 veranschaulicht ein Universalverarbeitungscluster („GPC“)3100 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform handelt es sich bei dem GPC 3100 um den GPC 3018 aus **Fig. 30**. In mindestens einer Ausführungsform beinhaltet jeder GPC 3100 ohne Einschränkung eine Anzahl von Hardware-Einheiten zur Verarbeitung von Aufgaben und beinhaltet jeder GPC 3100 ohne Einschränkung einen Pipeline-Manager 3102, eine Vor-Rasteroperationseinheit (pre-raster operations unit - „PROP“-Einheit) 3104, eine Raster-Engine 3108, eine Arbeitsverteilungskreuzschiene (work distribution crossbar - „WDX“) 3116, eine Speicherverwaltungseinheit („MMU“) 3118, ein oder mehrere Datenverarbeitungscluster (Data Processing Clusters - „DPCs“) 3106 und eine beliebige geeignete Kombination von Teilen.

[0314] In mindestens einer Ausführungsform wird der Betrieb des GPC 3100 durch den Pipeline-Manager 3102 gesteuert. In mindestens einer Ausführungsform verwaltet der Pipeline-Manager 3102 die Konfiguration eines oder mehrerer DPCs 3106 für die Verarbeitung von Aufgaben, die dem GPC 3100 zugewiesen sind. In mindestens einer Ausführungsform konfiguriert der Pipeline-Manager 3102 mindestens einen von einem oder mehreren DPCs 3106 dazu, mindestens einen Abschnitt einer Grafik-Rendering-Pipeline zu implementieren. In mindestens einer Ausführungsform ist der DPC 3106 dazu konfiguriert, ein Vertex-Shader-Programm auf einem programmierbaren Streaming-Multiprozessor (streaming multi-processor - „SM“) 3114 auszuführen. In mindestens einer Ausführungsform ist der Pipeline-Manager 3102 dazu konfiguriert, die von einer Arbeitsverteilungseinheit empfangenen Pakete in mindestens einer Ausführungsform an die entsprechenden logischen Einheiten innerhalb des GPC 3100 zu leiten und einige Pakete können an Festfunktions-Hardwareeinheiten in der PROP 3104 und/oder die Raster-Engine 3108 geleitet werden, während andere Pakete an die DPCs 3106 zur Verarbeitung durch eine Primitiv-Engine 3112 oder den SM 3114 geleitet werden können. In mindestens einer Ausführungsform konfiguriert der Pipeline-Manager 3102 mindestens einen der DPCs 3106 zum Implementieren eines Modells eines neuronalen Netzes und/oder einer Rechenpipeline.

[0315] In mindestens einer Ausführungsform ist die PROP-Einheit 3104 dazu konfiguriert, in mindestens einer Ausführungsform die von der Raster-Engine 3108 und den DPCs 3106 erzeugten Daten an eine Rasteroperations-(„ROP“-)Einheit in der Partitionseinheit 3022 zu leiten, die vorstehend in Verbindung mit **Fig. 30** detaillierter beschrieben ist. In mindestens einer Ausführungsform ist die PROP-Einheit 3104 dazu konfiguriert, Optimierungen für die Farbmischung durchzuführen, Pixeldaten zu organisieren, Adressübersetzungen durchzuführen usw. In mindestens einer Ausführungsform beinhaltet die Raster-Engine 3108 ohne Einschränkung eine Anzahl von Festfunktions-Hardwareeinheiten, die dazu konfiguriert sind, verschiedene Rasteroperationen durchzuführen, und in mindestens einer Ausführungsform beinhaltet die Raster-Engine 3108 ohne Einschränkung eine Einrichtungs-Engine, eine Grobraster-Engine, eine Culling-Engine, eine Clipping-Engine, eine Feinraster-Engine, eine Kachelverschmelzungs-Engine und eine beliebige geeignete Kombination davon. In mindestens einer Ausführungsform empfängt die Einrichtungs-Engine transformierte Vertices und erzeugt Ebenengleichungen, die dem durch die Vertices definierten geometrischen Primitiv zugeordnet sind; die Ebenengleichungen werden an die Grobraster-Engine übertragen, um Abdeckungsinformationen (z. B. eine x-, y-Abdeckungsmaske für eine Kachel) für das Primitiv zu erzeugen; die Ausgabe der Grobraster-Engine wird an die Culling-Engine übertragen, in der Fragmente, die dem Primitiv zugeordnet sind und einen z-Test nicht bestehen, Culling unterzogen werden und an eine Clipping-Engine übertragen werden, in der Fragmente, die außerhalb eines Sichtkegelstumpfes liegen, Clipping unterzogen werden. In mindestens einer Ausführungsform werden Fragmente, die das Clipping und Culling überstehen, an die Feinraster-Engine übergeben, um Attribute für Pixelfragmente basierend auf Ebenengleichungen zu erzeugen, die von der Einrichtungs-Engine erzeugt werden. In mindestens einer Ausführungsform umfasst die Ausgabe der Raster-Engine 3108 Fragmente, die von einer beliebigen geeigneten Einheit, wie etwa von einem in dem DPC 3106 implementierten Fragment-Shader, verarbeitet werden.

[0316] In mindestens einer Ausführungsform umfasst jeder DPC 3106, der in dem GPC 3100 enthalten ist, ohne Einschränkung eine M-Pipe-Steuerung (M-Pipe Controller - „MPC“) 3110; eine Primitiv-Engine 3112; einen oder mehrere SMs 3114 und eine beliebige geeignete Kombination davon. In mindestens einer Ausführungsform steuert die MPC 3110 den Betrieb des DPC 3106 und leitet von dem Pipeline-Manager 3102 empfangene Pakete an die entsprechenden Einheiten in dem DPC 3106. In mindestens einer Ausführungsform werden Pakete, die einem Vertex zugeordnet sind, an die Primitiv-Engine 3112 geleitet, die dazu konfiguriert ist, Vertexattribute, die dem Vertex zugeordnet sind, aus dem Speicher abzurufen; im Gegensatz dazu können Pakete, die einem Shader-Programm zugeordnet sind, an den SM 3114 übertragen werden.

[0317] In mindestens einer Ausführungsform umfasst der SM 3114 ohne Einschränkung einen programmierbaren Streaming-Prozessor, der dazu konfiguriert ist, Aufgaben zu verarbeiten, die durch eine Anzahl von Threads dargestellt sind. In mindestens einer Ausführungsform weist der SM 3114 mehrere Threads auf und ist dazu konfiguriert, eine Vielzahl von Threads (z.B. 32 Threads) aus einer konkreten Gruppe von Threads gleichzeitig auszuführen und eine Single-Instruction-Multiple-Data-(„SIMD“-)Architektur zu implementieren, bei der jeder Thread in einer Gruppe von Threads (z. B. ein Warp) dazu konfiguriert ist, einen anderen Datensatz auf Grundlage des gleichen Anweisungssatzes zu verarbeiten. In mindestens einer Ausführungsform führen alle Threads in einer Gruppe von Threads dieselben Anweisungen aus. In mindestens einer Ausführungsform implementiert der SM 3114 eine Single-Instruction-Multiple-Thread-(„SIMT“-)Architektur, bei der jeder Thread in einer Gruppe von Threads dazu konfiguriert ist, einen anderen Datensatz auf Grundlage desselben Anweisungssatzes zu verarbeiten, wobei jedoch die einzelnen Threads in der Gruppe von Threads während der Ausführung divergieren dürfen. In mindestens einer Ausführungsform werden ein Programmzähler, ein Aufrufstapel und ein Ausführungszustand für jeden Warp beibehalten, wodurch Nebenläufigkeit zwischen Warps und serielle Ausführung innerhalb von Warps ermöglicht wird, wenn Threads innerhalb eines Warps divergieren. In einer anderen Ausführungsform werden ein Programmzähler, ein Aufrufstapel und ein Ausführungszustand für jeden einzelnen Thread beibehalten, wodurch gleichwertige Nebenläufigkeit zwischen allen Threads, innerhalb und zwischen Warps, ermöglicht wird. In mindestens einer Ausführungsform wird der Ausführungszustand für jeden einzelnen Thread beibehalten und können Threads, die dieselben Anweisungen ausführen, zur besseren Effizienz konvergieren und parallel ausgeführt werden. Mindestens eine Ausführungsform des SM 3114 wird nachstehend detaillierter beschrieben.

[0318] In mindestens einer Ausführungsform stellt die MMU 3118 eine Schnittstelle zwischen dem GPC 3100 und der Speicherpartitionseinheit (z. B. der Partitionseinheit 3022 aus **Fig. 30**) bereit und stellt die MMU 3118 die Übersetzung virtueller Adressen in physische Adressen, den Speicherschutz und die Vermittlung von Speicheranforderungen bereit. In mindestens einer Ausführungsform stellt die MMU 3118 einen oder mehrere Adressübersetzungspuffer („TLBs“) zum Durchführen der Übersetzung von virtuellen Adressen in physische Adressen in Speicher bereit.

[0319] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform wird der Deep-Learning-Anwendungsprozessor verwendet, um ein Modell für maschinelles Lernen, wie etwa ein neuronales Netz, zu trainieren, dem GPC 3100 bereitgestellte Informationen vorherzusagen oder abzuleiten. In mindestens einer Ausführungsform wird der GPC 3100 verwendet, um Informationen auf Grundlage eines trainierten Modells des maschinellen Lernens (z. B. eines neuronalen Netzes) abzuleiten oder vorherzusagen, das durch einen anderen Prozessor oder ein anderes System oder durch den GPC 3100 trainiert wurde. In mindestens einer Ausführungsform kann der GPC 3100 verwendet werden, um einen oder mehrere der in dieser Schrift beschriebenen Anwendungsfälle für neuronale Netze durchzuführen.

[0320] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0321] **Fig. 32** veranschaulicht eine Speicherpartitionseinheit 3200 einer Parallelverarbeitungseinheit („PPU“) gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform beinhaltet die Speicherpartitionseinheit 3200 ohne Einschränkung eine Rasteroperations-(„ROP“-)Einheit 3202, einen Cache 3204 der Ebene zwei („L2“-), eine Speicherschnittstelle 3206 und eine beliebige geeignete Kombination davon. In mindestens einer Ausführungsform ist die Speicherschnittstelle 3206 an Speicher gekoppelt. In mindestens einer Ausführungsform kann die Speicherschnittstelle 3206 32-, 64-, 128-, 1024-Bit-Datenbusse oder ähnliche Implementierungen für Hochgeschwindigkeitsdatenübertragung implementieren. In min-

destens einer Ausführungsform beinhaltet die PPU U Speicherschnittstellen 3206, eine Speicherschnittstelle 3206 pro Paar von Partitionseinheiten 3200, wobei jedes Paar von Partitionseinheiten 3200 mit einer entsprechenden Speichervorrichtung verbunden ist. In mindestens einer Ausführungsform kann die PPU zum Beispiel mit bis zu Y Speichervorrichtungen verbunden sein, wie etwa mit Speicherstapeln mit hoher Bandbreite oder mit einem synchronen dynamischen Direktzugriffsspeicher mit Grafik-Double-Data-Rate, Version 5 („GDDR5-SDRAM“).

[0322] In mindestens einer Ausführungsform implementiert die Speicherschnittstelle 3206 eine Speicherschnittstelle mit Speicher mit hoher Bandbreite der zweiten Generation (high bandwidth memory second generation - „HBM2“) und ist Y gleich der Hälfte von U. In mindestens einer Ausführungsform befinden sich HBM2-Speicherstapel in demselben physischen Gehäuse wie die PPU, was im Vergleich zu herkömmlichen GDDR5-SDRAM-Systemen erhebliche Leistungs- und Flächeneinsparungen bereitstellt. In mindestens einer Ausführungsform beinhaltet jeder HBM2-Stapel ohne Einschränkung vier Speicherchips und ist Y gleich 4, wobei jeder HBM2-Stapel zwei 128-Bit-Kanäle pro Chip für insgesamt 8 Kanäle und eine Datenbusbreite von 1024 Bit beinhaltet. In mindestens einer Ausführungsform unterstützt der Speicher Single-Error-Correcting-Double-Error-Detecting-(„SECDED“-)Fehlerkorrekturcode (Error Correction Code - „ECC“) zum Schützen von Daten. In mindestens einer Ausführungsform stellt der ECC eine höhere Zuverlässigkeit für Rechenanwendungen bereit, die empfindlich auf Datenkorruption reagieren.

[0323] In mindestens einer Ausführungsform implementiert die PPU eine Speicherhierarchie mit mehreren Ebenen. In mindestens einer Ausführungsform unterstützt die Speicherpartitionseinheit 3200 einen vereinheitlichten Speicher, um einen einzelnen vereinheitlichten virtuellen Adressraum für den Speicher der zentralen Verarbeitungseinheit („CPU“) und der PPU bereitzustellen, wodurch die gemeinsame Nutzung von Daten zwischen virtuellen Speichersystemen ermöglicht wird. In mindestens einer Ausführungsform wird die Häufigkeit von Zugriffen von einer PPU auf Speicher, der sich auf anderen Prozessoren befindet, verfolgt, um sicherzustellen, dass Speicherseiten in den physischen Speicher der PPU verschoben werden, die häufiger auf Seiten zugreift. In mindestens einer Ausführungsform unterstützt die Hochgeschwindigkeits-GPU-Zusammenschaltung 3008 Adressübersetzungsdienste, die es der PPU ermöglichen, direkt auf die Seitentabellen der CPU zuzugreifen, und der PPU vollen Zugriff auf den CPU-Speicher bereitstellen.

[0324] In mindestens einer Ausführungsform übertragen Kopier-Engines Daten zwischen mehreren PPUs oder zwischen PPUs und CPUs. In mindestens einer Ausführungsform können Kopier-Engines Seitenfehler für Adressen erzeugen, die nicht in Seitentabellen zugeordnet sind, und die Speicherpartitionseinheit 3200 bedient dann Seitenfehler, indem sie die Adressen in der Seitentabelle zuordnet, woraufhin die Kopier-Engine die Übertragung durchführt. In mindestens einer Ausführungsform ist der Speicher für mehrere Kopier-Engine-Operationen zwischen mehreren Prozessoren gepinnt (d. h. nicht auslagerbar), was den verfügbaren Speicher erheblich reduziert. In mindestens einer Ausführungsform können bei Hardware-Seitenfehlern Adressen an Kopier-Engines übergeben werden, ohne Rücksicht darauf, ob Speicherseiten speicherresident sind und der Kopierprozess transparent ist.

[0325] Daten aus dem Speicher 3004 aus **Fig. 30** oder einem anderen Systemspeicher werden gemäß mindestens einer Ausführungsform durch die Speicherpartitionseinheit 3200 abgerufen und in L2-Cache 3204 gespeichert, der sich auf dem Chip befindet und von verschiedenen GPCs gemeinsam genutzt wird. Jede Speicherpartitionseinheit 3200 beinhaltet in mindestens einer Ausführungsform ohne Einschränkung mindestens einen Abschnitt des L2-Caches, der einer entsprechenden Speichervorrichtung zugeordnet ist. In mindestens einer Ausführungsform sind die Caches der unteren Ebene in verschiedenen Einheiten innerhalb der GPCs implementiert. In mindestens einer Ausführungsform kann jeder der SMs 3114 einen Cache der Ebene eins („L1“) implementieren, wobei der L1-Cache ein privater Speicher ist, der für einen konkreten SM 3114 dediziert ist, und Daten aus dem L2-Cache 3204 abgerufen und in jedem der LI-Caches zur Verarbeitung in funktionellen Einheiten der SMs 3114 gespeichert werden. In mindestens einer Ausführungsform ist der L2-Cache 3204 an die Speicherschnittstelle 3206 und die XBar 3020 gekoppelt.

[0326] In mindestens einer Ausführungsform führt die ROP-Einheit 3202 Grafikrasteroperationen durch, die mit der Pixelfarbe in Bezug stehen, wie etwa Farbkomprimierung, Pixel-Blending und andere. In mindestens einer Ausführungsform implementiert die ROP-Einheit 3202 die Tiefenprüfung in Verbindung mit der Raster-Engine 3108, wobei sie eine Tiefe für eine Abtastposition, die einem Pixelfragment zugeordnet ist, von der Culling-Engine der Raster-Engine 3108 empfängt. In mindestens einer Ausführungsform wird die Tiefe gegen eine entsprechende Tiefe in einem Tiefenpuffer für eine dem Fragment zugeordnete Abtastposition geprüft. In mindestens einer Ausführungsform, wenn das Fragment die Tiefenprüfung für die Abtastposition besteht, aktualisiert die ROP-Einheit 3202 dann den Tiefenpuffer und überträgt ein Ergebnis der Tiefenprü-

fung an die Raster-Engine 3108. Es versteht sich, dass sich die Anzahl der Partitionseinheiten 3200 von der Anzahl der GPCs unterscheiden kann, und daher kann jede ROP-Einheit 3202 in mindestens einer Ausführungsform mit jedem der GPCs gekoppelt sein. In mindestens einer Ausführungsform verfolgt die ROP-Einheit 3202 die von verschiedenen GPCs empfangenen Pakete und bestimmt, ob ein durch die ROP-Einheit 3202 erzeugtes Ergebnis an die XBar 3020 geleitet werden soll.

[0327] Fig. 33 veranschaulicht einen Streaming-Multiprozessor („SM“) 3300 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist der SM 3300 der SM 3114 aus **Fig. 31**. In mindestens einer Ausführungsform beinhaltet der SM 3300 ohne Einschränkung einen Anweisungs-Cache 3302; eine oder mehrere Planer-Einheiten 3304; eine Registerdatei 3308; einen oder mehrere Verarbeitungskerne („Kerne“) 3310; eine oder mehrere Spezialfunktionseinheiten (special function units - „SFUs“) 3312; eine oder mehrere Lade-/Speichereinheiten (load/store units - „LSUs“) 3314; ein Zusammenschaltungsnetz 3316; einen gemeinsam genutzten Speicher/Cache der Ebene eins („L1“) 3318 und eine beliebige geeignete Kombination davon. In mindestens einer Ausführungsform teilt eine Arbeitsverteilungseinheit Aufgaben zur Ausführung auf Universalverarbeitungsclustern („GPCs“) von Parallelverarbeitungseinheiten („PPUs“) zu und wird jede Aufgabe einem konkreten Datenverarbeitungscluster („DPC“) innerhalb eines GPCs zugewiesen und wird, wenn die Aufgabe einem Shader-Programm zugeordnet ist, die Aufgabe einem der SMs 3300 zugewiesen. In mindestens einer Ausführungsform empfängt die Planer-Einheit 3304 Aufgaben von der Arbeitsverteilungseinheit und verwaltet die Anweisungsplanung für einen oder mehrere Thread-Blöcke, die dem SM 3300 zugewiesen sind. In mindestens einer Ausführungsform plant die Planer-Einheit 3304 Thread-Blöcke für die Ausführung als Warps aus parallelen Threads, wobei jedem Thread-Block mindestens ein Warp zugewiesen ist. In mindestens einer Ausführungsform führt jeder Warp Threads aus. In mindestens einer Ausführungsform verwaltet die Planer-Einheit 3304 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 3310, SFUs 3312 und LSUs 3314) zuteilt.

[0328] In mindestens einer Ausführungsform können sich kooperative Gruppen auf ein Programmiermodell zum Organisieren von Gruppen kommunizierender Threads beziehen, das es Entwicklern erlaubt, die Granularität auszudrücken, mit der Threads kommunizieren, und so den Ausdruck satterer, effizienterer paralleler Dekompositionen ermöglicht. In mindestens einer Ausführungsform unterstützen kooperative Launch-APIs die Synchronisation zwischen Thread-Blöcken zur Ausführung paralleler Algorithmen. In mindestens einer Ausführungsform bieten Anwendungen herkömmlicher Programmiermodelle ein einzelnes, einfaches Konstrukt für die Synchronisation kooperierender Threads: eine Sperre über alle Threads eines Thread-Blocks (z. B. die Funktion syncthreads()). In mindestens einer Ausführungsform können Programmierer jedoch Gruppen von Threads mit einer geringeren als Thread-Block-Granularität definieren und innerhalb definierter Gruppen synchronisieren, um eine höhere Rechenleistung, Ausgestaltungsflexibilität und Software-Wiederverwendung in Form von kollektiven gruppenweiten Funktionsschnittstellen zu ermöglichen. In mindestens einer Ausführungsform ermöglichen es kooperative Gruppen Programmierern, Gruppen von Threads explizit mit Teilblock- (d. h. so klein wie ein einzelner Thread) und Mehrblock-Granularität zu definieren und kollektive Operationen, wie etwa Synchronisation, an Threads in einer kooperativen Gruppe durchzuführen. In mindestens einer Ausführungsform unterstützt das Programmiermodell eine saubere Komposition über Software-Grenzen hinweg, sodass sich Bibliotheken und Nutzenfunktionen innerhalb ihres lokalen Kontexts sicher synchronisieren können, ohne Annahmen über Konvergenz treffen zu müssen. In mindestens einer Ausführungsform ermöglichen Primitive der kooperativen Gruppen neue Muster der kooperativen Parallelität, einschließlich, aber nicht beschränkt auf, Produzent-Konsument-Parallelität, opportunistischer Parallelität und globaler Synchronisation über ein gesamtes Gitter von Thread-Blöcken.

[0329] In mindestens einer Ausführungsform ist eine Zuteilungseinheit 3306 dazu konfiguriert, Anweisungen an eine oder mehrere der funktionellen Einheiten zu übertragen und die Planer-Einheit 3304 beinhaltet ohne Einschränkung zwei Zuteilungseinheiten 3306, die ermöglichen, dass zwei unterschiedliche Anweisungen aus demselben Warp während jedes Taktzyklus versendet werden. In mindestens einer Ausführungsform beinhaltet jede Planer-Einheit 3304 eine einzelne Zuteilungseinheit 3306 oder zusätzliche Zuteilungseinheiten 3306.

[0330] In mindestens einer Ausführungsform beinhaltet jeder SM 3300 in mindestens einer Ausführungsform ohne Einschränkung die Registerdatei 3308, die einen Satz von Registern für funktionelle Einheiten des SM 3300 bereitstellt. In mindestens einer Ausführungsform ist die Registerdatei 3308 derart zwischen den einzelnen funktionellen Einheiten aufgeteilt, dass jeder funktionellen Einheit ein dedizierter Abschnitt der Registerdatei 3308 zugewiesen ist. In mindestens einer Ausführungsform ist die Registerdatei 3308 auf unterschiedli-

che Warps aufgeteilt, die durch den SM 3300 ausgeführt werden, und die Registerdatei 3308 stellt temporären Datenspeicher für Operanden bereit, die mit Datenpfaden von funktionellen Einheiten verbunden sind. In mindestens einer Ausführungsform umfasst jeder SM 3300 ohne Einschränkung eine Vielzahl von L Verarbeitungskernen 3310. In mindestens einer Ausführungsform beinhaltet der SM 3300 ohne Einschränkung eine große Anzahl (z. B. 128 oder mehr) von unterschiedlichen Verarbeitungskernen 3310. In mindestens einer Ausführungsform beinhaltet jeder Verarbeitungskern 3310 in mindestens einer Ausführungsform ohne Einschränkung eine Vollpipelineverarbeitungseinheit mit einfacher Genauigkeit, doppelter Genauigkeit und/oder gemischter Genauigkeit, die ohne Einschränkung eine arithmetisch-logische Einheit für Gleitkommazahlen und eine arithmetisch-logische Einheit für Integer beinhaltet. In mindestens einer Ausführungsform implementieren die arithmetisch-logischen Einheiten für Gleitkommazahlen den Standard IEEE 754-2008 für Gleitkommaarithmetik. In mindestens einer Ausführungsform beinhalten die Verarbeitungskerne 3310 ohne Einschränkung 64 Gleitkommakeme mit einfacher Genauigkeit (32 Bit), 64 Integerkerne, 32 Gleitkommakeme mit doppelter Genauigkeit (64 Bit) und 8 Tensorkerne.

[0331] Tensorkerne sind gemäß mindestens einer Ausführungsform dazu konfiguriert, Matrixoperationen durchzuführen. In mindestens einer Ausführungsform sind ein oder mehrere Tensorkerne in den Verarbeitungskernen 3310 enthalten. In mindestens einer Ausführungsform sind die Tensorkerne dazu konfiguriert, Deep-Learning-Matrixarithmetik durchzuführen, wie etwa Faltungsoperationen zum Trainieren und Inferenzieren neuronaler Netze. In mindestens einer Ausführungsform arbeitet jeder Tensorkem an einer 4x4-Matrix und er führt eine Matrixmultiplikations- und -akkumulationsoperation D = A X B + C durch, wobei A, B, C und D 4x4-Matrizen sind.

[0332] In mindestens einer Ausführungsform sind die Matrixmultiplikationseingaben A und B 16-Bit-Gleitkommamatrizen und sind die Akkumulationsmatrizen C und D 16-Bit-Gleitkomma- oder 32-Bit-Gleitkommamatrizen. In mindestens einer Ausführungsform arbeiten die Tensorkerne an 16-Bit-Gleitkommaeingabedaten mit 32-Bit-Gleitkommaakkumulation. In mindestens einer Ausführungsform verwendet die 16-Bit-Gleitkommamultiplikation 64 Operationen und ergibt ein Produkt mit voller Genauigkeit, das dann unter Verwendung von 32-Bit-Gleitkommaaddition mit anderen Zwischenprodukten für eine 4x4x4-Matrixmultiplikation akkumuliert wird. In mindestens einer Ausführungsform werden die Tensorkerne verwendet, um viel größere zweidimensionale oder höherdimensionale Matrixoperationen durchzuführen, die aus diesen kleineren Elementen aufgebaut sind. In mindestens einer Ausführungsform macht eine API, wie etwa eine CUDA 9 C++ API, spezialisierte Matrixlade-, Matrixmultiplikations- und -akkumulations- sowie Matrixspeicheroperationen verfügbar, um die Tensorkerne anhand eines Programms mit CUDA-C++ effizient zu verwenden. In mindestens einer Ausführungsform nimmt die Schnittstelle auf Warp-Ebene auf der CUDA-Ebene Matrizen der Größe 16x16 an, die sich über alle 32 Threads des Warp erstrecken.

[0333] In mindestens einer Ausführungsform umfasst jeder SM 3300 ohne Einschränkung M SFUs 3312, die Spezialfunktionen durchführen (z. B. Attributauswertung, reziproke Quadratwurzel usw.). In mindestens einer Ausführungsform beinhalten die SFUs 3312 ohne Einschränkung eine Baumtraversierungseinheit, die dazu konfiguriert ist, eine hierarchische Baumdatenstruktur zu traversieren. In mindestens einer Ausführungsform beinhalten die SFUs 3312 ohne Einschränkung eine Textureinheit, die dazu konfiguriert ist, Texturkartenfilteroperationen durchzuführen. In mindestens einer Ausführungsform sind die Textureinheiten dazu konfiguriert, Texturkarten (z. B. ein 2D-Array von Texeln) aus Speicher zu laden und Texturkarten abzutasten, um abgetastete Texturwerte zur Verwendung in Shader-Programmen zu erzeugen, die durch den SM 3300 ausgeführt werden. In mindestens einer Ausführungsform werden die Texturkarten in dem gemeinsam genutzten Speicher/L1 -Cache 3318 gespeichert. In mindestens einer Ausführungsform implementieren die Textureinheiten Texturoperationen, wie etwa Filteroperationen unter Verwendung von Mip-Karten (z.B. Texturkarten mit variierenden Detailgraden), gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform beinhaltet jeder SM 3300 ohne Einschränkung zwei Textureinheiten.

[0334] Jeder SM 3300 umfasst in mindestens einer Ausführungsform ohne Einschränkung N LSUs 3314, die Lade- und Speicheroperationen zwischen dem gemeinsam genutzten Speicher/L1-Cache 3318 und der Registerdatei 3308 implementieren. Jeder SM 3300 beinhaltet in mindestens einer Ausführungsform ohne Einschränkung ein Zusammenschaltungsnetz 3316, das jede der funktionellen Einheiten mit der Registerdatei 3308 und der LSU 3314 mit der Registerdatei 3308 und dem gemeinsam genutzten Speicher/L1-Cache 3318 verbindet. In mindestens einer Ausführungsform ist das Zusammenschaltungsnetz 3316 eine Kreuzschiene, die dazu konfiguriert sein kann, beliebige funktionelle Einheiten mit beliebigen Registern in der Registerdatei 3308 zu verbinden und LSUs 3314 mit der Registerdatei 3308 und Speicherorten im gemeinsam genutzten Speicher/L1-Cache 3318 zu verbinden.

[0335] In mindestens einer Ausführungsform ist der gemeinsam genutzte Speicher/L1-Cache 3318 ein Array von chipinternem Speicher, das in mindestens einer Ausführungsform die Datenspeicherung und die Kommunikation zwischen dem SM 3300 und der Primitiv-Engine sowie zwischen Threads in dem SM 3300 ermöglicht. In mindestens einer Ausführungsform umfasst der gemeinsam genutzte Speicher/L1-Cache 3318 ohne Einschränkung eine Speicherkapazität von 128 KB und befindet sich in einem Pfad von dem SM 3300 zu einer Partitionseinheit. In mindestens einer Ausführungsform wird der gemeinsam genutzte Speicher/L1-Cache 3318 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 3318, L2-Cache und Speicher Ergänzungsspeicher.

[0336] Das Kombinieren der Funktionalität des Daten-Cache und des gemeinsam genutzten Speichers in einem einzelnen Speicherblock stellt in mindestens einer Ausführungsform eine verbesserte Rechenleistung für beide Typen von Speicherzugriffen bereit. In mindestens einer Ausführungsform wird die Kapazität durch Programme, die keinen gemeinsam genutzten Speicher verwenden, als Cache verwendet oder sie kann so verwendet werden, wie etwa, falls der gemeinsam genutzte Speicher dazu konfiguriert ist, die Hälfte einer Kapazität verwendet Textur- und Lade-/Speicheroperationen die verbleibende Kapazität verwenden können. Die Integration in den gemeinsam genutzten Speicher/L1-Cache 3318 ermöglicht dem gemeinsam genutzten Speicher/L1-Cache 3318 gemäß mindestens einer Ausführungsform als Leitung mit hohem Durchsatz für Streaming-Daten zu fungieren, während simultan Zugriff mit hoher Bandbreite und niedriger Latenz auf häufig wiederverwendete Daten bereitgestellt wird. In mindestens einer Ausführungsform kann bei einer Konfiguration für Universalparallelberechnung eine einfachere Konfiguration im Vergleich zur Grafikverarbeitung verwendet werden. In mindestens einer Ausführungsform werden Festfunktions-Grafikverarbeitungseinheiten umgangen, wodurch ein wesentlich einfacheres Programmiermodell entsteht. Bei der Konfiguration für Universalparallelberechnungen weist die Arbeitsverteilungseinheit in mindestens einer Ausführungsform Blöcke von Threads direkt DPCs zu und verteilt sie an diese. In mindestens einer Ausführungsform führen Threads in einem Block dasselbe Programm aus, wobei eine eindeutige Thread-ID bei der Berechnung verwendet wird, um sicherzustellen, dass jeder Thread eindeutige Ergebnisse erzeugt, wobei der SM 3300 zur Ausführung des Programms und zur Durchführung von Berechnungen, der gemeinsam genutzte Speicher/L1-Cache 3318 zur Kommunikation zwischen den Threads und die LSU 3314 zum Lesen des und Schreiben in den globalen Speicher durch den gemeinsam genutzten Speicher/L1-Cache 3318 und die Speicherpartitionseinheit verwendet werden. Bei der Konfiguration für Universalparallelberechnungen schreibt der SM 3300 in mindestens einer Ausführungsform Befehle, die durch die Planer-Einheit 3304 verwendet werden können, um neue Arbeit in den DPCs zu starten.

[0337] In mindestens einer Ausführungsform ist die PPU in einem Desktop-Computer, einem Laptop-Computer, einem Tablet-Computer, Servern, Supercomputern, einem Smartphone (z. B. einer drahtlosen tragbaren Vorrichtung), einem persönlichen digitalen Assistenten („PDA“), einer Digitalkamera, einem Fahrzeug, einer am Kopf befestigten Anzeige, einer elektronischen tragbaren Vorrichtung und anderen enthalten oder damit gekoppelt. In mindestens einer Ausführungsform ist die PPU auf einem einzelnen Halbleitersubstrat verkörpert. In mindestens einer Ausführungsform ist die PPU in einem System auf einem Chip („SoC“) zusammen mit einer oder mehreren anderen Vorrichtungen enthalten, wie etwa zusätzlichen PPUs, Speicher, einer Reduced-Instruction-Set-Computer-(„RISC“-)CPU, einer Speicherverwaltungseinheit („MMU“), einem Digital-Analog-Wandler (digital-to-analog converter - „DAC“) und dergleichen.

[0338] In mindestens einer Ausführungsform kann die PPU auf einer Grafikkarte enthalten sein, die eine oder mehrere Speichervorrichtungen beinhaltet. Eine Grafikkarte kann dazu konfiguriert sein, mit einem PCIe-Steckplatz auf einer Hauptplatine eines Desktop-Computers eine Schnittstelle zu bilden. In mindestens einer Ausführungsform kann die PPU eine integrierte Grafikverarbeitungseinheit („iGPU“) sein, die im Chipsatz der Hauptplatine enthalten ist.

[0339] Die Inferenz- und/oder Trainingslogik 915 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Details bezüglich der Inferenz- und/oder Trainingslogik 915 sind nachstehend in Verbindung mit den **Fig. 9A** und/oder 9B bereitgestellt. In mindestens einer Ausführungsform wird der Deep-Learning-Anwendungsprozessor verwendet, um ein Modell für maschinelles Lernen, wie etwa ein neuronales Netz, zu trainieren, dem SM 3300 bereitgestellte Informationen vorherzusagen oder abzuleiten. In mindestens einer Ausführungsform wird der SM 3300 verwendet, um Informationen auf Grundlage eines trainierten Modells für maschinelles Lernen (z. B. eines neuronalen Netzes) abzuleiten oder vorherzusagen, das durch einen anderen Prozessor oder ein anderes System oder durch den SM 3300 trainiert wurde. In mindestens einer Ausführungsform kann der SM 3300

verwendet werden, um einen oder mehrere der in dieser Schrift beschriebenen Anwendungsfälle für neuronale Netze durchzuführen.

[0340] In mindestens einer Ausführungsform können derartige Komponenten verwendet werden, um unter Verwendung eines oder mehrerer neuronaler Netze ein verbessertes Video zu erzeugen, wie etwa um ein Video mit höherer Bildrate aus Bildern eines Videos mit niedrigerer Bildrate zu erzeugen.

[0341] In mindestens einer Ausführungsform kann sich eine einzelne Halbleiterplattform auf eine einzige einheitliche halbleiterbasierte integrierte Schaltung oder einen solchen Chip beziehen. In mindestens einer Ausführungsform können Mehrchipmodule mit erhöhter Konnektivität verwendet werden, die den chipinternen Betrieb simulieren und wesentliche Verbesserungen gegenüber der Verwendung einer herkömmlichen zentralen Verarbeitungseinheit („CPU“) und einer Busimplementierung bieten. In mindestens einer Ausführungsform können verschiedene Module je nach Wunsch des Benutzers auch separat oder in verschiedenen Kombinationen von Halbleiterplattformen untergebracht sein.

[0342] In mindestens einer Ausführungsform sind Computerprogramme in Form von maschinenlesbarem ausführbaren Code oder Computersteuerlogik-Algorithmen im Hauptspeicher 1304 und/oder im Sekundärspeicher gespeichert. Bei Ausführung durch einen oder mehrere Prozessoren ermöglichen es Computerprogramme dem System 1300, verschiedene Funktionen gemäß mindestens einer Ausführungsform durchzuführen. In mindestens einer Ausführungsform sind der Arbeitsspeicher 1304, der Datenspeicher und/oder ein beliebiger anderer Datenspeicher mögliche Beispiele für computerlesbare Medien. In mindestens einer Ausführungsform kann sich der Sekundärspeicher auf ein(e) beliebige(s) geeignete(s) Speichervorrichtung oder -system beziehen, wie etwa ein Festplattenlaufwerk und/oder ein Wechselspeicherlaufwerk, das ein Diskettenlaufwerk, ein Magnetbandlaufwerk, ein Compact-Disk-Laufwerk, ein Digital-Versatile-Disk-(„DVD“-)Laufwerk, eine Aufzeichnungsvorrichtung, einen Universal-Serial-Bus-(„USB“-)Flash-Speicher usw. darstellt. In mindestens einer Ausführungsform werden die Architektur und/oder die Funktionalität verschiedener vorhergehender Figuren im Kontext der CPU 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 dazu ausgestaltet ist, als Einheit zur Durchführung zugehöriger Funktionen zu arbeiten und verkauft zu werden usw.); und einer beliebigen geeigneten Kombination aus (einer) integrierten Schaltung(en) implementiert.

[0343] In mindestens einer Ausführungsform werden die Architektur und/oder die Funktionalität verschiedener vorhergehender Figuren im Kontext eines allgemeinen Computersystems, eines Leiterplattensystems, eines für Unterhaltungszwecke dedizierten Spielekonsolensystems, eines anwendungsspezifischen Systems und anderer implementiert. In mindestens einer Ausführungsform kann das Computersystem 1300 die Form eines Desktop-Computers, eines Laptop-Computers, eines Tablet-Computers, von Servern, von Supercomputern, eines Smartphones (z. B. einer drahtlosen tragbaren Vorrichtung), eines persönlichen digitalen Assistenten („PDA“), einer Digitalkamera, eines Fahrzeugs, einer am Kopf befestigten Anzeige, einer tragbaren elektronischen Vorrichtung, einer Mobiltelefonvorrichtung, eines Fernsehgeräts, einer Arbeitsstation, von Spielekonsolen, eines eingebetteten Systems und/oder eines beliebigen anderen Typs von Logik annehmen.

[0344] In mindestens einer Ausführungsform beinhaltet das Parallelverarbeitungssystem 1312 ohne Einschränkung eine Vielzahl von Parallelverarbeitungseinheiten („PPUs“) 1314 und zugeordneten Speichern 1316. In mindestens einer Ausführungsform sind die PPUs 1314 über eine Zusammenschaltung 1318 und einen Switch 1320 oder Multiplexer mit einem Host-Prozessor oder anderen Peripherievorrichtungen verbunden. In mindestens einer Ausführungsform verteilt das Parallelverarbeitungssystem 1312 Rechenaufgaben auf PPUs 1314, die parallelisierbar sein können - zum Beispiel als Teil der Verteilung von Rechenaufgaben auf mehrere Thread-Blöcke einer Grafikverarbeitungseinheit („GPU“). In mindestens einer Ausführungsform wird der Speicher von einigen oder allen PPUs 1314 gemeinsam genutzt und ist er für diese zugänglich (z. B. für Lese- und/oder Schreibzugriff), obwohl ein derartiger gemeinsam genutzter Speicher Einbußen bei der Rechenleistung in Bezug auf die Verwendung von lokalem Speicher und Registern, die in einer PPU 1314 resident sind, mit sich bringen kann. In mindestens einer Ausführungsform wird der Betrieb der PPUs 1314 durch Verwendung eines Befehls wie etwa _syncthreads() synchronisiert, wobei alle Threads in einem Block (z. B. über mehrere PPUs 1314 hinweg ausgeführt) einen bestimmten Punkt der Codeausführung erreichen müssen, bevor sie fortfahren.

[0345] Andere Variationen liegen innerhalb des Wesens der vorliegenden Offenbarung. Somit können zwar bezüglich der offenbarten Techniken diverse Modifikationen und alternative Konstruktionen vorgenommen

werden, bestimmte veranschaulichte Ausführungsformen davon werden jedoch in den Zeichnungen gezeigt und wurden vorangehend ausführlich beschrieben. Allerdings versteht es sich, dass nicht die Absicht verfolgt wird, die Offenbarung auf die konkrete(n) offenbarte(n) Form oder Formen einzuschränken, sondern die Absicht ganz im Gegenteil darin besteht, sämtliche Modifikationen, alternativen Konstruktionen und Äquivalente abzudecken, die in den Geist und Umfang der wie in den beigefügten Ansprüchen definierten Offenbarung fallen.

[0346] Die Verwendung der Ausdrücke „ein“ und „eine“ und „der/die/das“ und ähnlicher Bezeichnungen im Kontext mit der Beschreibung offenbarter Ausführungsformen (insbesondere im Kontext mit den folgenden Ansprüchen) ist so auszulegen, dass sie sowohl den Singular als auch den Plural abdeckt, sofern in dieser Schrift nicht anders angegeben oder durch den Kontext eindeutig widerlegt, und nicht als Definition eines Ausdrucks. Die Ausdrücke „umfassend“, „aufweisend“, „beinhaltend“ und „enthaltend“ sind als offene Ausdrücke auszulegen (d. h. „einschließlich, aber nicht begrenzt auf“), sofern nicht anderweitig angegeben. Der Ausdruck „verbunden“ ist als teilweise oder vollständig ineinander enthalten, aneinander befestigt oder aneinander angefügt auszulegen, wenn er unmodifiziert vorliegt und sich auf physische Verbindungen bezieht, selbst, wenn ein Element dazwischen eingefügt ist. Die Nennung von Wertebereichen in dieser Schrift soll lediglich als schnelles Verfahren des einzelnen Bezugnehmens auf jeden separaten Wert dienen, der in den Bereich fällt, es sei denn, in dieser Schrift ist etwas anderes angegeben, und jeder separate Wert ist in die Beschreibung eingeschlossen, als ob er einzeln in dieser Schrift wiedergegeben wäre. Die Verwendung des Ausdrucks „Satz“ (z. B. „ein Satz von Objekten“) oder „Teilmenge“ ist als eine nicht leere Zusammenstellung auszulegen, die ein oder mehrere Elemente umfasst, es sei denn, es ist etwas anderes angemerkt oder dies widerspricht dem Kontext. Sofern nichts anderes angegeben ist oder der Kontext dem widerspricht, bezeichnet ferner der Ausdruck „Teilmenge“ eines entsprechenden Satzes nicht notwendigerweise eine richtige Teilmenge des entsprechenden Satzes, sondern die Teilmenge und der entsprechende Satz können gleich sein.

[0347] Sofern nicht spezifisch etwas anderes genannt ist oder der Kontext dem eindeutig widerspricht, ist verbindende Sprache, wie etwa Formulierungen der Form „wenigstens eines von A, B und C“ oder „mindestens eines von A, B und C“, andernfalls in dem Kontext zu verstehen, in dem sie allgemein verwendet werden, um darzustellen, dass ein Gegenstand, ein Ausdruck usw. entweder A oder B oder C oder eine beliebige nicht leere Teilmenge des Satzes aus A und B und C sein kann. Zum Beispiel beziehen sich in dem veranschaulichenden Beispiel für einen Satz, der drei Elemente aufweist, die verbindenden Formulierungen „wenigstens eines von A, B und C“ und „mindestens eines von A, B und C“ auf einen beliebigen der folgenden Sätze: {A}, {B}, {C}, {A, B}, {A, C}, {B, C}, {A, B, C}. Somit soll solche verbindende Sprache im Allgemeinen nicht ausdrücken, dass bestimmte Ausführungen erforderlich machen, dass zumindest eines von A, zumindest eines von B und zumindest eines von C vorhanden ist. Außerdem, sofern nicht anders angemerkt oder durch den Kontext widerlegt, bezeichnet der Ausdruck „Vielzahl“ einen Zustand der Pluralität (z. B. gibt „eine Vielzahl von Objekten“ mehrere Objekte an). Bei einer Vielzahl handelt es sich um mindestens zwei Objekte, es können aber auch mehr sein, wenn dies entweder explizit oder durch den Kontext angegeben wird. Sofern nicht anders angegeben oder anderweitig aus dem Kontext ersichtlich, bedeutet „basierend auf“-mindestens teilweise basierend auf und nicht „ausschließlich basierend auf“.

[0348] In dieser Schrift beschriebene Operationen von Prozessen können in einer beliebigen geeigneten Reihenfolge durchgeführt werden, sofern in dieser Schrift nicht anders angegeben oder der Kontext dem anderweitig eindeutig widerspricht. In mindestens einer Ausführungsform wird ein Prozess, wie zum Beispiel die in dieser Schrift beschriebenen Prozesse (oder Variationen und/oder Kombinationen davon), unter der Kontrolle von einem oder mehreren Computersystemen ausgeführt, die mit ausführbaren Anweisungen konfiguriert sind, und ist als Code (z. B. ausführbare Anweisungen, ein oder mehrere Computerprogramme oder eine oder mehrere Anwendungen), die kollektiv auf einem oder mehreren Prozessoren ausgeführt werden, durch Hardware oder Kombinationen davon implementiert. In mindestens einer Ausführungsform ist der Code auf einem computerlesbaren Speichermedium gespeichert, zum Beispiel in Form eines Computerprogramms, das eine Vielzahl von Anweisungen umfasst, die durch einen oder mehrere Prozessoren ausgeführt werden können. In mindestens einer Ausführungsform ist ein computerlesbares Speichermedium ein nicht transitorisches computerlesbares Speichermedium, das transitorische Signale (z. B. eine sich ausbreitende transiente elektrische oder elektromagnetische Übertragung) ausschließt, aber nicht transitorische Datenspeicherschaltungen (z. B. Puffer, Cache und Warteschlangen) innerhalb der Transceiver von transitorischen Signalen einschließt. In mindestens einer Ausführungsform ist der Code (z. B. ausführbarer Code oder Quellcode) auf einem Satz von einem oder mehreren nicht transitorischen computerlesbaren Speichermedien gespeichert, auf denen ausführbare Anweisungen gespeichert sind (oder ein anderer Speicher zum Speichern ausführbarer Anweisungen), die bei Ausführung (d. h. als Ergebnis der Ausführung) durch einen oder mehrere Prozessoren eines Computersystems das Computersystem dazu veranlassen, in dieser Schrift

beschriebene Operationen durchzuführen. Ein Satz von nicht transitorischen computerlesbaren Speichermedien umfasst in mindestens einer Ausführungsform mehrere nicht transitorische computerlesbare Speichermedien und einem oder mehreren der einzelnen nicht transitorischen Speichermedien mehrerer nicht transitorischer computerlesbarer Speichermedien fehlt der gesamte Code, während mehrere nicht transitorische computerlesbare Speichermedien kollektiv den gesamten Code speichern. In mindestens einer Ausführungsform werden die ausführbaren Anweisungen so ausgeführt, dass unterschiedliche Anweisungen durch unterschiedliche Prozessoren ausgeführt werden - zum Beispiel speichert ein nichttransitorisches computerlesbares Speichermedium Anweisungen und eine hauptsächliche zentrale Verarbeitungseinheit („CPU“) führt einige der Anweisungen aus, während eine Grafikverarbeitungseinheit („GPU“) andere Anweisungen ausführt. In mindestens einer Ausführungsform weisen unterschiedliche Komponenten eines Computersystems separate Prozessoren auf und führen unterschiedliche Prozessoren unterschiedliche Teilmengen von Anweisungen aus.

[0349] Dementsprechend sind in mindestens einer Ausführungsform Computersysteme dazu konfiguriert, einen oder mehrere Dienste zu implementieren, die einzeln oder zusammen Operationen der in dieser Schrift beschriebenen Prozesse durchführen, und sind derartige Computersysteme mit geeigneter Hardware und/oder Software konfiguriert, die eine Durchführung der Operationen ermöglicht. Ferner ist ein Computersystem, das mindestens eine Ausführungsform der vorliegenden Offenbarung implementiert, eine einzelne Vorrichtung und in einer anderen Ausführungsform ein verteiltes Computersystem, das mehrere Vorrichtungen umfasst, die unterschiedlich arbeiten, sodass das verteilte Computersystem die in dieser Schrift beschriebenen Operationen durchführt und sodass eine einzelne Vorrichtung nicht alle Operationen durchführt.

[0350] Die Verwendung jeglicher Beispiele oder beispielhafter Wortwahl (z. B. „wie etwa“), die in dieser bereitgestellt sind, soll lediglich die Ausführungsformen der Offenbarung besser veranschaulichen und stellt keine Einschränkung des Umfangs der Offenbarung dar, es sei denn, es ist etwas anderes beansprucht. Keinerlei Wortwahl in der Beschreibung sollte so ausgelegt werden, dass sie ein beliebiges nicht beanspruchtes Element als für die Umsetzung der Offenbarung wesentlich angibt.

[0351] Jegliche Referenzen, einschließlich Veröffentlichungen, Patentanmeldungen und Patenten, die in dieser Schrift erwähnt werden, sind hiermit durch Bezugnahme in demselben Maße aufgenommen, als wäre jede Referenz einzeln und spezifisch als durch Referenz eingeschlossen angegeben und in ihrer Gesamtheit in dieser Schrift ausgeführt.

[0352] In der Beschreibung und den Ansprüchen können die Ausdrücke „gekoppelt“ und „verbunden“ sowie deren Ableitungen verwendet werden. Es versteht sich, dass diese Ausdrücke nicht als Synonyme füreinander bestimmt sein können. Vielmehr kann in bestimmten Beispielen „verbunden“ oder „gekoppelt“ verwendet werden, um anzuzeigen, dass zwei oder mehr Elemente in direktem oder indirektem physischen oder elektrischen Kontakt miteinander stehen. Mit „gekoppelt“ kann auch gemeint sein, dass zwei oder mehr Elemente nicht in direktem Kontakt miteinander stehen, jedoch trotzdem miteinander zusammenwirken oder interagieren.

[0353] Sofern nicht ausdrücklich anders angegeben, versteht es sich, dass sich Ausdrücke wie „Verarbeitung“, „Berechnung“, „Berechnen“, „Bestimmen“ oder dergleichen in der gesamten Beschreibung auf Handlungen und/oder Prozesse eines Computers oder Rechensystems oder einer ähnlichen elektronischen Rechenvorrichtung, die Daten, die als physische, wie etwa elektronische, Größen in den Registern und/oder Speichern des Rechensystems dargestellt sind, manipulieren und/oder in andere Daten umwandeln, die auf ähnliche Weise als physische Größen in den Speichern, Registern oder anderen derartigen Informationsspeicher-, -übertragungs- oder - anzeigevorrichtungen des Rechensystems dargestellt sind.

[0354] Auf ähnliche Weise kann sich der Ausdruck „Prozessor“ auf eine beliebige Vorrichtung oder einen Abschnitt einer Vorrichtung beziehen, die/der elektronische Daten aus Registern und/oder dem Speicher verarbeitet und diese elektronischen Daten in andere elektronische Daten umwandelt, die in Registern und/oder im Speicher gespeichert werden können. Als nicht einschränkende Beispiele kann der „Prozessor“ eine CPU oder eine GPU sein. Eine „Rechenplattform“ kann einen oder mehrere Prozessoren umfassen. Im vorliegenden Zusammenhang können „Software“-Prozesse zum Beispiel Software- und/oder Hardware-Entitäten beinhalten, die im Laufe der Zeit Arbeit durchführen, wie etwa Aufgaben, Threads und intelligente Agenten. Außerdem kann sich jeder Prozess auf mehrere Prozesse beziehen, um Anweisungen nacheinander oder parallel, kontinuierlich oder intermittierend auszuführen. Die Ausdrücke „System“ und „Verfahren“ werden in dieser Schrift insofern austauschbar verwendet, dass ein System ein oder mehrere Verfahren verkörpern kann und die Verfahren als System betrachtet werden können.

[0355] Im vorliegenden Dokument kann auf Erhalten, Erfassen, Empfangen oder Eingeben von analogen oder digitalen Daten in ein Teilsystem, Computersystem oder eine computerimplementierte Maschine Bezug genommen werden. Das Erhalten, Erfassen, Empfangen oder Eingeben von analogen und digitalen Daten kann auf vielfältige Weise erfolgen, wie etwa durch Empfangen von Daten als Parameter eines Funktionsaufrufs oder eines Aufrufs einer Anwendungsprogrammierschnittstelle. In einigen Implementierungen kann der Prozess des Erhaltens, Erfassens, Empfangens oder Eingebens von analogen oder digitalen Daten durch Übertragen von Daten über eine serielle oder parallele Schnittstelle erfolgen. In einer anderen Implementierung kann der Prozess des Erhaltens, Erfassens, Empfangens oder Eingebens von analogen oder digitalen Daten durch Übertragen von Daten über ein Computernetzwerk von der bereitstellenden Entität zur erfassenden Entität erfolgen. Es kann auch auf Bereitstellen, Ausgeben, Übertragen, Senden oder Darstellen von analogen oder digitalen Daten Bezug genommen werden. In verschiedenen Beispielen kann der Prozess des Bereitstellens, Ausgebens, Übertragens, Sendens oder Darstellens von analogen oder digitalen Daten durch Übertragen von Daten als Eingabe- oder Ausgabeparameter eines Funktionsaufrufs, Parameter einer Anwendungsprogrammierschnittstelle oder eines Interprozess-Kommunikationsmechanismus erfolgen.

[0356] Obwohl die vorstehende Erörterung beispielhafte Implementierung der beschriebenen Techniken darlegt, können auch andere Architekturen verwendet werden, um die beschriebene Funktionalität zu implementieren, und sie sollen im Umfang dieser Offenbarung liegen. Darüber hinaus könnten, obwohl spezifische Verteilungen von Zuständigkeiten vorstehend zum Zwecke der Erörterung definiert sind, verschiedene Funktionen und Zuständigkeiten in Abhängigkeit von den Umständen anders verteilt und aufgeteilt werden.

[0357] Obwohl der Gegenstand in für Strukturmerkmale und/oder Verfahrenshandlungen spezifischer Sprache beschrieben wurde, versteht es sich ferner, dass der in den beigefügten Ansprüchen beanspruchte Gegenstand nicht unbedingt auf die beschriebenen konkreten Merkmale oder Handlungen beschränkt ist. Vielmehr werden spezifische Merkmale und Handlungen als beispielhafte Formen zum Implementieren der Ansprüche offenbart.

Patentansprüche

1. Prozessor, umfassend:
eine oder mehrere arithmetisch-logische Einheiten (arithmetic logic units - ALUs), die dazu konfiguriert werden sollen, unter Verwendung eines oder mehrerer neuronaler Netze, die unter Verwendung einer oder mehrerer zeitlicher Posendarstellungen trainiert sind, Bilder für ein Video abzuleiten.

2. Prozessor nach Anspruch 1, wobei die Videobilder, die zum Trainieren des einen oder der mehreren neuronalen Netze verwendet werden, verwendet werden, um die eine oder die mehreren zeitlichen Posendarstellungen und eine zeitinvariante Erscheinungsbilddarstellung zu erzeugen.

3. Prozessor nach Anspruch 2, wobei Farb-Jittering und Thin-Plate-Spline-(TPS-)Verkrümmung die Erscheinungsbildinvarianz und die Lokalisierungseigenschaften für die eine oder die mehreren zeitlichen Posendarstellungen erzwingen.

4. Prozessor nach Anspruch 2, wobei die eine oder die mehreren zeitlichen Posendarstellungen Merkmale umfassen, die durch Gauß-Wärmekarten dargestellt und als Mittelwert und Kovarianz parametrisiert sind.

5. Prozessor nach Anspruch 2, wobei die eine oder die mehreren zeitlichen Posendarstellungen und die zeitinvariante Erscheinungsbilddarstellung verwendet werden, um ein Eingangsvideobild zu rekonstruieren und wobei ein Verlustwert, der sich aus dem Vergleichen des Eingangsvideobilds mit dem rekonstruierten Videobild ergibt, durch Anpassen eines oder mehrerer Netzparameter des einen oder der mehreren neuronalen Netze minimiert wird.

6. Prozessor nach Anspruch 1, wobei die Bewegung der einen oder der mehreren zeitlichen Posendarstellungen im Laufe der Zeit unter Verwendung eines zeitlichen Codierers, der ein oder mehrere Netze mit langem Kurzzeitgedächtnis (long short-term memory networks - LSTM-Netze) beinhaltet, modelliert wird.

7. Prozessor nach Anspruch 1, wobei die abgeleiteten Bilder verwendet werden, um ein Video mit einer höheren Bildrate, weniger ausgelassenen Bildern oder zusätzlichem Inhalt zu erzeugen.

8. System, umfassend:
einen oder mehrere Prozessoren, die dazu konfiguriert werden sollen, unter Verwendung eines oder mehrerer neuronaler Netze, die unter Verwendung einer oder mehrerer zeitlicher Posendarstellungen trainiert sind, Bilder für ein Video abzuleiten; und
einen oder mehrere Speicher, um das eine oder die mehreren neuronalen Netze zu speichern.

9. System nach Anspruch 8, wobei die Videobilder, die zum Trainieren des einen oder der mehreren neuronalen Netze verwendet werden, verwendet werden, um eine Posendarstellung und eine zeitinvariante Erscheinungsbilddarstellung zu erzeugen.

10. System nach Anspruch 9, wobei Farb-Jittering und Thin-Plate-Spline-(TPS-)Verkrümmung die Erscheinungsbildinvarianz und die Lokalisierungseigenschaften für die zeitlichen Posendarstellungen erzwingen.

11. System nach Anspruch 9, wobei die eine oder die mehreren zeitlichen Posendarstellungen Merkmale umfassen, die durch Gauß-Wärmekarten dargestellt sind, die als Mittelwert und Kovarianz parametrisiert sind.

12. System nach Anspruch 9, wobei die Posendarstellung und die zeitinvariante Erscheinungsbilddarstellung verwendet werden, um ein Eingangsvideobild zu rekonstruieren und wobei ein Verlustwert, der sich aus dem Vergleichen des Eingangsvideobilds mit dem rekonstruierten Videobild ergibt, durch Anpassen eines oder mehrerer Netzparameter des einen oder der mehreren neuronalen Netze minimiert wird.

13. System nach Anspruch 8, wobei die Bewegung der einen oder der mehreren zeitlichen Posendarstellungen im Laufe der Zeit unter Verwendung eines zeitlichen Codierers, der ein oder mehrere Netze mit langem Kurzzeitgedächtnis (LSTM-Netze) beinhaltet, modelliert wird.

14. Maschinenlesbares Medium, auf dem ein Satz von Anweisungen gespeichert ist, die bei Durchführung durch einen oder mehrere Prozessoren den einen oder die mehreren Prozessoren mindestens zu Folgendem veranlassen:
Ableiten von Bildern für ein Video unter Verwendung eines oder mehrerer neuronaler Netze, die unter Verwendung einer oder mehrerer zeitlicher Posendarstellungen trainiert sind.

15. Maschinenlesbares Medium nach Anspruch 14, wobei die Videobilder, die zum Trainieren des einen oder der mehreren neuronalen Netze verwendet werden, verwendet werden, um eine Posendarstellung und eine zeitinvariante Erscheinungsbilddarstellung zu erzeugen.

16. Maschinenlesbares Medium nach Anspruch 15, wobei Farb-Jittering und Thin-Plate-Spline-(TPS-) Verkrümmung die Erscheinungsbildinvarianz und die Lokalisierungseigenschaften für die zeitlichen Posendarstellungen erzwingen.

17. Maschinenlesbares Medium nach Anspruch 15, wobei die eine oder die mehreren zeitlichen Posendarstellungen Merkmale umfassen, die durch Gauß-Wärmekarten dargestellt sind, die als Mittelwert und Kovarianz parametrisiert sind.

18. Maschinenlesbares Medium nach Anspruch 15, wobei die Posendarstellung und die zeitinvariante Erscheinungsbilddarstellung verwendet werden, um ein Eingangsvideobild zu rekonstruieren und wobei ein Verlustwert, der sich aus dem Vergleichen des Eingangsvideobilds mit dem rekonstruierten Videobild ergibt, durch Anpassen eines oder mehrerer Netzparameter des einen oder der mehreren neuronalen Netze minimiert wird.

19. Maschinenlesbares Medium nach Anspruch 14, wobei die Bewegung der einen oder der mehreren zeitlichen Posendarstellungen im Laufe der Zeit unter Verwendung eines zeitlichen Codierers, der ein oder mehrere Netze mit langem Kurzzeitgedächtnis (LSTM-Netze) beinhaltet, modelliert wird.

20. Prozessor, umfassend:
eine oder mehrere arithmetisch-logische Einheiten (ALUs), um ein oder mehrere neuronale Netze zu trainieren, zumindest teilweise unter Verwendung eines oder mehrerer neuronaler Netze, die unter Verwendung einer oder mehrerer zeitlicher Posendarstellungen trainiert sind, Bilder für ein Video abzuleiten.

21. Prozessor nach Anspruch 20, wobei die Videobilder, die zum Trainieren des einen oder der mehreren neuronalen Netze verwendet werden, verwendet werden, um eine Posendarstellung und eine zeitinvariante Erscheinungsbilddarstellung zu erzeugen.

22. Prozessor nach Anspruch 21, wobei Farb-Jittering und Thin-Plate-Spline-(TPS-)Verkrümmung die Erscheinungsbildinvarianz und die Lokalisierungseigenschaften für die zeitlichen Posendarstellungen erzwingen.

23. Prozessor nach Anspruch 21, wobei die eine oder die mehreren zeitlichen Posendarstellungen Merkmale umfassen, die durch Gauß-Wärmekarten dargestellt sind, die als Mittelwert und Kovarianz parametrisiert sind.

24. Prozessor nach Anspruch 21, wobei die Posendarstellung und die zeitinvariante Erscheinungsbilddarstellung verwendet werden, um ein Eingangsvideobild zu rekonstruieren und wobei ein Verlustwert, der sich aus dem Vergleichen des Eingangsvideobilds mit dem rekonstruierten Videobild ergibt, durch Anpassen eines oder mehrerer Netzparameter des einen oder der mehreren neuronalen Netze minimiert wird.

25. Prozessor nach Anspruch 20, wobei die Bewegung der einen oder der mehreren zeitlichen Posendarstellungen im Laufe der Zeit unter Verwendung eines zeitlichen Codierers, der ein oder mehrere Netze mit langem Kurzzeitgedächtnis (LSTM-Netze) beinhaltet, modelliert wird.

26. System, umfassend:
einen oder mehrere Prozessoren, um Parameter zu berechnen, die einem oder mehreren neuronalen Netzen entsprechen, zumindest teilweise, um unter Verwendung einer oder mehrerer zeitlicher Posendarstellungen Bilder für ein Video abzuleiten; und
einen oder mehrere Speicher, um die Parameter zu speichern.

27. System nach Anspruch 26, wobei die Videobilder, die zum Trainieren des einen oder der mehreren neuronalen Netze verwendet werden, verwendet werden, um eine Posendarstellung und eine zeitinvariante Erscheinungsbilddarstellung zu erzeugen.

28. System nach Anspruch 27, wobei Farb-Jittering und Thin-Plate-Spline-(TPS-)Verkrümmung die Erscheinungsbildinvarianz und die Lokalisierungseigenschaften für die zeitlichen Posendarstellungen erzwingen.

29. System nach Anspruch 27, wobei die eine oder die mehreren zeitlichen Posendarstellungen Merkmale umfassen, die durch Gauß-Wärmekarten dargestellt sind, die als Mittelwert und Kovarianz parametrisiert sind.

30. System nach Anspruch 27, wobei die Posendarstellung und die zeitinvariante Erscheinungsbilddarstellung verwendet werden, um ein Eingangsvideobild zu rekonstruieren und wobei ein Verlustwert, der sich aus dem Vergleichen des Eingangsvideobilds mit dem rekonstruierten Videobild ergibt, durch Anpassen eines oder mehrerer Netzparameter des einen oder der mehreren neuronalen Netze minimiert wird.

31. System nach Anspruch 26, wobei die Bewegung der einen oder der mehreren zeitlichen Posendarstellungen im Laufe der Zeit unter Verwendung eines zeitlichen Codierers, der ein oder mehrere Netze mit langem Kurzzeitgedächtnis (LSTM-Netze) beinhaltet, modelliert wird.

32. Maschinenlesbares Medium, auf dem ein Satz von Anweisungen gespeichert ist, die bei Durchführung durch einen oder mehrere Prozessoren den einen oder die mehreren Prozessoren mindestens zu Folgendem veranlassen:
Veranlassen, dass ein oder mehrere neuronale Netze trainiert werden, um zumindest teilweise unter Verwendung einer oder mehrerer zeitlicher Posendarstellungen Bilder für ein Video abzuleiten; und
einen oder mehrere Speicher, um die Parameter zu speichern.

33. Maschinenlesbares Medium nach Anspruch 32, wobei die Videobilder, die zum Trainieren des einen oder der mehreren neuronalen Netze verwendet werden, verwendet werden, um eine Posendarstellung und eine zeitinvariante Erscheinungsbilddarstellung zu erzeugen.

34. Maschinenlesbares Medium nach Anspruch 33, wobei Farb-Jittering und Thin-Plate-Spline-(TPS-) Verkrümmung die Erscheinungsbildinvarianz und die Lokalisierungseigenschaften für die zeitlichen Posendarstellungen erzwingen.

35. Maschinenlesbares Medium nach Anspruch 33, wobei die eine oder die mehreren zeitlichen Posendarstellungen Merkmale umfassen, die durch Gauß-Wärmekarten dargestellt sind, die als Mittelwert und Kovarianz parametrisiert sind.

36. Maschinenlesbares Medium nach Anspruch 33, wobei die Posendarstellung und die zeitinvariante Erscheinungsbilddarstellung verwendet werden, um ein Eingangsvideobild zu rekonstruieren und wobei ein Verlustwert, der sich aus dem Vergleichen des Eingangsvideobilds mit dem rekonstruierten Videobild ergibt, durch Anpassen eines oder mehrerer Netzparameter des einen oder der mehreren neuronalen Netze minimiert wird.

37. Maschinenlesbares Medium nach Anspruch 32, wobei die Bewegung der einen oder der mehreren zeitlichen Posendarstellungen im Laufe der Zeit unter Verwendung eines zeitlichen Codierers, der ein oder mehrere Netze mit langem Kurzzeitgedächtnis (LSTM-Netze) beinhaltet, modelliert wird.

Es folgen 46 Seiten Zeichnungen

Anhängende Zeichnungen

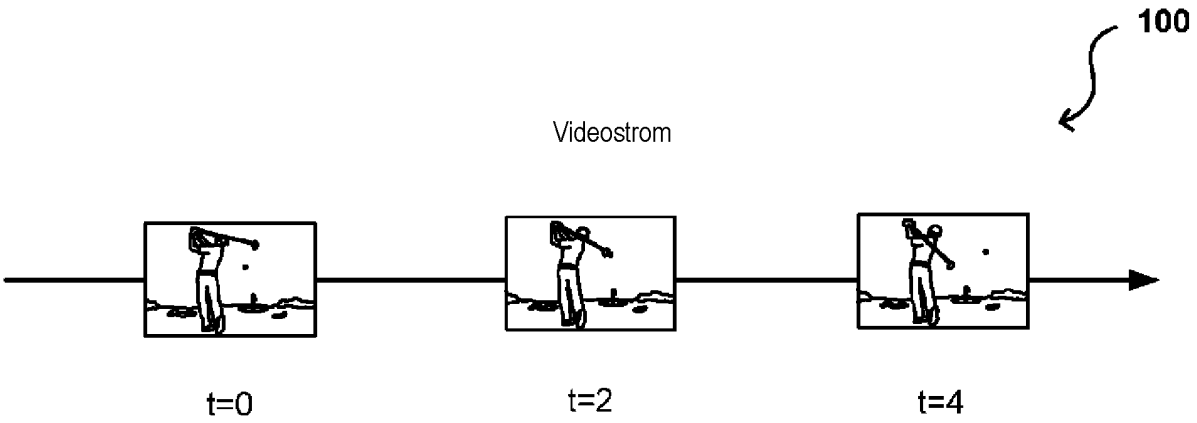


FIG. 1A

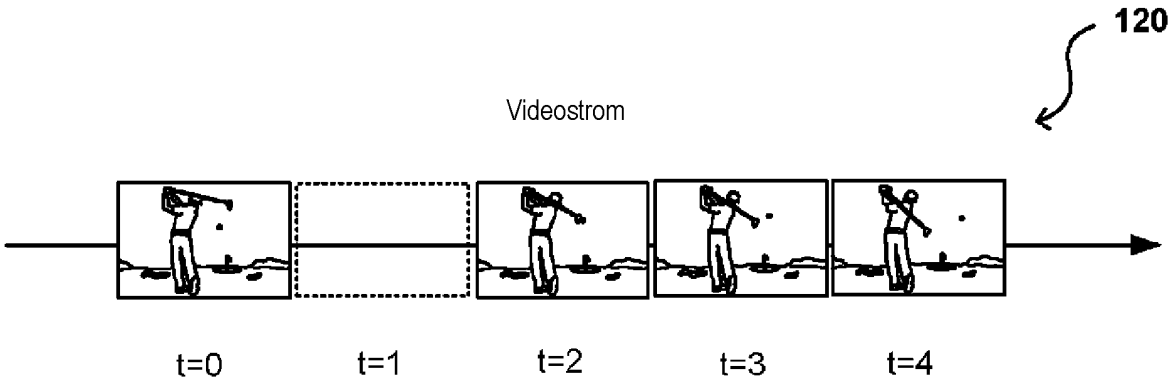


FIG. 1B

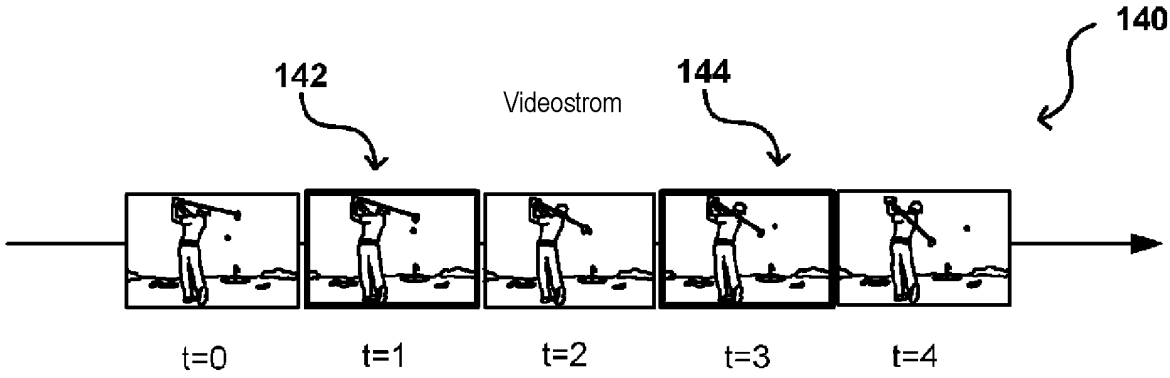


FIG. 1C

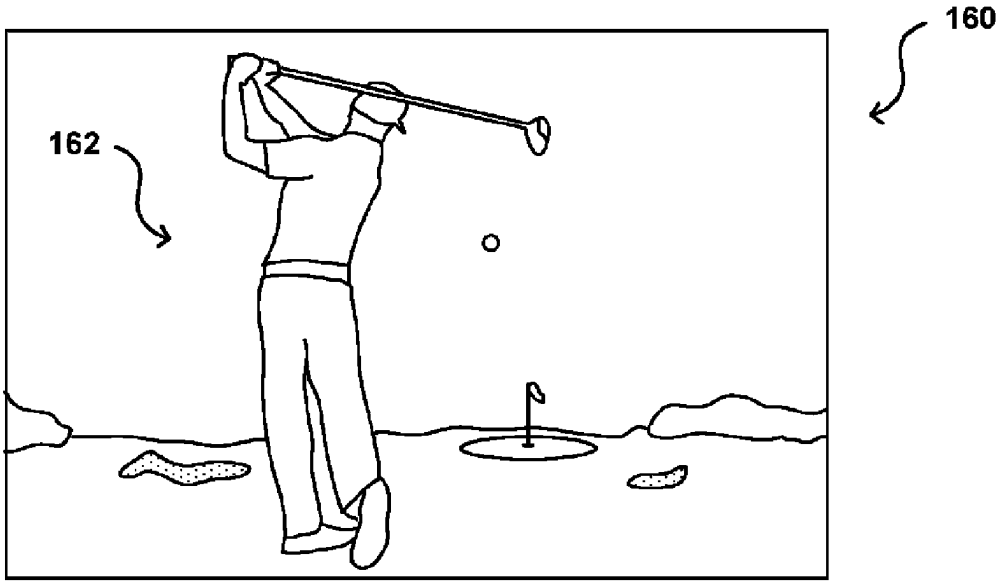


FIG. 1D

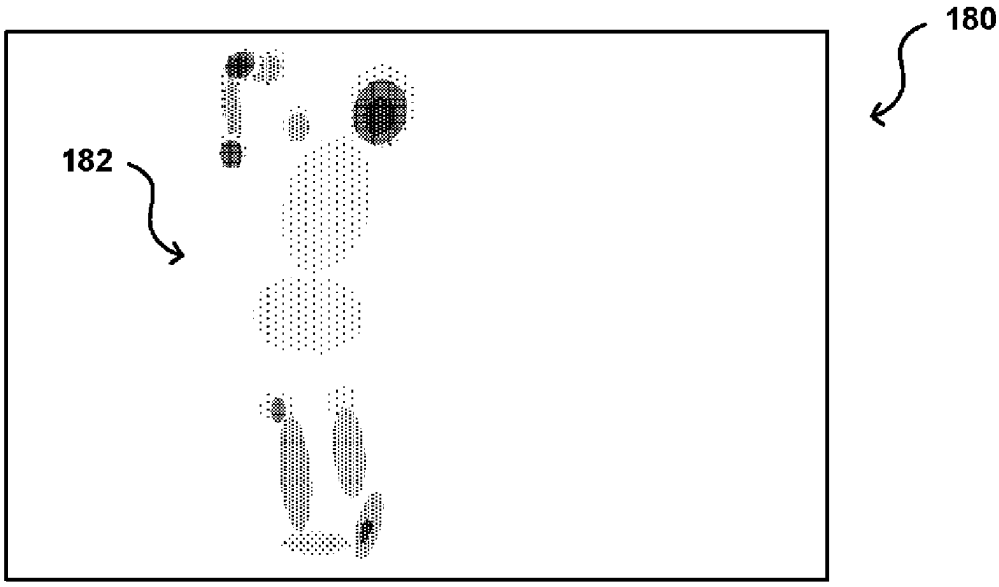


FIG. 1E

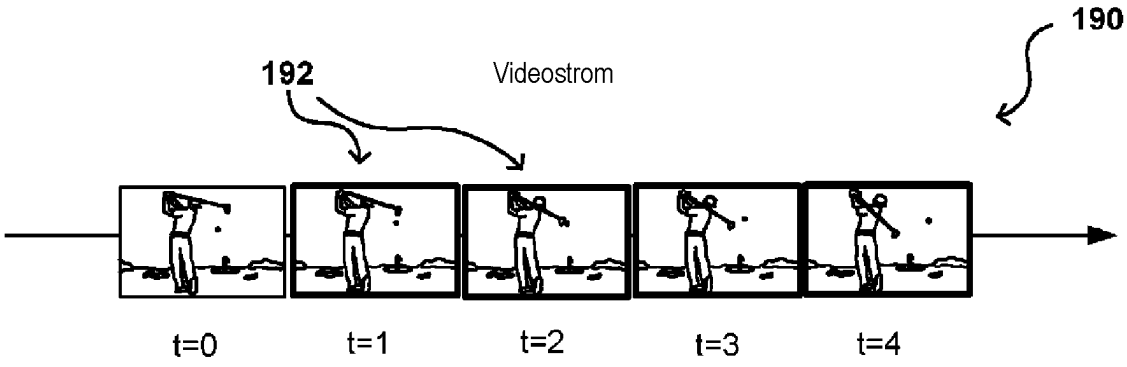


FIG. 1F

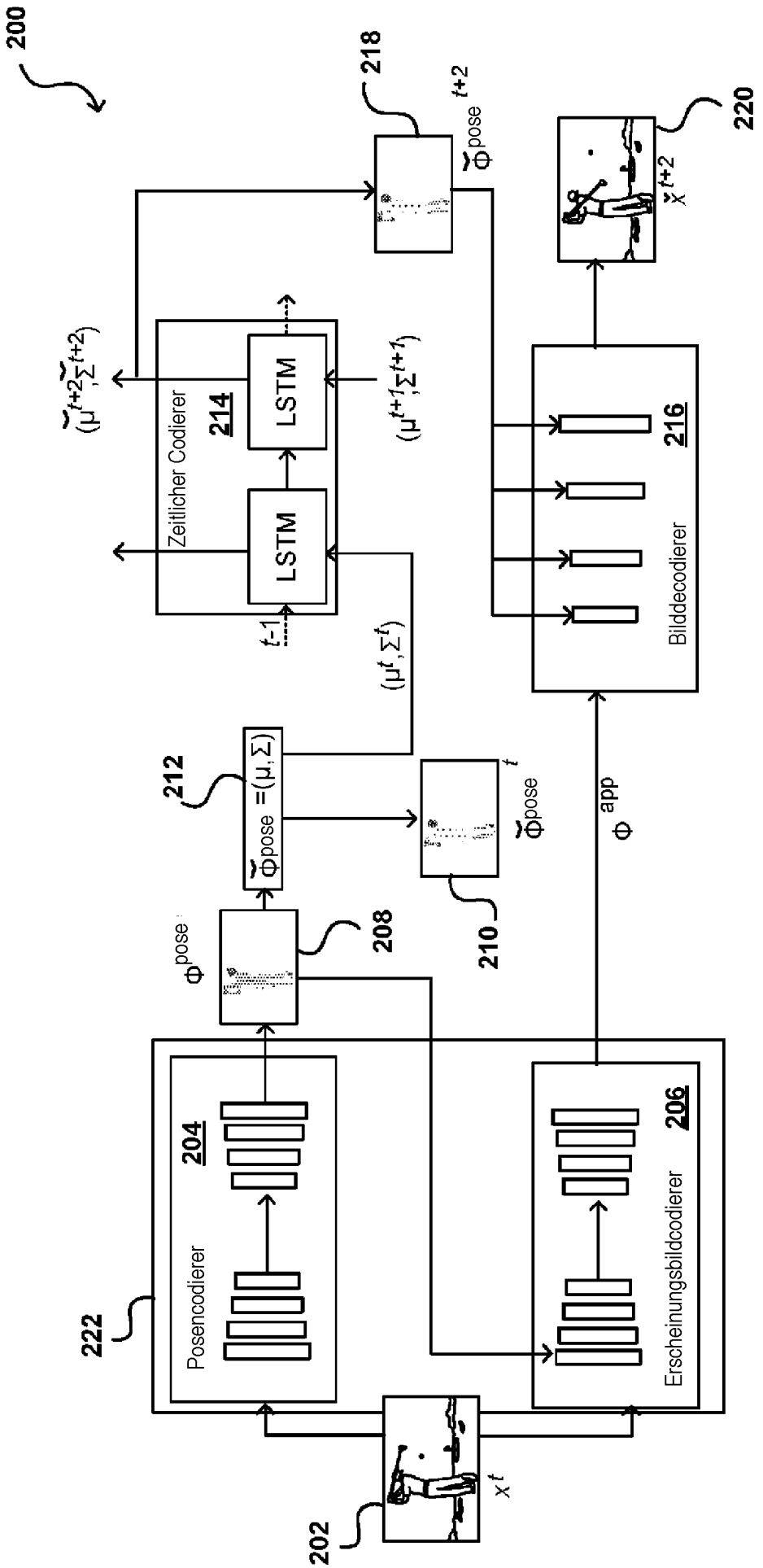

200
218
220
LSTM
LSTM
Zeitlicher Codierer
214
216
Bilddecodierer
212
208
210
204
Posencodierer
206
Erscheinungsbildcodierer
222
202

FIG. 2

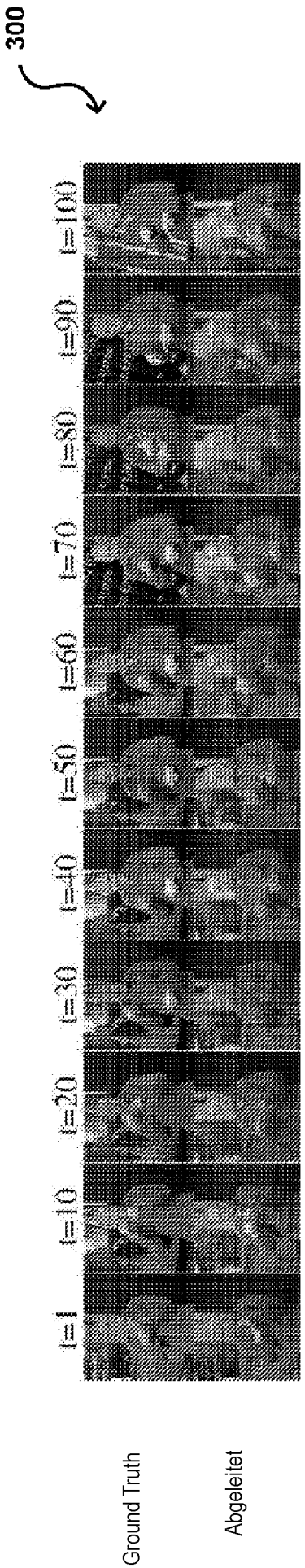

300
t=1
t=10
t=20
t=30
t=40
t=50
t=60
t=70
t=80
t=90
t=100
Ground Truth
Abgeleitet

FIG. 3A

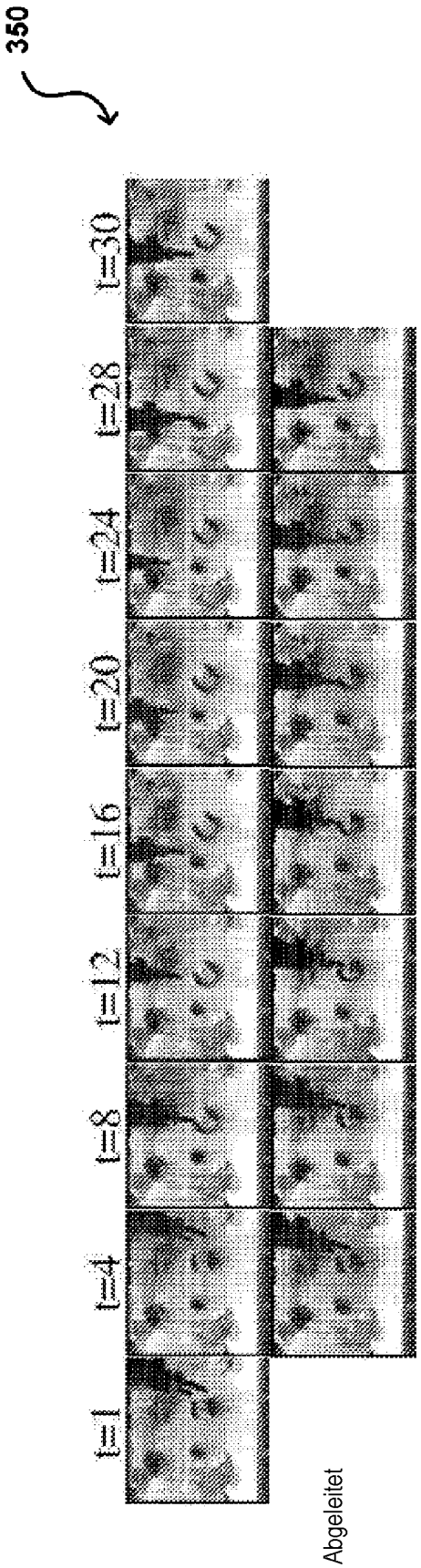

350
t=1
t=4
t=8
t=12
t=16
t=20
t=24
t=28
t=30
Videostrom
Abgeleitet

FIG. 3B

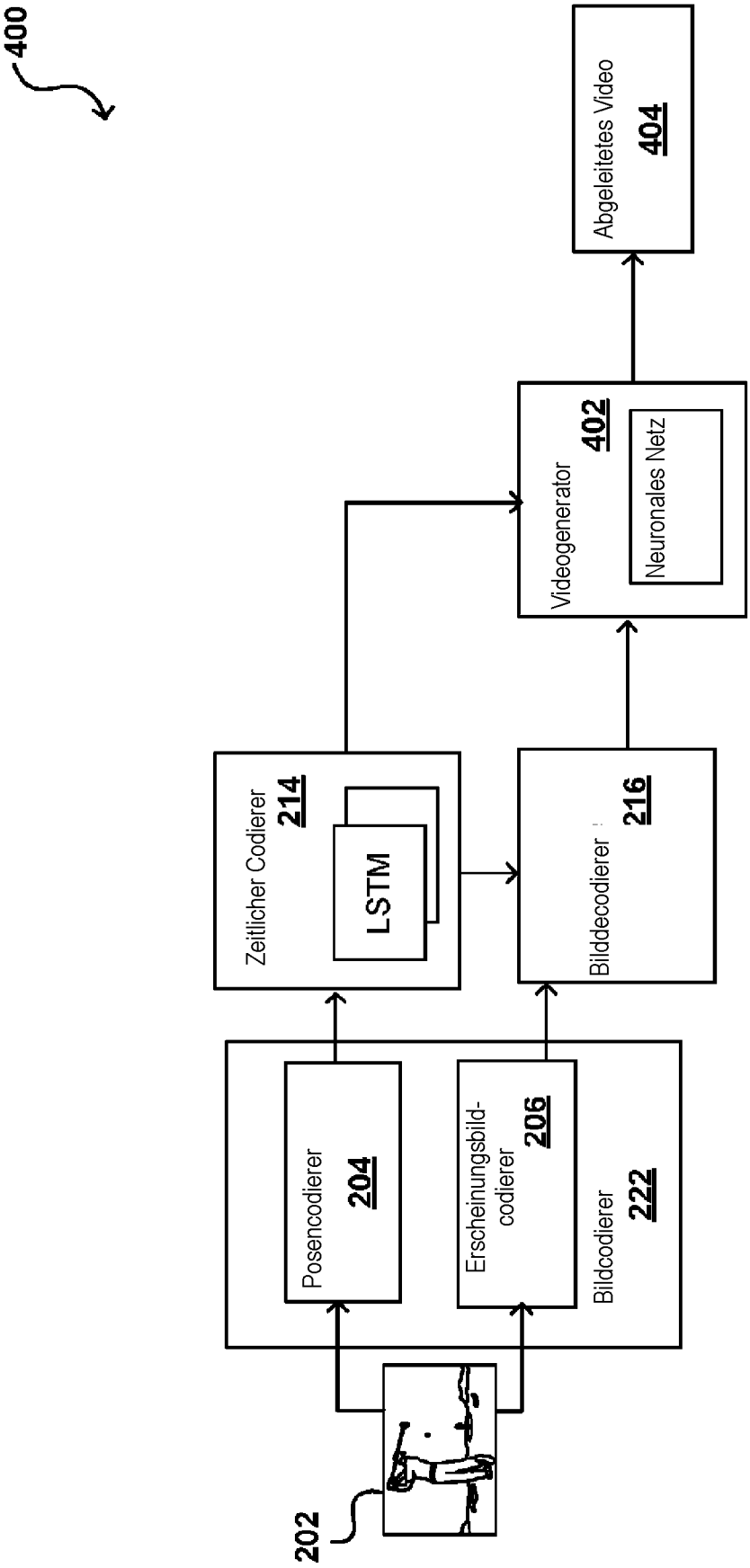

400
Abgeleitetes Video
404
Videogenerator
402
Neuronales Netz
Zeitlicher Codierer
214
LSTM
Bilddecodierer
216
Posencodierer
204
Erscheinungsbild-codierer
206
Bildcodierer
222
202

FIG. 4

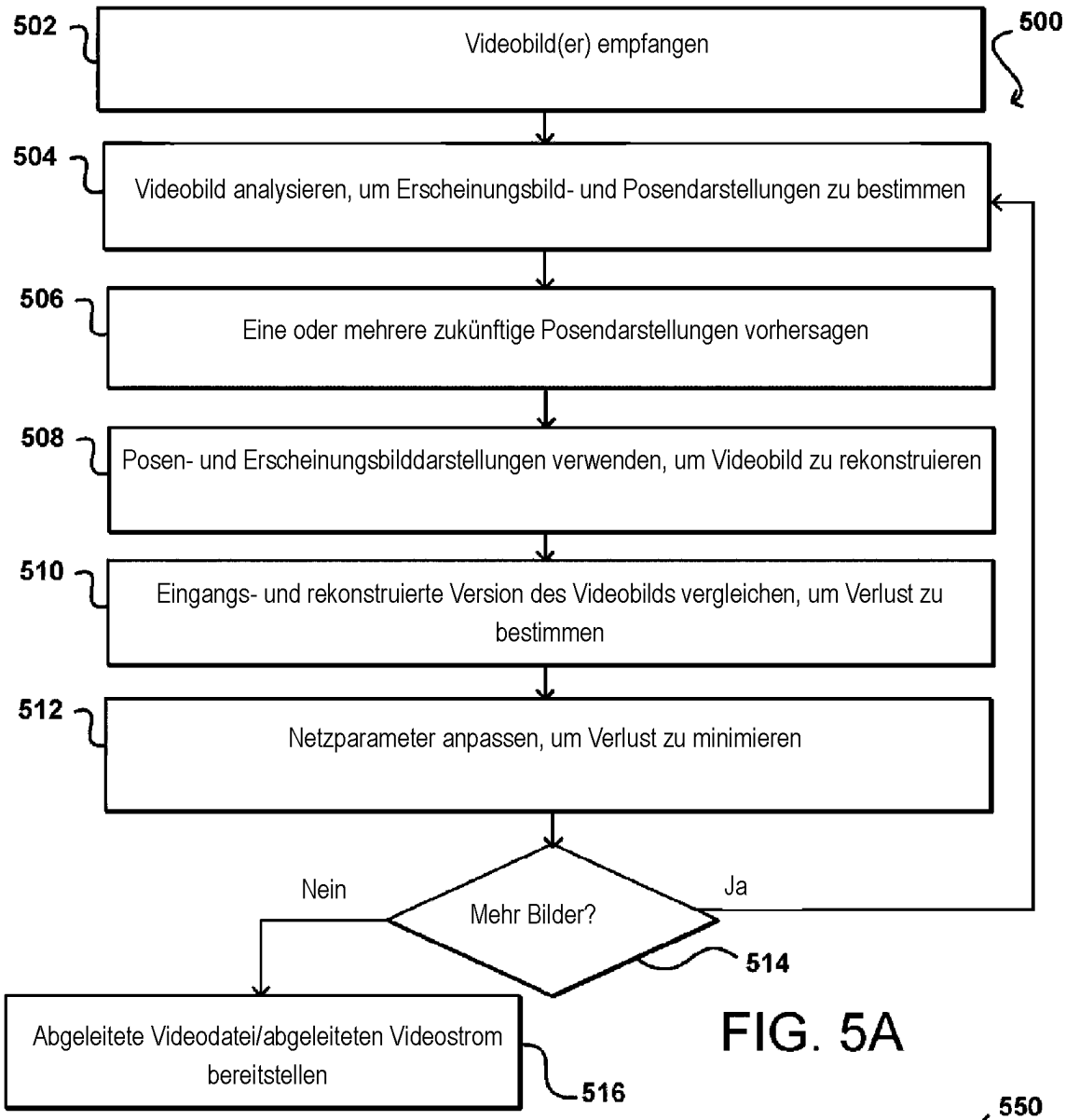

502
Videobild(er) empfangen
500
504
Videobild analysieren, um Erscheinungsbild- und Posendarstellungen zu bestimmen
506
Eine oder mehrere zukünftige Posendarstellungen vorhersagen
508
Posen- und Erscheinungsbilddarstellungen verwenden, um Videobild zu rekonstruieren
510
Eingangs- und rekonstruierte Version des Videobilds vergleichen, um Verlust zu bestimmen
512
Netzparameter anpassen, um Verlust zu minimieren
Nein
Mehr Bilder?
Ja
514
Abgeleitete Videodatei/abgeleiteten Videostrom bereitstellen
516

FIG. 5A

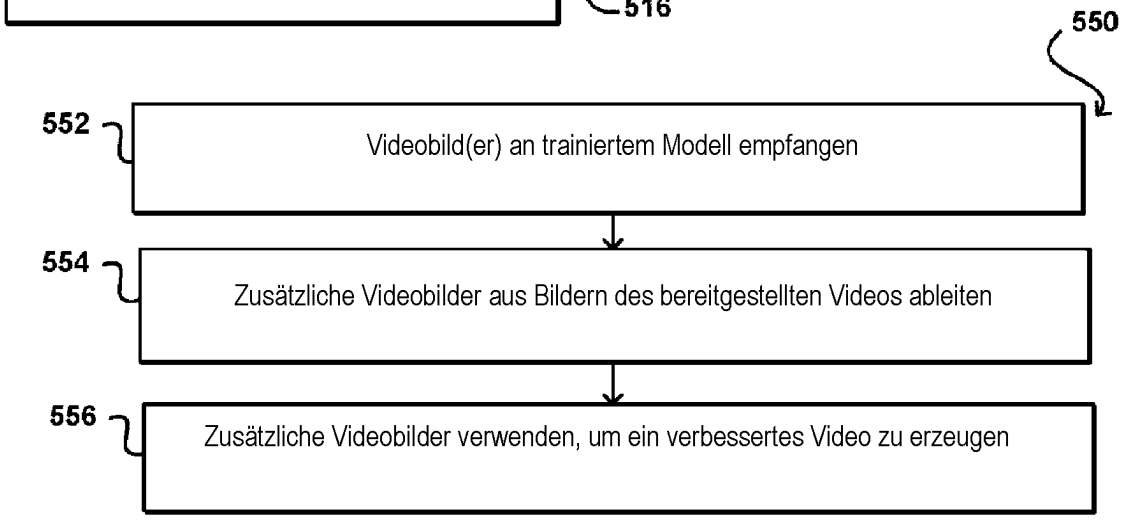

550
552
Videobild(er) an trainiertem Modell empfangen
554
Zusätzliche Videobilder aus Bildern des bereitgestellten Videos ableiten
556
Zusätzliche Videobilder verwenden, um ein verbessertes Video zu erzeugen

FIG. 5B

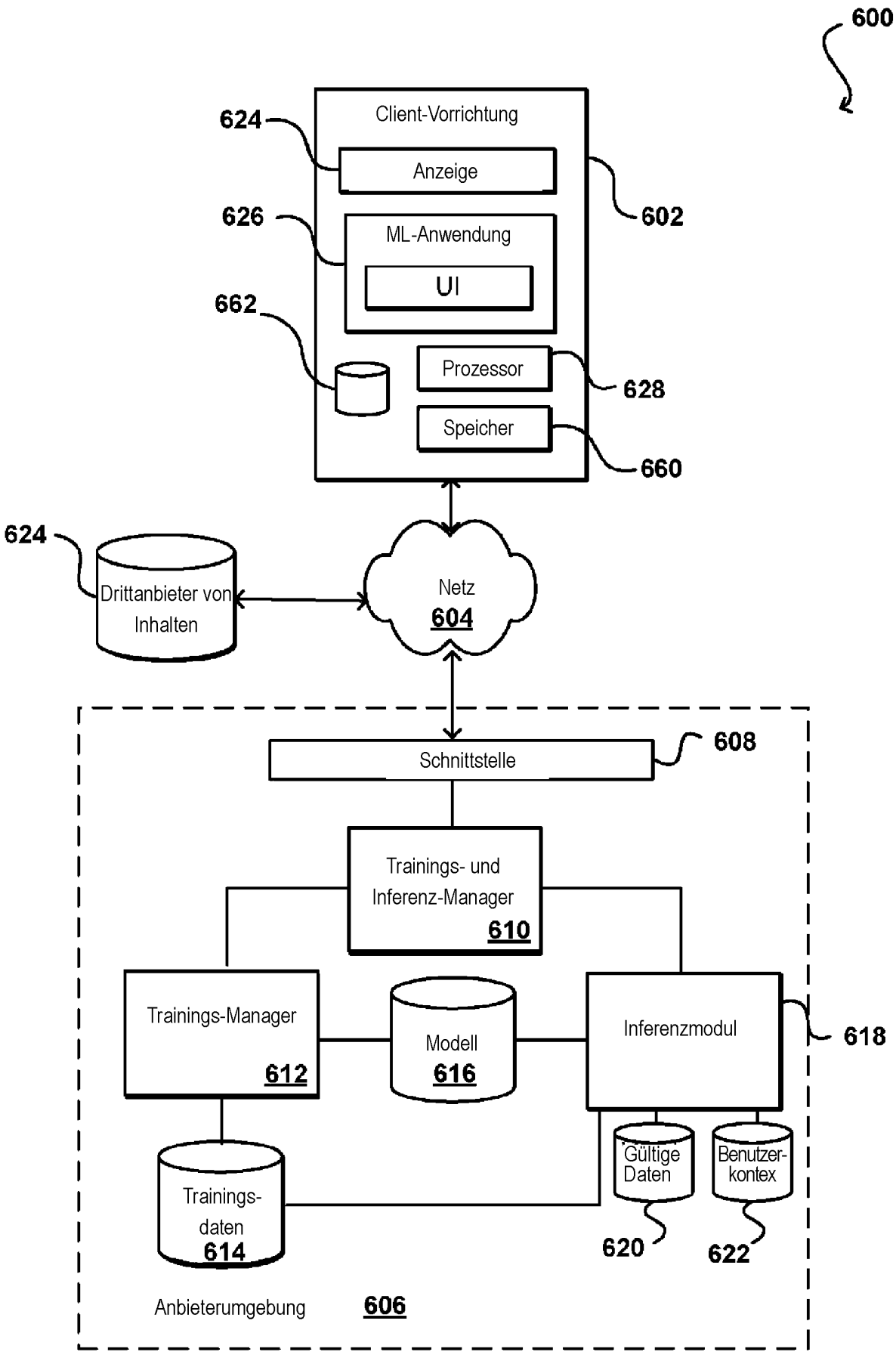

600
Client-Vorrichtung
624
Anzeige
602
626
ML-Anwendung
UI
662
Prozessor
628
Speicher
660
624
Drittanbieter von Inhalten
Netz
604
608
Schnittstelle
Trainings- und Inferenz-Manager
610
Trainings-Manager
612
Modell
616
Inferenzmodul
618
Gültige Daten
Benutzer-kontex
Trainings-daten
614
620
622
Anbieterumgebung
606

FIG. 6

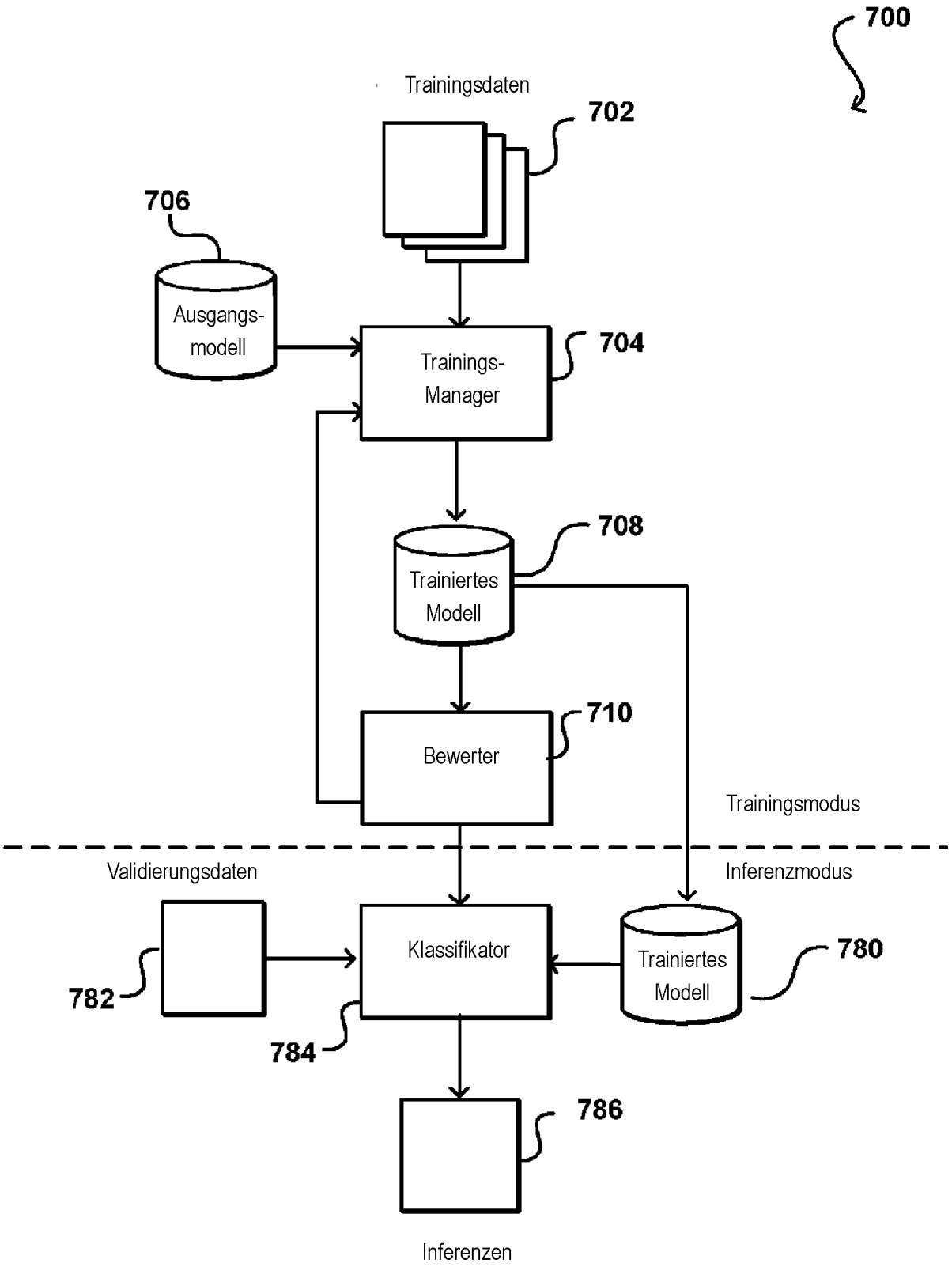

700
Trainingsdaten
702
706
Ausgangs-
modell
Trainings-
Manager
704
708
Trainiertes
Modell
710
Bewerter
Trainingsmodus
Validierungsdaten
Inferenzmodus
Klassifikator
Trainiertes
Modell
780
782
784
786
Inferenzen

FIG. 7

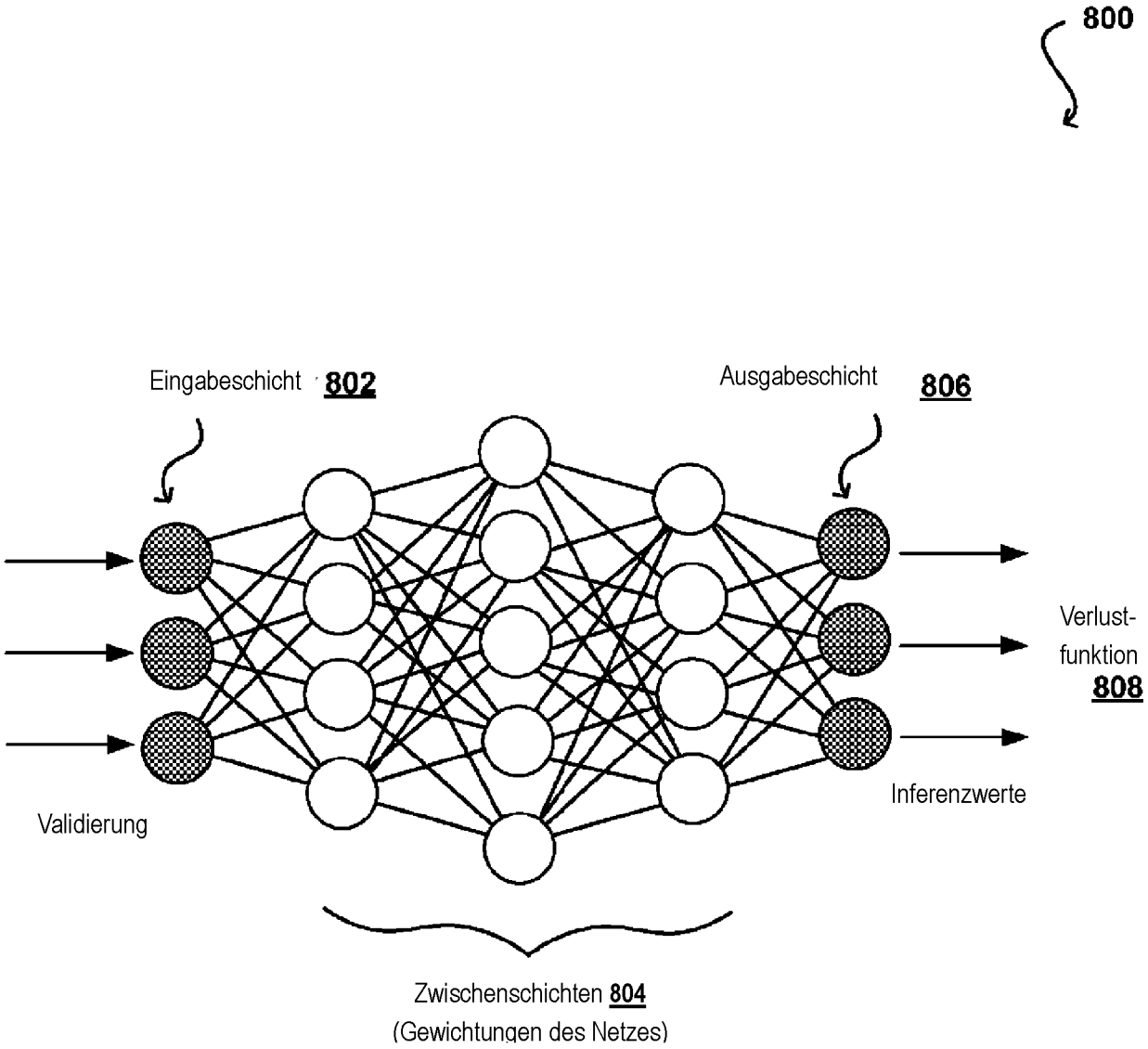

800
Eingabeschicht 802
Ausgabeschicht 806
Verlust-
funktion
808
Validierung
Inferenzwerte
Zwischenschichten 804
(Gewichtungen des Netzes)

FIG. 8

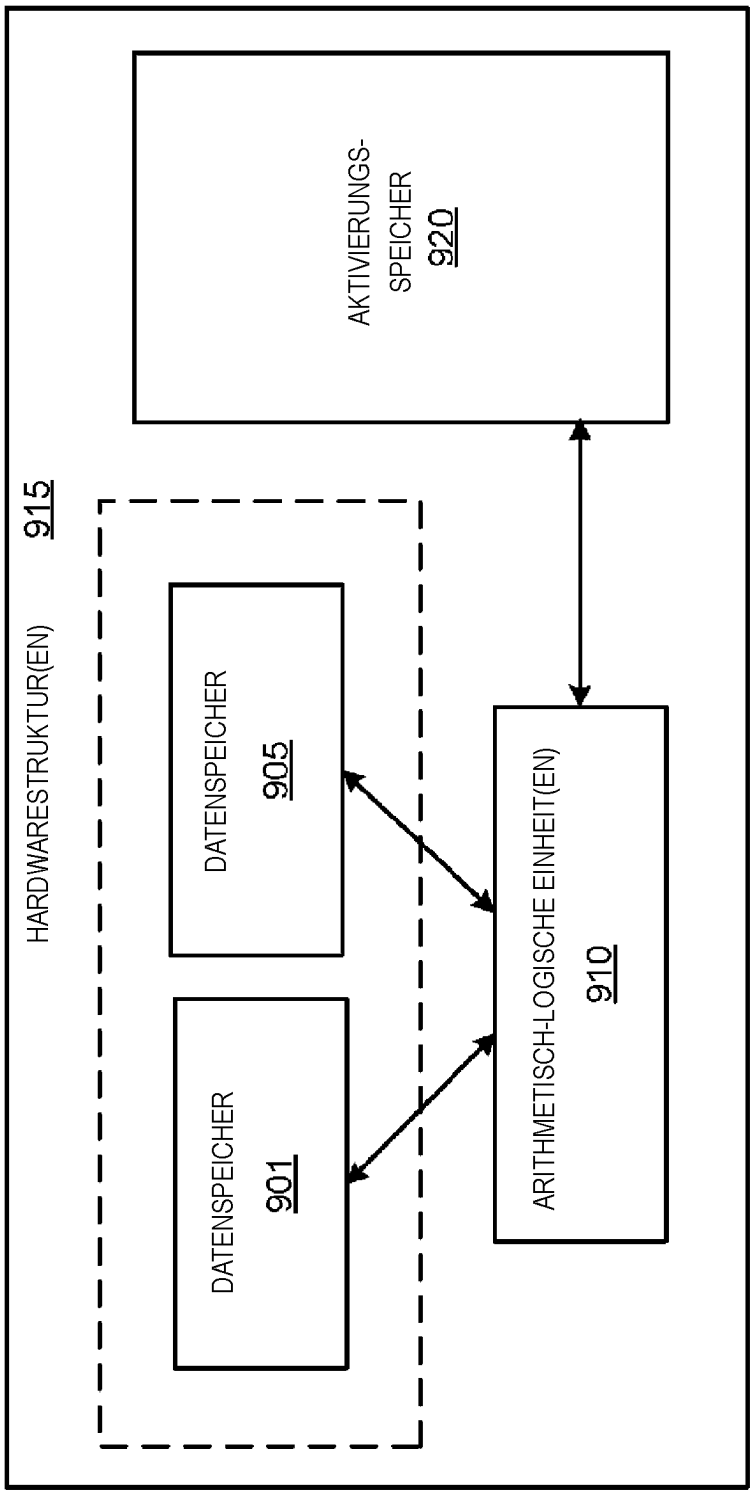

AKTIVIERUNGS-
SPEICHER
920
HARDWARESTRUKTUR(EN)
915
DATENSPEICHER
905
DATENSPEICHER
901
ARITHMETISCH-LOGISCHE EINHEIT(EN)
910

FIG. 9A

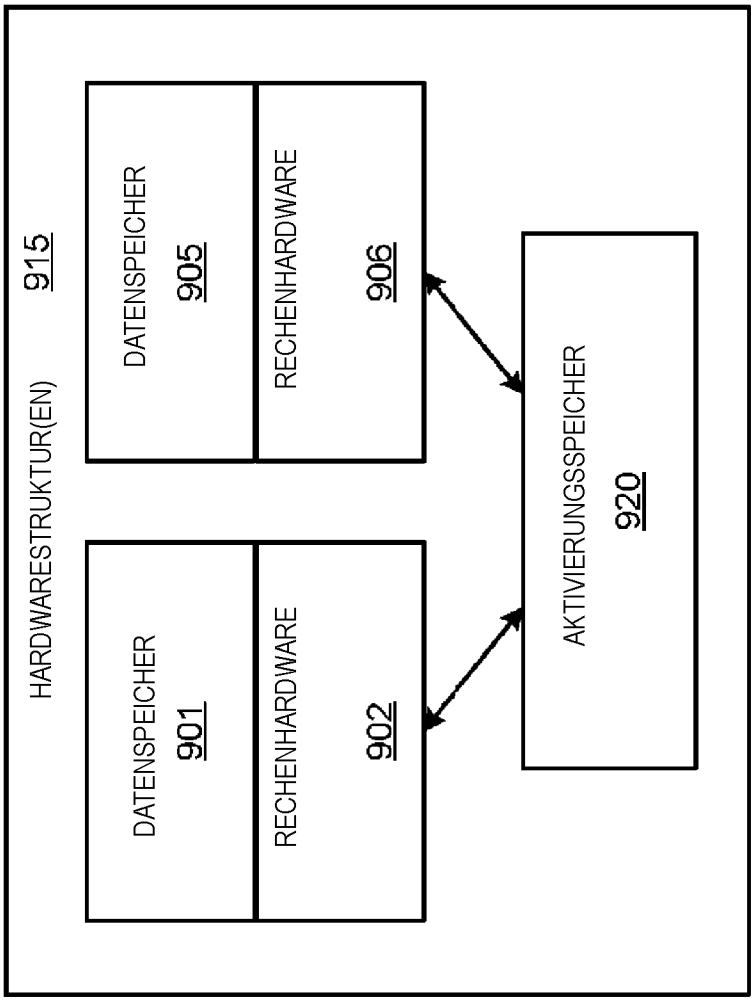

HARDWARESTRUKTUR(EN) 915
DATENSPEICHER 905
RECHENHARDWARE 906
DATENSPEICHER 901
RECHENHARDWARE 902
AKTIVIERUNGSSPEICHER 920

FIG. 9B

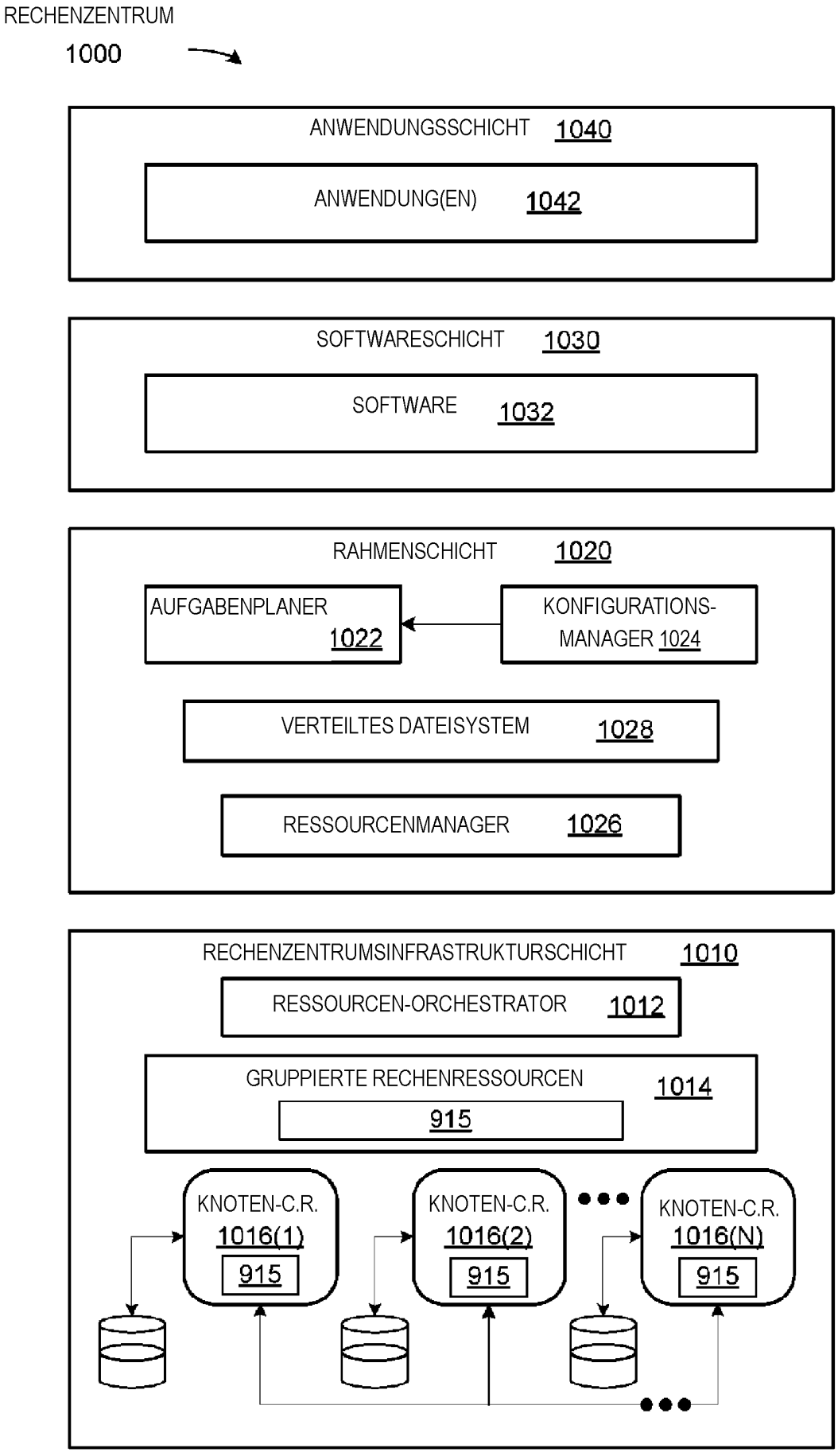

RECHENZENTRUM
1000
ANWENDUNGSSCHICHT 1040
ANWENDUNG(EN) 1042
SOFTWARESCHICHT 1030
SOFTWARE 1032
RAHMENSCHICHT 1020
AUFGABENPLANER 1022
KONFIGURATIONS-MANAGER 1024
VERTEILTES DATEISYSTEM 1028
RESSOURCENMANAGER 1026
RECHENZENTRUMSINFRASTRUKTURSCHICHT 1010
RESSOURCEN-ORCHESTRATOR 1012
GRUPPIERTE RECHENRESSOURCEN 1014
915
KNOTEN-C.R. 1016(1) 915
KNOTEN-C.R. 1016(2) 915
KNOTEN-C.R. 1016(N) 915

FIG. 10

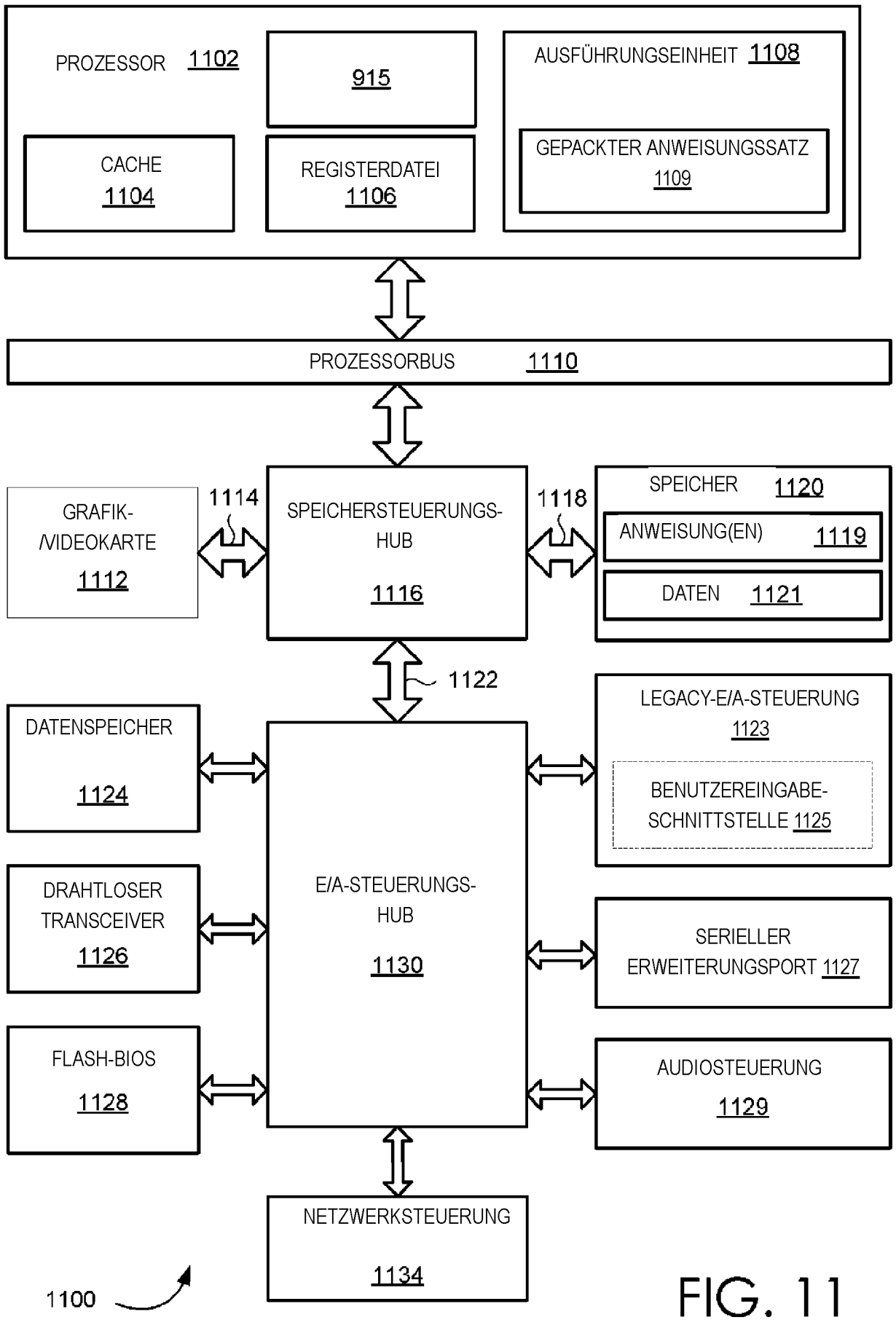

PROZESSOR 1102
915
AUSFÜHRUNGSEINHEIT 1108
CACHE 1104
REGISTERDATEI 1106
GEPACKTER ANWEISUNGSSATZ 1109
PROZESSORBUS 1110
GRAFIK-/VIDEOKARTE 1112
1114
SPEICHERSTEUERUNGS-HUB 1116
1118
SPEICHER 1120
ANWEISUNG(EN) 1119
DATEN 1121
1122
DATENSPEICHER 1124
LEGACY-E/A-STEUERUNG 1123
BENUTZEREINGABE-SCHNITTSTELLE 1125
DRAHTLOSER TRANSCEIVER 1126
E/A-STEUERUNGS-HUB 1130
SERIELLER ERWEITERUNGSPORT 1127
FLASH-BIOS 1128
AUDIOSTEUERUNG 1129
NETZWERKSTEUERUNG 1134
1100

FIG. 11

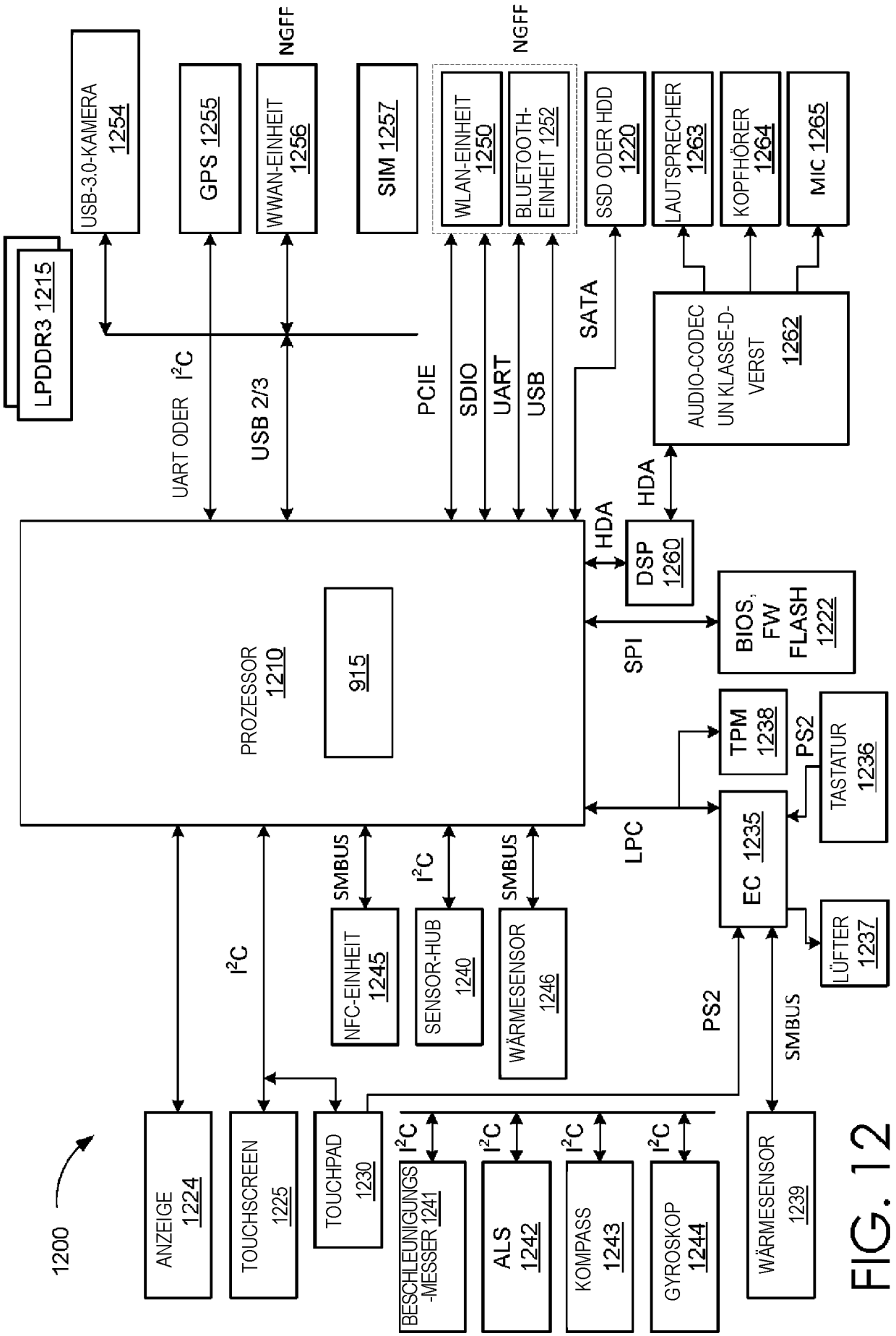

1200
PROZESSOR 1210
915
LPDDR3 1215
USB-3.0-KAMERA 1254
GPS 1255
WWAN-EINHEIT 1256
NGFF
SIM 1257
WLAN-EINHEIT 1250
BLUETOOTH-EINHEIT 1252
NGFF
SSD ODER HDD 1220
LAUTSPRECHER 1263
KOPFHÖRER 1264
MIC 1265
AUDIO-CODEC UN KLASSE-D-VERST 1262
UART ODER I²C
USB 2/3
PCIE
SDIO
UART
USB
SATA
HDA
HDA
DSP 1260
SPI
BIOS, FW FLASH 1222
TPM 1238
PS2
TASTATUR 1236
LPC
EC 1235
LÜFTER 1237
SMBUS
I²C
SMBUS
NFC-EINHEIT 1245
SENSOR-HUB 1240
WÄRMESENSOR 1246
I²C
PS2
SMBUS
ANZEIGE 1224
TOUCHSCREEN 1225
TOUCHPAD 1230
I²C
I²C
I²C
I²C
BESCHLEUNIGUNGS-MESSER 1241
ALS 1242
KOMPASS 1243
GYROSKOP 1244
WÄRMESENSOR 1239

FIG. 12

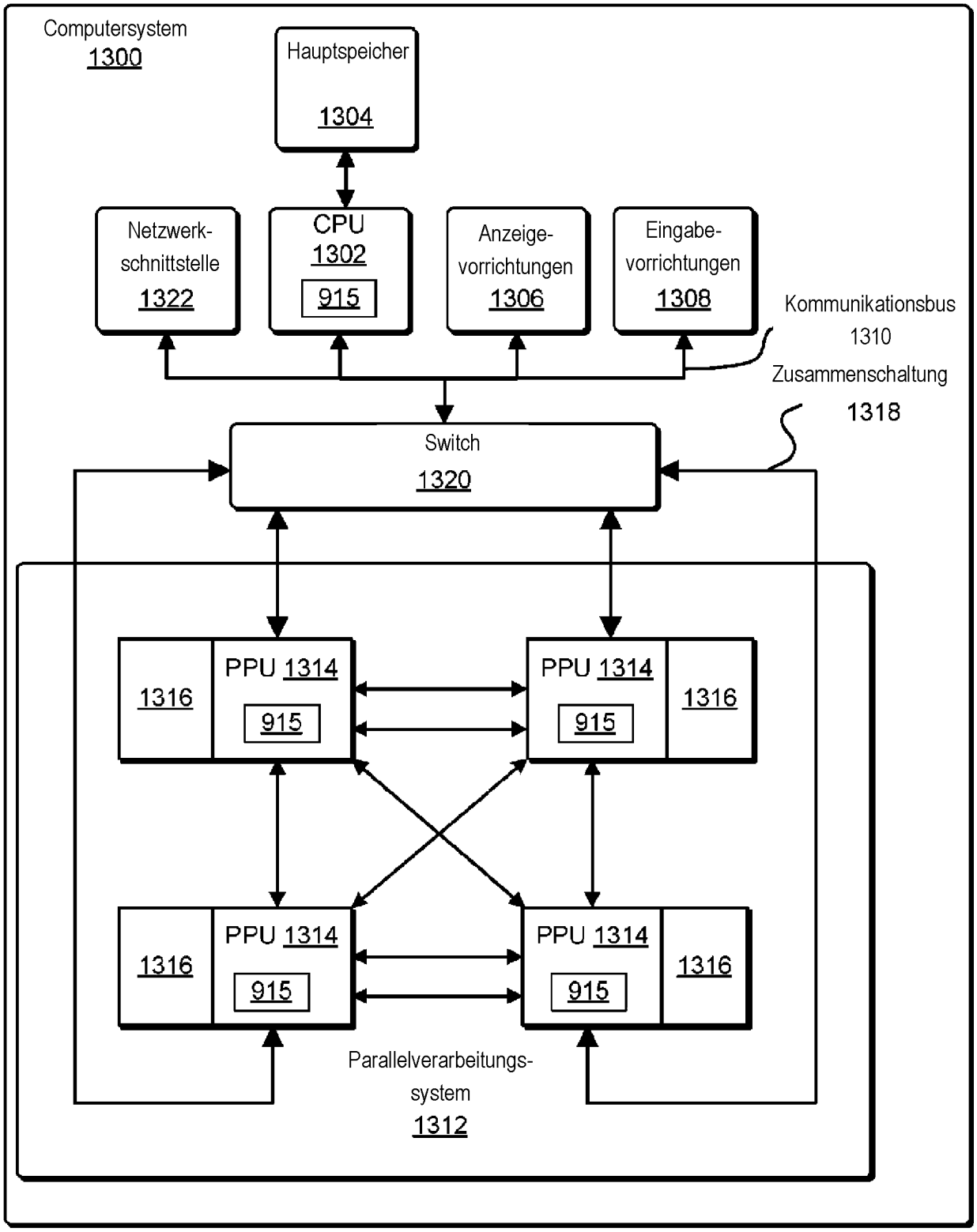

Computersystem
1300
Hauptspeicher
1304
Netzwerk-
schnittstelle
1322
CPU
1302
915
Anzeige-
vorrichtungen
1306
Eingabe-
vorrichtungen
1308
Kommunikationsbus
1310
Zusammenschaltung
1318
Switch
1320
1316
PPU 1314
915
PPU 1314
915
1316
1316
PPU 1314
915
PPU 1314
915
1316
Parallelverarbeitungs-
system
1312

FIG. 13

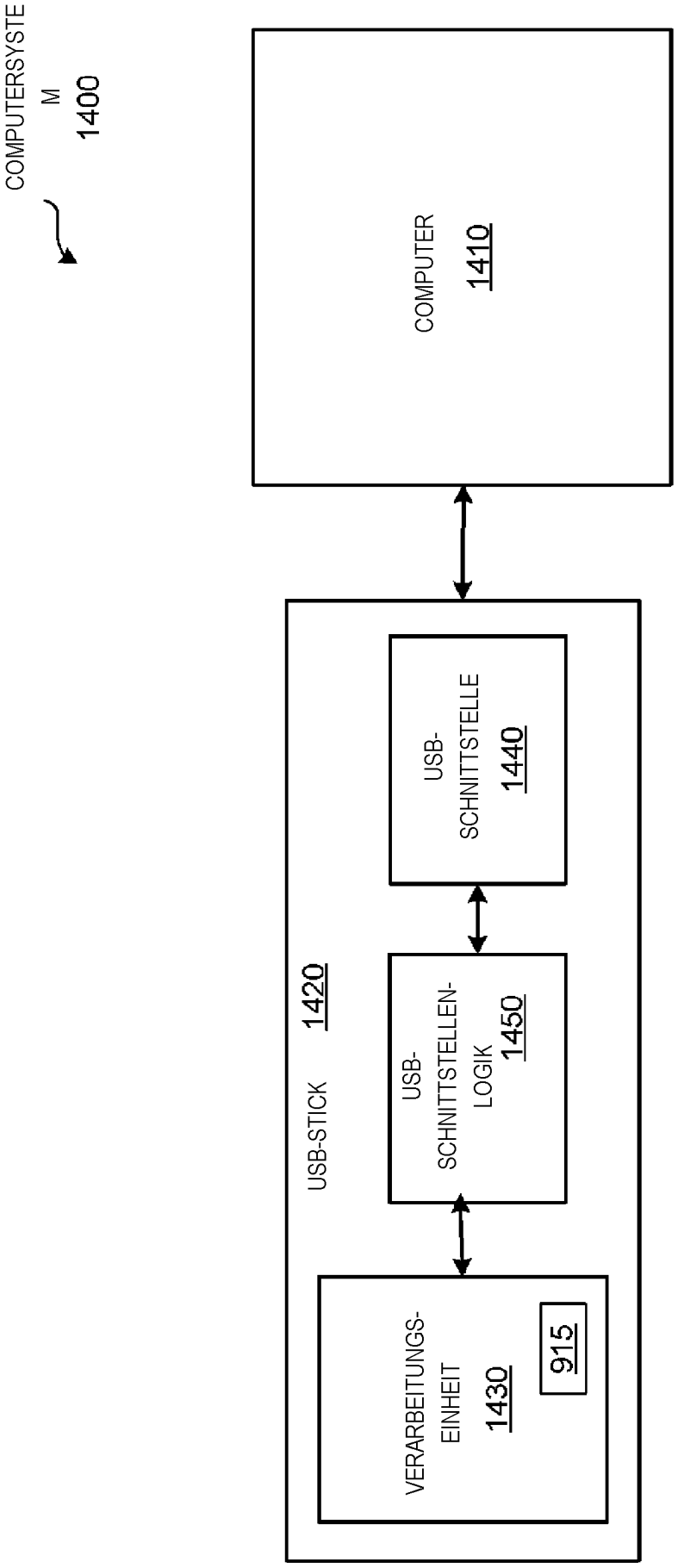

COMPUTERSYSTEM
1400
COMPUTER
1410
USB-STICK
1420
USB-SCHNITTSTELLE
1440
USB-SCHNITTSTELLEN-LOGIK
1450
VERARBEITUNGS-EINHEIT
1430
915

FIG. 14

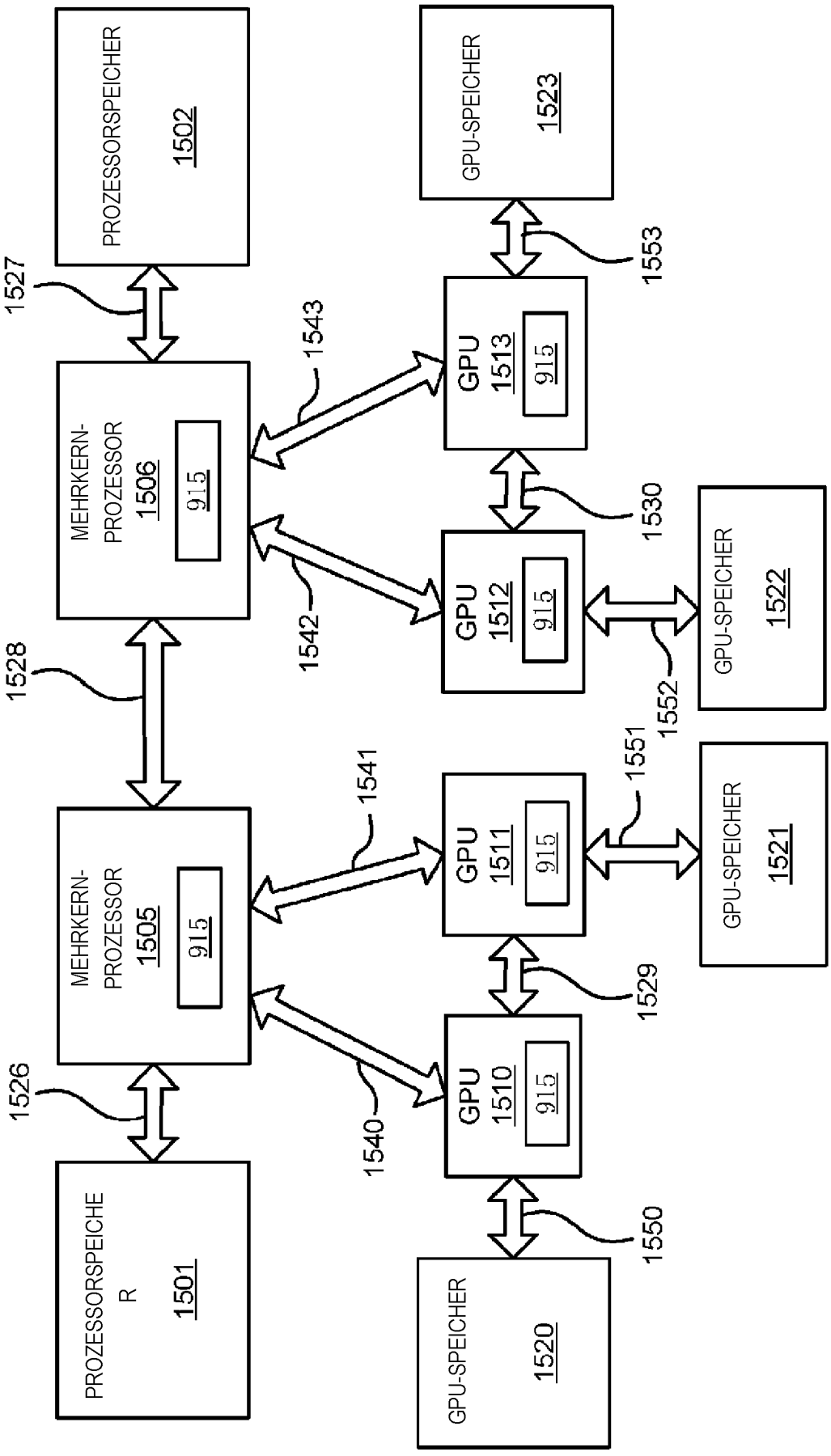

PROZESSORSPEICHER 1502
GPU-SPEICHER 1523
1527
1553
1543
MEHRKERN-PROZESSOR 1506
915
GPU 1513
915
1530
1542
GPU 1512
915
GPU-SPEICHER 1522
1528
1552
1551
1541
MEHRKERN-PROZESSOR 1505
915
GPU 1511
915
GPU-SPEICHER 1521
1529
1526
1540
GPU 1510
915
PROZESSORSPEICHER 1501
1550
GPU-SPEICHER 1520

FIG. 15A

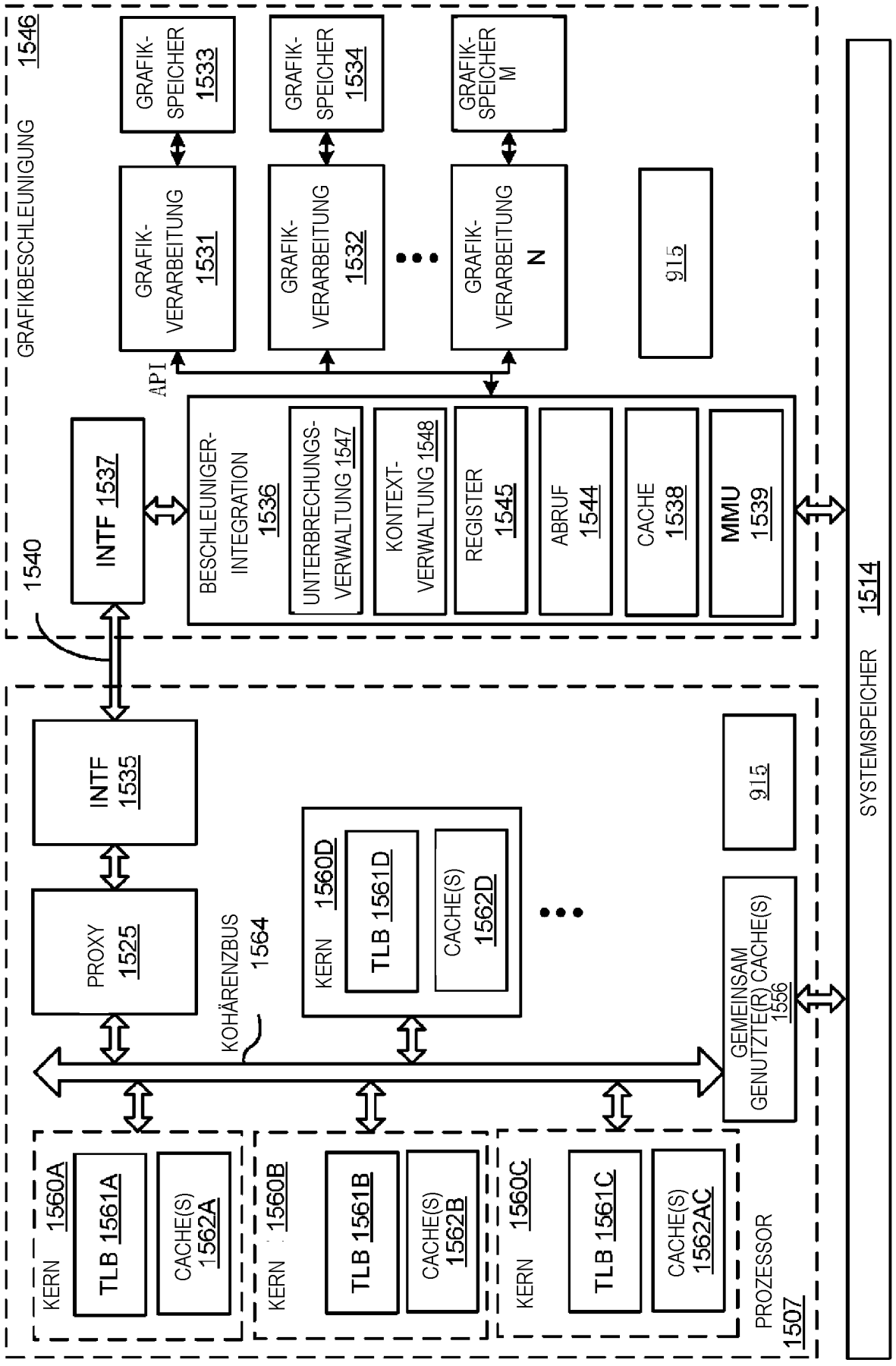

GRAFIKBESCHLEUNIGUNG 1546
GRAFIK-SPEICHER 1533
GRAFIK-SPEICHER 1534
GRAFIK-SPEICHER M
GRAFIK-VERARBEITUNG 1531
GRAFIK-VERARBEITUNG 1532
GRAFIK-VERARBEITUNG N
915
API
INTF 1537
1540
BESCHLEUNIGER-INTEGRATION 1536
UNTERBRECHUNGS-VERWALTUNG 1547
KONTEXT-VERWALTUNG 1548
REGISTER 1545
ABRUF 1544
CACHE 1538
MMU 1539
INTF 1535
PROXY 1525
KOHÄRENZBUS 1564
KERN 1560D
TLB 1561D
CACHE(S) 1562D
915
GEMEINSAM GENUTZTE(R) CACHE(S) 1556
SYSTEMSPEICHER 1514
KERN 1560A
TLB 1561A
CACHE(S) 1562A
KERN 1560B
TLB 1561B
CACHE(S) 1562B
KERN 1560C
TLB 1561C
CACHE(S) 1562AC
PROZESSOR 1507

FIG. 15B

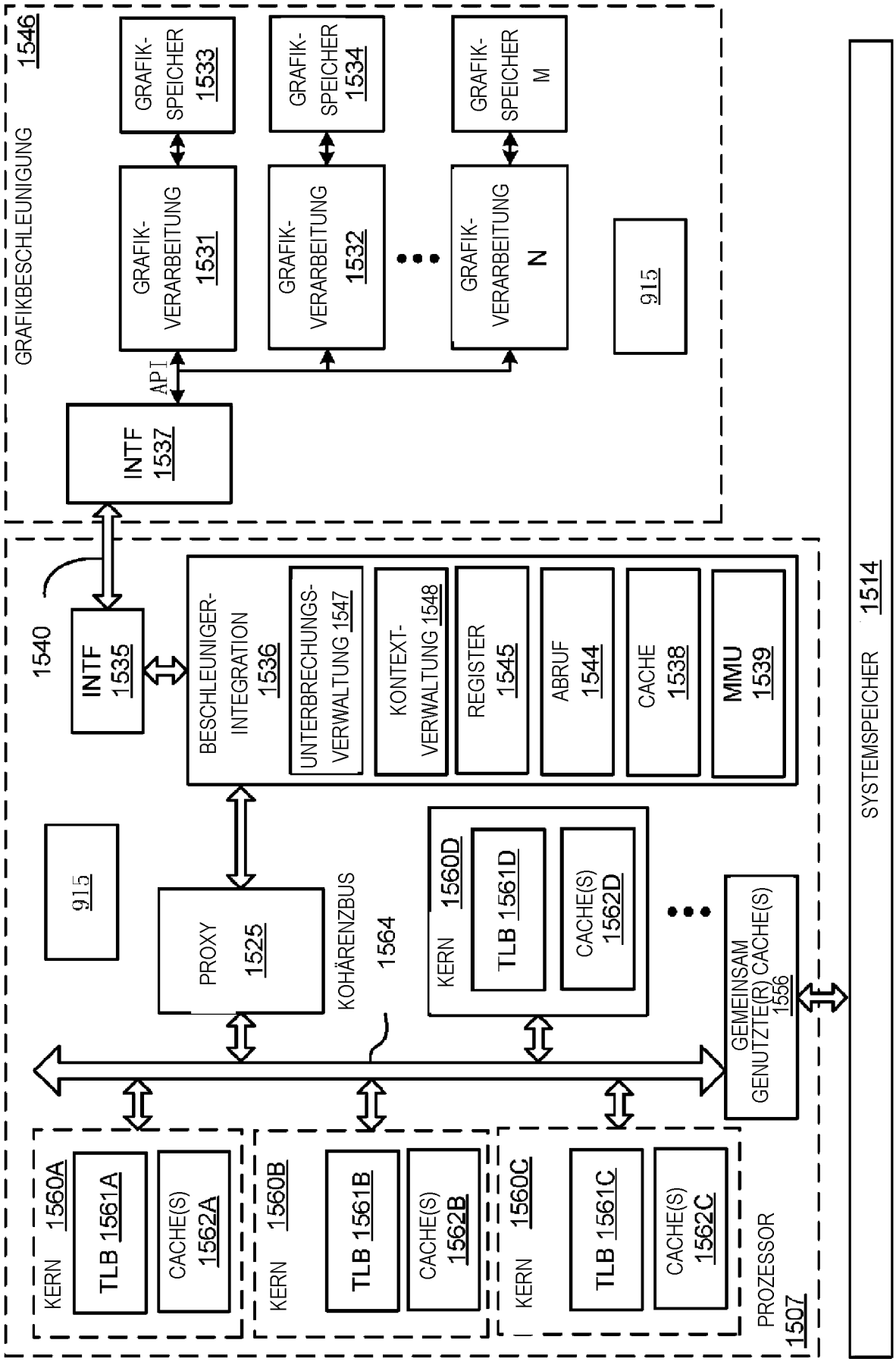

GRAFIKBESCHLEUNIGUNG 1546
GRAFIK-SPEICHER 1533
GRAFIK-SPEICHER 1534
GRAFIK-SPEICHER M
GRAFIK-VERARBEITUNG 1531
GRAFIK-VERARBEITUNG 1532
GRAFIK-VERARBEITUNG N
915
API
INTF 1537
1540
INTF 1535
BESCHLEUNIGER-INTEGRATION 1536
UNTERBRECHUNGS-VERWALTUNG 1547
KONTEXT-VERWALTUNG 1548
REGISTER 1545
ABRUF 1544
CACHE 1538
MMU 1539
915
PROXY 1525
KOHÄRENZBUS 1564
KERN 1560D
TLB 1561D
CACHE(S) 1562D
GEMEINSAM GENUTZTE(R) CACHE(S) 1556
SYSTEMSPEICHER 1514
KERN 1560A
TLB 1561A
CACHE(S) 1562A
KERN 1560B
TLB 1561B
CACHE(S) 1562B
KERN 1560C
TLB 1561C
CACHE(S) 1562C
PROZESSOR 1507

FIG. 15C

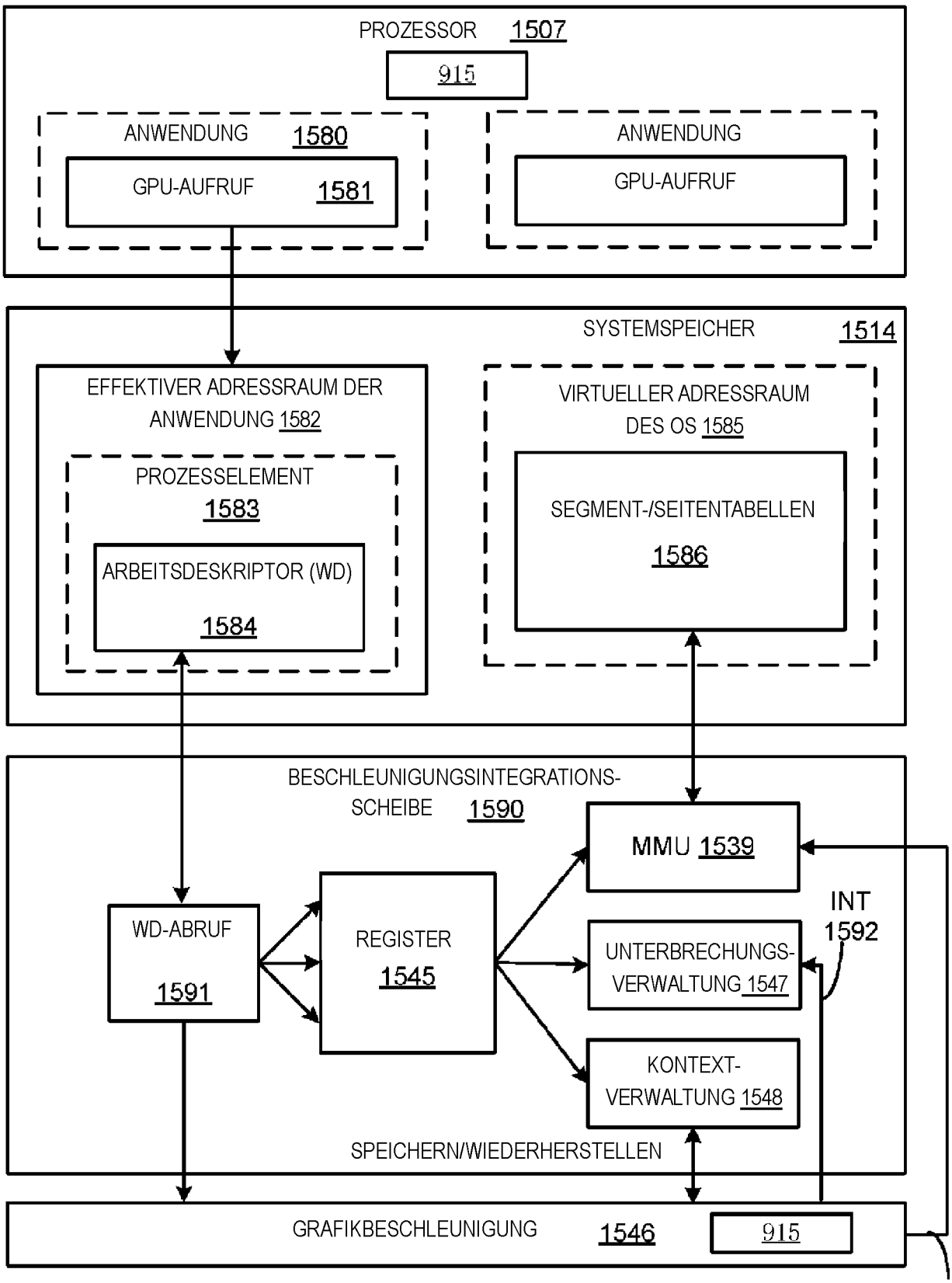

PROZESSOR 1507
915
ANWENDUNG 1580
GPU-AUFRUF 1581
ANWENDUNG
GPU-AUFRUF
SYSTEMSPEICHER 1514
EFFEKTIVER ADRESSRAUM DER ANWENDUNG 1582
PROZESSELEMENT 1583
ARBEITSDESKRIPTOR (WD) 1584
VIRTUELLER ADRESSRAUM DES OS 1585
SEGMENT-/SEITENTABELLEN 1586
BESCHLEUNIGUNGSINTEGRATIONS-SCHEIBE 1590
MMU 1539
INT 1592
WD-ABRUF 1591
REGISTER 1545
UNTERBRECHUNGS-VERWALTUNG 1547
KONTEXT-VERWALTUNG 1548
SPEICHERN/WIEDERHERSTELLEN
GRAFIKBESCHLEUNIGUNG 1546
915
EFFEKTIVE ADRESSE 1593

FIG. 15D

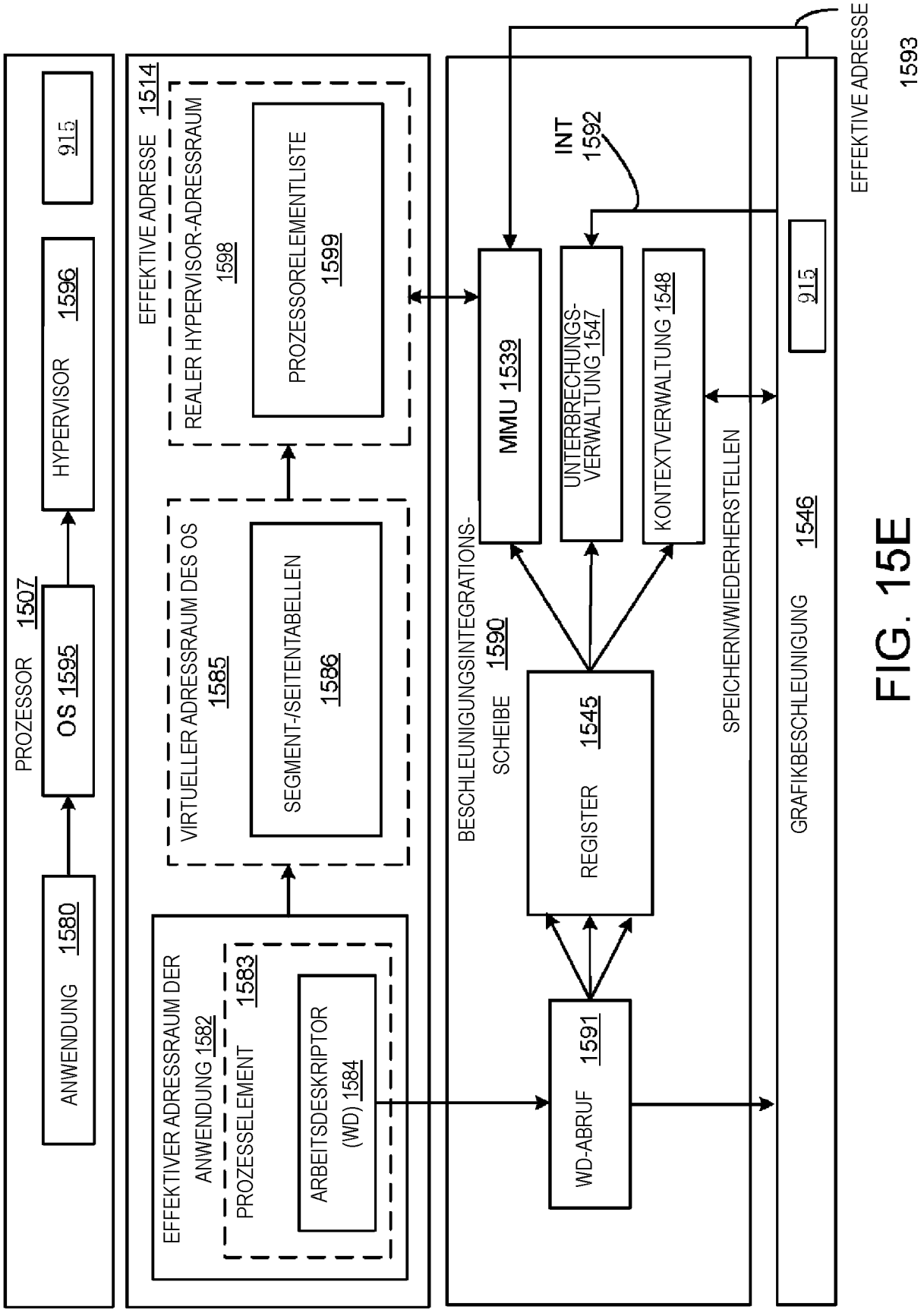

PROZESSOR 1507
ANWENDUNG 1580
OS 1595
HYPERVISOR 1596
915
EFFEKTIVE ADRESSE 1514
REALER HYPERVISOR-ADRESSRAUM 1598
PROZESSORELEMENTLISTE 1599
VIRTUELLER ADRESSRAUM DES OS 1585
SEGMENT-/SEITENTABELLEN 1586
EFFEKTIVER ADRESSRAUM DER ANWENDUNG 1582
PROZESSELEMENT 1583
ARBEITSDESKRIPTOR (WD) 1584
BESCHLEUNIGUNGSINTEGRATIONS-SCHEIBE 1590
MMU 1539
UNTERBRECHUNGS-VERWALTUNG 1547
KONTEXTVERWALTUNG 1548
INT 1592
REGISTER 1545
WD-ABRUF 1591
SPEICHERN/WIEDERHERSTELLEN
GRAFIKBESCHLEUNIGUNG 1546
915
EFFEKTIVE ADRESSE 1593

FIG. 15E

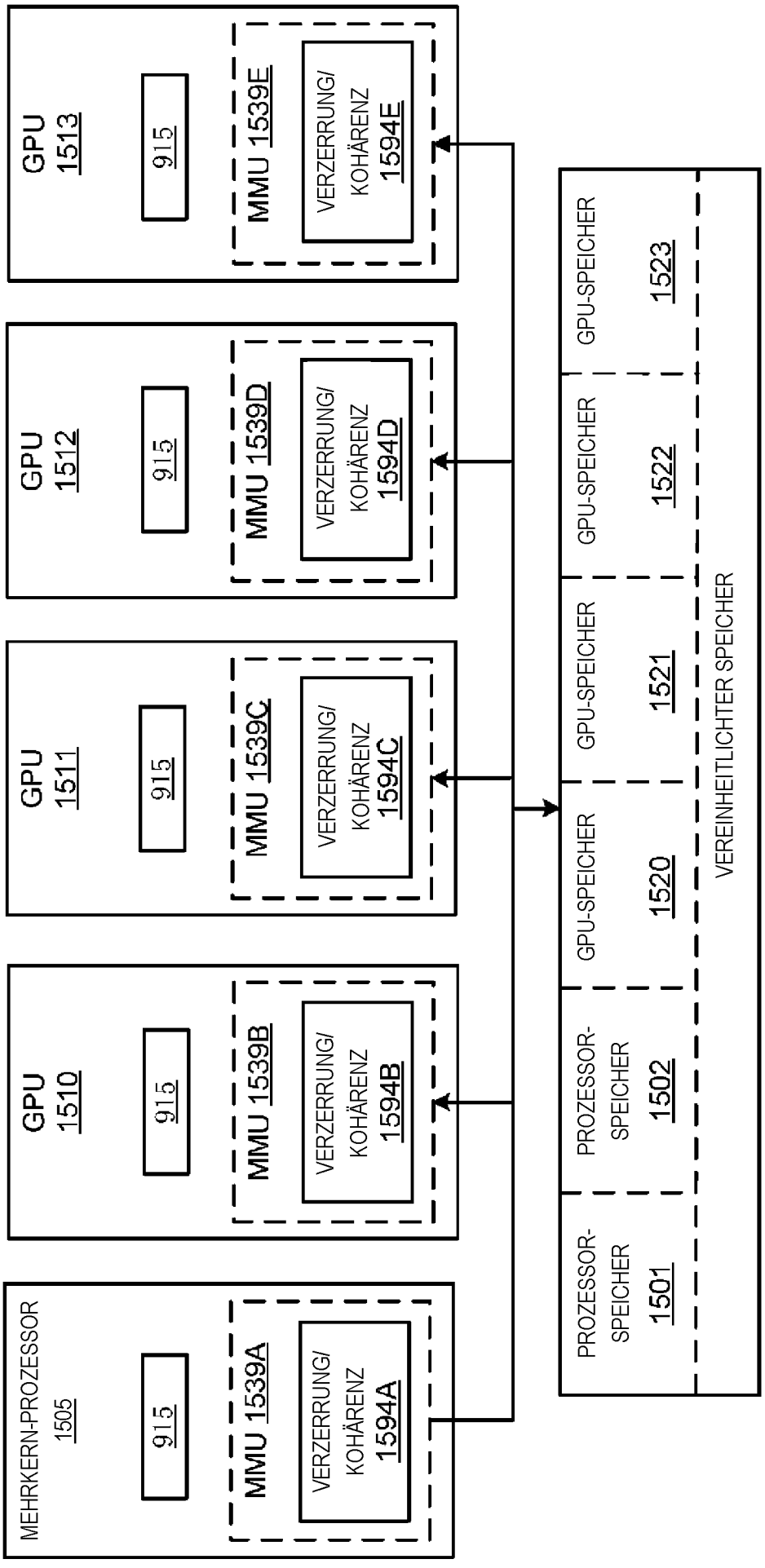

MEHRKERN-PROZESSOR
1505
915
MMU 1539A
VERZERRUNG/
KOHÄRENZ
1594A
GPU
1510
915
MMU 1539B
VERZERRUNG/
KOHÄRENZ
1594B
GPU
1511
915
MMU 1539C
VERZERRUNG/
KOHÄRENZ
1594C
GPU
1512
915
MMU 1539D
VERZERRUNG/
KOHÄRENZ
1594D
GPU
1513
915
MMU 1539E
VERZERRUNG/
KOHÄRENZ
1594E
PROZESSOR-
SPEICHER
1501
PROZESSOR-
SPEICHER
1502
GPU-SPEICHER
1520
GPU-SPEICHER
1521
GPU-SPEICHER
1522
GPU-SPEICHER
1523
VEREINHEITLICHTER SPEICHER

FIG. 15F

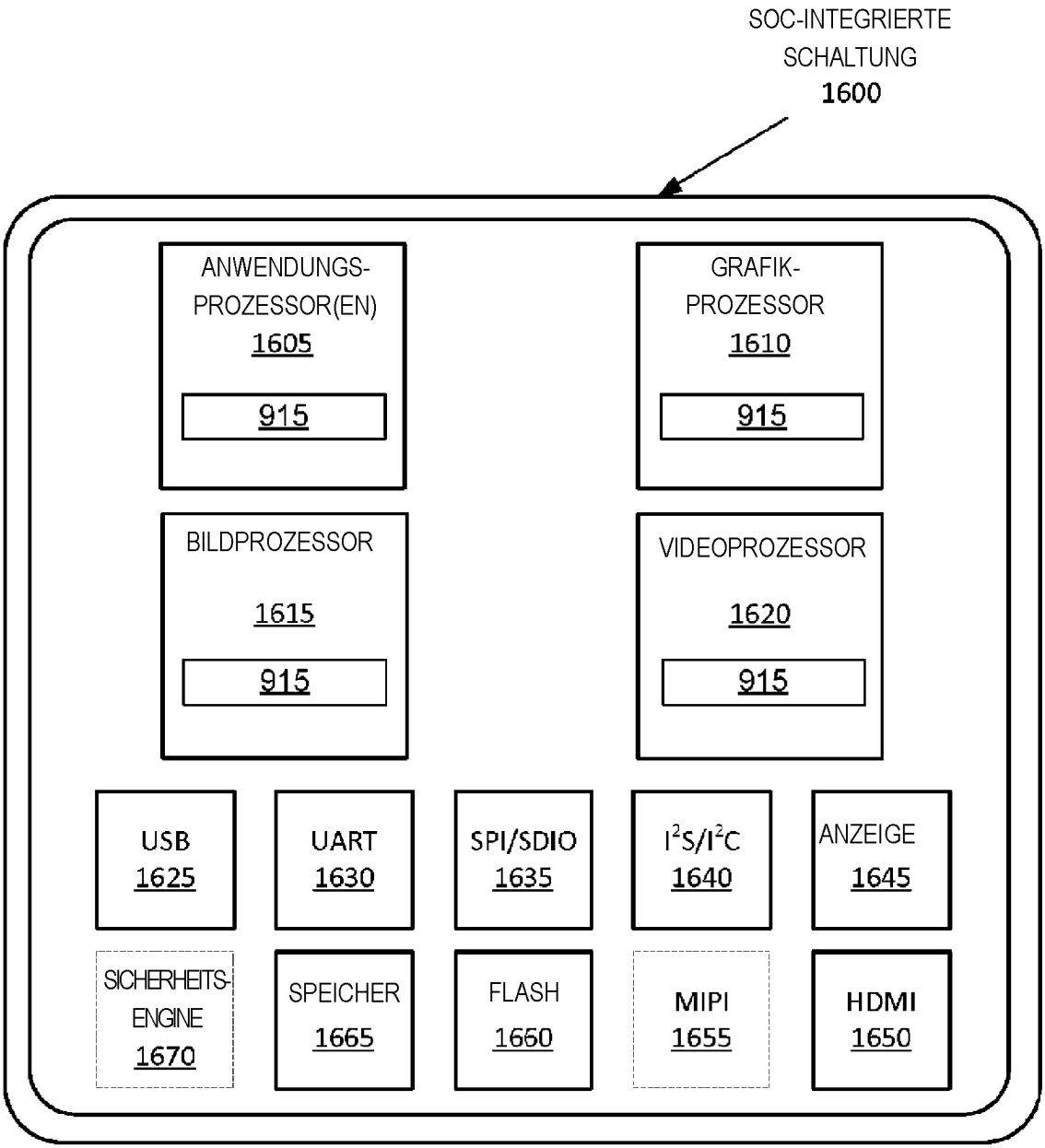

SOC-INTEGRIERTE SCHALTUNG
1600
ANWENDUNGS-PROZESSOR(EN)
1605
915
GRAFIK-PROZESSOR
1610
915
BILDPROZESSOR
1615
915
VIDEOPROZESSOR
1620
915
USB
1625
UART
1630
SPI/SDIO
1635
I2S/I2C
1640
ANZEIGE
1645
SICHERHEITS-ENGINE
1670
SPEICHER
1665
FLASH
1660
MIPI
1655
HDMI
1650

FIG. 16

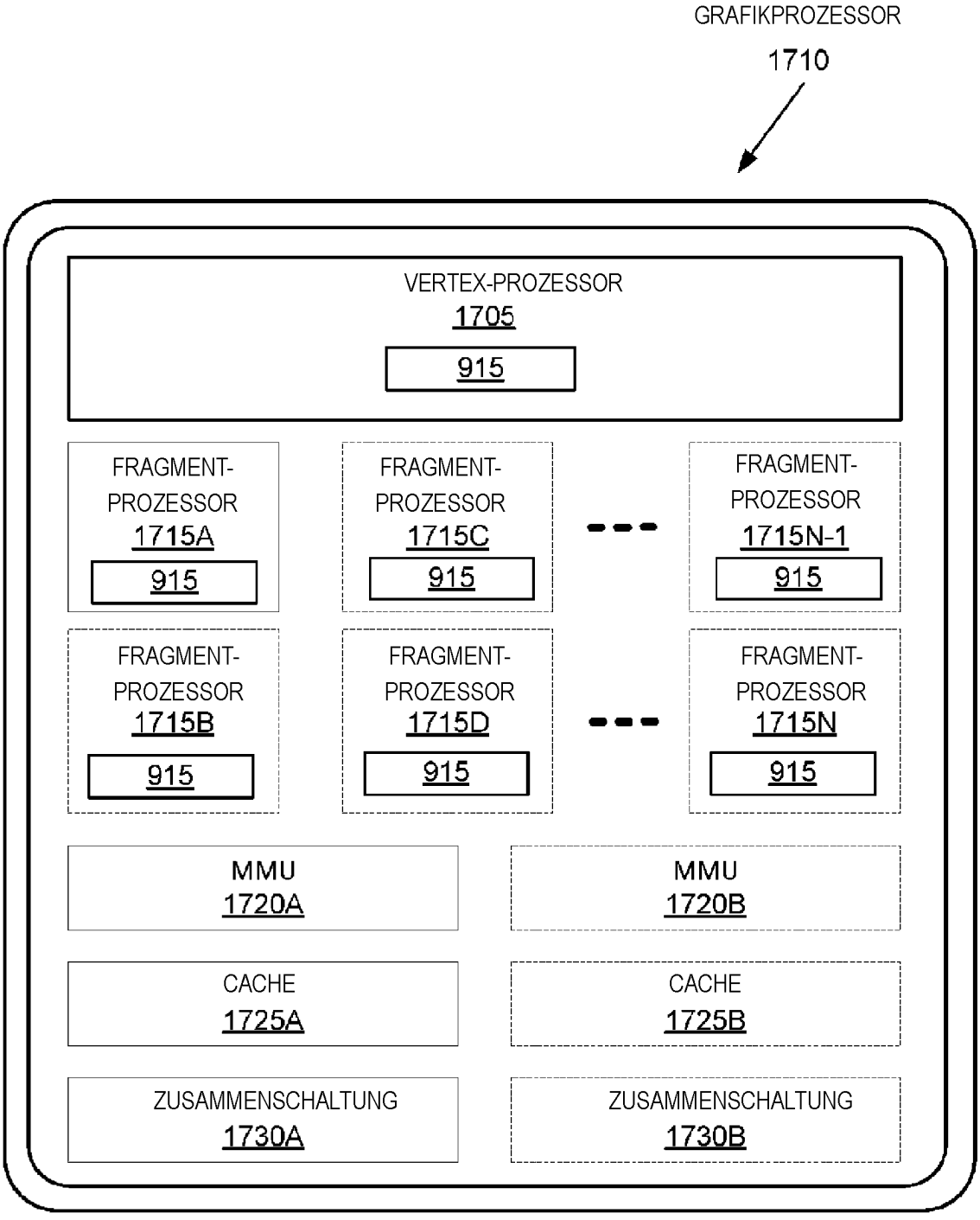

GRAFIKPROZESSOR
1710
VERTEX-PROZESSOR
1705
915
FRAGMENT-PROZESSOR
1715A
915
FRAGMENT-PROZESSOR
1715C
915
FRAGMENT-PROZESSOR
1715N-1
915
FRAGMENT-PROZESSOR
1715B
915
FRAGMENT-PROZESSOR
1715D
915
FRAGMENT-PROZESSOR
1715N
915
MMU
1720A
MMU
1720B
CACHE
1725A
CACHE
1725B
ZUSAMMENSCHALTUNG
1730A
ZUSAMMENSCHALTUNG
1730B

FIG. 17A

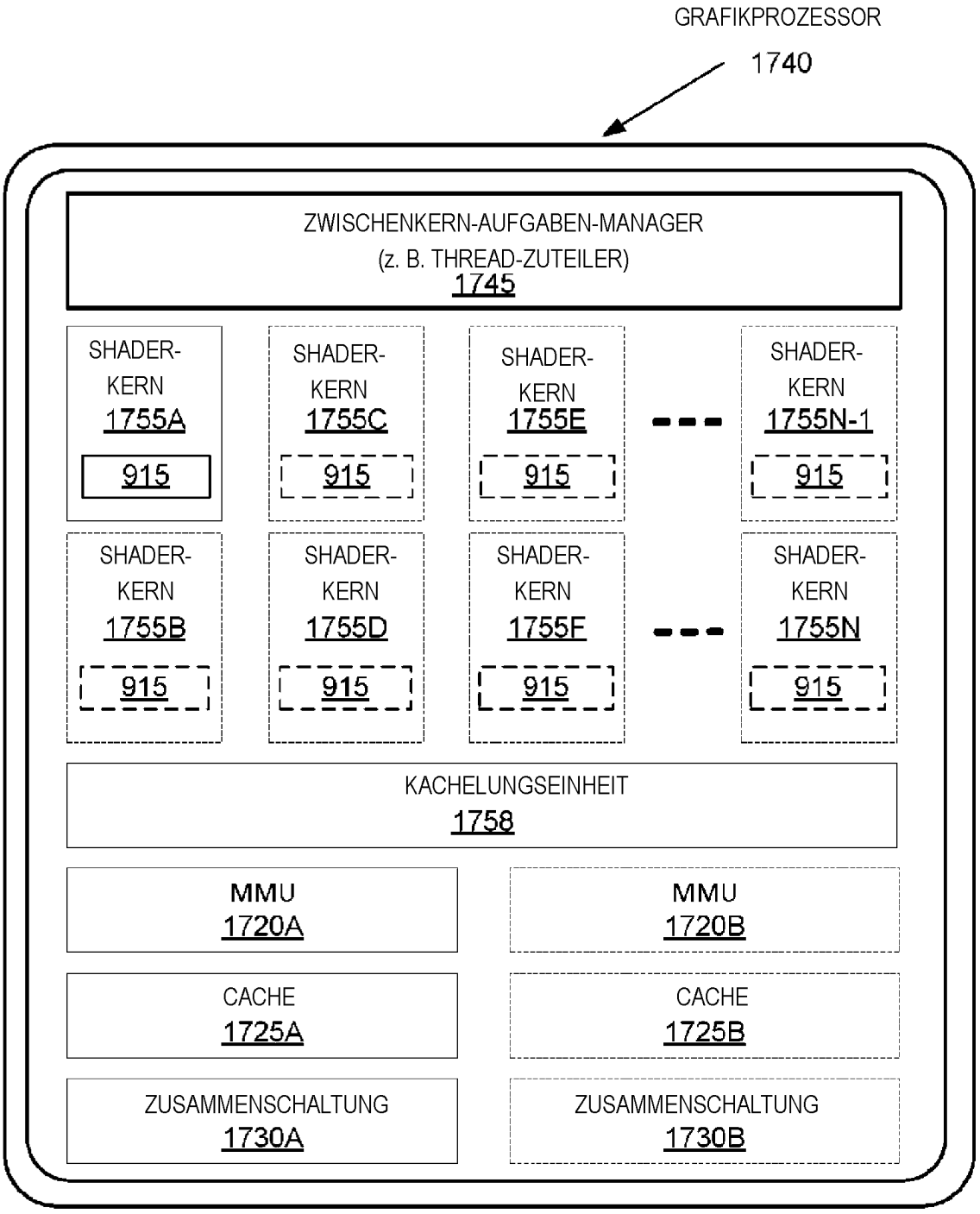

GRAFIKPROZESSOR
1740
ZWISCHENKERN-AUFGABEN-MANAGER
(z. B. THREAD-ZUTEILER)
1745
SHADER-KERN 1755A
915
SHADER-KERN 1755C
915
SHADER-KERN 1755E
915
SHADER-KERN 1755N-1
915
SHADER-KERN 1755B
915
SHADER-KERN 1755D
915
SHADER-KERN 1755F
915
SHADER-KERN 1755N
915
KACHELUNGSEINHEIT
1758
MMU 1720A
MMU 1720B
CACHE 1725A
CACHE 1725B
ZUSAMMENSCHALTUNG 1730A
ZUSAMMENSCHALTUNG 1730B

FIG. 17B

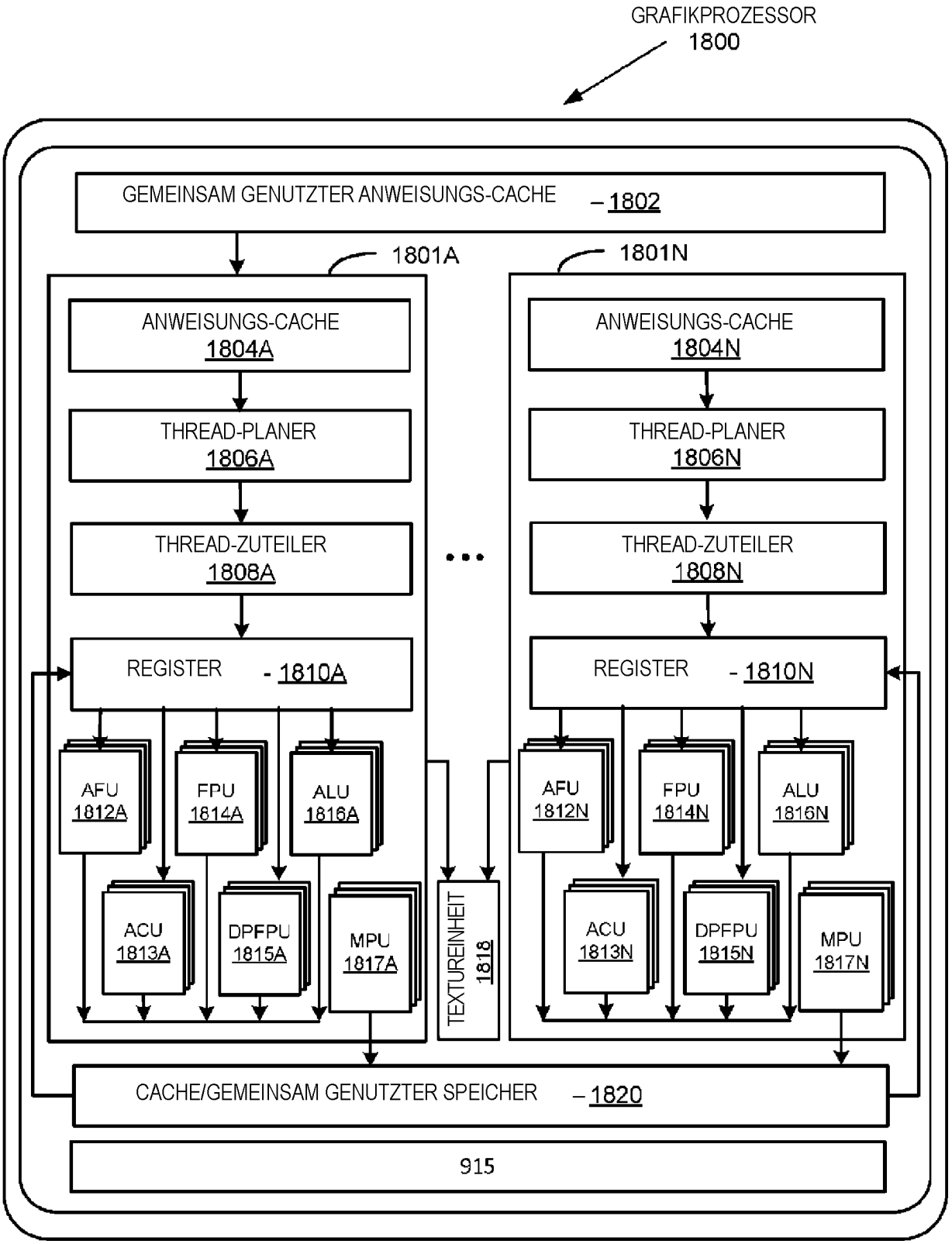

GRAFIKPROZESSOR
1800
GEMEINSAM GENUTZTER ANWEISUNGS-CACHE – 1802
1801A
1801N
ANWEISUNGS-CACHE
1804A
ANWEISUNGS-CACHE
1804N
THREAD-PLANER
1806A
THREAD-PLANER
1806N
THREAD-ZUTEILER
1808A
THREAD-ZUTEILER
1808N
REGISTER - 1810A
REGISTER - 1810N
AFU
1812A
FPU
1814A
ALU
1816A
AFU
1812N
FPU
1814N
ALU
1816N
ACU
1813A
DPFPU
1815A
MPU
1817A
TEXTUREINHEIT
1818
ACU
1813N
DPFPU
1815N
MPU
1817N
CACHE/GEMEINSAM GENUTZTER SPEICHER – 1820
915

FIG. 18A

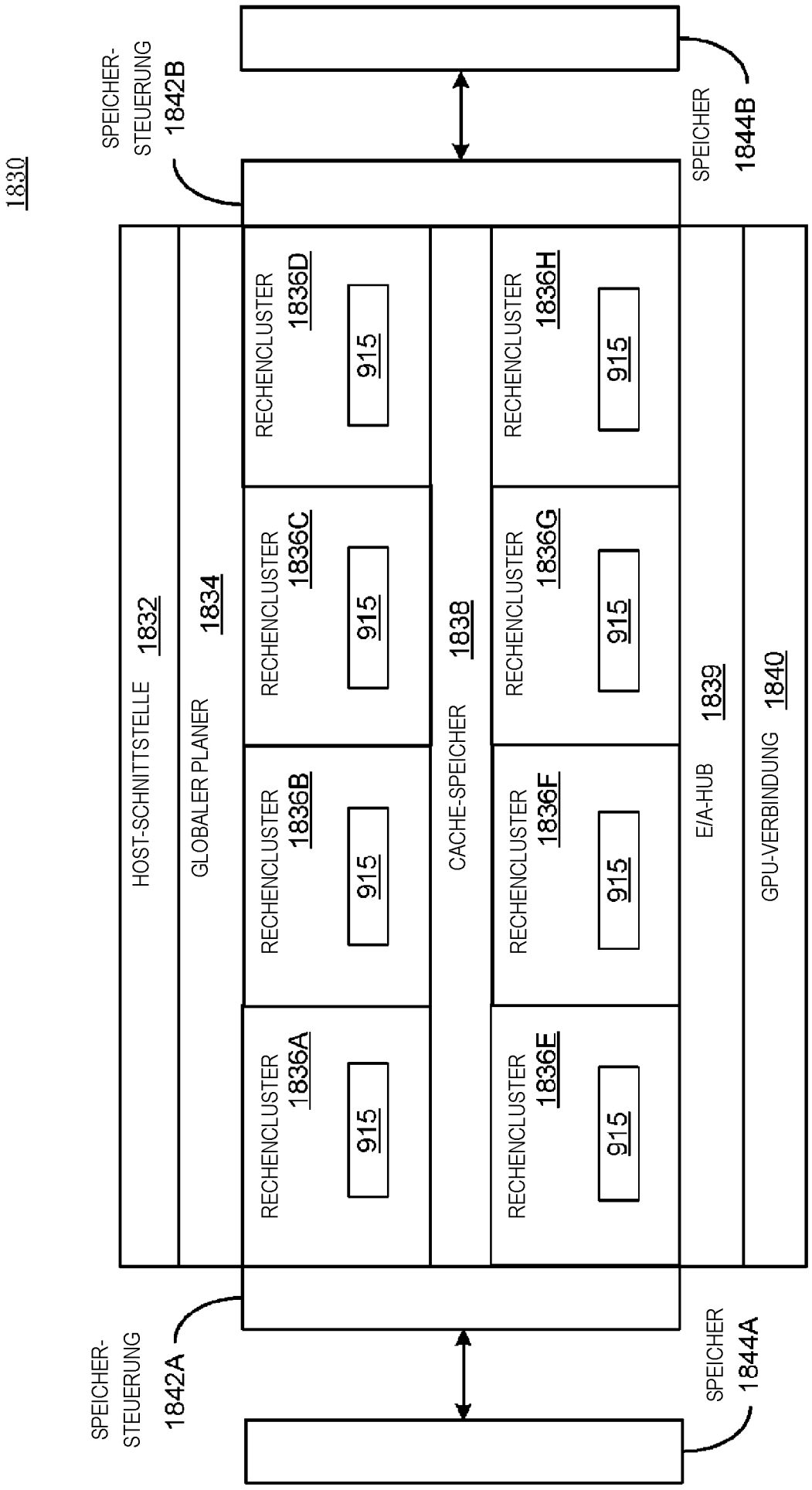

1830
SPEICHER-STEUERUNG 1842B
SPEICHER 1844B
HOST-SCHNITTSTELLE 1832
GLOBALER PLANER 1834
RECHENCLUSTER 1836A
RECHENCLUSTER 1836B
RECHENCLUSTER 1836C
RECHENCLUSTER 1836D
RECHENCLUSTER 1836E
RECHENCLUSTER 1836F
RECHENCLUSTER 1836G
RECHENCLUSTER 1836H
915
CACHE-SPEICHER 1838
E/A-HUB 1839
GPU-VERBINDUNG 1840
SPEICHER-STEUERUNG 1842A
SPEICHER 1844A

FIG. 18B

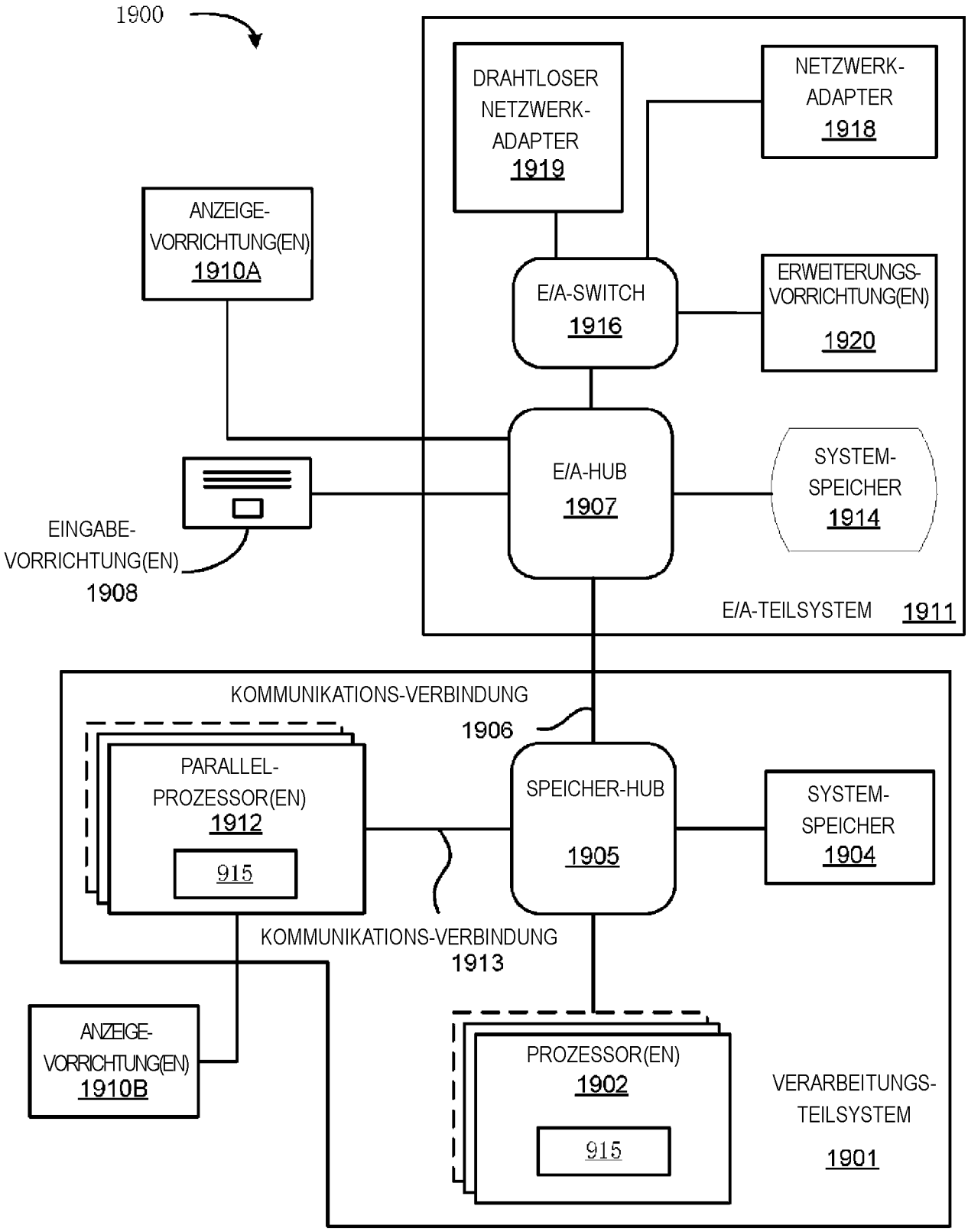

1900
DRAHTLOSER NETZWERK-ADAPTER 1919
NETZWERK-ADAPTER 1918
ANZEIGE-VORRICHTUNG(EN) 1910A
E/A-SWITCH 1916
ERWEITERUNGS-VORRICHTUNG(EN) 1920
E/A-HUB 1907
SYSTEM-SPEICHER 1914
EINGABE-VORRICHTUNG(EN) 1908
E/A-TEILSYSTEM 1911
KOMMUNIKATIONS-VERBINDUNG 1906
PARALLEL-PROZESSOR(EN) 1912
915
SPEICHER-HUB 1905
SYSTEM-SPEICHER 1904
KOMMUNIKATIONS-VERBINDUNG 1913
ANZEIGE-VORRICHTUNG(EN) 1910B
PROZESSOR(EN) 1902
915
VERARBEITUNGS-TEILSYSTEM 1901

FIG. 19

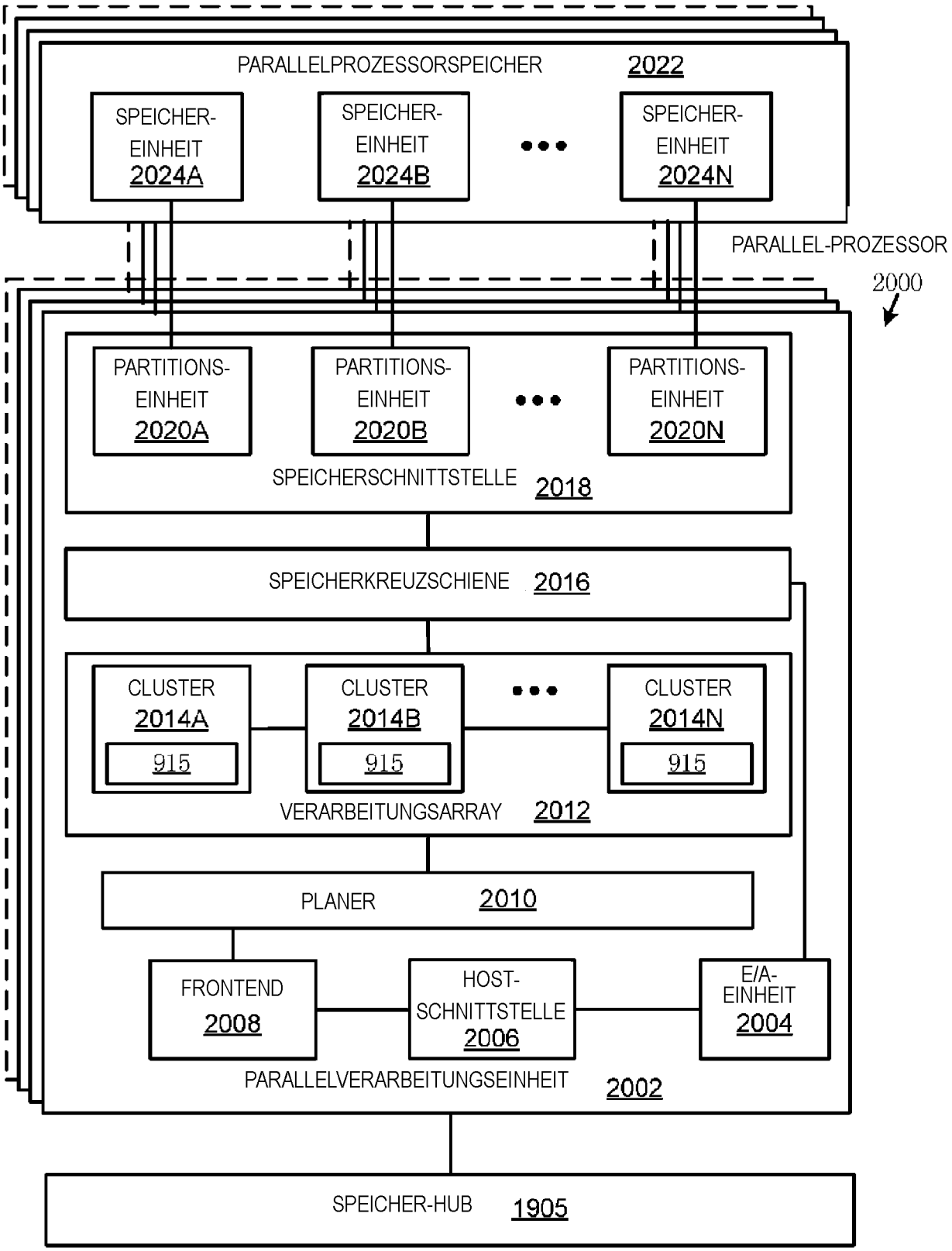

PARALLELPROZESSORSPEICHER 2022
SPEICHER-EINHEIT 2024A
SPEICHER-EINHEIT 2024B
SPEICHER-EINHEIT 2024N
PARALLEL-PROZESSOR 2000
PARTITIONS-EINHEIT 2020A
PARTITIONS-EINHEIT 2020B
PARTITIONS-EINHEIT 2020N
SPEICHERSCHNITTSTELLE 2018
SPEICHERKREUZSCHIENE 2016
CLUSTER 2014A
CLUSTER 2014B
CLUSTER 2014N
915
915
915
VERARBEITUNGSARRAY 2012
PLANER 2010
FRONTEND 2008
HOST-SCHNITTSTELLE 2006
E/A-EINHEIT 2004
PARALLELVERARBEITUNGSEINHEIT 2002
SPEICHER-HUB 1905

FIG. 20A

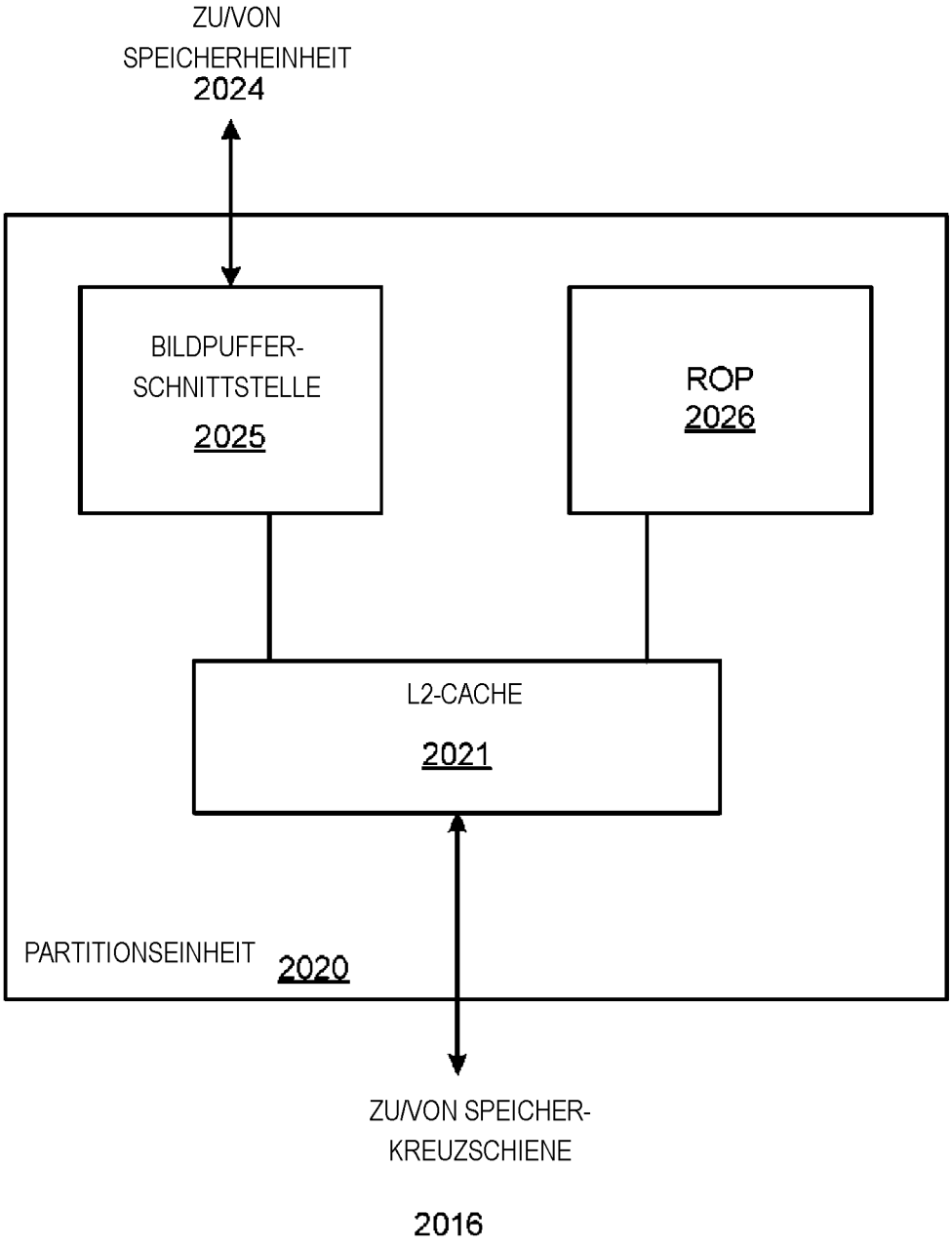

ZU/VON
SPEICHERHEINHEIT
2024
BILDPUFFER-
SCHNITTSTELLE
2025
ROP
2026
L2-CACHE
2021
PARTITIONSEINHEIT
2020
ZU/VON SPEICHER-
KREUZSCHIENE
2016

FIG. 20B

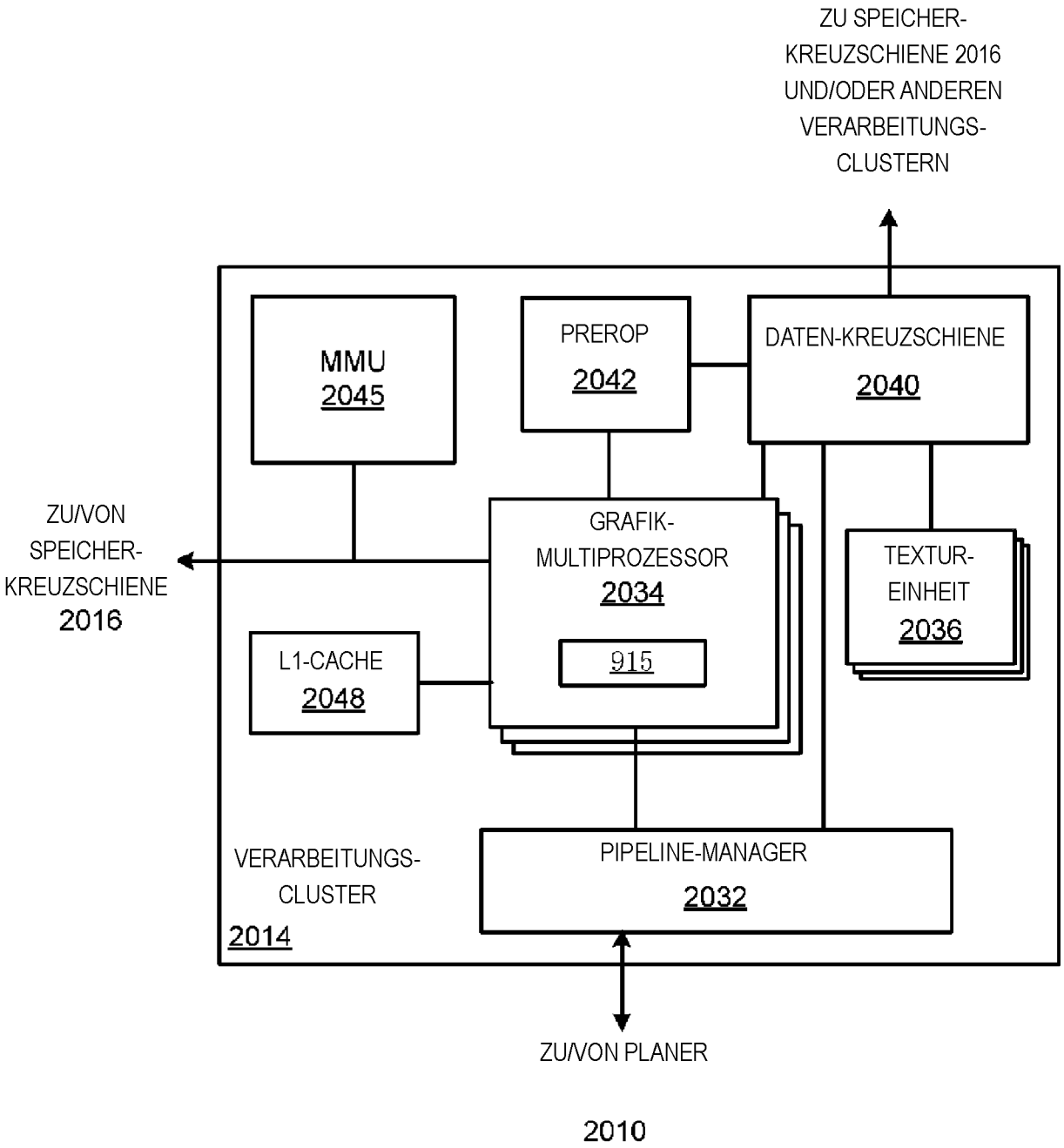

ZU SPEICHER-
KREUZSCHIENE 2016
UND/ODER ANDEREN
VERARBEITUNGS-
CLUSTERN
MMU
2045
PREROP
2042
DATEN-KREUZSCHIENE
2040
ZU/VON
SPEICHER-
KREUZSCHIENE
2016
GRAFIK-
MULTIPROZESSOR
2034
915
TEXTUR-
EINHEIT
2036
L1-CACHE
2048
VERARBEITUNGS-
CLUSTER
2014
PIPELINE-MANAGER
2032
ZU/VON PLANER
2010

FIG. 20C

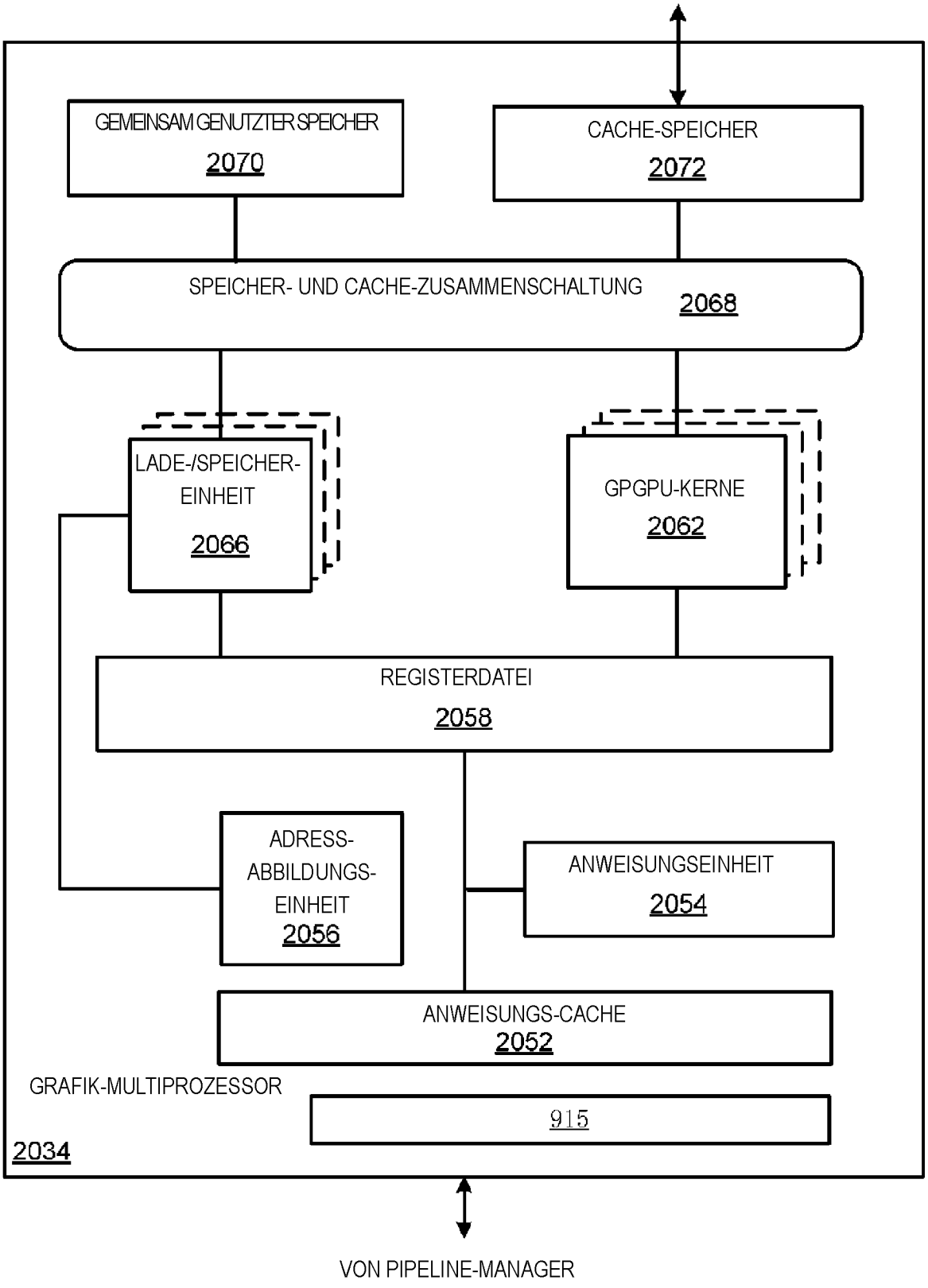

GEMEINSAM GENUTZTER SPEICHER
2070
CACHE-SPEICHER
2072
SPEICHER- UND CACHE-ZUSAMMENSCHALTUNG
2068
LADE-/SPEICHER-EINHEIT
2066
GPGPU-KERNE
2062
REGISTERDATEI
2058
ADRESS-ABBILDUNGS-EINHEIT
2056
ANWEISUNGSEINHEIT
2054
ANWEISUNGS-CACHE
2052
GRAFIK-MULTIPROZESSOR
915
2034
VON PIPELINE-MANAGER
2032

FIG. 20D

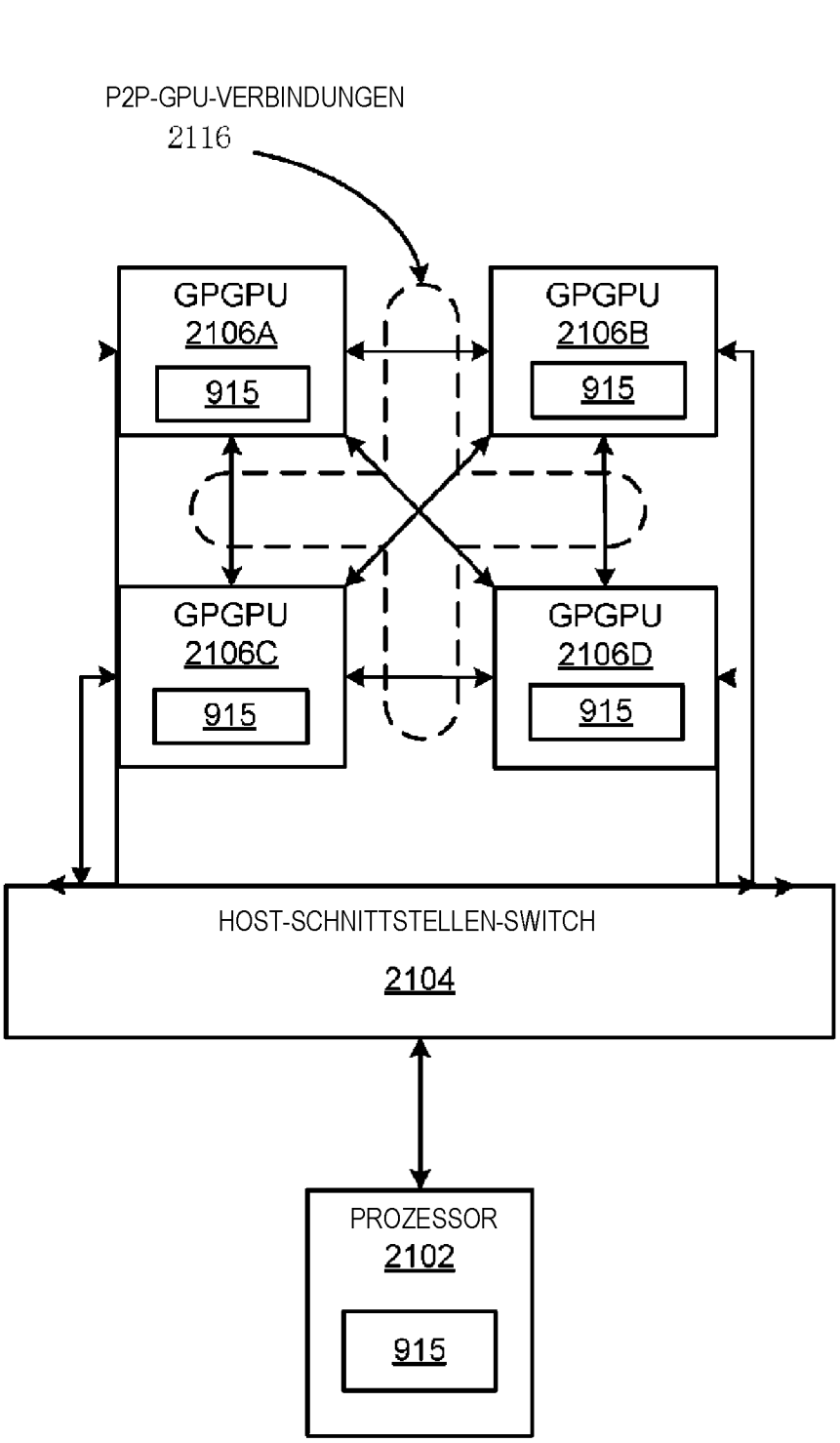

2100
P2P-GPU-VERBINDUNGEN
2116
GPGPU
2106A
915
GPGPU
2106B
915
GPGPU
2106C
915
GPGPU
2106D
915
HOST-SCHNITTSTELLEN-SWITCH
2104
PROZESSOR
2102
915

FIG. 21

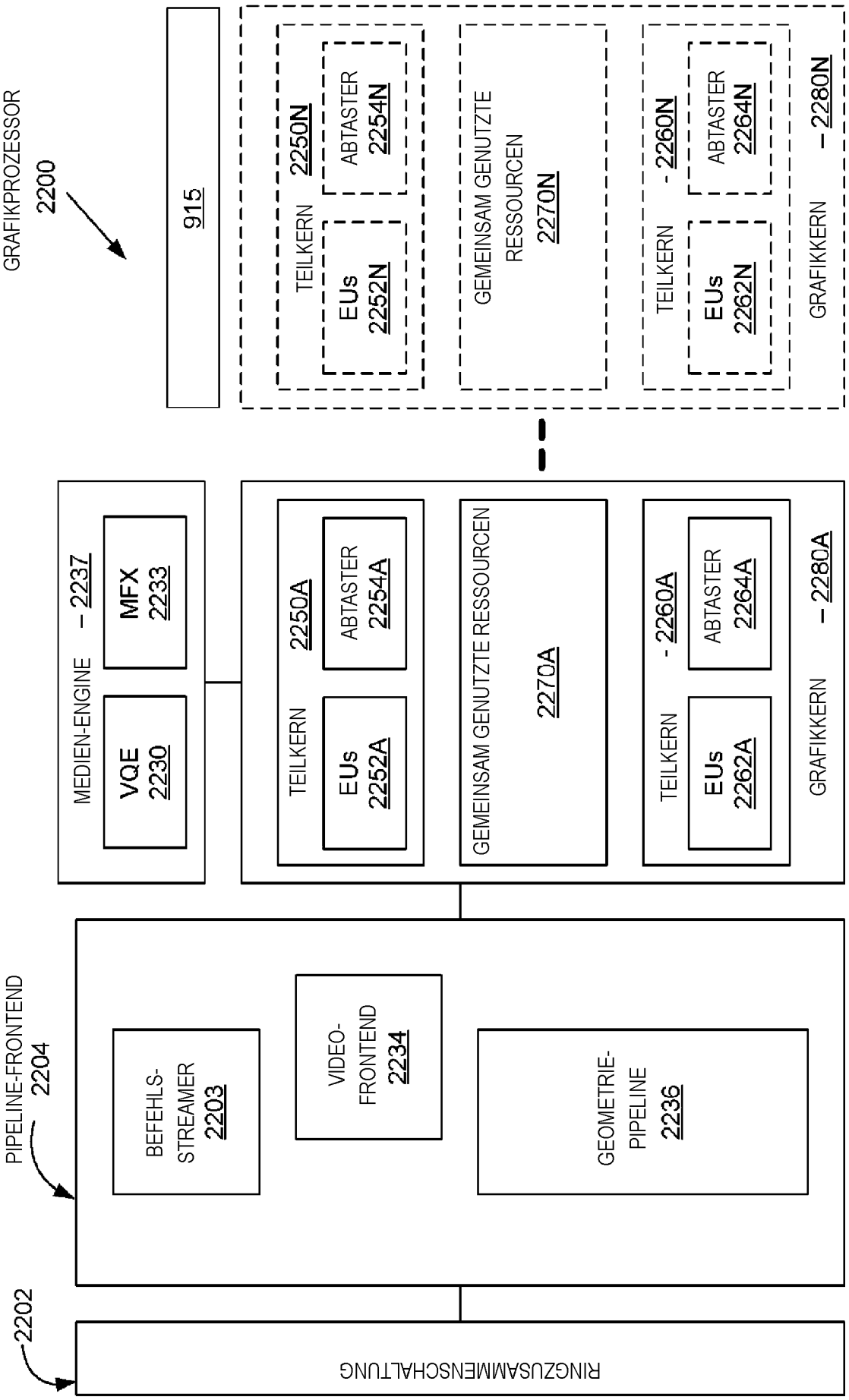

GRAFIKPROZESSOR
2200
915
TEILKERN 2250N
EUs 2252N
ABTASTER 2254N
GEMEINSAM GENUTZTE RESSOURCEN 2270N
TEILKERN 2260N
EUs 2262N
ABTASTER 2264N
GRAFIKKERN 2280N
MEDIEN-ENGINE 2237
VQE 2230
MFX 2233
TEILKERN 2250A
EUs 2252A
ABTASTER 2254A
GEMEINSAM GENUTZTE RESSOURCEN 2270A
TEILKERN 2260A
EUs 2262A
ABTASTER 2264A
GRAFIKKERN 2280A
PIPELINE-FRONTEND 2204
BEFEHLS-STREAMER 2203
VIDEO-FRONTEND 2234
GEOMETRIE-PIPELINE 2236
2202
RINGZUSAMMENSCHALTUNG

FIG. 22

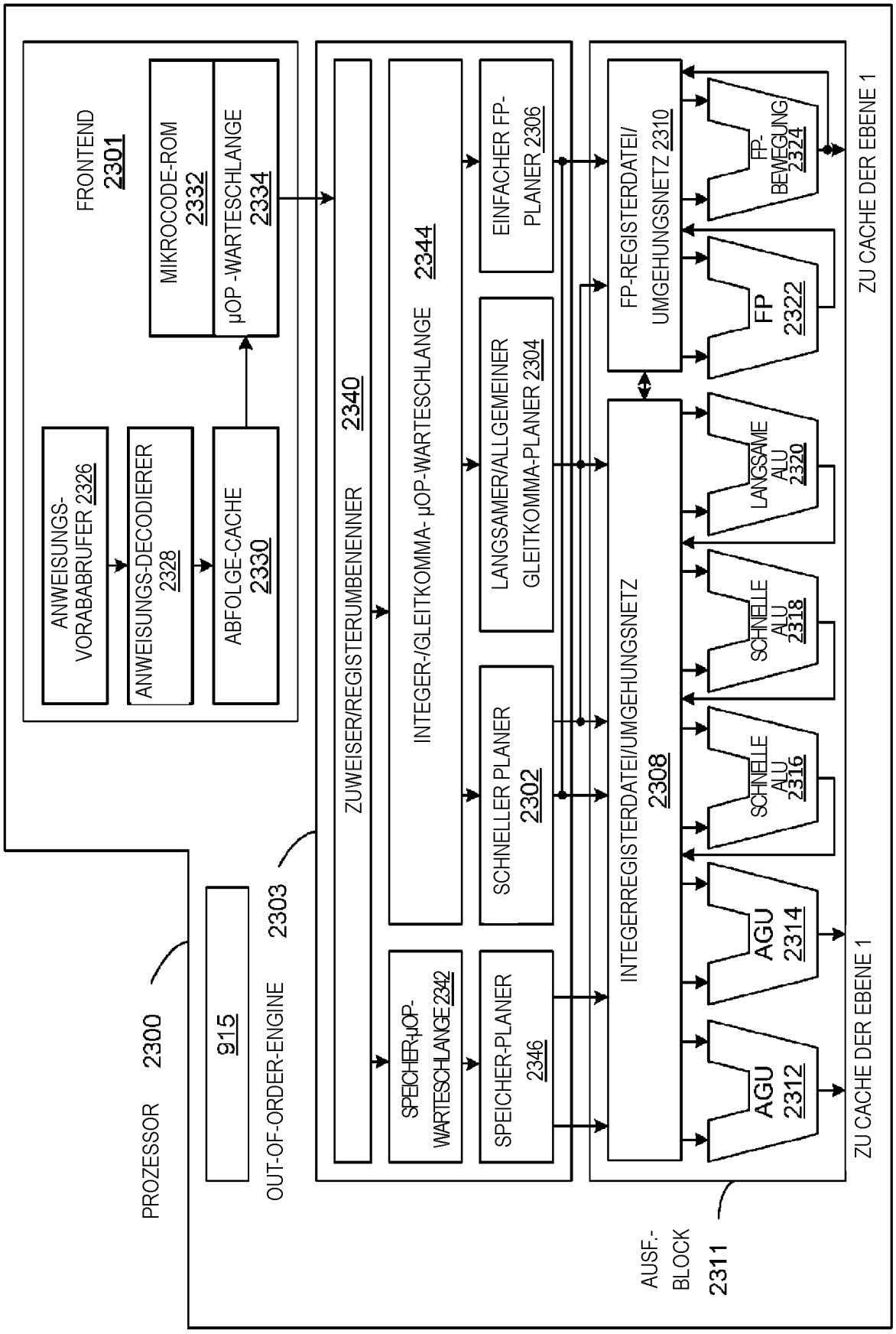

PROZESSOR 2300
915
OUT-OF-ORDER-ENGINE 2303
FRONTEND 2301
ANWEISUNGS-VORABABRUFER 2326
ANWEISUNGS-DECODIERER 2328
ABFOLGE-CACHE 2330
MIKROCODE-ROM 2332
µOP -WARTESCHLANGE 2334
ZUWEISER/REGISTERUMBENENNER 2340
INTEGER-/GLEITKOMMA- µOP-WARTESCHLANGE 2344
SPEICHER-µOP-WARTESCHLANGE 2342
SCHNELLER PLANER 2302
SPEICHER-PLANER 2346
LANGSAMER/ALLGEMEINER GLEITKOMMA-PLANER 2304
EINFACHER FP-PLANER 2306
FP-REGISTERDATEI/UMGEHUNGSNETZ 2310
INTEGERREGISTERDATEI/UMGEHUNGSNETZ 2308
AGU 2312
AGU 2314
SCHNELLE ALU 2316
SCHNELLE ALU 2318
LANGSAME ALU 2320
FP 2322
FP-BEWEGUNG 2324
ZU CACHE DER EBENE 1
ZU CACHE DER EBENE 1
AUSF.-BLOCK 2311

FIG. 23

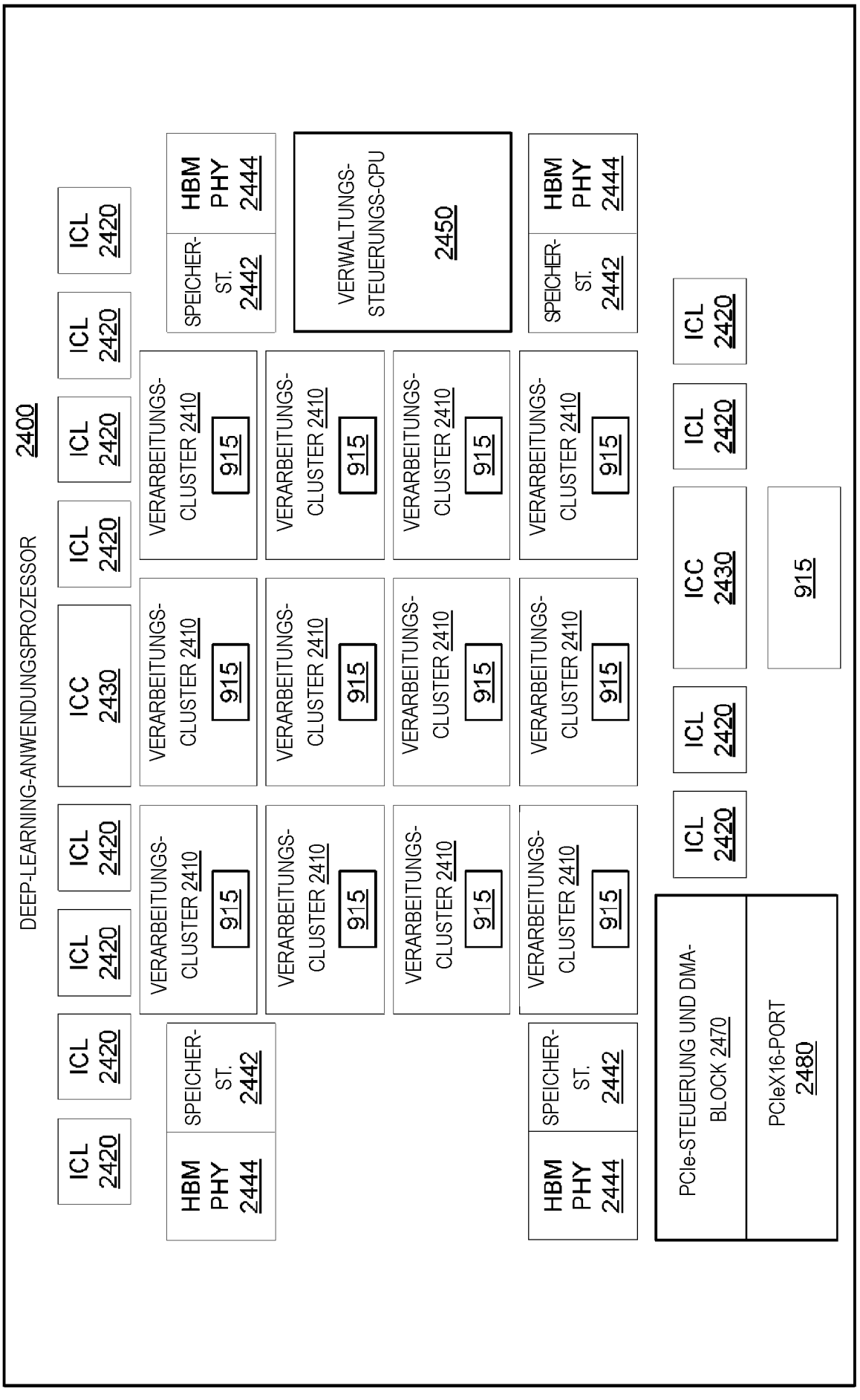

DEEP-LEARNING-ANWENDUNGSPROZESSOR 2400
ICL 2420
ICC 2430
VERARBEITUNGS-CLUSTER 2410
915
SPEICHER-ST. 2442
HBM PHY 2444
VERWALTUNGS-STEUERUNGS-CPU 2450
PCIe-STEUERUNG UND DMA-BLOCK 2470
PCIeX16-PORT 2480

FIG. 24

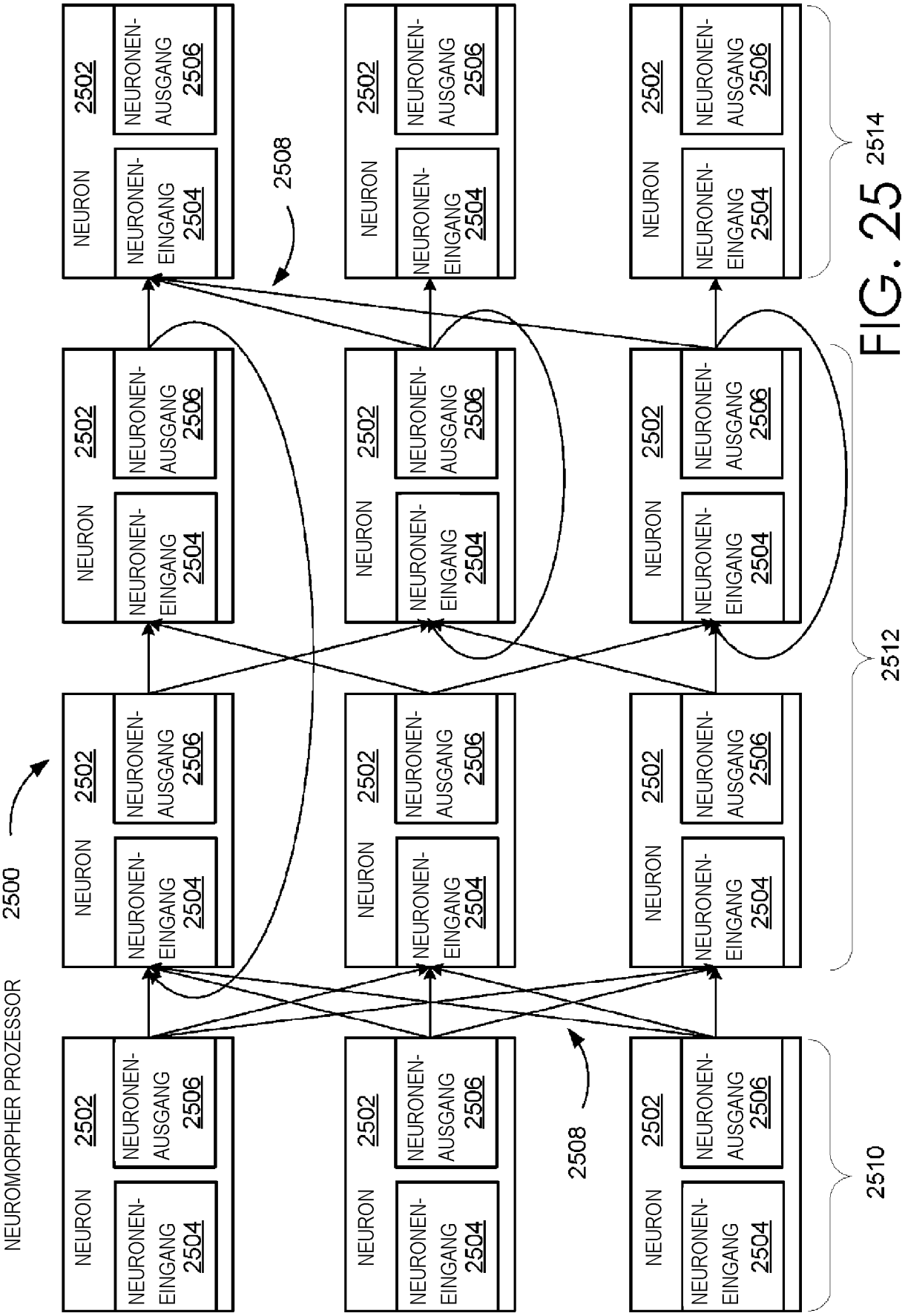

NEUROMORPHER PROZESSOR 2500
NEURON 2502
NEURONEN-EINGANG 2504
NEURONEN-AUSGANG 2506
2508
2510
2512
2514

FIG. 25

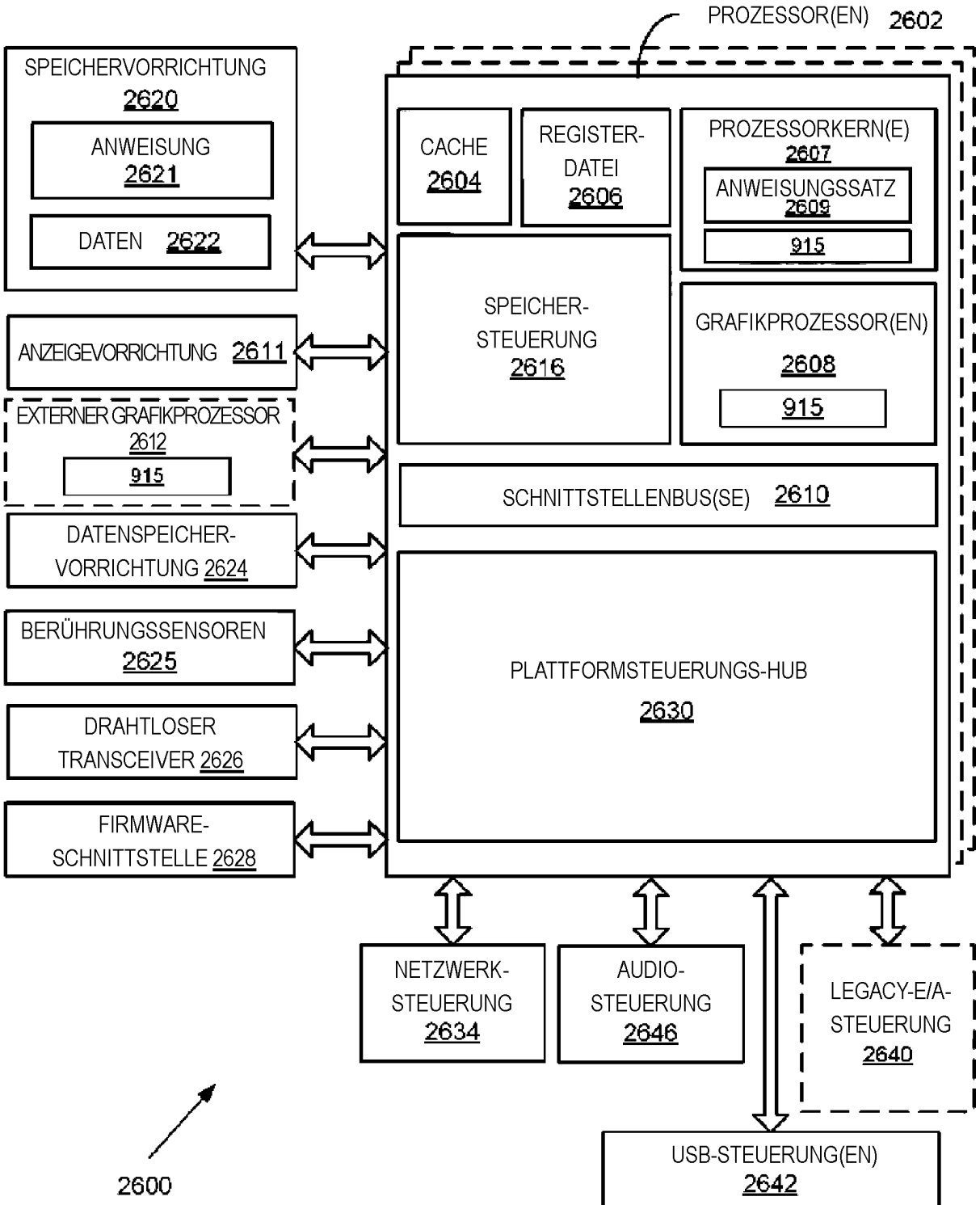

PROZESSOR(EN) 2602
SPEICHERVORRICHTUNG 2620
ANWEISUNG 2621
DATEN 2622
CACHE 2604
REGISTER-DATEI 2606
PROZESSORKERN(E) 2607
ANWEISUNGSSATZ 2609
915
SPEICHER-STEUERUNG 2616
GRAFIKPROZESSOR(EN) 2608
915
ANZEIGEVORRICHTUNG 2611
EXTERNER GRAFIKPROZESSOR 2612
915
SCHNITTSTELLENBUS(SE) 2610
DATENSPEICHER-VORRICHTUNG 2624
BERÜHRUNGSSENSOREN 2625
PLATTFORMSTEUERUNGS-HUB 2630
DRAHTLOSER TRANSCEIVER 2626
FIRMWARE-SCHNITTSTELLE 2628
NETZWERK-STEUERUNG 2634
AUDIO-STEUERUNG 2646
LEGACY-E/A-STEUERUNG 2640
2600
USB-STEUERUNG(EN) 2642
TASTATUR/MAUS 2643
KAMERA 2644

FIG. 26

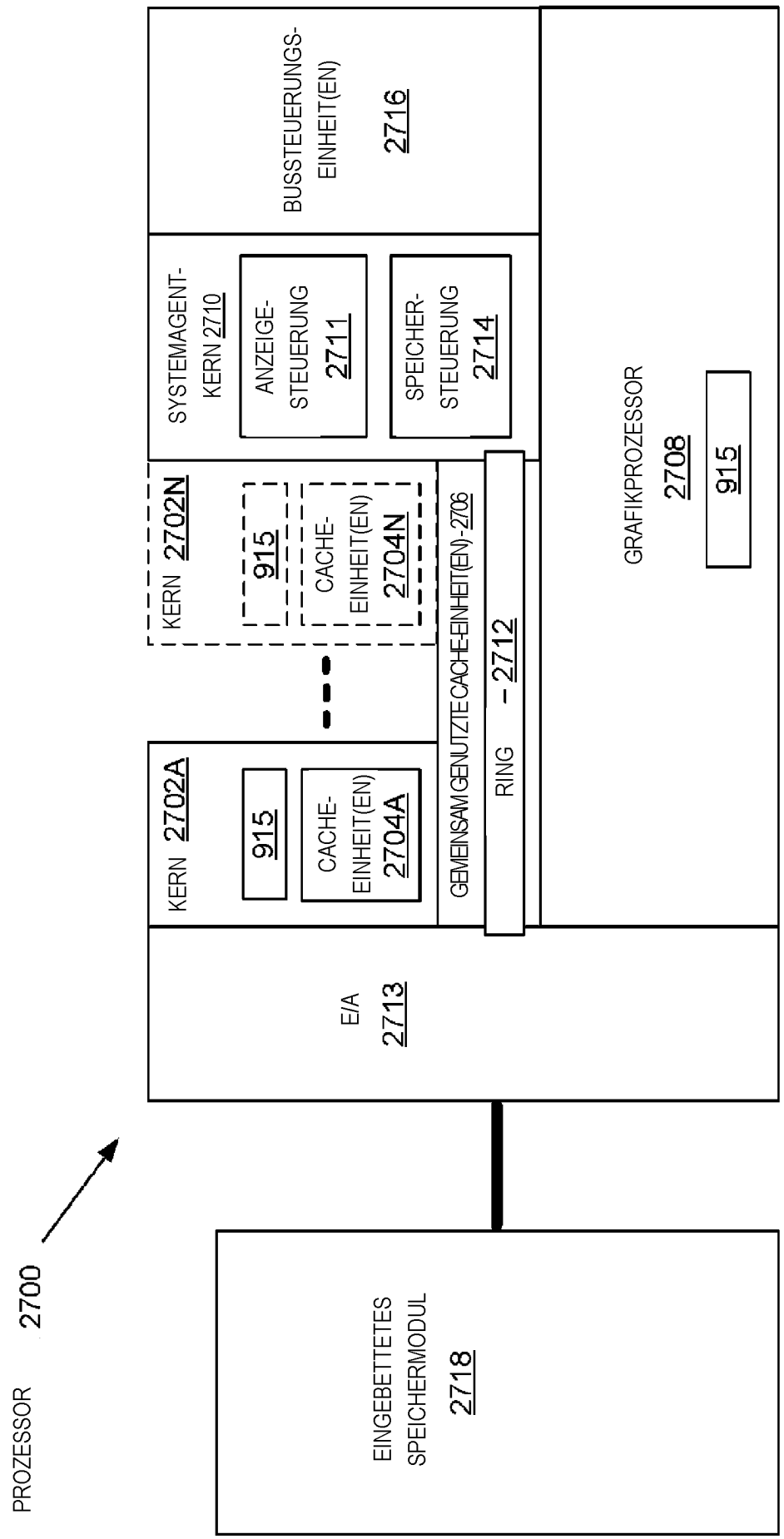

PROZESSOR
2700
EINGEBETTETES SPEICHERMODUL
2718
E/A
2713
KERN 2702A
915
CACHE-EINHEIT(EN) 2704A
KERN 2702N
915
CACHE-EINHEIT(EN) 2704N
GEMEINSAM GENUTZTE CACHE-EINHEIT(EN) 2706
RING 2712
SYSTEMAGENT-KERN 2710
ANZEIGE-STEUERUNG 2711
SPEICHER-STEUERUNG 2714
BUSSTEUERUNGS-EINHEIT(EN) 2716
GRAFIKPROZESSOR 2708
915

FIG. 27

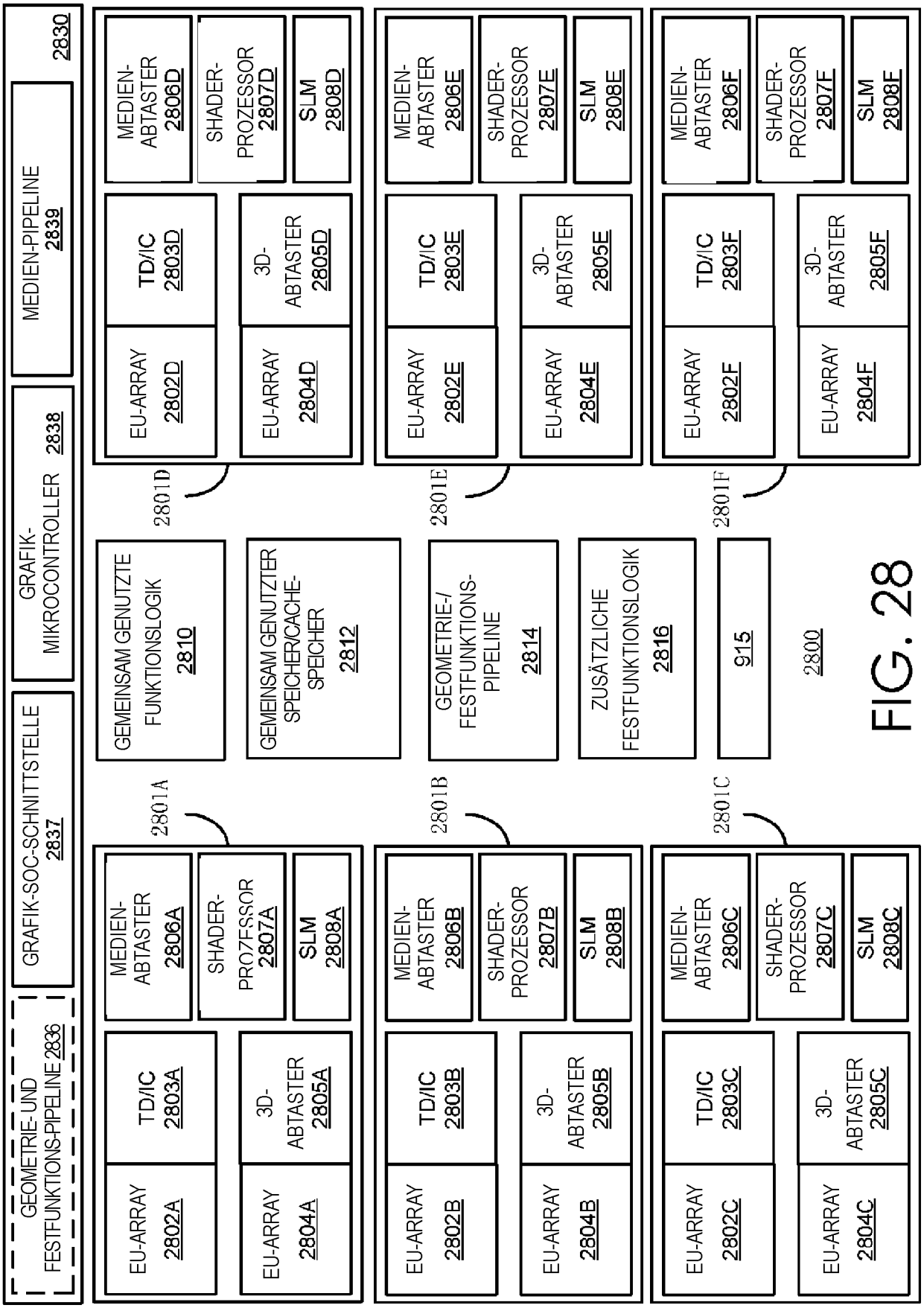

GEOMETRIE- UND FESTFUNKTIONS-PIPELINE 2836
GRAFIK-SOC-SCHNITTSTELLE 2837
GRAFIK-MIKROCONTROLLER 2838
MEDIEN-PIPELINE 2839
2830
2801A
2801B
2801C
2801D
2801E
2801F
EU-ARRAY 2802A
TD/IC 2803A
MEDIEN-ABTASTER 2806A
SHADER-PROZESSOR 2807A
3D-ABTASTER 2805A
EU-ARRAY 2804A
SLM 2808A
EU-ARRAY 2802B
TD/IC 2803B
MEDIEN-ABTASTER 2806B
SHADER-PROZESSOR 2807B
3D-ABTASTER 2805B
EU-ARRAY 2804B
SLM 2808B
EU-ARRAY 2802C
TD/IC 2803C
MEDIEN-ABTASTER 2806C
SHADER-PROZESSOR 2807C
3D-ABTASTER 2805C
EU-ARRAY 2804C
SLM 2808C
EU-ARRAY 2802D
TD/IC 2803D
MEDIEN-ABTASTER 2806D
SHADER-PROZESSOR 2807D
3D-ABTASTER 2805D
EU-ARRAY 2804D
SLM 2808D
EU-ARRAY 2802E
TD/IC 2803E
MEDIEN-ABTASTER 2806E
SHADER-PROZESSOR 2807E
3D-ABTASTER 2805E
EU-ARRAY 2804E
SLM 2808E
EU-ARRAY 2802F
TD/IC 2803F
MEDIEN-ABTASTER 2806F
SHADER-PROZESSOR 2807F
3D-ABTASTER 2805F
EU-ARRAY 2804F
SLM 2808F
GEMEINSAM GENUTZTE FUNKTIONSLOGIK 2810
GEMEINSAM GENUTZTER SPEICHER/CACHE-SPEICHER 2812
GEOMETRIE-/FESTFUNKTIONS-PIPELINE 2814
ZUSÄTZLICHE FESTFUNKTIONSLOGIK 2816
915
2800

FIG. 28

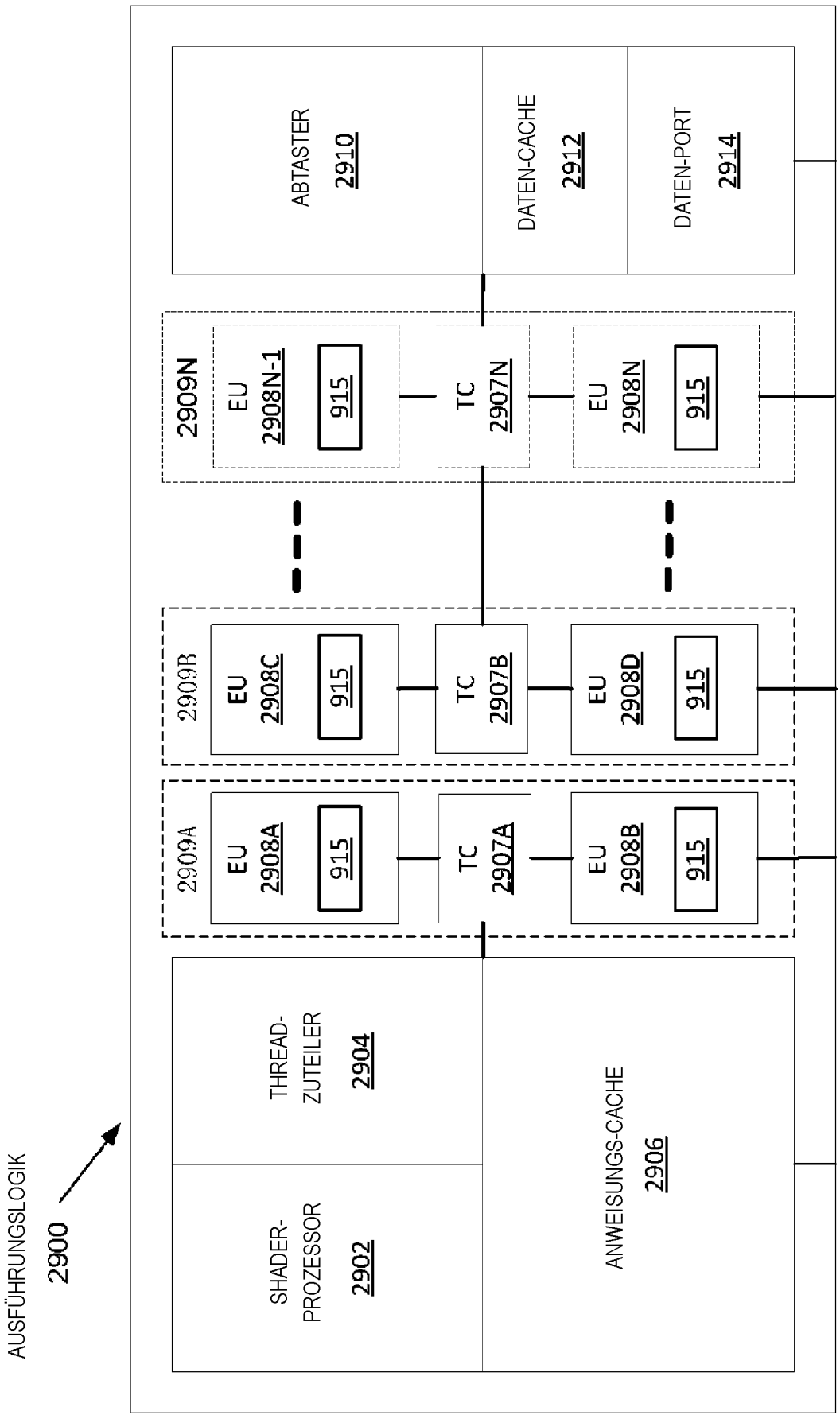

AUSFÜHRUNGSLOGIK
2900
SHADER-PROZESSOR 2902
THREAD-ZUTEILER 2904
ANWEISUNGS-CACHE 2906
2909A
EU 2908A
915
TC 2907A
EU 2908B
915
2909B
EU 2908C
915
TC 2907B
EU 2908D
915
2909N
EU 2908N-1
915
TC 2907N
EU 2908N
915
ABTASTER 2910
DATEN-CACHE 2912
DATEN-PORT 2914

FIG. 29A

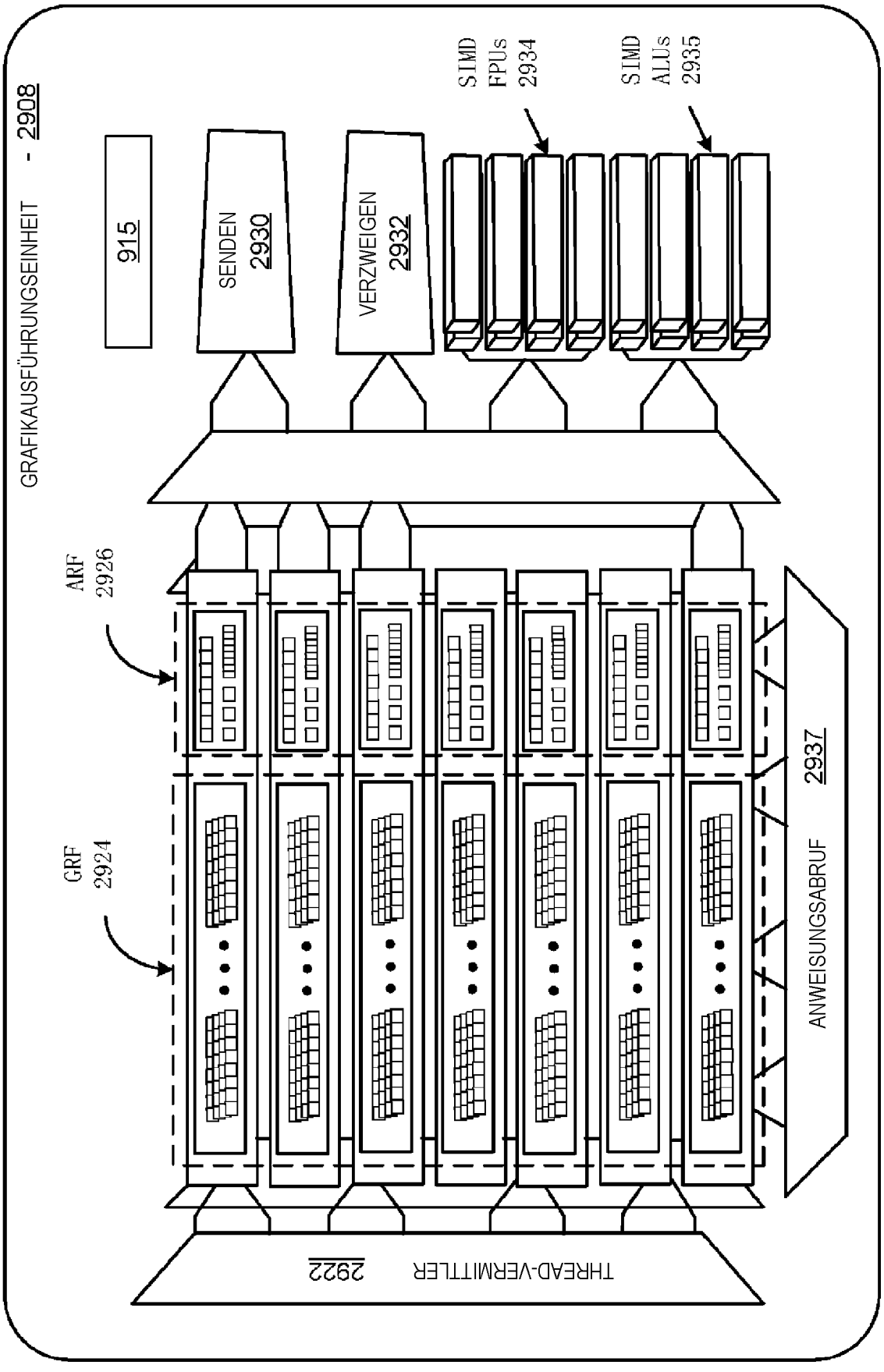

GRAFIKAUSFÜHRUNGSEINHEIT - 2908
915
SENDEN 2930
VERZWEIGEN 2932
SIMD FPUs 2934
SIMD ALUs 2935
ARF 2926
GRF 2924
ANWEISUNGSABRUF 2937
THREAD-VERMITTLER 2922

FIG. 29B

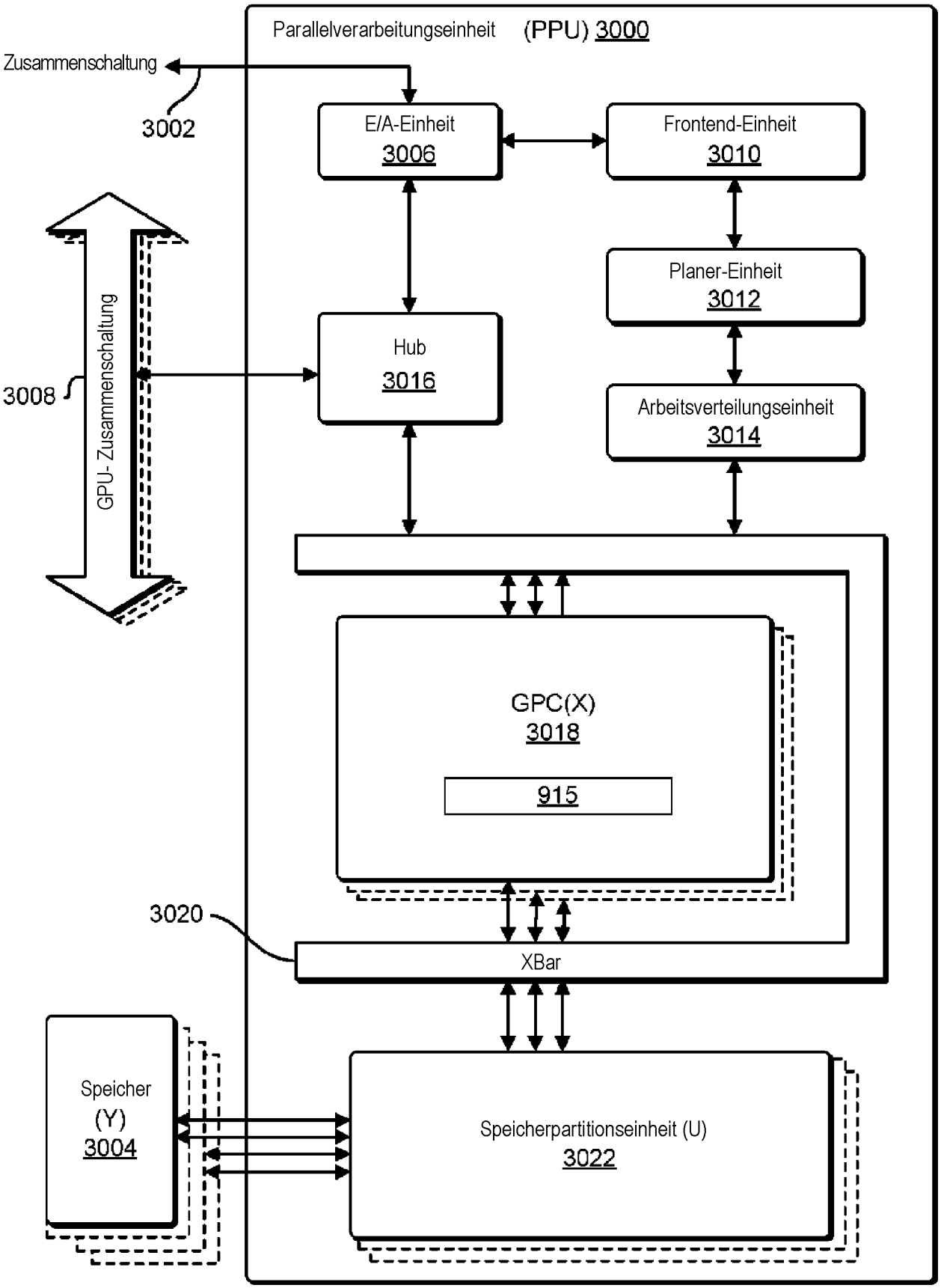

Parallelverarbeitungseinheit (PPU) 3000
Zusammenschaltung
3002
E/A-Einheit
3006
Frontend-Einheit
3010
Planer-Einheit
3012
Hub
3016
Arbeitsverteilungseinheit
3014
3008
GPU- Zusammenschaltung
GPC(X)
3018
915
3020
XBar
Speicher
(Y)
3004
Speicherpartitionseinheit (U)
3022

FIG. 30

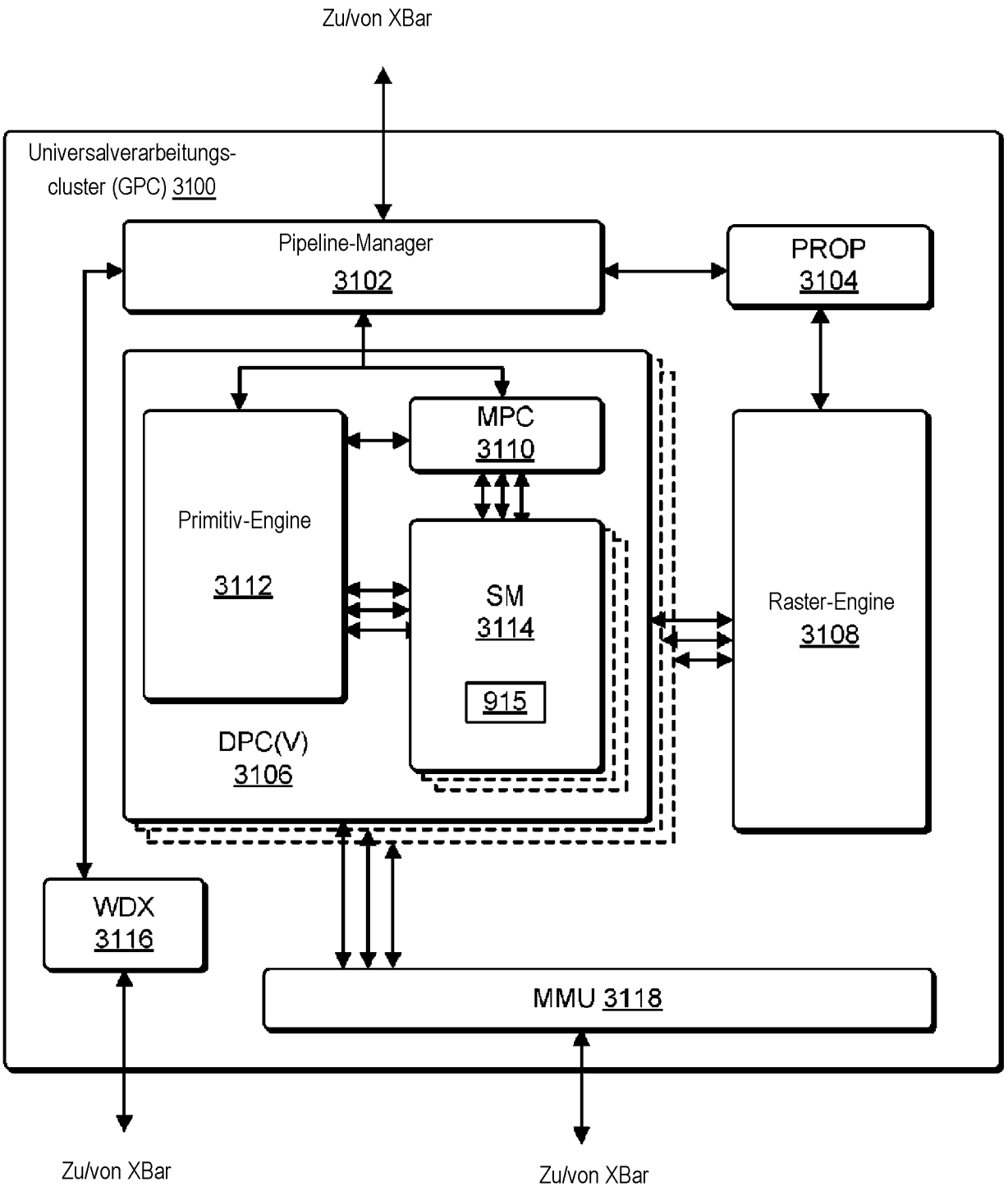

Zu/von XBar
Universalverarbeitungs-
cluster (GPC) 3100
Pipeline-Manager
3102
PROP
3104
MPC
3110
Primitiv-Engine
3112
SM
3114
915
Raster-Engine
3108
DPC(V)
3106
WDX
3116
MMU 3118
Zu/von XBar
Zu/von XBar

FIG. 31

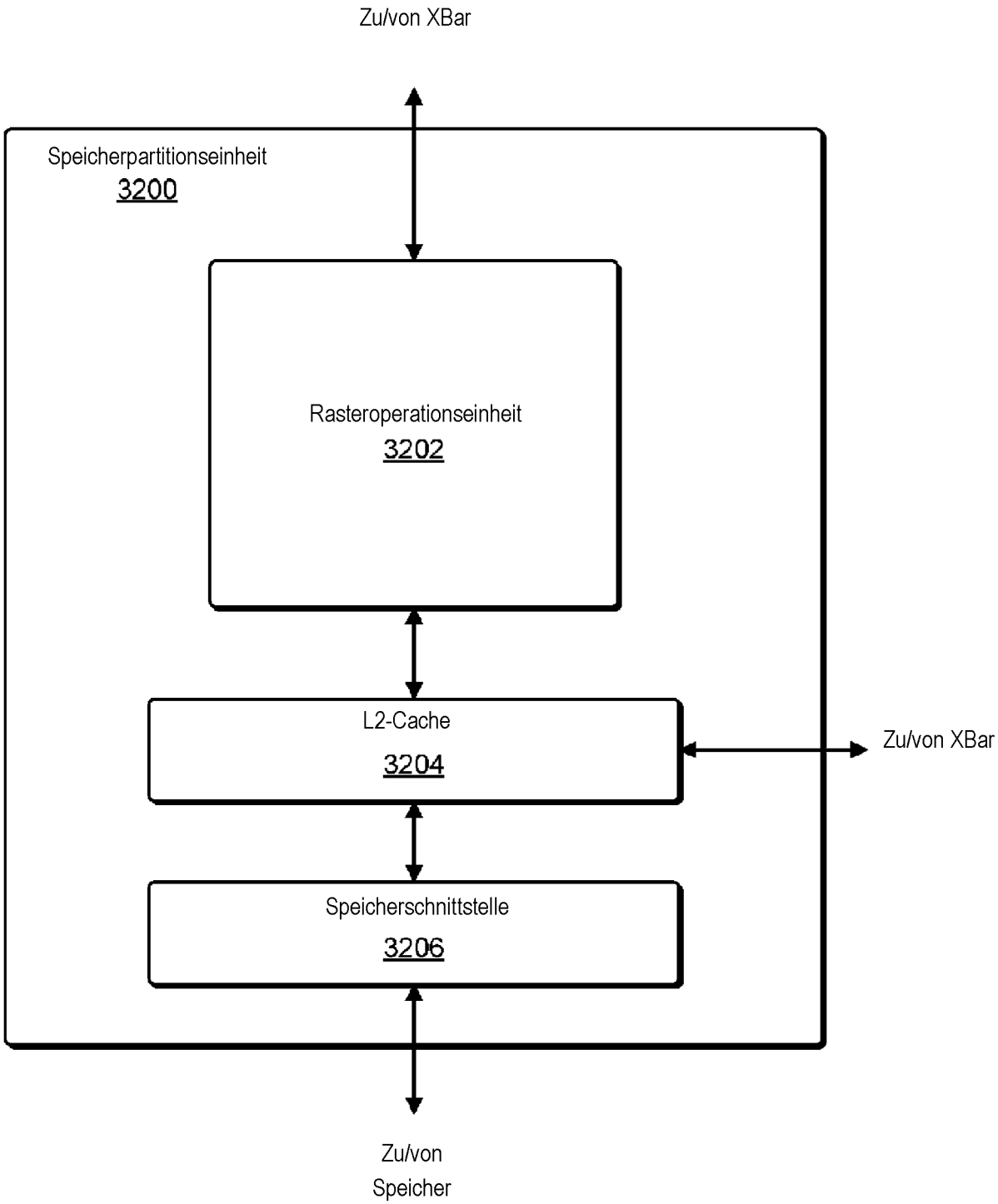

Zu/von XBar
Speicherpartitionseinheit
3200
Rasteroperationseinheit
3202
L2-Cache
3204
Zu/von XBar
Speicherschnittstelle
3206
Zu/von
Speicher

FIG. 32

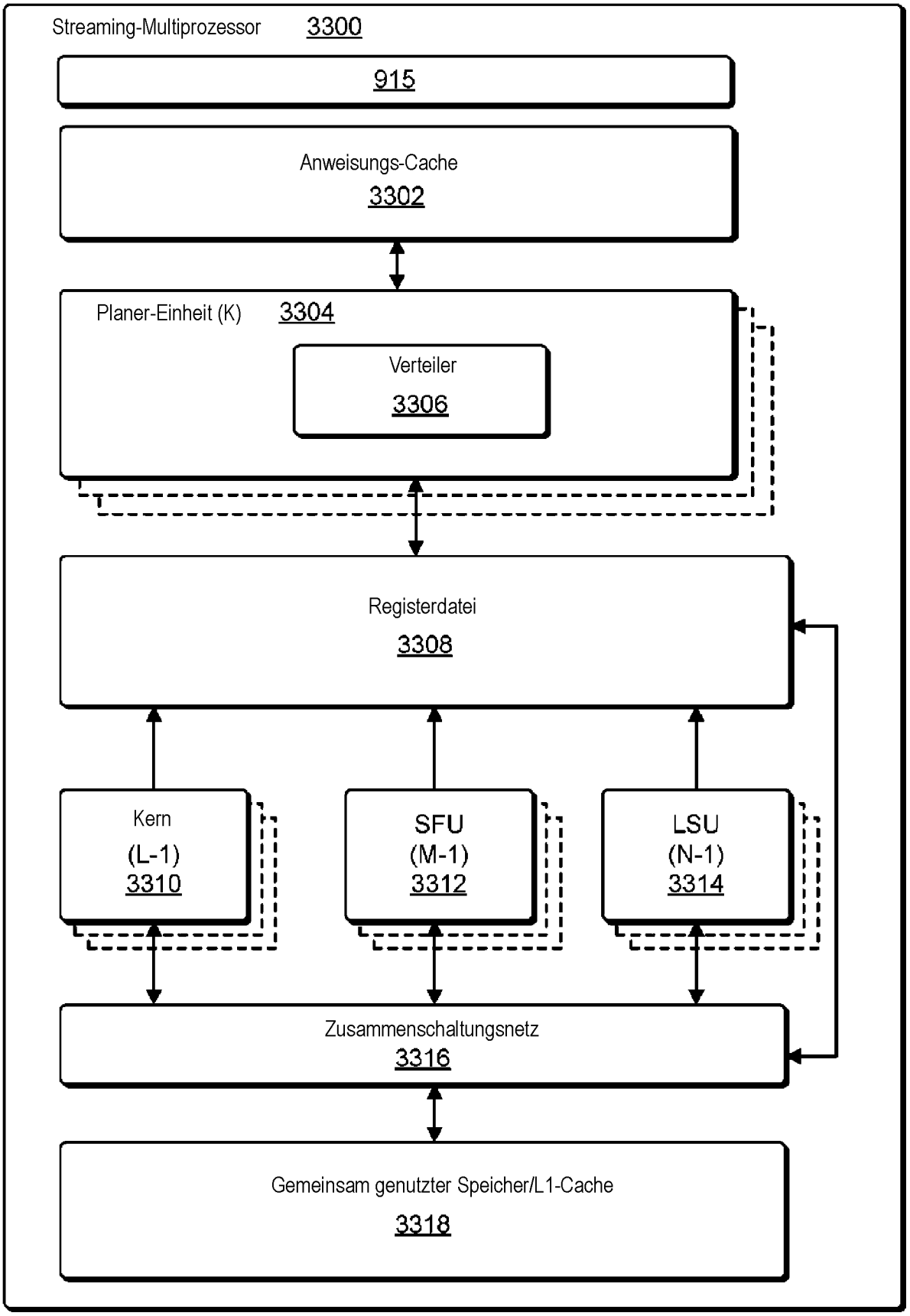


FIG. 33

 Das Dokument wurde durch die Firma Luminess hergestellt.