



(10) **DE 11 2020 003 832 T5** 2022.05.12

(12) **Veröffentlichung**

der internationalen Anmeldung mit der
(87) Veröffentlichungs-Nr.: **WO 2021/030640**
in der deutschen Übersetzung (Art. III § 8 Abs. 2
IntPatÜbkG)
(21) Deutsches Aktenzeichen: **11 2020 003 832.7**
(86) PCT-Aktenzeichen: **PCT/US2020/046252**
(86) PCT-Anmeldetag: **13.08.2020**
(87) PCT-Veröffentlichungstag: **18.02.2021**
(43) Veröffentlichungstag der PCT Anmeldung
in deutscher Übersetzung: **12.05.2022**

(51) Int Cl.: **G06T 7/30 (2017.01)**
G06T 1/40 (2006.01)

(30) Unionspriorität:
16/540,717 **14.08.2019** **US**

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

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

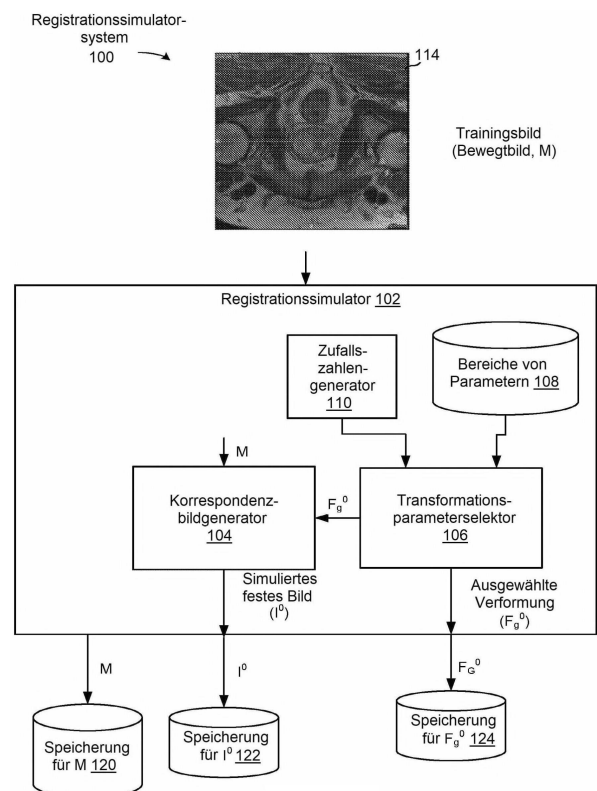
(72) Erfinder:
**Zhu, Wentao, Santa Clara, CA, US; Xu, Daguang,
Santa Clara, CA, US; Myronenko, Andriy, Santa
Clara, CA, US; Xu, Ziyue, Santa Clara, CA, US**

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **NEURONALE NETZWERKE ZUR BILDREGISTRIERUNG UND BILDSEGMENTIERUNG, DIE
UNTER VERWENDUNG EINES REGISTRIERUNGSSIMULATORS TRAINIERT WERDEN**

(57) Zusammenfassung: Einrichtungen, Systeme und Techniken, um eine Registrierung unter Bildern durchzuführen. In mindestens einer Ausführungsform werden eine oder mehrere neuronale Netzwerke trainiert, um die Registrierung von gemeinsamen Merkmalen unter mindestens zwei Bildern durch Erzeugen einer ersten Korrespondenz durch Simulieren eines Registrierungsprozesses zum Registrieren eines Bildes und Anwenden der mindestens zwei Bildern und der ersten Korrespondenz auf ein neuronales Netzwerk anzugeben, um eine zweite Korrespondenz gemeinsamer Merkmale unter den mindestens zwei Bildern herzuleiten.



Beschreibung

QUERVERWEIS AUF VERWANDTE ANMELDUNG

[0001] Diese Anmeldung beansprucht Priorität der U.S. Patentanmeldung Nr. 16/540,717, eingereicht am 14. August 2019, mit dem Titel „NEURAL NETWORK FOR IMAGE REGISTRATION AND IMAGE SEGMENTATION TRAINED USING A REGISTRATION SIMULATOR“, deren gesamter Inhalt hier durch Bezugnahme in seiner Gesamtheit und für alle Zwecke aufgenommen ist.

TECHNISCHES GEBIET

[0002] In mindestens einer Ausführungsform umfasst ein Prozessor eine oder mehrere Arithmetik-Logikeinheiten (ALUs), um Trainieren und/oder Inferenzieren unter Verwendung neuronaler Netzwerke durchzuführen. In mindestens einer Ausführungsform werden ein oder mehrere neuronale Netzwerke zur Bildregistrierung und/oder Bildsegmentierung unter Verwendung eines Registrierungssimulators trainiert, der aus einem Trainingsbild ein zweites Bild erzeugt, dass ein in dem Trainingsbild zu registrierendes Bild simulieren kann.

HINTERGRUND

[0003] Die Registrierung eines Bildes in einem anderen Bild kann ein rechenmäßig intensiver Prozess sein. In einigen Gebieten, wie beispielhafte medizinischer Bildregistrierung, kann es nötig sein, dass mehrere Bilder eines Patienten registriert werden, so dass ein medizinisches Bild mit einem anderen medizinischen Bild ausgerichtet wird, ungeachtet dessen, dass die Bilder unter unterschiedlichen Positionen, Winkeln und/oder Bedingungen aufgenommen worden sein könnten. Die Segmentierung eines Bildes, um Grenzen von Elementen in dem Bild zu identifizieren, ist ebenfalls eine Herausforderung.

Figurenliste

Fig. 1 veranschaulicht ein Beispiel eines Registrierungssimulatorsystems gemäß mindestens einer Ausführungsform;

Fig. 2 veranschaulicht ein Beispiel des Registrierungssimulators von **Fig. 1**, der zum Trainieren eines neuronalen Netzwerks verwendet wird, gemäß mindestens einer Ausführungsform;

Fig. 3 veranschaulicht ein Beispiel eines Registrierungssimulators, der zum Trainieren eines neuronalen Netzwerks verwendet wird, gemäß mindestens einer Ausführungsform;

Fig. 4 veranschaulicht ein Beispiel eines Registrierungssimulators, der zum Trainieren eines neuronalen Netzwerks unter Verwendung von Segmentierung verwendet wird, gemäß mindestens einer Ausführungsform;

Fig. 5 veranschaulicht ein Beispiel eines Registrierungssimulators, der zum Trainieren eines neuronalen Netzwerks verwendet wird, wobei Registrierungssimulation von einem Paar von Bildern und ihre Segmentierungen ist, gemäß mindestens einer Ausführungsform;

Fig. 6 veranschaulicht ein Beispiel des Verwendens eines trainierten neuronalen Netzwerks, um Inferenzierungen durchzuführen, wenn trainiert wie hier beschrieben, gemäß mindestens einer Ausführungsform;

Fig. 7 veranschaulicht ein Beispiel des Verwendens eines trainierten neuronalen Netzwerks, um Inferenzierungen durchzuführen, wenn trainiert wie hier beschrieben zur Registrierung und Segmentierung trainiert, gemäß mindestens einer Ausführungsform;

Fig. 8 veranschaulicht ein Beispiel einer Modifikation unter Verwendung einer Faltungsschicht gemäß mindestens einer Ausführungsform;

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

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

Fig. 10 veranschaulicht Training und Einsatz eines tiefen neuronalen Netzwerks gemäß mindestens einer Ausführungsform;

- Fig. 11** veranschaulicht ein beispielhaftes Rechenzentrumsystem gemäß mindestens einer Ausführungsform;
- Fig. 12A** veranschaulicht ein Beispiel eines autonomen Fahrzeugs gemäß mindestens einer Ausführungsform;
- Fig. 12B** veranschaulicht ein Beispiel von Kameraorten und Blickfeldern für ein autonomes Fahrzeug von **Fig. 12A** gemäß mindestens einer Ausführungsform;
- Fig. 12C** ist ein Blockdiagramm, das eine beispielhafte Systemarchitektur für ein autonomes Fahrzeug von **Fig. 12A** veranschaulicht, gemäß mindestens einer Ausführungsform;
- Fig. 12D** ist ein Diagramm, das ein System zur Kommunikation zwischen einem oder mehreren Cloud-basierten Servern und einem autonomen Fahrzeug von **Fig. 12A** veranschaulicht, gemäß mindestens einer Ausführungsform;
- Fig. 13** ist ein Blockdiagramm, das ein Computersystem veranschaulicht, gemäß mindestens einer Ausführungsform;
- Fig. 14** ist ein Blockdiagramm, das ein Computersystem veranschaulicht, gemäß mindestens einer Ausführungsform;
- Fig. 15** veranschaulicht ein Computersystem, gemäß mindestens einer Ausführungsform;
- Fig. 16** veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;
- Fig. 17** veranschaulicht beispielhafte integrierte Schaltungen und zugeordnete Graphikprozessoren, die unter Verwendung eines oder mehrerer IP-Kerne hergestellt werden können, gemäß mindestens einer Ausführungsform;
- Fig. 18A** veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;
- Fig. 18B** veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;
- Fig. 18C** veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;
- Fig. 18D** veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;
- Fig. 18E** veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;
- Fig. 18F** veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;
- Fig. 19A** und **Fig. 19B** veranschaulichen beispielhafte integrierte Schaltungen und zugeordnete Graphikprozessoren, die unter Verwendung einer oder mehrerer IP-Kerne hergestellt werden können, gemäß mindestens einer Ausführungsform;
- Fig. 20A** und **Fig. 20B** veranschaulichen zusätzliche beispielhafte Graphikprozessorlogik gemäß mindestens einer Ausführungsform;
- Fig. 21** veranschaulicht ein Computersystem gemäß mindestens einer Ausführungsform;
- Fig. 22A** veranschaulicht einen Parallelprozessor gemäß mindestens einer Ausführungsform;
- Fig. 22B** veranschaulicht eine Partitionseinheit gemäß mindestens einer Ausführungsform;
- Fig. 22C** veranschaulicht einer Verarbeitungs-Cluster gemäß mindestens einer Ausführungsform;
- Fig. 22D** veranschaulicht einen Graphik-Multiprozessor gemäß mindestens einer Ausführungsform;
- Fig. 23** veranschaulicht ein System mit mehreren Graphikverarbeitungseinheiten (GPUs) gemäß mindestens einer Ausführungsform;
- Fig. 24** veranschaulicht einen Graphikprozessor gemäß mindestens einer Ausführungsform;
- Fig. 25** ist ein Blockdiagramm, das eine Prozessor-Mikroarchitektur für einen Prozessor veranschaulicht, gemäß mindestens einer Ausführungsform;
- Fig. 26** veranschaulicht einen Deep-Learning-Anwendungsprozessor gemäß mindestens einer Ausführungsform;
- Fig. 27** ist ein Blockdiagramm, das ein Beispiel eines neuromorphen Prozessors veranschaulicht, gemäß mindestens einer Ausführungsform;

Fig. 28 und Fig. 29 veranschaulichen mindestens Abschnitte eines Graphikprozessors gemäß mindestens einer Ausführungsform;

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

Fig. 31A und Fig. 31B veranschaulichen eine Thread-Ausführungslogik gemäß mindestens einer Ausführungsform;

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

Fig. 33 veranschaulicht einen allgemeinen Verarbeitungscluster („GPC“) gemäß mindestens einer Ausführungsform;

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

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

AUSFÜHRLICHE BESCHREIBUNG

[0004] Zwei oder mehrere Bilder können unter Verwendung eines Prozesses registriert werden, der Korrespondenzen zwischen Patches, Objekten, Kanten, Oberflächen oder Orientierungspunkte festlegt, die in den beiden oder mehreren Bildern erscheinen, in Anbetracht dessen, dass die Bilder unterschiedliche Bilder einer Szene in einer Umgebung sein könnten. Eine derartige Umgebung ist eine medizinische Bildgebung, wobei zwei Bilder, die von einer Szene genommen werden, die Teil eines Patientenkörpers ist, vielleicht unter Verwendung unterschiedlicher Erfassungstechnologien aus Informationen Nutzen ziehen könnten, die ein Bild in dem anderen registriert, wie beispielhafte Bildfusion, Organatlaserzeugung, Tumorwachstumüberwachung usw. Manuelle Bildregistrierung ist mühsam und mangelt an Reproduzierbarkeit, was potenziell klinische Nachteile verursacht. Eine Registrierung könnte als Teil einer Analyse eines Paares von Bildern von unterschiedlichen Modalitäten, die zu unterschiedlichen Zeiten erfasst wurden, von unterschiedlichen Blickpunkten oder von unterschiedlichen Patienten ausgeführt werden. Allgemeiner gesagt könnten die Bilder einer Szene in einer biologischen Umgebung sein, die keine Menschen beinhaltet.

[0005] Bilder könnten zweidimensionale Bilder oder dreidimensionale Bilder sein. Für ein zweidimensionales Bild könnte die Registrierung ein Prozess des Findens einer zweidimensionalen Vektoranordnung oder eines Verschiebungsfeldes sein, das angibt, wie sich Pixeln eines Bildes bewegen würden, damit sich Elemente dieses Bildes mit entsprechenden Elementen eines anderen Bildes ausrichten. In einigen Fällen werden diese als ein Bewegtbild bzw. ein festes Bild, bezeichnet. Mit dreidimensionalen Bildern könnte die Registrierung ein dreidimensionales Verschiebungsfeld finden, das angibt, wie sich Voxel von einem Bild in Ausrichtung bewegen würden.

[0006] Segmentierung ist ein Prozess des Bestimmens von einem Bild, wo Grenzen von Elementen sind, die in dem Bild anschaulich dargestellt werden, und möglicherweise ebenfalls der Beschriftung dieser Objekte. Menschen können häufig ohne Weiteres identifizieren, wo ein Objekt in einem Bild beginnt und andere Objekte enden, wobei dies durchzuführen jedoch ein schwerer Prozess für ein Computersystem sein kann. Eine Segmentierung eines Bildes könnte im Speicher oder beim Datentransfer als eine Segmentierungsmaske dargestellt werden, die Grenzen von einigen in diesem Bild sichtbaren Objekten angeben, wobei Beschriftungen in dieser Segmentierungsmaske ebenfalls möglicherweise eine textliche oder Referenzbeschriftung angeben, die identifizierbaren Objekten in dem Bild zugeordnet ist. Segmentierung ist in medizinischen Gebieten sowie auch anderen Gebieten der Bildverarbeitung nützlich.

[0007] Neuronale Netzwerke können nützlich sein, um eine Bildregistrierung autonom oder semiautonom durchzuführen. In mindestens einer Ausführungsform wird ein neuronales Netzwerk unter Verwendung von Paaren von Bildern und einem Verschiebungsfeld trainiert, wobei dieses Verschiebungsfeld durch einen Registrierungssimulator erzeugt wird und eines von diesem Paar von Bildern von dem anderen Bild und dem Verschiebungsfeld erzeugt wird. In mindestens einer Ausführungsform ist das Training des neuronalen Netzwerks bestrebt, eine Verlustfunktion zu verringern oder anderweitig zu optimieren, die Differenzen zwischen dem durch den Registrierungssimulator erzeugten Verschiebungsfeld und einem durch dieses neuronale Netzwerk erzeugten Verschiebungsfeld darstellt, wobei jenes Paar von Bildern und dieses Verschiebungsfeld bekannt sind. In mindestens einer Ausführungsform ist eine Verlustfunktion ein Hybridverlust, der einen Registrierungs-feld-Überwachungsverlust und einen Datenähnlichkeitsverlust umfasst.

[0008] In mindestens einer Ausführungsform kann ein neuronales Netzwerk mit Segmentierungsmasken trainiert werden, in dem dieses neuronale Netzwerk mit Segmentierungsmasken von bereitgestellten Bilder versehen wird.

[0009] Fig. 1 veranschaulicht ein Beispiel eines Registrierungssimulatorsystems 100 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform umfasst ein Registrierungssimulator 102 einen Korrespondenzbildgenerator 104, eine Transformationsparameter-Auswahleinrichtung 106, Speicherung für Parameterbereiche 108 und einen Zufallszahlengenerator 110. In mindestens einer Ausführungsform empfängt der Registrierungssimulator 102 als Eingabe ein Trainingsbild 114 M, das als ein Bewegtbild in einem Bildregistrierungsprozess dienen kann. In mindestens einer Ausführungsform gibt der Registrierungssimulator 102 das Trainingsbild 114 M an die Speicherung 120 aus, falls nicht bereits gespeichert, oder gibt, falls benötigt, ein simuliertes festes Bild, I^0 , an die Speicherung 122 aus, und gibt eine ausgewählte Verformung, F_g^0 , an die Speicherung 124 aus.

[0010] In mindestens einer Ausführungsform wählt die Transformationsparameter-Auswahleinrichtung 106 Transformationsparameter aus. In mindestens einer Ausführungsform könnten die Transformationsparameter einer Translation, einer Drehung, einer Skalierung und/oder einer elastischen Verformung entsprechen. In mindestens einer Ausführungsform erhält die Transformationsparameter-Auswahleinrichtung 106 eine oder mehrerer Zufallszahlen von einem Zufallszahlengenerator 110 und verwendet diese, um aus einem Bereich von Werten unter Verwendung einer gleichmäßigen Wahrscheinlichkeitsverteilung oder einer anderen Wahrscheinlichkeitsverteilung auszuwählen. In mindestens einer Ausführungsform könnte die Transformationsparameter-Auswahleinrichtung 106 eine Translation in jeder Dimension unter einem Bereich von möglichen Translationen auswählen und einen Drehwinkel, α , unter einem Bereich von Drehwinkeln von 0 auf einen Winkel, A , für jede Dimension auswählen. In mindestens einer Ausführungsform könnte die Transformationsparameter-Auswahleinrichtung 106 einen Skalierungsfaktor, c_m , aus einem Bereich eines minimalen Skalierungsfaktors, C_{min} , und eines maximalen Skalierungsfaktors, C_{max} , auswählen. In mindestens einer Ausführungsform entspricht ein Wert von c , der kleiner als 1 ist, einem Schrumpfen des Bewegtbildes, und c , der größer als 1 ist, entspricht einem Vergrößern des Bewegtbildes. In mindestens einer Ausführungsform könnten elastische Verzerrungen durch zufälliges Auswählen eines Koordinaten-Offsets einer Gaußschen Verteilung mit einer Standardabweichung γ von einem Bereich von Werten von 0 bis Γ und Anwenden eines multidimensionalen Gaußschen Filters mit einer Standardabweichung von σ von ausgewählt aus einem Bereich von Σ_{min} bis Σ_{max} modelliert werden, um erzeugte Koordinaten-Offsets glatt und realistisch zu machen.

[0011] In mindestens einer Ausführungsform wird, sobald Transformationsparameter ausgewählt sind, ein Verschiebungsfeld, das einer Transformation F_g^0 entspricht, berechnet und kann ausgegeben werden. In mindestens einer Ausführungsform erzeugt der Korrespondenzbildgenerator 104 ein simuliertes festes Bild I^0 unter Verwendung dieses Verschiebungsfelds. In mindestens einer Ausführungsform könnte, da das simulierte feste Bild von dem Bewegtbild erzeugt wird, ein hoch trainiertes neuronales Netzwerk versuchen, das Bewegtbild und das simulierte feste Bild zu registrieren, erwartet werden, ein Verschiebungsfeld nahe einer ausgewählte Verformung, F_g^0 , auszugeben. In mindestens einer Ausführungsform kann die ausgewählte Verformung F_g^0 als eine Bodenwahrheit zum Trainieren eines neuronalen Netzwerks mit überwachten Trainieren mit dem Bewegtbild und dem simulierten festen Bild dienen, das Verschiebungsfeld zu schätzen.

[0012] Fig. 2 veranschaulicht ein Beispiel des Registrierungssimulators 102, der zum Trainieren eines neuronalen Netzwerks 202 verwendet wird, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform gibt der Registrierungssimulator 102 ein Bewegtbild, M, ein simuliertes festes Bild I^0 und eine ausgewählte Verformung, F_g^0 , aus. In mindestens einer Ausführungsform trainiert das neuronale Netzwerk 202 auf dem Bewegtbild und dem simulierten festen Bild unter Annahme einer ausgewählten Verformung F_g^0 als Bodenwahrheit für das Verschiebungsfeld und gibt ein geschätztes Verschiebungsfeld, F^0 , aus. In mindestens einer Ausführungsform empfängt ein Verlustbewerter 204 als Eingaben eine ausgewählte Verformung F_g^0 und ein geschätztes Verschiebungsfeld, F^0 , und berechnet eine Verlustfunktion, $L_F(F^0, F_g^0; \theta)$, wobei θ Parameter des neuronalen Netzwerks 202 darstellen, und stellt die Verlustfunktion als Rückkopplung zu dem neuronalen Netzwerk 202 bereit. In mindestens einer Ausführungsform wird ein Ausgang des neuronalen Netzwerks 202 in Gleichung 1 gezeigt und eine beispielhafte Verlustfunktion wird als Gleichung 2 gezeigt, wobei p eine Pixel/Voxel-Position in einem Bildkoordinatenraum mit Q Pixeln/Voxeln in diesem Bildkoordinatenraum ist.

$$F^0 = N(M, I^0, \theta) \quad (\text{Gl. 1})$$

$$\mathcal{L}_F(F^0, F_g^0; \theta) = \frac{1}{\|\Omega\|} \sum_{p \in \Omega} \|F^0(p) - F_g^0(p)\|_{L_2} \quad (\text{Gl. 2})$$

[0013] In mindestens einer Ausführungsform ist der Verlust in Gleichung 2 eine feldüberwachte Verlustfunktion für Bildabgleich der Messausrichtung von zwei Verschiebungsfeldern. In mindestens einer Ausführungsform kann, wo eine Verschiebungsfeld-Bodenwahrheit glatt ist, ein neuronales Netzwerk eine Glätte eines geschätzten Verschiebungsfelds von einer Registrierungssimulation lernen. In mindestens einer Ausführungsform kann der Verlust mit der Abnahme von Datenähnlichkeitsverlust beständig abnehmen. In mindestens einer Ausführungsform kann ein neuronales Netzwerk auf mehreren Bewegungsbildern trainiert werden und ein gegebenes Bewegungsbild kann erneut zum Trainieren unter Verwendung unterschiedlicher zufällig ausgewählter Verschiebungsfelder verwendet werden. In mindestens einer Ausführungsform kann die Auswahl von Transformationsparameters zufällig oder unter Verwendung eines anderen statistischen Verfahrens gemacht werden.

[0014] In mindestens einer Ausführungsform kann ein Modell eines neuronalen Netzwerks von der Qualität und Diversität eines simulierten Registrierungsfeldes abhängen. In mindestens einer Ausführungsform kann ein Modell verallgemeinert werden, um Datenähnlichkeitsverlust als einen Hilfsabschnitt einer Verlustfunktion zu umfassen. In mindestens einer Ausführungsform wird ein neuronales Netzwerk unter Verwendung rekonstruierter Bilder, I_R^0 , trainiert, die durch Verzerren eines Bewegungsbildes M mit geschätztem Registrierungsfeld F^0 rekonstruiert werden. In mindestens einer Ausführungsform wird ein Modell eines neuronalen Netzwerks auf mehreren Trainingsbildern trainiert, jede Eingabe als Bewegungsbild M der Reihe nach.

[0015] Fig. 3 veranschaulicht ein Beispiel davon gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform wird, wie dort gezeigt, ein Registrierungssimulator 302 zum Trainieren eines neuronalen Netzwerks 304 verwendet. In mindestens einer Ausführungsform gibt der Registrierungssimulator 302 ein Bewegungsbild, M, ein simuliertes festes Bild, I^0 , und eine ausgewählte Verformung, F_g^0 , aus. In mindestens einer Ausführungsform trainiert das neuronale Netzwerk 304 auf dem Bewegungsbild M und simulierten festen Bild I^0 unter Annahme der ausgewählten Verformung F_g^0 als Bodenwahrheit für das Verschiebungsfeld, und gibt ein geschätztes Verschiebungsfeld, F^0 , aus. In mindestens einer Ausführungsform empfängt ein Verlustbewerter 306 als Eingaben eine ausgewählte Verformung F_g^0 , ein geschätztes Verschiebungsfeld, F^0 , ein simuliertes festes Bild I^0 und ein rekonstruiertes Bild, I_R^0 , und berechnet eine Registrierungsfeldfunktion, $\mathcal{L}_F(F^0, F_g^0; \theta)$, jedoch ebenfalls eine Datenähnlichkeitsfunktion, wie durch Gleichung 3 gezeigt, die eine Form einer negativen normierten lokalen Kreuzkorrelation ist, wobei p_i eine Pixelposition innerhalb einer Fensters um p und $I^0(p)$ und $I_R^0(p)$ jeweils lokale Mittel innerhalb einer Fensters um die Pixelposition p_i in I^0 und I_R^0 sind.

$$\mathcal{L}_{sim}(I^0, I_R^0; \theta) = -\frac{1}{\|\Omega\|} \sum_{p \in \Omega} \frac{\left(\sum_{p_i} \left(I^0(p_i) - \overline{I^0(p)} \right) \left(I_R^0(p_i) - \overline{I_R^0(p)} \right) \right)^2}{\sum_{p_i} \left(I^0(p_i) - \overline{I^0(p)} \right)^2 \sum_{p_i} \left(I_R^0(p_i) - \overline{I_R^0(p)} \right)^2} \quad (\text{Gl. 3})$$

[0016] In mindestens einer Ausführungsform stellt der Verlustbewerter 306 einer Hybridverlustfunktion als Rückkopplung dem neuronalen Netzwerk 304 bereit, wobei diese Hybridverlustfunktion eine gewichtete Summe einer Registrierungsfeldfunktion und einer Datenähnlichkeitsfunktion sein könnte, wie in Gleichung 4, wobei X ein Hyperparameter ist, der zum Abstimmen eines Ausgleichs zwischen zwei Verlusten nützlich ist.

$$\mathcal{L}_{reg}(F^0, F_g^0, I^0, I_R^0; \theta) = \mathcal{L}_F(F^0, F_g^0; \theta) + \lambda \mathcal{L}_{sim}(I^0, I_R^0; \theta) \quad (\text{Gl. 4})$$

[0017] In mindestens einer Ausführungsform kann, wenn das neuronale Netzwerk 304 zum Inferenzieren bei einem gegebenen Bewegt-/Atlasbild M und festen Bild I verwendet wird, ein Verschiebungsfeld von Gleichung 2 geschätzt werden und ein rekonstruiertes Bild, I_R , als eine Transformation von M und ein geschätztes Verschiebungsfeld berechnet werden. In mindestens einer Ausführungsform könnte ein Verformer 308 verwendet werden, um ein rekonstruiertes Bild und eine rekonstruierte Segmentierung zu rekonstruieren.

[0018] Fig. 4 veranschaulicht ein Beispiel eines Registrierungssimulators, der zum Trainieren eines neuronalen Netzwerks verwendet wird, wobei Registrierungssimulation, S^M , eine Segmentierungs-Bodenwahrheit eines Bewegt-/Atlasbildes M sowie auch Bild M ist, gemäß mindestens einer Ausführungsform. In mindestens

einer Ausführungsform wird, wie dort gezeigt, ein Registrierungssimulator 402 zum Trainieren eines neuronalen Netzwerks 404 verwendet. In mindestens einer Ausführungsform empfängt der Registrierungssimulator 402 ein Trainingsbild sowie auch eine Segmentierungsmaske, die eine Segmentierung von Bild M darstellt. In mindestens einer Ausführungsform gibt der Registrierungssimulator 402 ein Bewegtbild, M , ein simuliertes festes Bild I^0 , eine ausgewählte Verformung, F_g^0 als einer-Bodenwahrheit einer Verschiebung zwischen M und I^0 und eine Segmentierung-Bodenwahrheit, S_g^0 , aus, die einer Transformation von S^M durch F_g^0 entspricht.

[0019] In mindestens einer Ausführungsform trainiert das neuronale Netzwerk 404 auf einem Bewegtbild und einem simulierten festen Bild unter der Annahme einer ausgewählten Verformung F_g^0 als Bodenwahrheit für das Verschiebungsfeld und trainiert ebenfalls auf der Segmentierung-Bodenwahrheit. In mindestens einer Ausführungsform gibt das neuronale Netzwerk 404 ein geschätztes Verschiebungsfeld, F^0 , sowie auch eine geschätzte Segmentierung, S^0 , aus. In mindestens einer Ausführungsform empfängt ein Verlustbewerter 406 als Eingaben eine ausgewählte Verformung F_g^0 , ein geschätztes Verschiebungsfeld, F^0 , ein simuliertes festes Bild I^0 , ein rekonstruiertes Bild, I_R^0 , das durch Transformieren M um F^0 rekonstruiert wurde, sowie auch eine rekonstruierte Segmentierung, S^0 , die durch Transformieren S^M um F^0 rekonstruiert wurde. In mindestens einer Ausführungsform könnte ein Verformer 408 verwendet werden, um ein rekonstruiertes Bild und eine rekonstruierte Segmentierung zu rekonstruieren.

[0020] In mindestens einer Ausführungsform kann ein geschätztes Verschiebungsfeld auf das Transformieren von Segmentierungsmasken für Bildsegmentierungen angewandt werden, wenn ein neuronales Netzwerk trainiert wird und Inferenzen durchführt. In mindestens einer Ausführungsform können Bodenwahrheiten der Segmentierung zum Trainieren eines neuronalen Netzwerks verwendet werden, um einer Registrierungsnetzwerk fein abzustimmen.

[0021] In mindestens einer Ausführungsform verwendet die Registrierung-basierte Segmentierung durch Multi-Task-Learning (MTL) ein Dual-Registrierungsschema unter Verwendung eines Registrierungssimulators. In mindestens einer Ausführungsform kann ein trainiertes neuronales Netzwerk ein Beweg-/Atlasbild M mit einem beliebigen Bild I^0 von einem Registrierungssimulator sowie auch einem beliebigen Bild, das von einem Datensatz von Bildern ausgewählt wird, registrieren. In mindestens einer Ausführungsform kann in einem Segmentierungsszenario ein Registrierungssimulator als ein Datenaugmentierungswerkzeug wirken, das mit medizinischen Bildsegmentierung nützlich sein kann, wobei medizinische Bilddatensätze klein sind und eine dichte Annotation von Segmentierung aufwändig ist.

[0022] Fig. 5 veranschaulicht ein Beispiel eines Registrierungssimulators, der zum Trainieren eines neuronalen Netzwerks verwendet wird, wobei Registrierungssimulation ein Paar von Bildern und ihre Segmentierungen ist, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform wird, wie dort gezeigt, ein Registrierungssimulator 502 zum Trainieren eines neuronalen Netzwerks 504 verwendet. In mindestens einer Ausführungsform empfängt der Registrierungssimulator 502 zwei Trainingsbilder, die als M^0 und M^1 bezeichnet werden, sowie auch zwei Segmentierungsmasken, die jeweils Segmentierungen S^{M^0} und S^{M^1} von Bildern M^0 und M^1 darstellen. In mindestens einer Ausführungsform gibt der Registrierungssimulator 502 Bewegtbilder, M^0 und M^1 , die simulierten festen Bilder I^0 und I^1 entsprechen, eine ausgewählte Verformung, F_g^0 als einer-Bodenwahrheit einer Verschiebung zwischen Bewegtbildern und ihren jeweiligen simulierten festen Bildern und Segmentierung-Bodenwahrheiten, S_g^0 und S_g^1 aus, die Transformationen von Segmentierungen um F_g^0 entsprechen.

[0023] In mindestens einer Ausführungsform trainiert das neuronale Netzwerk 504 auf diesen Eingaben und gibt zwei geschätzte Verschiebungsfelder, F^0 und F^1 , sowie auch zwei geschätzte Segmentierungen, S^0 , S^1 aus. In mindestens einer Ausführungsform empfängt ein Verlustbewerter 506 als Eingaben eine ausgewählte Verformung F_g^0 , ein geschätztes Verschiebungsfeld, F^0 , simulierte feste Bilder I^0 und I^1 , rekonstruierte Bilder I_R^0 und I_R^1 , die durch Transformieren M^0 durch F^0 und M^1 durch F^1 rekonstruiert wurden, sowie auch rekonstruierte Segmentierungen, S^0 , aus, die durch Transformieren S^{M^0} durch F^0 und S^1 durch Transformieren S^{M^1} durch F^1 rekonstruiert wurden.

[0024] In mindestens einer Ausführungsform könnte ein Verformer 508 verwendet werden, um jene rekonstruierten Bilder und jene rekonstruierten Segmentierungen zu rekonstruieren. In mindestens einer Ausführungsform gibt bei Anwenden des Eingangspaares M^0 und I^1 auf das neuronale Netzwerk 504 das neuronale Netzwerk 504 ein Verschiebungsfeld, F^1 , basierend auf seinem Training zur Registrierung von Bildern aus. In mindestens einer Ausführungsform sollte abhängig vom Training eines neuronalen Netzwerks das Verschiebungsfeld F^1 nahe dem Verschiebungsfeld F_g^0 sein. In mindestens einer Ausführungsform ergibt das Anwen-

den des Verschiebungsfelds F^1 , um $M1$ zu verzerren, $I_R^1 = T(M1, F^1)$. In mindestens einer Ausführungsform sollte abhängig vom Trainieren eines neuronalen Netzwerks I_R^1 nahe I^1 sein. In mindestens einer Ausführungsform gibt ein Anwenden des Eingangspaares $M0$ und $M1$ auf ein neuronales Netzwerk ein neuronales Netzwerk ein Verschiebungsfeld F^0 basierend auf seinem Training zur Registrierung von Bildern aus. In mindestens einer Ausführungsform ergibt das Anwenden des Verschiebungsfeldes F^0 , um $M0$ zu verzerren, $I_R^0 = T(M0, F^0)$. In mindestens einer Ausführungsform sollte, abhängig von einem Training eines neuronalen Netzwerks, I_R^0 nahe I^0 sein.

[0025] In mindestens einer Ausführungsform berechnet ein Verlustbewerter 506 eine mehrteilige Verlustfunktion, die vielleicht eine Registrierungs-Verlustfunktion, eine Simulations-Verlustfunktion und eine Segmentierungs-Verlustfunktion umfasst. In mindestens einer Ausführungsform könnte eine Segmentierungs-Verlustfunktion vielleicht Funktionen von zwei Bildern umfassen und eine Simulations-Verlustfunktion könnte Funktionen von zwei Bildern umfassen. In mindestens einer Ausführungsform könnte eine Segmentierungs-Verlustfunktion eine erste Funktion einer ersten Differenzfunktion von Differenzen zwischen einer ersten geschätzten Segmentierungsmaske und einer ersten rekonstruierten Segmentierungsmaske und/oder eine zweite Differenzfunktion von Differenzen zwischen einer zweiten geschätzten Segmentierungsmaske und einer zweiten rekonstruierten Segmentierungsmaske sein.

[0026] In mindestens einer Ausführungsform kann die Registrierung einer anderen Registrierung in einer dualen Registrierung für Bewegt-/Atlasbild $M0$ und ein beliebiges Bild $M1$ von einem Datensatz unter Verwendung $F^1 = N(M, I^1; \theta)$ erhalten werden und das rekonstruierte Bild I_R^1 kann unter Verwendung von $I_R^1 = T(M1, F^1)$ erhalten werden. In mindestens einer Ausführungsform kann die Segmentierung-Bodenwahrheit S_g^0 unter Verwendung $S_g^0 = T(S^{M0}, F^0)$ erhalten werden und die Segmentierung-Bodenwahrheit S_g^1 kann unter Verwendung $S_g^1 = T(S^{M1}, F^1)$ erhalten werden. In mindestens einer Ausführungsform könnte eine Gesamtregistrierungs-Verlustfunktion gemäß Gleichungen 5-8 bestimmt werden.

$$\mathcal{L}_{reg}(F^0, F_g^0, I^0, I_R^0, I^1, I_R^1; \theta) = \mathcal{L}_F(\quad) + \lambda(\mathcal{L}_{sim}(0) + \mathcal{L}_{sim}(1)) + \beta \mathcal{L}_{seg}(\quad) \quad (\text{Gl. 5})$$

$$\mathcal{L}_F(\quad) = \mathcal{L}_F(F^0, F_g^0; \theta) \quad (\text{Gl. 6})$$

$$\mathcal{L}_{sim}(i) = \mathcal{L}_{sim}(I^i, I_R^i; \theta) \quad (\text{Gl. 7})$$

$$\mathcal{L}_{seg}(\quad) = \mathcal{L}_{seg}(S^0, S_g^0, S^1, S_g^1) \quad (\text{Gl. 8})$$

[0027] In mindestens einer Ausführungsform könnte eine Segmentierungs-Verlustfunktion von Gleichung 8 berechnet werden, wie in Gleichungen 9 und 10 gezeigt, wobei S^0 , und S^1_g eine Form von Segmentierung-Bodenwahrheit und S^0 und S^1 kontinuierliche Werte mit Interpolationsreihenfolge eins in einer räumlichen Transformer T und C Klassen sind.

$$\mathcal{L}_{seg}(S^0, S_g^0, S^1, S_g^1) = D(S^0, S_g^0) + D(S^1, S_g^1) \quad (\text{Gl. 9})$$

$$D(S^0, S_g^0) = -\frac{1}{C} \sum_{C=0}^{C-1} \frac{\sum_p (S^0(p) S_g^0(p))}{\sum_p (S^0(p) + S_g^0(p) - S^0(p) S_g^0(p))} \quad (\text{Gl. 10})$$

[0028] In mindestens einer Ausführungsform können, wie in Gleichung 5 veranschaulicht, Parameter γ und β eingestellt werden, um Gewichte von verschiedenen Komponenten einer Verlustfunktion zu steuern. In mindestens einer Ausführungsform könnte bei Inferenzzeit eine Ausgabe eines neuronalen Netzwerks eine endgültige Segmentierungsvorhersage durch Nehmen von $\arg\max$ entlang einer Klassendimension bereitstellen.

[0029] Fig. 6 veranschaulicht ein Beispiel einer Verwendung eines trainierten neuronalen Netzwerks, um Inferenzen durchzuführen, wenn trainiert, wie hier beschrieben, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform empfängt ein trainiertes neuronales Netzwerk 602 zwei Bilder, $M1$ und $M2$, ohne notwendigerweise irgendeine Bodenwahrheit bezüglich ihrer Beziehung aufzuweisen. In mindestens einer Ausführungsform gibt das trainierte neuronale Netzwerk 602 ein geschätztes Verschiebungsfeld aus, das eine Bewertung des trainierten neuronalen Netzwerks 602 hinsichtlich dessen angibt, wie eines der beiden Bilder transformiert werden würden, um sich mit dem anderen Bild auszurichten.

[0030] Fig. 7 veranschaulicht ein Beispiel einer Verwendung eines trainierten neuronalen Netzwerks, um Inferenzen durchzuführen, wenn wie hier beschrieben zur Registrierung und Segmentierung trainiert, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform empfängt ein trainiertes neuronales Netzwerk 702 zwei Bilder, M1 und M2, ohne notwendigerweise irgendeine Bodenwahrheit bezüglich ihrer Beziehung aufzuweisen, und eine Segmentierungsmaske, SM^1 , für Bild M1. In mindestens einer Ausführungsform gibt ein trainiertes neuronales Netzwerk 702 ein geschätztes Verschiebungsfeld aus, das eine Bewertung des trainierten neuronalen Netzwerks 702 bezüglich dessen darstellt, wie eines der beiden Bilder transformiert werden würde, um sich mit dem anderen Bild auszurichten, sowie auch einer Segmentierung von Bild M2 bei gegebener Segmentierung von Bild M1. In mindestens einer Ausführungsform verwendet das trainierte neuronale Netzwerk 702 Inferenzen der Verschiebung von Bildern, um bei der Segmentierung von Bild M2 anzukommen.

[0031] Fig. 8 veranschaulicht ein Beispiel einer Modifikation unter Verwendung einer Faltungsschicht gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform wird ein neuronales Netzwerk modifiziert, um eine Faltungsschicht zum Modifizieren einer letzten neuronalen Netzwerkschicht bereitzustellen, die eine Faltung einer letzten Schicht eines neuronalen Netzwerks und einer vorhergesagten Segmentierung bereitstellt. In mindestens einer Ausführungsform kann eine Softmax-Aktivierungsschicht zum Modifizieren einer vorhergesagten Segmentierung gemäß einer Softmax-Operation auf eine Faltung verwendet werden. In mindestens einer Ausführungsform verwendet eine zusätzliche Faltungsschicht mit einer vorhergesagten Segmentierung und ein Merkmal einer letzten Schicht vom neuronalen Netzwerk als Eingabe eine Softmax-Aktivierungsfunktion, um eine Segmentierung weiter zu verbessern.

[0032] Fig. 9A veranschaulicht die Inferenzierungs- und/oder Trainingslogik 915, die verwendet wird, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehrerer Ausführungsformen zugeordnet sind. Einzelheiten der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit Fig. 9A und/oder Fig. 9B bereitgestellt.

[0033] In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915, ohne Beschränkung, Code und/oder Datenspeicherung 901 umfassen, um Vorwärts- und/oder Ausgangsgewichtung und/oder Eingangs-/Ausgangsdaten und/oder andere Parameter zu speichern, um Neuronen oder Schichten eines neuronalen Netzwerks zu konfigurieren, trainiert und/oder verwendet zum Inferenzieren in Aspekten einer oder mehrerer Ausführungsformen. In mindestens einer Ausführungsform kann Trainingslogik 915 Code und/oder Datenspeicherung 901 umfassen oder damit gekoppelt sein, um Graphencode oder andere Software zu speichern, um das Timing und/oder die Reihenfolge zu steuern, bei der Gewichtungen und/oder andere Parameterinformationen zu laden sind, um Logik, einschließlich Ganzzahl- und/oder Gleitkomma-Einheiten (kollektiv Arithmetik-Logikeinheiten (ALUs), zu konfigurieren. In mindestens einer Ausführungsform lädt Code, wie beispielhaften Graphencode, Gewichtung oder andere Parameterinformationen in Prozessor ALUs basierend auf einer Architektur eines neuronalen Netzwerks, dem der Code entspricht. In mindestens einer Ausführungsform speichert Code und/oder Datenspeicherung 901 Gewichtungsparameter und/oder Eingangs-/Ausgangsdaten von jeder Schicht eines neuronalen Netzwerks, das trainiert oder in Verbindung mit einer oder mehrerer Ausführungsformen verwendet wird, während Vorwärtspropagation von Eingangs-/Ausgangsdaten, und/oder Gewichtungsparameter während des Trainierens und/oder Inferenzierens unter Verwendung von Aspekten einer oder mehrerer Ausführungsformen. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt des Codes und/oder der Datenspeicherung 901 mit anderer On-Chip- oder Off-Chip-Datenspeicherung, einschließlich L1-, L2- oder L3-Cache eines Prozessors oder Systemspeichers, enthalten sein.

[0034] In mindestens einer Ausführungsform kann jeder beliebige Abschnitt der Datenspeicherung 901 intern oder extern zu einem oder mehreren Prozessoren oder anderen HardwareLogikvorrichtungen oder Schaltungen sein. In mindestens einer Ausführungsform kann die Datenspeicherung 901 Cache-Speicher, dynamisches RAM („DRAM“), statisches RAM („SRAM“), nichtflüchtiger Speicher (z.B. Flash-Speicher) oder eine andere Speicherung sein. In mindestens einer Ausführungsform kann die Wahl, ob die Datenspeicherung 901 beispielsweise intern oder extern zu einem Prozessor ist oder aus DRAM, SRAM, Flash-Speicher oder einem anderen Speichertyp besteht, von auf dem Chip bzw. on-chip gegenüber nicht auf dem Chip bzw. off-chip verfügbarem Speicher, Latenzanforderungen der durchgeführten Trainings- und/oder Inferenzierungsfunktionen, der Losgröße der bei der Inferenzierung und/oder dem Training eines neuronalen Netzwerks verwendeten Daten oder einer Kombination dieser Faktoren abhängen.

[0035] In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915, ohne darauf beschränkt zu sein, einen Code und/oder eine Datenspeicherung 905 zum Speichern von Rückwärts-

und/oder Ausgangsgewichtung und/oder Eingangs-/Ausgangsdaten umfassen, die Neuronen oder Schichten eines neuronalen Netzwerks entsprechen, das zur Inferenzierung in Aspekten einer oder mehrerer Ausführungsformen trainiert und/oder verwendet wird. In mindestens einer Ausführungsform speichert Code und/oder Datenspeicherung 905 Gewichtungsparameter und/oder Eingangs-/Ausgangsdaten von jeder Schicht eines neuronalen Netzwerks, das trainiert oder in Verbindung mit einer oder mehrerer Ausführungsformen während Rückwärtspropagation von Eingangs-/Ausgangsdaten und/oder Gewichtungsparameter während Trainieren und/oder Inferenzieren unter Verwendung Aspekten einer oder mehrerer Ausführungsformen verwendet wird. In mindestens einer Ausführungsform kann die Trainingslogik 915 Code und/oder Datenspeicherung 905 umfassen oder gekoppelt sein mit, um Graphencode oder andere Software zu speichern, um das Timing und/oder die Reihenfolge zu steuern, bei denen Gewichtung und/oder andere Parameterinformationen zu laden sind, um Logik, einschließlich Ganzzahl- und/oder Gleitkomma-Einheiten (zusammen, Arithmetik-Logikeinheiten (ALUs), zu konfigurieren. In mindestens einer Ausführungsform lädt Code, wie beispielhafte Graphencode, Gewichtung oder andere Parameterinformationen in Prozessor-ALUs basierend auf einer Architektur eines neuronalen Netzwerks, dem der Code entspricht. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt von Code und/oder Datenspeicherung 905 mit anderer on-chip oder off-chip Datenspeicherung enthalten sein, einschließlich eines L1-, L2- oder L3-Cache eines Prozessors oder Systemspeichers. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt von Code und/oder Datenspeicherung 905 intern oder extern auf einer oder mehreren Prozessoren oder andere Hardwarelogikvorrichtungen oder Schaltungen sein. In mindestens einer Ausführungsform kann die Code- und/oder Datenspeicherung 905 Cache-Speicher, DRAM, SRAM, nichtflüchtiger Speicher (z. B., Flash Speicher) oder eine andere Speicherung sein. In mindestens einer Ausführungsform kann die Wahl, ob die Code- und/oder Datenspeicherung 605 beispielsweise intern oder extern zu einem Prozessor ist oder aus DRAM, SRAM, Flash-Speicher oder einem anderen Speichertyp besteht, von dem On-Chip gegenüber dem Off-Chip verfügbaren Speicher, Latenzanforderungen von durchgeführten Trainings- und/oder Inferenzfunktionen, der Losgröße der bei dem Inferenzieren und/oder dem Training eines neuronalen Netzwerks verwendeten Daten oder einer Kombination dieser Faktoren abhängen.

[0036] In mindestens einer Ausführungsform können die Datenspeicherung 901 und Datenspeicherung 905 separate Speicherungsstrukturen sein. In mindestens einer Ausführungsform können die Datenspeicherung 901 und Datenspeicherung 905 von der gleichen Speicherungsstruktur sein. In mindestens einer Ausführungsform kann die Datenspeicherung 901 und Datenspeicherung 905 teilweise von gleicher Speicherungsstruktur und teilweise separater Speicherungsstrukturen sein. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt der Datenspeicherung 901 und Datenspeicherung 905 mit anderer on-chip oder off-chip Datenspeicherung enthalten sein, einschließlich L1-, L2- oder L3-Cache oder Systemspeicher eines Prozessors.

[0037] In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915, ohne darauf beschränkt zu sein, eine oder mehrere Arithmetiklogikeinheiten („ALU(s)“) 910 umfassen, einschließlich Ganzzahl- und/oder Gleitkomma-Einheiten, um logische und/oder mathematische Operationen basierend mindestens teilweise auf oder angegeben durch Training und/oder Inferenzcode (z. B., Graphencode) durchzuführen, wobei ein Ergebnis davon Aktivierungen erzeugen kann (z. B., Ausgangswerte von Schichten oder Neuronen innerhalb eines neuronalen Netzwerks), die in einer Aktivierungsspeicherung 920 gespeichert sind, die Funktionen von Eingang/Ausgang und/oder Gewichtungsparameterdaten sind, in Code und/oder Datenspeicherung 901 und/oder Code und/oder Datenspeicherung 905 gespeichert sind. In mindestens einer Ausführungsform werden Aktivierungen, die in der Aktivierungsspeicherung 920 gespeichert sind, gemäß von ALU(s) 910 durchgeführter linearer algebraischer und oder matrixbasierter Mathematik als Antwort auf das Durchführen von Anweisungen oder anderen Code erzeugt, wobei Gewichtungswerte, die in Code und/oder Datenspeicherung 905 und/oder Daten 901 gespeichert sind, als Operanden zusammen mit anderen Werten, wie beispielweise Bias-Werte, Gradienteninformationen, Impulswerte oder andere Parameter oder Hyperparameter, verwendet werden, wobei beliebige oder alle davon in Code und/oder Datenspeicherung 905 oder Code und/oder Datenspeicherung 901 oder einer anderen Speicherung on- oder off-chip gespeichert werden können.

[0038] In mindestens einer Ausführungsform sind ALU(s) 910 innerhalb einer oder mehrerer Prozessoren oder anderen Hardwarelogikvorrichtungen oder Schaltungen enthalten, wohingegen in einer anderen Ausführungsform eine oder mehrere ALUs 910 extern zu einem Prozessor oder einer andere Hardwarelogikvorrichtung oder Schaltung sein können, die sie verwenden (z. B., ein Coprozessor). In mindestens einer Ausführungsform können ALUs 910 innerhalb von Ausführungseinheiten eines Prozessors oder anderweitig innerhalb einer Bank von ALUs enthalten sein, die von Ausführungseinheiten entweder innerhalb des gleichen Prozessors zugänglich oder zwischen unterschiedlichen Prozessoren von unterschiedlichen Typen verteilt sind (z. B., Zentralverarbeitungseinheiten, Graphikverarbeitungseinheiten, feste Funktionseinheiten usw.). In

mindestens einer Ausführungsform können die Datenspeicherung 901, die Datenspeicherung 905 und die Aktivierungsspeicherung 920 auf dem gleichen Prozessor oder einer anderen Hardwarelogikvorrichtung oder Schaltung sein, wohingegen sie in einer anderen Ausführungsform in unterschiedlichen Prozessoren oder anderen Hardwarelogikvorrichtungen oder Schaltungen oder einer Kombination von gleichen und unterschiedlichen Prozessoren oder anderen Hardwarelogikvorrichtungen oder Schaltungen sein können. In mindestens einer Ausführungsform kann ein beliebiger Abschnitt der Aktivierungsspeicherung 920 mit anderer on-chip oder off-chip Datenspeicherung, einschließlich L1-, L2- oder L3-Cache oder Systemspeicher eines Prozessors, enthalten sein. Des Weiteren können Inferenzierungs- und/oder Trainingscode mit einem anderem Code gespeichert sein, der einem Prozessor oder anderer Hardwarelogik oder Schaltung zugänglich ist, und unter Verwendung von Abruf-, Decodier-, Planungs-, Ausführungs-, Ausmusterung- und/oder anderen logischen Schaltungen eines Prozessors geholt und/oder verarbeitet werden.

[0039] In mindestens einer Ausführungsform kann die Aktivierungsspeicherung 920 ein Cache-Speicher, DRAM, SRAM, nichtflüchtiger Speicher (z. B., Flash-Speicher) oder eine andere Speicherung sein. In mindestens einer Ausführungsform kann die Aktivierungsspeicherung 920 vollständig oder teilweise innerhalb oder außerhalb einer oder mehrerer Prozessoren oder anderen logischen Schaltungen sein. In mindestens einer Ausführungsform kann die Wahl, ob die Aktivierungsspeicherung 920 beispielsweise intern oder extern zu einem Prozessor ist oder aus DRAM, SRAM, Flash oder einem anderen Speichertyp gebildet wird, von der verfügbaren Speicherung on-chip versus off-chip, Latenzanforderungen von durchgeführten Trainings- und/oder Inferenzierungsfunktionen, Losgröße von Daten, die beim Inferenzieren und/oder Trainieren eines neuronalen Netzwerks verwendet werden, oder eine Kombination dieser Faktoren abhängen. In mindestens einer Ausführungsform kann die in **Fig. 9A** veranschaulichte Inferenzierungs- und/oder Trainingslogik 915 in Verbindung mit einer anwendungsspezifischen integrierten Schaltung („ASIC“), wie beispielhafte Tensorflow® Verarbeitung Unit von Google, einer Inferenzverarbeitungseinheit (IPU) von Graphcore™ oder einem Nervana® (z. B., „Lake Crest“) Prozessor von Intel Corp. verwendet werden. In mindestens einer Ausführungsform kann die in **Fig. 9A** veranschaulichte Inferenzierungs- und/oder Trainingslogik 915 in Verbindung mit Zentralverarbeitung („CPU“)-Hardware, Graphikverarbeitungseinheit („GPU“)-Hardware oder anderer Hardware, wie beispielhafte feldprogrammierbare Gate-Arrays (Field Programmable Gate Arrays; „FPGAs“) verwendet werden.

[0040] **Fig. 9B** veranschaulicht die Inferenzierungs- und/oder Trainingslogik 915 gemäß mindestens einer verschiedenen Ausführungsform. In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915, ohne darauf beschränkt zu sein, Hardwarelogik umfassen, bei welcher Rechenressourcen fest zugeordnet sind oder anderweitig exklusiv in Verbindung mit Gewichtungswerten oder anderen Informationen verwendet werden, die einer oder mehrerer Schichten von Neuronen innerhalb eines neuronalen Netzwerks entsprechen. In mindestens einer Ausführungsform kann die in **Fig. 9B** veranschaulichte Inferenzierungs- und/oder Trainingslogik 915 in Verbindung mit einer anwendungsspezifischen integrierten Schaltung (ASIC) verwendet werden, wie beispielhafte Tensorflow® Verarbeitungseinheit von Google, einer Inferenzverarbeitungseinheit (IPU) von Graphcore™ oder einem Nervana® (z. B., „Lake Crest“) Prozessor von Intel Corp. In mindestens einer Ausführungsform kann die in **Fig. 9B** veranschaulichte Inferenzierungs- und/oder Trainingslogik 915 in Verbindung mit Zentralverarbeitung (CPU)-Hardware, Graphikverarbeitungseinheit (GPU)-Hardware oder andere Hardware, wie beispielhafte feldprogrammierbare Gate-Arrays (Field Programmable Gate Arrays; FPGAs) verwendet werden. In mindestens einer Ausführungsform umfasst die Inferenzierungs- und/oder Trainingslogik 915, ohne darauf beschränkt zu sein, Code und/oder Datenspeicherung 901 und Code und/oder Datenspeicherung 905, die verwendet werden können, um Code (z. B., Graphencode), Gewichtungswerte und/oder andere Informationen, einschließlich Bias-Werte, Gradienteninformationen, Impulswerte und/oder andere Parameter- oder Hyperparameter-Informationen, zu speichern. In mindestens einer in **Fig. 9B** veranschaulichten Ausführungsform ist jeweils Code und/oder Datenspeicherung 901 und Code und/oder Datenspeicherung 905 einer fest zugeordneten Rechenressource zugeordnet, wie die beispielhafte Rechenhardware 902 und Rechenhardware 906. In mindestens einer Ausführungsform umfassen die Rechenhardware 902 und Rechenhardware 906 jeweils eine oder mehrere ALUs, die mathematische Funktionen, wie beispielhafte Funktionen der linearen Algebra, jeweils nur auf in Code und/oder Datenspeicherung 901 und Code und/oder Datenspeicherung 905 gespeicherten Informationen durchführen, deren Ergebnis in der Aktivierungsspeicherung 920 gespeichert ist.

[0041] In mindestens einer Ausführungsform entsprechen die Datenspeicherung 901 und 905 und die entsprechende Rechenhardware 902 und 906 jeweils unterschiedlichen Schichten eines neuronalen Netzwerks, so dass die resultierende Aktivierung eines „Speicher/Rechen-Paares 901/902“ der Datenspeicherung 901 und Rechenhardware 902 als eine Eingabe in das nächste „Speicher/Rechen Paar 905/906“ der Datenspeicherung 905 und Rechenhardware 906 bereitgestellt wird, um die konzeptionelle Organisation eines neuro-

nen Netzes zu spiegeln. In mindestens einer Ausführungsform kann jedes der Speicher/Rechenpaare 901/902 und 905/906 mehr als einer neuronalen Netzwerkschicht entsprechen. In mindestens einer Ausführungsform können zusätzliche Speicher/Rechenpaare (nicht gezeigt) anschließend an oder parallel mit Speicherungs-Rechenpaare 901/902 und 905/906 in der Inferenzierungs- und/oder Trainingslogik 915 enthalten sein.

TRAINING UND EINSATZ EINES NEURONALEN NETZWERKS

[0042] Fig. 10 veranschaulicht Training und Einsatz eines tiefen neuronalen Netzes gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform wird ein untrainiertes neuronales Netzwerk 1006 unter Verwendung eines Trainingsdatensatzes 1002 trainiert. In mindestens einer Ausführungsform ist das Training-Framework 1004 ein PyTorch-Framework, wohingegen in anderen Ausführungsformen das Training-Framework 1004 ein Tensorflow-, Boost-, Caffe-, Microsoft Cognitive Toolkit/CNTK-, MXNet-, Chainer-, Keras-, Deeplearning4j- oder ein anderes Training-Framework ist. In mindestens einer Ausführungsform trainiert das Training-Framework 1004 ein untrainiertes neuronales Netzwerk 1006 und ermöglicht dessen Training unter Verwendung von hier beschriebenen Verarbeitungsressourcen, um ein trainiertes neuronales Netzwerk 1008 zu erzeugen. In mindestens einer Ausführungsform können Gewichte zufällig oder durch Vor-training unter Verwendung eines Netzes tiefen Glaubens gewählt werden. In mindestens einer Ausführungsform kann das Training entweder in einer beaufsichtigten, teilweise beaufsichtigten oder unbeaufsichtigten Art und Weise durchgeführt werden.

[0043] In mindestens einer Ausführungsform wird ein untrainiertes neuronales Netzwerk 1006 unter Verwendung von beaufsichtigtem Lernen trainiert, wobei der Trainingsdatensatz 1002 eine Eingabe umfasst, die mit einer gewünschten Ausgabe für eine Eingabe gepaart wird, oder wobei der Trainingsdatensatz 1002 eine Eingabe mit einer bekannten Ausgabe umfasst und eine Ausgabe des neuronalen Netzes 1006 manuell eingestuft wird. In mindestens einer Ausführungsform wird ein untrainiertes neuronales Netzwerk 1006 in einer beaufsichtigten Art und Weise trainiert, verarbeitet Eingaben von dem Trainingsdatensatz 1002 und vergleicht resultierende Ausgaben gegen einen Satz von erwarteten oder gewünschten Ausgaben. In mindestens einer Ausführungsform werden Fehler dann durch das untrainierte neuronale Netzwerk 1006 zurück propagiert. In mindestens einer Ausführungsform stellt das Training-Framework 1004 Gewichte ein, die das untrainierte neuronale Netzwerk 1006 steuern. In mindestens einer Ausführungsform umfasst das Training-Framework 1004 Tools, um zu überwachen, wie gut das untrainierte neuronale Netzwerk 1006 zu einem Modell hin konvergiert, wie beispielsweise das trainierte neuronale Netzwerk 1008, das geeignet ist, korrekte Antworten, wie beispielsweise im Ergebnis 1014, basierend auf bekannten Eingangsdaten, wie beispielsweise neue Daten 1012 zu erzeugen. In mindestens einer Ausführungsform trainiert das Training-Framework 1004 das untrainierte neuronale Netzwerk 1006 wiederholt, während Gewichte eingestellt werden, um eine Ausgabe des untrainierten neuronalen Netzes 1006 unter Verwendung einer Verlustfunktion und eines Einstellalgorithmus, wie beispielsweise eines stochastischen Gradientenabstiegs, zu verfeinern. In mindestens einer Ausführungsform trainiert das Training-Framework 1004 das untrainierte neuronale Netzwerk 1006, bis das untrainierte neuronale Netzwerk 1006 eine gewünschte Genauigkeit erreicht. In mindestens einer Ausführungsform kann dann das trainierte neuronale Netzwerk 1008 eingesetzt werden, um eine beliebige Anzahl von Maschinenlernoperationen zu implementieren.

[0044] In mindestens einer Ausführungsform wird das untrainierte neuronale Netzwerk 1006 unter Verwendung unbeaufsichtigten Lernens trainiert, wobei das untrainierte neuronale Netzwerk 1006 versucht, sich selbst unter Verwendung von ungekennzeichneten Daten zu trainieren. In mindestens einer Ausführungsform wird der Trainingsdatensatz für unbeaufsichtigtes Lernen 1002 Eingangsdaten ohne irgendwelche zugeordneten Ausgangsdaten oder „Ground-Truth“-Daten umfassen. In mindestens einer Ausführungsform kann das untrainierte neuronale Netzwerk 1006 Gruppierungen innerhalb des Trainingsdatensatzes 1002 lernen und kann bestimmen, wie einzelne Eingaben mit dem untrainierten Datensatz 1002 in Beziehung stehen. In mindestens einer Ausführungsform kann unbeaufsichtigtes Training verwendet werden, um eine selbstorganisierende Karte zu erzeugen, die ein Typ eines trainierten neuronalen Netzes 1008 ist, das imstande ist, Operationen durchzuführen, die beim Verringern der Dimensionalität neuer Daten 1012 nützlich sind. In mindestens einer Ausführungsform kann das unbeaufsichtigte Training ebenfalls verwendet werden, um eine Anomalieerkennung durchzuführen, die eine Identifikation von Datenpunkten in neuen Daten 1012 ermöglicht, die von normalen Mustern des neuen Datensatzes 1012 abweichen.

[0045] In mindestens einer Ausführungsform kann halbbeaufsichtigtes Lernen verwendet werden, das eine Technik ist, bei welcher ein Trainingsdatensatz 1002 eine Mischung aus gekennzeichneten und nicht gekennzeichneten Daten umfasst. In mindestens einer Ausführungsform kann das Training-Framework 1004

verwendet werden, um inkrementelles Lernen durchzuführen, wie beispielsweise durch transferierte Lern-techniken. In mindestens einer Ausführungsform ermöglicht das inkrementelle Lernen dem trainierten neuronalen Netzwerk 1008, sich an neue Daten 1012 anzupassen, ohne das Wissen zu vergessen, das während des anfänglichen Trainings in das Netzwerk eingebracht wurde.

RECHENZENTRUM

[0046] Fig. 11 veranschaulicht ein beispielhaftes Rechenzentrum 1100, in welchem mindestens eine Ausführungsform verwendet werden kann. In mindestens einer Ausführungsform umfasst ein Rechenzentrum 1100 eine Rechenzentrum-Infrastrukturschicht 1110, eine Framework-Schicht 1120, eine Softwareschicht 1130 und eine Anwendungsschicht 1140.

[0047] In mindestens einer Ausführungsform kann, wie in Fig. 11 gezeigt, eine Rechenzentrum-Infrastrukturschicht 1110 einen Ressourcenorchestrator 1112, gruppierte Rechenressourcen 1114 und Knotenrechenressourcen („Knoten-C.R.“) 1116(1)-1116(N) umfassen, wobei „N“ eine beliebige positive ganze Zahl darstellt. In mindestens einer Ausführungsform können die Knoten-C.R.s 1116(1)-1116(N), sind jedoch nicht beschränkt auf, eine beliebige Anzahl von zentralen Verarbeitungseinheiten („CPUs“) oder anderen Prozessoren (einschließlich Beschleunigern, feldprogrammierbaren Gate-Arrays (FPGAs), Graphikprozessoren usw.), Speichervorrichtungen, (z.B. dynamischer Nur-Lese-Speicher), Speichervorrichtungen (z.B. Solid-State- oder Festplattenlaufwerke), Netzwerk-Eingabe-/Ausgabe („NW E/A“-)Vorrichtungen, Netzwerkschalter, virtuelle Maschinen („VMs“), Leistungsmodul und Kühlmodul usw. umfassen. In mindestens einer Ausführungsform können ein oder mehrere Knoten-C.R.s aus den Knoten-C.R.s 1116(1)-1116(N) ein Server mit einer oder mehreren der oben erwähnten Rechenressourcen sein.

[0048] In mindestens einer Ausführungsform können gruppierte Rechenressourcen 1114 separate Gruppierungen von Knoten-C.R.s. umfassen, die in einem oder mehreren Racks (nicht gezeigt) untergebracht sind, oder in vielen Racks, die in Rechenzentren an verschiedenen geografischen Standorten untergebracht sind (ebenfalls nicht gezeigt). Separate Gruppierungen von Knoten-C.R.s. können innerhalb gruppierter Rechenressourcen 1114 gruppierte Rechen-, Netzwerk-, Speicher- oder Speicherungsressourcen umfassen, die konfiguriert oder zugeteilt sein können, um eine oder mehrere Arbeitslasten zu unterstützen. In mindestens einer Ausführungsform können mehrere Knoten-C.R.s einschließlich CPUs oder Prozessoren innerhalb eines oder mehrerer Racks gruppiert werden, um Rechenressourcen zur Unterstützung einer oder mehrerer Arbeitslasten bereitzustellen. In mindestens einer Ausführungsform können ein oder mehrere Racks ebenfalls eine beliebige Anzahl von Leistungsmodulen, Kühlmodulen und in beliebiger Kombination umfassen.

[0049] In mindestens einer Ausführungsform kann ein Ressourcenorchestrator 1112 eine oder mehrere Knoten-C.R.s 1116(1)-1116(N) und/oder gruppierte Rechenressourcen 1114 konfigurieren oder anderweitig steuern. In mindestens einer Ausführungsform kann der Ressourcenorchestrator 1112 eine Software-Design-Infrastruktur („SDI“-)Verwaltungseinheit für ein Rechenzentrum 1100 umfassen. In mindestens einer Ausführungsform kann ein Ressourcenorchestrator Hardware, Software oder eine Kombination davon umfassen.

[0050] In mindestens einer Ausführungsform umfasst, wie in Fig. 11 gezeigt, die Framework-Schicht 1120 einen Job-Scheduler 1122, einen Konfigurationsverwalter 1124, einen Ressourcenverwalter 1126 und ein verteiltes Dateisystem 1128. In mindestens einer Ausführungsform kann die Framework-Schicht 1120 ein Framework zur Unterstützung der Software 1132 der Softwareschicht 1130 und/oder eine oder mehrere Anwendungen 1142 der Anwendungsschicht 1140 umfassen. In mindestens einer Ausführungsform kann die Software 1132 oder können die Anwendung(en) 1142 jeweils webbasierte Dienstsoftware oder Anwendungen umfassen, wie beispielsweise diejenigen, die von Amazon Web Services, Google Cloud und Microsoft Azure bereitgestellt werden. In mindestens einer Ausführungsform kann, ohne darauf beschränkt zu sein, die Framework-Schicht 1120 eine Art Framework für eine freie und quelloffene Software-Webanwendung-Framework wie beispielsweise Apache Spark™ (nachfolgend als „Spark“ bezeichnet) sein, welches das verteilte Dateisystem 1128 für eine groß angelegte Datenverarbeitung (z.B. „big data“) nutzen kann. In mindestens einer Ausführungsform kann der Job-Scheduler 1132 einen Spark-Treiber umfassen, um die Planung von Arbeitslasten zu erleichtern, die von verschiedenen Schichten des Rechenzentrums 1100 unterstützt werden. In mindestens einer Ausführungsform kann der Konfigurationsverwalter 1124 in der Lage sein, verschiedene Schichten wie beispielsweise die Softwareschicht 1130 und die Framework-Schicht 1120 einschließlich Spark und des verteilten Dateisystems 1128 zur Unterstützung der groß angelegten Datenverarbeitung zu konfigurieren. In mindestens einer Ausführungsform kann der Ressourcenverwalter 1126 in der Lage sein, geclusterte oder gruppierte Rechenressourcen zu verwalten, die zur Unterstützung des verteilten Dateisystems 1128 und des Job-Planers 1122 gemappt oder zugeteilt sind. In mindestens einer Ausführungsform

rungsform können geclusterte oder gruppierte Rechenressourcen die gruppierte Rechenressource 1114 auf der Rechenzentrum-Infrastrukturschicht 1110 umfassen. In mindestens einer Ausführungsform kann sich der Ressourcenverwalter 1126 mit dem Ressourcenorchestrator 1112 koordinieren, um diese gemappten oder zugewiesenen Rechenressourcen zu verwalten.

[0051] In mindestens einer Ausführungsform kann die in der Softwareschicht 1130 enthaltene Software 1132 Software, die von mindestens Teilen der Knoten-C.R.s 1116(1)-1116(N) verwendet wird, gruppierte Rechenressourcen 1114 und/oder das verteilte Dateisystem 1128 der Framework-Schicht 1120 umfassen. Eine oder mehrere Arten von Software können, ohne darauf beschränkt zu sein, Internet-Webseiten-Suchsoftware, E-Mail-Virenschanner-Software, Datenbanksoftware und Streaming-Videoinhalt-Software umfassen.

[0052] In mindestens einer Ausführungsform kann(können) die in der Anwendungsschicht 1140 enthaltene(n) Anwendung(en) 1142 eine oder mehrere Arten von Anwendungen, die von mindestens Teilen der Knoten-C.R.s 1116(1)-1116(N) verwendet werden, gruppierte Rechenressourcen 1114 und/oder das verteilte Dateisystem 1128 der Framework-Schicht 1120 umfassen. Eine oder mehrere Arten von Anwendungen können, ohne darauf beschränkt zu sein, eine beliebige Anzahl von genomischen Anwendungen, ein kognitives Berechnen und maschinellen Lernen, einschließlich Trainings- oder Inferenzsoftware, Framework-Software für maschinelles Lernen (z.B. PyTorch, TensorFlow, Caffe usw.) oder andere Anwendungen maschinellen Lernens, umfassen, die in Verbindung mit einer oder mehreren Ausführungsformen verwendet werden.

[0053] In mindestens einer Ausführungsform kann jeder der Konfigurationsmanager 1134, des Ressourcenmanager 1136 und des Ressourcenorchestrators 1112, basierend auf einer beliebigen Menge und der Art von Daten, die in beliebig technisch machbarer Weise erfasst wurden, eine beliebige Anzahl und Art von selbstmodifizierenden Aktionen implementieren. In mindestens einer Ausführungsform können selbstmodifizierende Aktionen einen Betreiber des Rechenzentrums 1100 davon befreien, möglicherweise schlechte Konfigurationsentscheidungen zu treffen und möglicherweise ungenutzte und/oder schlecht funktionierende Abschnitte eines Rechenzentrums zu vermeiden.

[0054] In mindestens einer Ausführungsform kann ein Rechenzentrum 1100 Tools, Dienste, Software oder andere Ressourcen umfassen, um ein oder mehrere Modelle maschinellen Lernens zu trainieren oder Informationen unter Verwendung eines oder mehrerer Modelle maschinellen Lernens vorherzusagen oder abzuleiten, gemäß einer oder mehrerer hier beschriebener Ausführungsformen. Beispielsweise kann in mindestens einer Ausführungsform ein Modell maschinellen Lernens trainiert werden, indem Gewichtsparameter gemäß einer neuronalen Netzwerkarchitektur unter Verwendung der vorstehend in Bezug auf ein Rechenzentrum 1100 beschriebenen Software und Rechenressourcen berechnet werden. In mindestens einer Ausführungsform können trainierte Modelle maschinellen Lernens, die einem oder mehreren neuronalen Netzwerken entsprechen, verwendet werden, um Informationen unter Verwendung der vorstehend in Bezug auf ein Rechenzentrum 1100 beschriebenen Ressourcen unter Verwendung von Gewichtsparametern, die durch eine oder mehrere der hierin beschriebenen Trainingstechniken berechnet wurden, abzuleiten oder vorherzusagen.

[0055] In mindestens einer Ausführungsform kann ein Rechenzentrum CPUs, anwendungsspezifische integrierte Schaltungen (ASICs), GPUs, FPGAs oder andere Hardware verwenden, um Training und/oder Inferenzieren unter Verwendung der vorstehend beschriebenen Ressourcen durchzuführen. Außerdem können eine oder mehrere der vorstehend beschriebenen Software- und/oder Hardware-Ressourcen als ein Dienst konfiguriert sein, um Benutzern zu erlauben, Informationen, wie beispielsweise Bilderkennung, Spracherkennung oder andere Dienste künstlicher Intelligenz, zu trainieren oder eine Inferenzierung derselben durchzuführen.

[0056] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einem oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten zur Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform kann in dem System von **Fig. 11** die Inferenzierungs- und/oder Trainingslogik 915 zum Inferenzieren oder Vorhersagen von Operationen verwendet werden, die mindestens teilweise auf Gewichtsparametern basieren, die unter Verwendung von Trainingsoperationen für neuronale Netzwerke, Funktionen und/oder Architekturen neuronaler Netzwerke oder hier beschriebenen Anwendungsfällen neuronaler Netzwerke berechnet wurden.

AUTONOMES FAHRZEUG

[0057] Fig. 12A veranschaulicht ein Beispiel eines autonomen Fahrzeugs 1200 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann das autonome Fahrzeug 1200 (alternativ hier als „Fahrzeug 1200“ bezeichnet), ohne darauf beschränkt zu sein, ein Personenkraftwagen, wie beispielsweise ein Pkw, ein Lastkraftwagen, ein Bus und/oder ein anderer Typ von Fahrzeug sein, das einen oder mehrere Fahrgäste aufnehmen kann. In mindestens einer Ausführungsform kann ein Fahrzeug 1200 ein Halb-Zugmaschinen-Anhänger-Lastwagen sein, der zum Ziehen von Fracht verwendet wird. In mindestens einer Ausführungsform kann ein Fahrzeug 1200 ein Flugzeug, ein Roboterfahrzeug oder ein anderer Typ von Fahrzeug sein.

[0058] Autonome Fahrzeuge können im Allgemeinen hinsichtlich Automatisierungsgrade beschrieben werden, die von der National Highway Traffic Safety Administration („NHTSA“), einer Abteilung des US Department of Transportation, und dem Standard „Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles“ der Society of Automotive Engineers („SAE“) (z. B. Standard Nr. J3016-201806, veröffentlicht am 15. Juni 2018, Standard Nr. J3016-201609, veröffentlicht am 30. September 2016, und frühere und zukünftige Versionen dieser Norm) definiert wird. In einer oder mehreren Ausführungsformen kann ein Fahrzeug 1200 in Übereinstimmung mit einer oder mehreren der autonomen Fahrstufen 1 bis 5 funktionsfähig sein. Beispielsweise kann ein Fahrzeug 1200 in mindestens einer Ausführungsform in der Lage sein, eine bedingte Automatisierung (Stufe 3), eine hohe Automatisierung (Stufe 4) und/oder eine vollständige Automatisierung (Stufe 5) abhängig von der Ausführungsform durchzuführen.

[0059] In mindestens einer Ausführungsform kann ein Fahrzeug 1200 Komponenten, wie beispielsweise ein Fahrgestell, eine Fahrzeugkarosserie, Räder (z. B. 2, 4, 6, 8, 18 usw.), Reifen, Achsen und andere Komponenten eines Fahrzeugs umfassen. In mindestens einer Ausführungsform kann ein Fahrzeug 1200 ein Antriebssystem 1250, wie beispielsweise einen Verbrennungsmotor, eine Hybrid-Elektroanlage, einen vollelektrischen Motor und/oder eine andere Art eines Antriebssystems umfassen. In mindestens einer Ausführungsform kann das Antriebssystem 1250 mit einem Antriebsstrang des Fahrzeugs 1200 verbunden sein, der ein Getriebe umfassen kann, ohne darauf beschränkt zu sein, um den Antrieb eines Fahrzeugs 1200 zu ermöglichen. In mindestens einer Ausführungsform kann ein Antriebssystem 1250 als Antwort auf das Empfangen von Signalen von einer oder mehreren Drossel/einem Beschleuniger 1252 gesteuert werden.

[0060] In mindestens einer Ausführungsform wird ein Lenksystem 1254, das ein Lenkrad, ohne darauf beschränkt zu sein, umfassen kann, verwendet, um ein Fahrzeug 1200 (z. B. entlang eines gewünschten Wegs oder Route) zu lenken, wenn ein Antriebssystem 1250 in Betrieb ist (z. B., wenn ein Fahrzeug 1200 in Bewegung ist). In mindestens einer Ausführungsform kann ein Lenksystem 1254 Signale von einem oder mehreren Lenkaktuatoren 1256 empfangen. Ein Lenkrad kann für die Funktionalität der Vollautomatisierung (Stufe 5) optional sein. In mindestens einer Ausführungsform kann ein Bremssensorsystem 1246 verwendet werden, um Fahrzeugbremsen als Antwort auf das Empfangen von Signalen von dem(den) Bremsaktuator(en) 1248 und/oder Bremssensoren zu betätigen.

[0061] In mindestens einer Ausführungsform stellen der(die) Controller 1236, der(die) ein oder mehrere Systeme on Chips („SoCs“) (in Fig. 12A nicht gezeigt) und/oder Graphikverarbeitungseinheiten („GPU(s)“) umfassen kann(können), Signale (z. B. welche Befehle repräsentieren) einer oder mehreren Komponenten und/oder Systemen des Fahrzeugs 1200 bereit. Beispielsweise kann(können) der(die) Controller 1236 Signale zum Betätigen von Fahrzeugbremsen über ein oder mehrere Bremsaktuatoren 1248, zum Betätigen des Lenksystems 1254 über ein oder mehrere Lenkaktuatoren 1256 und zum Betätigen des Antriebssystems 1250 über ein oder mehrere Drosseln/Beschleuniger 1252 senden. In mindestens einer Ausführungsform kann(können) der(die) Controller 1236 eine oder mehrere eingebaute (z. B. integrierte) Rechenvorrichtungen (z. B. Supercomputer) umfassen, die Sensorsignale verarbeiten und Betriebsbefehle (z. B. Signale, die Befehle repräsentieren) ausgeben, um autonomes Fahren zu ermöglichen und/oder einen menschlichen Fahrer beim Fahren des Fahrzeugs 1200 zu unterstützen. In mindestens einer Ausführungsform kann(können) der(die) Controller 1236 einen ersten Controller für autonome Fahrfunktionen, einen zweiten Controller 1236 für funktionale Sicherheitsfunktionen, einen dritten Controller 1236 für Funktionen der künstlichen Intelligenz (z. B. Computer-Vision), einen vierten Controller 1236 für Infotainment-Funktionalität, einen fünften Controller 1236 für Redundanz bei Notfällen und/oder andere Controller umfassen. In mindestens einer Ausführungsform kann ein einzelner Controller 1236 zwei oder mehrere oben beschriebene Funktionalitäten handhaben, zwei oder mehr Controller 1236 können eine einzelne Funktionalität und/oder eine beliebige Kombination davon handhaben.

[0062] In mindestens einer Ausführungsform kann(können) der(die) Controller 1236 Signale zum Steuern einer oder mehrerer Komponenten und/oder Systeme des Fahrzeugs 1200 als Antwort auf Sensordaten bereitstellen, die von einem oder mehreren Sensoren (z. B. Sensoreingaben) empfangen werden. In mindestens einer Ausführungsform können die Sensordaten, beispielsweise und ohne darauf beschränkt zu sein, von Sensor(en) von globalen Navigationssatellitensystemen 1258 (z. B. Global Positioning System Sensor (s); „GNSS“), RADAR-Sensor(en) 1260, Ultraschallsensor(en) 1262, LIDAR-Sensor(en) 1264, Inertial Measurement Unit (IMU) Sensor(en) 1266 (z. B. Beschleunigungssensor(en), Gyroskop(e), Magnetkompass(e), Magnetometer(e), usw.), Mikrophon(e) 1296, Stereokamera(s) 1268, Weitwinkelkamera(s) 1270 (z. B. Fischaugenkameras), Infrarot-Kamera(s) 1072, Surround-Kamera(s) 1274 (z. B. 360-Grad-Kameras), Fernbereichskameras (in **Fig. 12A** nicht gezeigt), Mittelbereichskamera(s) (in **Fig. 12A** nicht gezeigt), Geschwindigkeitssensor(en) 1244 (z. B. zur Messung der Geschwindigkeit des Fahrzeugs 1200), Schwingungssensor(en) 1242, Lenksensor(en) 1240, Bremssensor(en) (z. B. als Teil des Bremssensorsystems 1246) und/oder anderen Sensortypen empfangen werden.

[0063] In mindestens einer Ausführungsform können ein oder mehrere Controller 1236 Eingaben (z. B. repräsentiert durch Eingabedaten) von einer Instrumentengruppe 1232 des Fahrzeugs 1200 empfangen und Ausgaben (z. B., repräsentiert durch Ausgabedaten, Anzeigedaten usw.) über eine Anzeige einer Mensch-Maschine-Schnittstelle (Human-Machine Interface; „HMI“) 1234, eine hörbare Meldevorrichtung, einen Lautsprecher und/oder über andere Komponenten des Fahrzeugs 1200 bereitstellen. In mindestens einer Ausführungsform können die Ausgaben Informationen, wie beispielsweise Fahrzeugvektorgeschwindigkeit, Geschwindigkeit, Zeit, Kartendaten (z. B. eine Hochdefinitions-Karte (in **Fig. 12A** nicht gezeigt), Standortdaten (z. B. den Standort des Fahrzeugs 1200, wie z. B. auf einer Karte), Richtung, Standort anderer Fahrzeuge (z. B. ein Belegungsgeber), Informationen über Objekte und Status von Objekten, wie von dem(den) Controllern 1236 wahrgenommen, usw. umfassen. Beispielsweise kann in mindestens einer Ausführungsform die HMI-Anzeige 1234 Informationen über das Vorhandensein eines oder mehrerer Objekte (z. B. eines Straßenschilds, eines Warnschilds, einer Ampeländerung usw.) und/oder Informationen über Fahrmanöver, die ein Fahrzeug durchführte, gerade durchführt oder durchführen wird (z. B. aktuelles Wechseln von Spuren, Nehmen einer Ausfahrt 34B in zwei Meilen, usw.), anzeigen.

[0064] In mindestens einer Ausführungsform umfasst ein Fahrzeug 1200 ferner eine Netzwerkschnittstelle 1224, die eine oder mehrere drahtlose Antenne(n) 1226 und/oder Modem(s) zur Kommunikation über ein oder mehrere Netzwerke verwenden kann. Beispielsweise kann in mindestens einer Ausführungsform eine Netzwerkschnittstelle 1224 imstande sein, über Long-Term Evolution („LTE“), Wide Code Division Multiple Access („WCDMA“), Universal Mobile Telecommunications System („UMTS“), Global System for Mobile Communication („GSM“), („CDMA2000“), IMT-CDMA Multi-Carrier („CDMA2000“) usw. zu kommunizieren. In mindestens einer Ausführungsform kann(können) die drahtlose(n) Antenne(n) 1226 ebenfalls eine Kommunikation zwischen Objekten in der Umgebung (z. B. Fahrzeugen, mobilen Vorrichtungen usw.) unter Verwendung von einem Lokalbereichsnetzwerk(en), wie beispielsweise Bluetooth, Bluetooth Low Energy („LE“), Z-Wave, ZigBee usw., und/oder einem Weitbereichsnetzwerk(en) mit geringer Leistung („LPWANs“), wie beispielsweise LoRaWAN, SigFox usw. Protokollen, ermöglichen.

[0065] Eine Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915 in einem System von **Fig. 12A** zum Inferenzieren oder Voraussagen von Operationen verwendet werden, die mindestens teilweise auf Gewichtungsparmeter basieren, die unter Verwendung von neuronalen Netzwerktrainingsoperationen, neuronalen Netzwerkfunktionen und/oder Architekturen, oder hier beschriebenen Anwendungsfällen eines neuronalen Netzwerks berechnet wurden.

[0066] **Fig. 12B** veranschaulicht ein Beispiel von Kamerastandorten und Sichtfeldern für das beispielhafte autonome Fahrzeug 1200 von **Fig. 12A** gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform sind die Kameras und jeweilige Sichtfelder eine beispielhafte Ausführungsform und sind nicht bestimmt, einschränkend zu sein. Beispielsweise können zusätzliche und/oder alternative Kameras umfasst sein und/oder die Kameras können an unterschiedlichen Stellen an einem Fahrzeug 1200 lokalisiert sein.

[0067] In mindestens einer Ausführungsform können Kameratypen für Kameras, sind jedoch nicht beschränkt darauf, Digitalkameras umfassen, die für eine Verwendung mit den Komponenten und/oder Systemen des Fahrzeugs 1200 angepasst sein können. In mindestens einer Ausführungsform kann(können) die Kamera(s) bei dem Automotive Safety Integrity Level (ASIL) B und/oder bei einem anderen ASIL arbeiten.

Die Kameratypen können für eine beliebige Bildaufnahmerate, z. B. 60 Einzelbilder pro Sekunde (fps), 1020 fps, 240 fps, usw. abhängig von der Umgebung fähig sein. In mindestens einer Ausführungsform können die Kameras in der Lage sein, rollende Verschlüsse, globale Verschlüsse, eine andere Art von Verschluss oder eine Kombination davon zu verwenden. In mindestens einer Ausführungsform kann ein Farbfilterarray ein Red Clear („RCCC“) Farbfilterarray, ein Red Clear Blue („RCCB“) Farbfilterarray, ein Red Blue Green Clear („RBGC“) Farbfilterarray, ein Foveon X3 Farbfilterarray, ein Bayer-Sensoren („RGGB“) Farbfilterarray, ein monochromes Sensorfarbfilterarray und/oder eine andere Art von Farbfilterarray umfassen. In mindestens einer Ausführungsform können klare Pixelkameras, wie beispielsweise Kameras mit einem RCCC-, einem RCCB- und/oder einem RBGC-Farbfilterarray, in einem Bemühen verwendet werden, die Lichtempfindlichkeit zu erhöhen.

[0068] In mindestens einer Ausführungsform kann(können) eine oder mehrere der Kameras verwendet werden, um Advanced Driver Assistance Systems („ADAS“)-Funktionen (z. B. als Teil eines redundanten oder ausfallsicheren Designs) durchzuführen. Beispielsweise kann in mindestens einer Ausführungsform eine Multifunktion-Monokamera installiert sein, um Funktionen bereitzustellen, die Spurabweichungswarnung, Verkehrszeichenunterstützung und intelligente Scheinwerfersteuerung umfassen. In mindestens einer Ausführungsform kann(können) eine oder mehrere Kameras (z. B. alle Kameras) Bilddaten (z. B. Video) gleichzeitig aufzeichnen und bereitstellen.

[0069] In mindestens einer Ausführungsform können eine oder mehrere Kameras in einer Montagebaugruppe, wie beispielsweise einer kundenspezifischen (dreidimensionalen („3D“) gedruckten) Baugruppe, angebracht sein, um Streulicht und Reflexionen aus dem Inneren eines Autos (z. B. Reflexionen vom Armaturenbrett, die in den Spiegeln der Windschutzscheibe reflektiert werden) abzubauen, welche die Fähigkeiten der Bilddatenerkennung der Kamera beeinträchtigen können. In Bezug auf die Montagebaugruppen der Außenspiegel können in mindestens einer Ausführungsform die Außenspiegelbaugruppen kundenspezifisch in 3D gedruckt werden, so dass eine Kameramontageplatte einer Form eines Außenspiegels entspricht. In mindestens einer Ausführungsform kann(können) die Kamera(s) in einem Außenspiegel integriert sein. Für Seitensichtkameras die Kamera(s) auch innerhalb vier Säulen an jeder Ecke einer Kabine integriert sein.

[0070] In mindestens einer Ausführungsform können Kameras mit einem Sichtfeld, das Teile einer Umgebung vor einem Fahrzeug 1200 umfasst (z. B. nach vorne gerichtete Kameras), für eine Rundumsicht verwendet werden, um dabei zu helfen, nach vorne gerichtete Wege und Hindernisse zu identifizieren, sowie auch dabei zu helfen, mit Hilfe eines oder mehrerer Controller 1236 und/oder Steuer-SoCs, wichtige Informationen zum Erzeugen eines Belegungsgitters und/oder zum Bestimmen der bevorzugten Fahrzeugwege bereitzustellen. In mindestens einer Ausführungsform können nach vorne gerichtete Kameras verwendet werden, um viele der gleichen ADAS-Funktionen, wie LIDAR, einschließlich, ohne darauf beschränkt zu sein, Notbremsung, Fußgängererkennung und Kollisionsvermeidung, durchzuführen. In mindestens einer Ausführungsform können nach vorne gerichtete Kameras auch für ADAS-Funktionen und -Systeme, einschließlich Spurverlassungswarnungen (Lane Departure Warnings; „LDW“), autonome Geschwindigkeitsregelung (Autonomous Cruise Control; „ACC“) und/oder andere Funktionen, wie beispielsweise Verkehrszeichenerkennung, verwendet werden.

[0071] In mindestens einer Ausführungsform kann eine Vielfalt von Kameras in einer nach vorne gerichteten Konfiguration verwendet werden, die beispielsweise eine monokulare Kameraplattform umfasst, die einen CMOS („complementary metal oxide semiconductor“)-Farbbildgeber umfasst. In mindestens einer Ausführungsform kann eine Weitwinkelkamera 1270 verwendet werden, um Objekte wahrzunehmen, die von einer Peripherie in Sicht kommen (z. B. Fußgänger, Kreuzverkehr oder Fahrräder). Obwohl nur eine Weitwinkelkamera 1270 in **Fig. 12B** veranschaulicht ist, kann es in anderen Ausführungsformen eine beliebige Anzahl (einschließlich null) von Weitwinkelkameras an einem Fahrzeug 1200 geben. In mindestens einer Ausführungsform kann eine beliebige Anzahl von Fernbereichskameras 1298 (z. B. ein Fernbereichs-Stereokamera-paar) zur tiefenbasierten Objekterkennung insbesondere für Objekte verwendet werden, für die ein neuronales Netzwerk noch nicht trainiert wurde. In mindestens einer Ausführungsform kann(können) eine Fernbereichskamera(s) 1298 ebenfalls zur Objekterkennung und -klassifizierung sowie auch zur grundlegenden Objektverfolgung eingesetzt werden.

[0072] In mindestens einer Ausführungsform kann eine beliebige Anzahl von Stereokameras 1268 ebenfalls in einer nach vorne gerichteten Konfiguration umfasst sein. In mindestens einer Ausführungsform kann eine beliebige Anzahl von Stereokameras 1268 eine integrierte Steuereinheit umfassen, die eine skalierbare Verarbeitungseinheit umfasst, die eine programmierbare Logik („FPGA“) und einen Mehrkernmikroprozessor mit einer integrierten Controller Area Network („CAN“) oder Ethernet-Schnittstelle auf einem einzigen Chip bereit-

stellen kann. In mindestens einer Ausführungsform kann eine derartige Einheit verwendet werden, um eine 3D-Karte der Umgebung des Fahrzeugs 1200 zu erzeugen, die eine Abstandsschätzung für alle Punkte in einem Bild umfasst. In mindestens einer Ausführungsform kann(können) eine alternative Stereokamera(s) 1268 einen kompakten Stereosichtsensor(en) umfassen, der zwei Kameraobjektive (je eine links und rechts) und einen Bildverarbeitungschip, ohne darauf beschränkt zu sein, umfassen kann, der den Abstand von einem Fahrzeug 1200 zu einem Zielobjekt messen und die erzeugten Informationen (z. B. Metadaten) verwenden kann, um die autonomen Funktionen der Notbremsung und Spurverlassungswarnung zu aktivieren. In mindestens einer Ausführungsform können andere Typen einer(von) Stereokamera(s) 1268 zusätzlich zu oder alternativ aus den hier beschriebenen verwendet werden.

[0073] In mindestens einer Ausführungsform können Kameras mit einem Sichtfeld, das Teile der Umgebung seitlich eines Fahrzeugs 1200 umfasst (z. B. Seitensichtkameras), für die Surround-Ansicht verwendet werden, die Informationen bereitstellt, die zum Erzeugen und Aktualisieren des Belegungsgitters sowie auch zum Erzeugen von Seitenaufprallwarnungen verwendet werden. Beispielsweise kann(können) in mindestens einer Ausführungsform die Surround-Kamera(s) 1274 (z. B. vier Surround-Kameras 1274, wie in **Fig. 12B** veranschaulicht) an einem Fahrzeug 1200 positioniert sein. Surround-Kamera(s) 1274 können, ohne darauf beschränkt zu sein, eine beliebige Anzahl und Kombination von Weitwinkelkamera(s) 1270, Fischaugenkamera(s), 360-Grad-Kamera(s) und/oder ähnliche Kameras umfassen. Beispielsweise können in mindestens einer Ausführungsform vier Fischaugenkameras an einer Vorderseite, einer Rückseite und den Seiten des Fahrzeugs 1200 positioniert werden. In mindestens einer Ausführungsform kann ein Fahrzeug 1200 drei Surround-Kameras 1274 (z. B. links, rechts und hinten) verwenden und kann eine oder mehrere andere Kameras (z. B. eine nach vorne gerichtete Kamera) als eine vierte Surround-View-Kamera wirksam einsetzen.

[0074] In mindestens einer Ausführungsform können Kameras mit einem Sichtfeld, das Teile einer Umgebung hinter einem Fahrzeug 1200 umfasst (z. B. Rückfahrkameras), für eine Einparkhilfe, eine Rundumsicht, Warnungen vor Heckkollision sowie zum Erzeugen und Aktualisieren eines Belegungsgitters verwendet werden. In mindestens einer Ausführungsform kann eine weite Vielfalt von Kameras verwendet werden, einschließlich, jedoch nicht beschränkt auf, Kameras, die auch als eine nach vorne gerichtete Kamera(s) geeignet sind (z. B. Fern- und/oder Mittelbereichskamera(s) 1276, Stereokamera(s) 1268, Infrarotkameras 1072 usw.), wie hier beschrieben.

[0075] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915 in dem System von **Fig. 12B** zum Inferenzieren oder Voraussagen von Operationen verwendet werden, die mindestens teilweise auf Gewichtungparameter basieren, die unter Verwendung von neuronalen Netzwerktrainingsoperationen, neuronalen Netzwerkfunktionen und/oder Architekturen, oder hier beschriebenen Anwendungsfällen eines neuronalen Netzwerks berechnet wurden.

[0076] **Fig. 12C** ist ein Blockdiagramm, das eine beispielhafte Systemarchitektur für das autonome Fahrzeug 1200 von **Fig. 12A** veranschaulicht, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform wird jede der Komponenten, Merkmale und Systeme des Fahrzeugs 1200 in **Fig. 12C** als über einen Bus 1202 verbunden veranschaulicht. In mindestens einer Ausführungsform kann ein Bus 1202 eine CAN(Controller Area Network)-Datenschnittstelle (alternativ hier als ein „CAN-Bus“ bezeichnet) umfassen. In mindestens einer Ausführungsform kann ein CAN ein Netzwerk innerhalb des Fahrzeugs 1200 sein, das verwendet wird, um bei der Steuerung unterschiedlicher Merkmale und der Funktionalität des Fahrzeugs 1200 zu helfen, wie beispielsweise der Betätigung von Bremsen, Beschleunigung, Bremsen, Lenkung, Scheibenwischern usw. In mindestens einer Ausführungsform kann der Bus 1202 konfiguriert sein, um Dutzende oder sogar Hunderte von Knoten jeweils mit seinem eigenen eindeutigen Identifikator (z. B. einer CAN-ID) aufzuweisen. In mindestens einer Ausführungsform kann ein Bus 1202 gelesen werden, um Lenkwinkel, Bodengeschwindigkeit, Motordrehzahlen pro Minute (U/min), Schalterpositionen und/oder andere Fahrzeugstatusanzeigen zu ermitteln. In mindestens einer Ausführungsform kann ein Bus 1202 ein CAN-Bus sein, der ASIL B konform ist.

[0077] In mindestens einer Ausführungsform kann zusätzlich zu oder alternativ von CAN, FlexRay- und/oder Ethernet-Protokolle verwendet werden. In mindestens einer Ausführungsform kann es eine beliebige Anzahl von Bussen geben, die den Bus 1202 bilden, der, ohne darauf beschränkt zu sein, null oder mehr CAN-Busse, null oder mehrere FlexRay-Busse, null oder mehrere Ethernet-Busse und/oder keinen und/oder null oder weitere andere Arten von Bussen mit einem unterschiedlichen Protokoll umfassen können. In mindes-

tens einer Ausführungsform können zwei oder mehr Busse 1202 verwendet werden, um unterschiedliche Funktionen auszuführen, und/oder können für Redundanz verwendet werden. Beispielsweise kann ein erster Bus 1202 für eine Kollisionsvermeidungsfunktionalität und ein zweiter Bus 1202 zur Betätigungssteuerung verwendet werden. In mindestens einer Ausführungsform kann jeder Bus 1202 mit beliebigen der Komponenten des Fahrzeugs 1200 kommunizieren, und zwei oder mehr Busse 1202 können mit gleichen Komponenten kommunizieren. In mindestens einer Ausführungsform kann jedes einer beliebigen Anzahl von System(en)-auf-Chip(s) (System on Chip(s); „SoC (s)“) (wie beispielsweise SoC 1204(A) und SoC 1204(B), jeder der Controller 1236 und/oder jeder Computer im Fahrzeug Zugriff auf die gleichen Eingangsdaten aufweisen (z. B. Eingaben von Sensoren des Fahrzeugs 1200) und mit einem gemeinsamen Bus, wie beispielsweise einem CAN-Bus, verbunden sein.

[0078] In mindestens einer Ausführungsform kann ein Fahrzeug 1200 einen oder mehrere Controller 1236 umfassen, wie beispielsweise jene, die hier in Bezug auf **Fig. 12A** beschrieben sind. In mindestens einer Ausführungsform kann(können) der(die) Controller 1236 für eine Vielfalt von Funktionen verwendet werden. In mindestens einer Ausführungsform kann(können) der(die) Controller 1236 mit einer von unterschiedlichen anderen Komponenten und Systemen des Fahrzeugs 1200 gekoppelt werden und kann(können) zur Steuerung des Fahrzeugs 1200, der künstlichen Intelligenz des Fahrzeugs 1200, des Infotainments für ein Fahrzeug 1200 und/oder dergleichen verwendet werden.

[0079] In mindestens einer Ausführungsform kann ein Fahrzeug 1200 eine beliebige Anzahl von SoCs 1204 umfassen. Jede der SoCs 1204 kann, ohne darauf beschränkt zu sein, zentrale Verarbeitungseinheiten („CPU(s)“) 1206, Graphikverarbeitungseinheiten („GPU(s)“) 1208, Prozessor(en) 1210, Cache-Speicher 1012, Beschleuniger 1214, Datenspeicher 1216 und/oder andere nicht veranschaulichte Komponenten und Merkmale umfassen. In mindestens einer Ausführungsform kann(können) das(die) SoC(s) 1204 zur Steuerung des Fahrzeugs 1200 in einer Vielfalt von Plattformen und Systemen verwendet werden. Beispielsweise kann(können) in mindestens einer Ausführungsform das(die) SoC(s) 1204 in einem System (z. B. dem System des Fahrzeugs 1200) mit einer High-Definition („HD“-Karte 1222 kombiniert werden, die Kartenauffrischungen und/oder -aktualisierungen über eine Netzwerkschnittstelle 1224 von einem oder mehreren Servern (in **Fig. 12C** nicht gezeigt) erhalten können.

[0080] In mindestens einer Ausführungsform kann(können) die CPU(s) 1206 einen CPU-Cluster oder CPU-Komplex (alternativ hier als ein „CCPLEX“ bezeichnet) umfassen. In mindestens einer Ausführungsform kann(können) die CPU(s) mehrere Kerne und/oder Level-2 („L2“-Caches umfassen. Beispielsweise kann(können) in mindestens einer Ausführungsform die CPU(s) 1206 acht Kerne in einer kohärenten Multiprozessor-Konfiguration umfassen. In mindestens einer Ausführungsform kann(können) die CPU(s) 1206 vier Dual-Core-Cluster umfassen, wobei jeder Cluster einen dedizierten L2-Cache (z. B. einen 2 MB L2-Cache) aufweist. In mindestens einer Ausführungsform kann(können) die CPU(s) 1206 (z. B. der CPLEX) so konfiguriert sein, dass sie simultane Clusteroperationen unterstützen, die einer beliebigen Kombination von Clustern der CPU(s) 1206 ermöglichen, zu einem gegebenen Zeitpunkt aktiv zu sein.

[0081] In mindestens einer Ausführungsform können eine oder mehrere der CPU(s) 1206 Energieverwaltungsfähigkeiten implementieren, die, ohne darauf beschränkt zu sein, ein oder mehrere der folgenden Merkmale umfassen: einzelne Hardwareblöcke können durch Clock-Gating automatisch im Leerlauf gesteuert werden, um dynamische Leistung zu sparen; jeder Kerntakt kann durch Gating gesteuert werden, wenn der Kern aufgrund der Ausführung von Wait for Interrupt („WFI“)/Wait for Event („WFE“-Anweisungen keine aktiven Anweisungen ausführt; jeder Kern kann unabhängig durch Power-Gating gesteuert werden; jeder Kerncluster kann durch Clock-Gating unabhängig gesteuert werden, wenn alle Kerne durch Clock-Gating oder Power-Gating gesteuert werden; und/oder jeder Kerncluster kann unabhängig durch Power-Gating gesteuert werden, wenn alle Kerne durch Power-Gating gesteuert werden. In mindestens einer Ausführungsform kann(können) die CPU(s) 1206 ferner einen erweiterten Algorithmus zur Verwaltung von Leistungszuständen implementieren, bei dem zulässige Leistungszustände und erwartete Aufwachzeiten spezifiziert sind, und die Hardware/der Mikrocode bestimmt den besten Leistungszustand, der für den Kern, den Cluster und den CPLEX einzugeben ist. In mindestens einer Ausführungsform können die Verarbeitungskerne vereinfachte Eingangssequenzen für den Leistungszustand in der Software unterstützen, wobei die Arbeit auf einen Mikrocode abgeladen wird.

[0082] In mindestens einer Ausführungsform kann(können) die GPU(s) 1208 eine integrierte GPU(s) umfassen, (alternativ hier als „iGPU“ bezeichnet). In mindestens einer Ausführungsform kann(können) die GPU(s) 1208 programmierbar und für parallele Arbeitslasten effizient sein. In mindestens einer Ausführungsform kann(können) die GPU(s) 1208 einen erweiterten Tensorbefehlssatz verwenden. In mindestens einer Ausführungsform

rungsform kann(können) die GPU(s) 1208 einen oder mehrere Streaming-Mikroprozessoren umfassen, wobei jeder Streaming-Mikroprozessor einen Level-1 („L1“-)Cache (z. B. einen L1-Cache mit mindestens 96KB Speicherkapazität) umfassen kann, und zwei oder mehrere der Streaming-Mikroprozessoren können einen Level-2 („L2“-)Cache (z. B. einen L2-Cache mit einer Speicherkapazität von 512 KB) gemeinsam nutzen. In mindestens einer Ausführungsform kann(können) die GPU(s) 1208 mindestens acht Streaming-Mikroprozessoren umfassen. In mindestens einer Ausführungsform kann(können) die GPU(s) 1208 eine Anwendungsprogrammierschnittstelle(n) (Application Programming Interface(s); „API(s)“) verwenden. In mindestens einer Ausführungsform kann(können) die GPU(s) 1208 eine oder mehrere parallele Computerplattformen und/oder Programmiermodelle (z. B. NVIDIA's CUDA) verwenden.

[0083] In mindestens einer Ausführungsform kann(können) die GPU(s) 1208 zur besten Leistung in Automobil- und eingebetteten Anwendungsfällen leistungsoptimiert sein. Beispielsweise könnte(n) in einer Ausführungsform die GPU(s) 1208 auf einem Fin-Feldeffekttransistor („FinFET“) hergestellt werden. In mindestens einer Ausführungsform kann jeder Streaming-Mikroprozessor eine Anzahl von gemischtpräzisen Verarbeitungskernen aufnehmen, die in mehrere Blöcke unterteilt sind. Beispielsweise, und ohne darauf beschränkt zu sein, könnten 64 PF32-Kerne und 32 PF64-Kerne in vier Verarbeitungsblöcke unterteilt werden. In mindestens einer Ausführungsform könnte jeder Verarbeitungsblock 16 FP32-Kernen, 8 FP64-Kernen, 16 INT32-Kernen, zwei NVIDIA TENSOR CORES mit gemischter Präzision für Deep-Learning-Matrix-Arithmetik, ein Level-0 („L“-)Befehls-Cache, ein Warp-Planer, eine Dispositionseinheit und/oder eine 64KB-Registerdatei zugeteilt werden. In mindestens einer Ausführungsform können Streaming-Mikroprozessoren unabhängige parallele Ganzzahl- und Gleitkomma-Datenpfade umfassen, um eine effiziente Ausführung von Arbeitslasten mit einer Mischung aus Rechen- und Adressierungsberechnungen vorzusehen. In mindestens einer Ausführungsform können Streaming-Mikroprozessoren eine unabhängige Thread-Scheduling-Funktionalität umfassen, um eine feinere Synchronisation und Zusammenarbeit zwischen parallelen Threads zu ermöglichen. In mindestens einer Ausführungsform können die Streaming-Mikroprozessoren einen kombinierten L1-Daten-Cache und eine gemeinsam genutzte Speichereinheit umfassen, um die Leistung zu verbessern, während die Programmierung vereinfacht wird.

[0084] In mindestens einer Ausführungsform kann(können) eine oder mehrere der GPU(s) 1208 einen Speicher mit hoher Bandbreite (High Bandwidth Memory; „HBM“) und/oder ein 16 GB HBM2-Speicherteilsystem umfassen, um in einigen Beispielen eine Spitzenspeicherbandbreite von etwa 900 GB/Sekunde bereitzustellen. In mindestens einer Ausführungsform kann zusätzlich zu oder alternativ von dem HBM-Speicher ein synchroner Graphik-Random-Access-Speicher („SGRAM“) verwendet werden, wie beispielsweise ein Typ 5 synchroner Graphik-Doppel-Datenraten-Speicher (Graphics Double Data Rate Type Five Synchronous Random-Access Memory; „GDDR5“).

[0085] In mindestens einer Ausführungsform kann(können) die GPU(s) 1208 eine vereinheitlichte Speichertechnologie umfassen. In mindestens einer Ausführungsform kann die Unterstützung von Adressübersetzungsdiensten (Address Translation Services; „ATS“) verwendet werden, um der(den) GPU(s) 1208 zu ermöglichen, direkt auf Seitentabellen der CPU(s) 1206 zugreifen. In mindestens einer Ausführungsform kann, wenn die Speicherverwaltungseinheit (Memory Management Unit; „MMU“) der GPU(s) 1208 einem Fehlzugriff unterliegt, eine Adressübersetzungsanforderung an die CPU(s) 1206 gesendet werden. Als Antwort darauf kann(können) in mindestens einer Ausführungsform die CPU(s) 1206 in ihren Seitentabellen nach der virtuellen-zu-physikalischen Adresszuordnung für die Adresse suchen und die Übersetzung zurück an die GPU(s) 1208 übertragen. In mindestens einer Ausführungsform kann die vereinheitlichte Speichertechnologie einen einzigen einheitlichen virtuellen Adressraum für Speicher von sowohl der CPU(s) 1206 als auch der GPU(s) 1208 ermöglichen, wodurch die Programmierung der GPU(s) 1208 und die Portierung von Anwendungen auf die GPU(s) 1208 vereinfacht wird.

[0086] In mindestens einer Ausführungsform kann(können) die GPU(s) 1208 eine beliebige Anzahl von Zugriffszählern umfassen, welche die Häufigkeit eines Zugriffs der GPU(s) 1208 auf den Speicher anderer Prozessoren nachverfolgen können. In mindestens einer Ausführungsform kann(können) der(die) Zugriffszähler beitragen, dass Speicherseiten in den physischen Speicher des Prozessors verschoben werden, der am häufigsten auf Seiten zugreift, um dadurch die Effizienz für zwischen Prozessoren gemeinsam genutzte Speicherbereiche zu verbessern.

[0087] In mindestens einer Ausführungsform kann ein oder mehrere der SoCs 1204 eine beliebige Anzahl von Caches 1012 umfassen, einschließlich derjenigen der hier beschriebenen. Beispielsweise kann(können) in mindestens einer Ausführungsform der(die) Cache(s) 1012 einen Level-3 („L3“)Cache umfassen, der sowohl für die CPU(s) 1206 als auch die GPU(s) 1208 verfügbar ist (z. B. der sowohl mit der(den) CPU(s)

1206 und der(den) GPU(s) 1208 verbinden ist). In mindestens einer Ausführungsform kann(können) der(die) Cache(s) 1012 einen Write-Back-Cache umfassen, der Zustände der Leitungen nachverfolgen kann, wie beispielsweise durch Verwendung eines Cache-Kohärenzprotokolls (z. B. MEI, MESI, MSI usw.). In mindestens einer Ausführungsform kann ein L3-Cache abhängig von der Ausführungsform 4 MB oder mehr umfassen, obwohl kleinere Cachegrößen verwendet werden können.

[0088] In mindestens einer Ausführungsform können ein oder mehrere der SoCs 1204 einen oder mehrere Beschleuniger 1214 umfassen (z. B. Hardwarebeschleuniger, Softwarebeschleuniger oder eine Kombination davon). In mindestens einer Ausführungsform kann(können) der(die) SoC(s) 1204 einen Hardwarebeschleunigungscluster umfassen, der optimierte Hardwarebeschleuniger und/oder einen großen On-Chip-Speicher umfassen kann. In mindestens einer Ausführungsform kann ein großer On-Chip-Speicher (z. B. 4 MB SRAM) einem Hardwarebeschleunigungscluster ermöglichen, neuronale Netzwerke und andere Berechnungen zu beschleunigen. In mindestens einer Ausführungsform kann der Hardwarebeschleunigungscluster verwendet werden, um die GPU(s) 1208 zu ergänzen und einige der Aufgaben der GPU(s) 1208 auszulagern (z. B., um mehr Zyklen der GPU(s) 1208 zum Durchführen anderer Aufgaben freizugeben). In mindestens einer Ausführungsform könnte(n) der(die) Beschleuniger 1214 für gezielte Arbeitslasten (z. B. Wahrnehmung, faltende neuronale Netzwerke (Convolutional Neural Networks; „CNNs“), rekurrente neuronale Netzwerke (Recurrent Neural Networks; „RNNs“), usw.) verwendet werden, die stabil genug sind, um für eine Beschleunigung zugänglich zu sein. In mindestens einer Ausführungsform kann ein CNN Regionen-basierte oder regionale faltende neuronale Netzwerke (Regional Convolutional Neural Networks; „RCNNs“) und Fast RCNNs (z. B. wie zur Objekterkennung verwendet) umfassen.

[0089] In mindestens einer Ausführungsform kann(können) der(die) Beschleuniger 1214 (z. B. der Hardwarebeschleunigungscluster) einen Beschleuniger für tiefes Lernen (Deep Learning Accelerator(s); „DLA(s)“) umfassen. DLA(s) kann(können), ohne darauf beschränkt zu sein, eine oder mehrere Tensor-Verarbeitungseinheiten (Tensor Processing Units; „TPUs“) umfassen, die konfiguriert sein können, um zusätzliche zehn Billionen Operationen pro Sekunde für Deep-Learning-Anwendungen und Inferenzieren bereitzustellen. Die TPUs können Beschleuniger sein, die für die Durchführung von Bildverarbeitungsfunktionen konfiguriert und optimiert sind (z. B. für CNNs, RCNNs usw.). Der(die) DLA(s) kann(können) ferner für einen bestimmten Satz von neuronalen Netzwerktypen und Gleitkommaoperationen sowie zum Inferenzieren optimiert sein. In mindestens einer Ausführungsform kann die Ausgestaltung der DLA(s) mehr Leistung pro Millimeter bereitstellen als ein typischer Allzweck-Graphikprozessor und übersteigt typischerweise bei weitem die Leistung einer CPU. In mindestens einer Ausführungsform kann(können) TPU(s) mehrere Funktionen ausführen, einschließlich einer Einzelinstanz-Faltungsfunktion, die beispielsweise, und ohne darauf beschränkt zu sein, sowohl Merkmale und Gewichtungen bei den Datentypen INT8, INT16 und FP16 sowie auch Postprozessorfunktionen unterstützt. In mindestens einer Ausführungsform kann(können) DLA(s) neuronale Netzwerke, insbesondere CNNs, schnell und effizient auf verarbeiteten oder unverarbeiteten Daten für beliebige einer Vielfalt von Funktionen ausführen, einschließlich beispielsweise und ohne darauf beschränkt zu sein: ein CNN zur Objektklassifizierung und -erkennung unter Verwendung von Daten von Kamerasensoren; ein CNN zur Abstandsschätzung unter Verwendung von Daten von Kamerasensoren; ein CNN zur Fahrzeugerkennung und -identifikation und -erkennung unter Verwendung von Daten aus Mikrofonen 1296; ein CNN zur Gesichtserkennung und Fahrzeughalteridentifikation unter Verwendung von Daten aus Kamerasensoren; und/oder ein CNN für Sicherheit und/oder sicherheitsbezogene Ereignisse.

[0090] In mindestens einer Ausführungsform kann(können) DLA(s) jede beliebige Funktion der GPU(s) 1208 ausführen und durch Verwenden eines Inferenzbeschleunigers kann ein Designer beispielsweise für jede Funktion entweder die DLA(s) oder die GPU(s) 1208 ansteuern. Beispielsweise kann sich in mindestens einer Ausführungsform der Designer auf die Verarbeitung von CNNs und Gleitkommaoperationen auf dem (den) DLA(s) konzentrieren und andere Funktionen dem(den) GPU(s) 1208 und/oder einem anderem(anderen) Beschleuniger(n) 1214 überlassen.

[0091] In mindestens einer Ausführungsform kann(können) der(die) Beschleuniger 1214 einen programmierbaren Visionsbeschleuniger (Programmable Vision Accelerator; „PVA“) umfassen, der hier alternativ als ein Computer-Vision-Beschleuniger bezeichnet werden kann. In mindestens einer Ausführungsform kann(können) ein PVA(s) ausgestaltet und konfiguriert sein, um Bildverarbeitungsalgorithmen für Anwendungen der fortgeschrittene Fahrerassistenzsysteme (Advanced Driver Assistance Systems; „ADAS“), des autonomen Fahrens, der Augmented Reality („AR“) Anwendungen und/oder Virtual Reality („VR“) Anwendungen zu beschleunigen. In mindestens einer Ausführungsform kann ein PVA ein Gleichgewicht zwischen Leistung und Flexibilität bereitstellen. Beispielsweise kann in mindestens einer Ausführungsform jeder PVA, beispielsweise und ohne darauf beschränkt zu sein, eine beliebige Anzahl von Computer mit reduziertem Befehlssatz-

kernen (Reduced Instruction Set Computer cores; „RISC“ cores), Direktzugriffsspeicher (Direct Memory Access; „DMA“) und/oder eine beliebige Anzahl von Vektorprozessoren umfassen.

[0092] In mindestens einer Ausführungsform können RISC-Kerne mit Bildsensoren (z. B. Bildsensoren von beliebigen der hier beschriebenen Kameras), Bildsignalprozessor(en) und/oder dergleichen wechselwirken. In mindestens einer Ausführungsform können RISC-Kerne eine beliebige Menge an Speicher umfassen. In mindestens einer Ausführungsform können RISC-Kerne eine beliebige Anzahl von Protokollen abhängig von der Ausführungsform verwenden. In mindestens einer Ausführungsform können RISC-Kerne ein Echtzeitbetriebssystem (Real-time Operating System; „RTOS“) ausführen. In mindestens einer Ausführungsform können RISC-Kerne unter Verwendung einer oder mehrerer integrierter Schaltungsvorrichtungen, anwendungsspezifischen integrierter Schaltungen (Application Specific Integrated Circuits; „ASICs“) und/oder Speichervorrichtungen implementiert werden. Beispielsweise können in mindestens einer Ausführungsform die RISC-Kerne einen Befehls-Cache und/oder einen eng gekoppelten RAM umfassen.

[0093] In mindestens einer Ausführungsform kann der DMA Komponenten des PVA ermöglichen, unabhängig von CPU(s) 1206 auf den Systemspeicher zuzugreifen. In mindestens einer Ausführungsform kann der DMA eine beliebige Anzahl von Merkmalen unterstützen, die verwendet werden, um eine Optimierung einem PVA bereitzustellen, einschließlich, jedoch nicht beschränkt auf, der Unterstützung multidimensionaler Adressierung und/oder zirkulärer Adressierung. In mindestens einer Ausführungsform kann der DMA bis zu sechs oder mehr Dimensionen der Adressierung unterstützen, welche, ohne darauf beschränkt zu sein, Blockbreite, Blockhöhe, Blocktiefe, horizontale Blockabstufung, vertikale Blockabstufung und/oder Tiefenabstufung umfassen können.

[0094] In mindestens einer Ausführungsform können die Vektorprozessoren programmierbare Prozessoren sein, die ausgestaltet sein können, um effizient und flexibel die Programmierung für Computer-Vision-Algorithmen durchzuführen und Signalverarbeitungsfähigkeiten bereitzustellen. In mindestens einer Ausführungsform kann der PVA einen PVA-Kern und zwei Vektorverarbeitungs-Teilsystem-Partitionen umfassen. In mindestens einer Ausführungsform kann der PVA ein Prozessorteilsystem, ein(mehrere) DMA-Engine(s) (z. B. zwei DMA-Engines) und/oder andere Peripheriegeräte umfassen. In mindestens einer Ausführungsform kann das Vektorverarbeitungs-Teilsystem als primäre Verarbeitungs-Engine des PVA arbeiten und kann eine Vektorverarbeitungseinheit (Vector Processing Unit; „VPU“), einen Befehls-Cache und/oder einen Vektorspeicher (z. B. Vector Memory; „VMEM“) umfassen. In mindestens einer Ausführungsform kann die VPU einen digitalen Signalprozessor, wie beispielsweise einen digitalen Signalprozessor mit mehreren Daten (Single Instruction, Multiple Data; „SIMD“) und einen digitalen Signalprozessor mit sehr langem Befehlswort (Very Long Instruction Word; „VLIW“) umfassen. In mindestens einer Ausführungsform kann die Kombination von SIMD und VLIW den Durchsatz und die Geschwindigkeit erhöhen.

[0095] In mindestens einer Ausführungsform kann jeder Vektorprozessoren einen Befehls-Cache umfassen und mit einem dedizierten Speicher gekoppelt sein. Als Ergebnis kann in mindestens einer Ausführungsform jeder Vektorprozessor konfiguriert sein, um unabhängig von anderen Vektorprozessoren ausführen zu können. In mindestens einer Ausführungsform können Vektorprozessoren, die in einem bestimmten PVA umfasst sind, konfiguriert sein, um Datenparallelität zu benutzen. Beispielsweise können in mindestens einer Ausführungsform mehrere, in einem einzigen PVA enthaltene Vektorprozessoren, den gleichen Computer-Vision-Algorithmus jedoch auf unterschiedlichen Bereichen eines Bildes ausführen. In mindestens einer Ausführungsform können die in einem bestimmten PVA enthaltenen Vektorprozessoren gleichzeitig unterschiedliche Computer-Vision-Algorithmen auf einem gleichen Bild ausführen oder sogar unterschiedliche Algorithmen auf sequenziellen Bildern oder Teilen eines Bildes ausführen. In mindestens einer Ausführungsform können unter anderem beliebig viele PVAs in dem Hardwarebeschleunigungscluster und beliebig viele Vektorprozessoren in jedem PVA enthalten sein. In mindestens einer Ausführungsform kann der PVA zusätzlich einen Speicher für einen fehlerkorrigierenden Code (Error Correcting Code; „ECC“) umfassen, um die Gesamtsystemsicherheit zu erhöhen.

[0096] In mindestens einer Ausführungsform kann(können) der(die) Beschleuniger (z. B. Hardwarebeschleunigungs-Cluster) ein On-Chip-Computer-Vision-Netzwerk und einen statischen Direktzugriffsspeicher (Static Random-Access Memory; „SRAM“) umfassen, um ein SRAM mit hoher Bandbreite und niedriger Latenz für den(die) Beschleuniger 1214 bereitzustellen. In mindestens einer Ausführungsform kann der On-Chip-Speicher mindestens 4MB SRAM umfassen, der beispielsweise und ohne Beschränkung aus acht feldkonfigurierbaren Speicherblöcken besteht, die sowohl für einen PVA als auch für einen DLA zugänglich sind. In mindestens einer Ausführungsform kann jedes Speicherblockpaar eine erweiterte periphere Busschnittstelle (Advanced Peripheral Bus interface; „APB“), Konfigurationsschaltungen, einen Controller und einen Multiplexer

umfassen. In mindestens einer Ausführungsform kann eine beliebige Art von Speicher verwendet werden. In mindestens einer Ausführungsform kann ein PVA und DLA über ein Rückgrat bzw. Backbone auf den Speicher zugreifen, das einem PVA und DLA einen Hochgeschwindigkeitszugriff auf den Speicher bereitstellt. In mindestens einer Ausführungsform kann das Backbone ein On-Chip Computer-Vision-Netzwerk umfassen, welches einen PVA und DLA mit dem Speicher (z. B. unter Verwendung der APB) verbindet.

[0097] In mindestens einer Ausführungsform kann das On-Chip-Computer-Vision-Netzwerk eine Schnittstelle umfassen, die vor der Übertragung irgendwelcher Steuersignale/Adressen/Daten bestimmt, dass sowohl PVA als auch DLA bereite und gültige Signale bereitstellen. In mindestens einer Ausführungsform kann eine Schnittstelle getrennte Phasen und getrennte Kanäle für die Übertragung von Steuersignalen/Adressen/Daten sowie eine Kommunikation der Burst-Art für einen kontinuierlichen Datentransfer vorsehen. In mindestens einer Ausführungsform kann eine Schnittstelle den Standards der International Organization for Standardization („ISO“) 26262 oder den Standards der International Electrotechnical Commission („IEC“) 61508 entsprechen, obwohl auch andere Standards und Protokolle verwendet werden können.

[0098] In mindestens einer Ausführungsform kann eine oder mehrere der SoC(s) 1204 einen Echtzeit-Strahlverfolgung-Hardwarebeschleuniger umfassen. In mindestens einer Ausführungsform kann ein Echtzeit-Strahlverfolgung-Hardwarebeschleuniger verwendet werden, um schnell und effizient die Positionen und die Ausmaße von Objekten (z. B. innerhalb eines Weltmodells) zu bestimmen, um Echtzeit-Visualisierungssimulationen zur RADAR-Signalinterpretation, zur Schallausbreitungssynthese und/oder -analyse, zur Simulation von SONAR-Systemen, zur Simulation einer allgemeinen Wellenausbreitung, zum Vergleich mit LIDAR-Daten zum Zwecke der Lokalisierung und/oder andere Funktionen und/oder für andere Anwendungen zu erzeugen.

[0099] In mindestens einer Ausführungsform kann(können) ein(die) Beschleuniger 1214 (z. B. Hardwarebeschleunigungs-Cluster(s)) ein breites Anwendungsspektrum für autonomes Fahren aufweisen. In mindestens einer Ausführungsform kann ein PVA ein programmierbarer Vision-Beschleuniger sein, der für Schlüsselverarbeitungsstufen in ADAS und autonomen Fahrzeugen verwendet werden kann. In mindestens einer Ausführungsform sind die Fähigkeiten eines PVA eine gute Übereinstimmung für algorithmische Domäne, die eine vorhersagbare Verarbeitung bei geringer Leistungsaufnahme und geringer Latenzzeit benötigen. Mit anderen Worten kann der PVA bei halbdichtem oder dichtem regulärem Rechnen selbst bei kleinen Datensätzen gut arbeiten, die vorhersagbare Laufzeiten mit geringer Latenzzeit und geringer Leistung benötigen. In mindestens einer Ausführungsform sind in autonomen Fahrzeuge, wie beispielsweise ein Fahrzeug 1200, PVAs ausgestaltet, um klassische Computer-Vision-Algorithmen auszuführen, da sie bei der Objekterkennung effizient sind und mit ganzzahliger Mathematik arbeiten.

[0100] Zum Beispiel wird gemäß mindestens einer Ausführungsform der Technologie der PVA verwendet, um Computer-Stereo-Vision durchzuführen. In mindestens einer Ausführungsform kann ein halbglobaler Abgleich-basierter Algorithmus verwendet werden, obwohl dies nicht bestimmt ist, einschränkend zu sein. In mindestens einer Ausführungsform erfordern Anwendungen für das autonome Fahren der Stufe 3-5 eine Bewegungsschätzung/on-the-fly Stereoabgleich (z. B. Struktur aus Bewegung, Fußgängererkennung, Spurerkennung usw.). In mindestens einer Ausführungsform kann ein PVA eine Computer-Stereo-Vision-Funktion an Eingaben aus zwei Monokularkameras ausführen.

[0101] In mindestens einer Ausführungsform kann der PVA zur Durchführung von dichtem optischem Fluss verwendet werden. Beispielsweise könnte in mindestens einer Ausführungsform der PVA RADAR-Rohdaten (z. B. unter Verwendung einer 4D Fast-Fourier-Transformation) verarbeiten, um verarbeitete RADAR-Daten bereitzustellen. In mindestens einer Ausführungsform wird ein PVA zur Flugzeit-Tiefenverarbeitung verwendet, indem Flugzeit-Rohdaten verarbeitet werden, um beispielsweise verarbeitete Flugzeit-Daten bereitzustellen.

[0102] In mindestens einer Ausführungsform kann ein DLA verwendet werden, um eine beliebige Art von Netzwerk zu betreiben, um die Kontrolle und Fahrsicherheit zu erhöhen, einschließlich beispielsweise, und ohne darauf beschränkt zu sein, eines neuronalen Netzwerks, das ein Maß an Vertrauen für jede Objekterkennung ausgibt. In mindestens einer Ausführungsform kann ein derartiger Vertrauenswert als eine Wahrscheinlichkeit oder als Bereitstellung einer relativen „Gewichtung“ jeder Erkennung im Vergleich mit anderen Erkennungen interpretiert werden. In mindestens einer Ausführungsform ermöglicht Vertrauen einem System, weitere Entscheidungen hinsichtlich dessen zu treffen, welche Erkennungen als wahre positive Erkennungen und nicht als falsch positive Erkennungen betrachtet werden sollten. Beispielsweise kann in mindestens einer Ausführungsform ein System einen Schwellenwert für das Vertrauen festlegen und nur die den

Schwellenwert übersteigenden Erkennungen als wahre positive Erkennungen betrachten. In einem automatischen Notbremssystem (Automatic Emergency Braking System; „AEB“-System) würden falsch positive Erkennungen dazu führen, dass ein Fahrzeug automatisch eine Notbremsung durchführt, was offensichtlich unerwünscht ist. In mindestens einer Ausführungsform können sehr vertrauenswürdige Erkennungen als Auslöser für ein AEB in Betracht kommen. In mindestens einer Ausführungsform kann der DLA ein neuronales Netzwerk zum Regressieren des Vertrauenswerts betreiben. In mindestens einer Ausführungsform kann das neuronale Netzwerk mindestens eine Teilmenge von Parametern als seine Eingabe verwenden, wie beispielsweise Abmessungen eines Begrenzungskastens, einer Ground-Ebenen-Schätzung (z. B. von einem anderen Teilsystem), eine Ausgabe von Sensoren der Trägheitsmesseinheit (IMU) 1266, die mit der Orientierung des Fahrzeugs 1200 korreliert, eine Entfernung, 3D-Ortsschätzungen des Objekts, die unter anderem von einem neuronalen Netzwerk und/oder von anderen Sensoren (z. B. LIDAR-Sensor(en) 1264 oder RADAR-Sensor(en) 1260) stammen.

[0103] In mindestens einer Ausführungsform kann(können) ein oder mehrere der(die) SoC(s) 1204 (einen) Datenspeicher 1216 (z. B. Speicher) umfassen. In mindestens einer Ausführungsform kann(können) der(die) Datenspeicher 1216 ein On-Chip-Speicher des(der) SoC(s) 1204 sein, der neuronale Netzwerke speichern kann, die auf der(den) GPU(s) 1208 und/oder einem DLA ausgeführt werden sollen. In mindestens einer Ausführungsform kann(können) der(die) Datenspeicher 1216 in der Kapazität groß genug sein, um mehrere Instanzen neuronaler Netzwerke für Redundanz und Sicherheit zu speichern. In mindestens einer Ausführungsform kann(können) der(die) Datenspeicher 1216 einen(mehrere) L2- oder L3-Cache(s) 1012 umfassen.

[0104] In mindestens einer Ausführungsform kann(können) ein oder mehrere SoC(s) 1204 eine beliebige Anzahl von Prozessor(en) 1210 (z. B. eingebettete Prozessoren) umfassen. Der(die) Prozessor(en) 1210 können einen Boot- und Leistungsmanagement-Prozessor umfassen, der ein dedizierter Prozessor und ein Teilsystem sein kann, um Boot-Leistungs- und Verwaltungs-Funktionen und die damit zugeordnete Sicherheitsdurchsetzung zu handhaben. In mindestens einer Ausführungsform kann der Boot- und ein Leistungsverwaltungs-Prozessor ein Teil einer Boot-Sequenz des(der) SoC(s) 1204 sein und zur Laufzeitleistungs-Verwaltungsdienste bereitstellen. In mindestens einer Ausführungsform kann ein Boot- und Leistungsverwaltungs-Prozessor Takt- und Spannungsprogrammierung, Unterstützung bei Systemübergängen mit niedrigem Leistungsverbrauch, Verwaltung von Thermik- und Temperatursensoren von SoC(s) 1204 und/oder Verwaltung der Leistungszustände von SoC(s) 1204 bereitstellen. In mindestens einer Ausführungsform kann jeder Temperatursensor als ein Ringoszillator implementiert sein, dessen Ausgangsfrequenz proportional zu der Temperatur ist, und das(die) SoC(s) 1204 kann(können) Ringoszillatoren verwenden, um Temperaturen der CPU(s) 1206, der GPU(s) 1208 und/oder des(der) Beschleuniger(s) 1214 zu erfassen. In mindestens einer Ausführungsform kann, wenn bestimmt wird, dass die Temperaturen einen Schwellenwert überschreiten, dann ein Boot- und Leistungsverwaltungs-Prozessor in eine Temperaturfehleroutine eintreten und das(die) SoC(s) 1204 in einen niedrigeren Leistungszustand versetzen und/oder ein Fahrzeug 1200 in einen Chauffeur-zu-sicheren-Stoppmodus versetzen (z. B. ein Fahrzeug 1200 zu einem sicheren Halt bringen).

[0105] In mindestens einer Ausführungsform kann(können) der(die) Prozessor(en) 1210 ferner einen Satz von eingebetteten Prozessoren umfassen, die als eine Audioverarbeitungs-Engine dienen können, die ein Audioteilsystem sein kann, das eine vollständige Hardwareunterstützung für Mehrkanal-Audio über mehrere Schnittstellen und einen breite und flexiblen Bereich von Audio-E/A-Schnittstellen ermöglicht. In mindestens einer Ausführungsform ist die Audioverarbeitungs-Engine ein dedizierter Prozessorkern mit einem Digitalsignalprozessor mit dediziertem RAM.

[0106] In mindestens einer Ausführungsform kann(können) der(die) Prozessor(en) 1210 ferner eine Always-On-Processor-Engine umfassen, die notwendige Hardware-Merkmale bereitstellen kann, um die Verwaltung von Sensoren mit niedriger Leistung und Aufwachenwendungsfälle zu unterstützen. In mindestens einer Ausführungsform kann die Always-On-Processor-Engine einen Prozessorkern, ein eng gekoppeltes RAM, unterstützende Peripheriegeräte (z. B. Zeitgeber und Interrupt-Controller), unterschiedliche E/A-Controller-Peripheriegeräte und Routinglogik umfassen.

[0107] In mindestens einer Ausführungsform kann(können) der(die) Prozessor(en) 1210 ferner eine Sicherheitscluster-Engine umfassen, die, ohne darauf beschränkt zu sein, ein dediziertes Prozessorteilsystem umfasst, um das Sicherheitsmanagement für Automobilanwendungen zu handhaben. In mindestens einer Ausführungsform kann eine Sicherheitscluster-Engine, ohne darauf beschränkt zu sein, zwei oder mehr Prozessorkerne, ein eng gekoppeltes RAM, Unterstützung von Peripheriegeräten (z. B. Zeitgeber, Interrupt-Controller usw.) und/oder RoutingLogik umfassen. In einem Sicherheitsmodus können zwei oder mehr Kerne, in mindestens einer Ausführungsform, in einem Lockstep-Modus arbeiten und als ein einzelner Kern

mit Vergleichslogik fungieren, um jegliche Unterschiede zwischen ihren Operationen zu erkennen. In mindestens einer Ausführungsform kann(können) der(die) Prozessor(en) 1210 ferner, ohne darauf beschränkt zu sein, eine Echtzeit-Kamera-Engine umfassen, die ein dediziertes Prozessor-Teilsystem für die Handhabung der Echtzeit-Kamera-Verwaltung umfassen kann. In mindestens einer Ausführungsform kann(können) der(die) Prozessor(en) 1210 ferner einen Signalprozessor mit hohem Dynamikbereich umfassen, der, ohne darauf beschränkt zu sein, einen Bildsignalprozessor umfassen kann, der eine Hardware-Engine ist, die Teil der Kameraverarbeitung-Pipeline ist.

[0108] In mindestens einer Ausführungsform kann(können) der(die) Prozessor(en) 1210 einen Videobild-Compositor umfassen, der ein Verarbeitungsblock (z. B. auf einem Mikroprozessor implementiert) sein kann, der Funktionen zur Videonachbearbeitung implementiert, die von einer Videowiedergabeanwendung benötigt werden, um das endgültige Bild für ein Abspielerfenster zu erzeugen. In mindestens einer Ausführungsform kann ein Videobild-Compositor eine Objektivverzerrungskorrektur an einer Weitwinkelkamera(s) 1270, einer Rundumsichtkamera(s) 1274 und/oder an einem kabineninternen Kamerasensor(en) durchführen. In mindestens einer Ausführungsform wird(werden) der(die) kabineninterne(n) Überwachungskamerasensor(en) bevorzugt von einem neuronalen Netzwerk überwacht, das auf einer anderen Instanz des SoC 1204 läuft und konfiguriert ist, Ereignisse in der Kabine zu identifizieren und entsprechend zu reagieren. In mindestens einer Ausführungsform kann ein kabineninternes System, ohne darauf beschränkt zu sein, Lippenlesen durchführen, um einen Mobilfunk zu aktivieren und einen Anruf zu tätigen, E-Mails zu diktieren, das Ziel eines Fahrzeugs zu ändern, ein Infotainmentsystem dessen Einstellungen eines Fahrzeugs zu aktivieren oder zu ändern, oder sprachaktiviertes Internetsurfen anzubieten. In mindestens einer Ausführungsform sind bestimmte Funktionen dem Fahrer verfügbar, wenn ein Fahrzeug in einem autonomen Modus arbeitet, und sind andernfalls deaktiviert.

[0109] In mindestens einer Ausführungsform kann ein Videobild-Compositor eine verbesserte zeitliche Rauschunterdrückung für sowohl räumliche als auch zeitliche Rauschunterdrückung umfassen. Wenn beispielsweise in mindestens einer Ausführungsform eine Bewegung in einem Video stattfindet, gewichtet die Rauschunterdrückung räumliche Information dementsprechend und verringert das Gewicht der von benachbarten Einzelbildern bereitgestellten Information. In mindestens einer Ausführungsform kann, wenn ein Bild oder ein Teil eines Bildes keine Bewegung umfasst, die von dem Videobild-Compositor durchgeführte zeitliche Rauschunterdrückung Information aus dem vorherigen Bild verwenden, um das Rauschen im aktuellen Bild zu verringern.

[0110] In mindestens einer Ausführungsform kann der Videobild-Compositor ebenfalls konfiguriert sein, um eine Stereogleichrichtung an eingegebenen Stereolinseneinzelbildern durchzuführen. In mindestens einer Ausführungsform kann der Videobild-Compositor ferner für eine Zusammensetzung der Benutzeroberfläche verwendet werden, wenn ein Betriebssystem des Desktop verwendet wird und die GPU(s) 1208 nicht erforderlich ist(sind), um kontinuierlich neue Oberflächen zu rendern. In mindestens einer Ausführungsform kann, wenn die GPU(s) 1208 eingeschaltet und aktiv 3D-Rendering durchführt(durchführen), der Videobild-Compositor verwendet werden, um die GPU(s) 1208 zu entlasten, um die Leistung und Reaktionsfähigkeit zu verbessern.

[0111] In mindestens einer Ausführungsform kann(können) ein oder mehrere SoC(s) 1204 ferner eine serielle MIPI (Mobile Industry Processor Interface; „MIPI“-Kameraschnittstelle zum Empfangen von Video und Eingaben von Kameras, eine Hochgeschwindigkeitsschnittstelle und/oder einen Videoeingangsbereich umfassen, der für Kamera- und zugehörige Pixeleingabefunktionen verwendet werden kann. In mindestens einer Ausführungsform können ein oder mehrere SoCs 1204 ferner einen Eingabe-/Ausgabe-Controller umfassen, der(die) durch Software gesteuert werden kann(können) und zum Empfangen von E/A-Signalen verwendet werden kann(können), die nicht an eine spezifische Rolle gebunden sind.

[0112] In mindestens einer Ausführungsform kann(können) das(die) SoC(s) 1204 ferner einen breiten Bereich von Peripherieschnittstellen umfassen, um die Kommunikation mit Peripheriegeräten, Audio-Codierern/Decodierern („Codecs“), einer Leistungsverwaltung und/oder anderen Vorrichtungen zu ermöglichen. Das(die) SoC(s) 1204 kann(können) verwendet werden, um Daten von Kameras (z. B. über Gigabit Multimedia Serial Link und Ethernet verbunden), Sensoren (z. B. LIDAR-Sensor(en) 1264, RADAR-Sensor(en) 1260 usw., die über Ethernet verbunden sein können), Daten von dem Bus 1202 (z. B. Fahrzeuggeschwindigkeit 1200, Lenkradposition usw.), Daten von GNSS-Sensor(en) 1258 (z. B. über Ethernet oder CAN-Bus verbunden) zu verarbeiten. In mindestens einer Ausführungsform kann(können) ein oder mehrere SoC(s) 1204 ferner dedizierte Hochleistungs-Massenspeicher-Controller umfassen, die ihre eigenen DMA-Engines umfassen können und die verwendet werden können, um die CPU(s) 1206 von Routinedatenverwaltungsaufgaben zu befreien.

[0113] In mindestens einer Ausführungsform kann(können) ein oder mehrere SoC(s) 1204 eine Ende-zu-Ende-Plattform mit einer flexiblen Architektur sein, die sich über die Automatisierungsstufen 3-5 erstreckt, um dadurch eine umfassende funktionale Sicherheitsarchitektur bereitzustellen, die Computer-Vision- und ADAS-Techniken für Diversität und Redundanz nutzt und effizient einsetzt, sowie eine Plattform für einen flexiblen, zuverlässigen Treiber-Software-Stapel zusammen mit Werkzeugen des tiefen Lernens bereitstellt. In mindestens einer Ausführungsform kann(können) das(die) SoC(s) 1204 schneller, zuverlässiger, und sogar energieeffizienter und platzsparender als herkömmliche Systeme sein. Beispielsweise kann(können) in mindestens einer Ausführungsform der(die) Beschleuniger 1214 in Kombination mit der(den) CPU(s) 1206, der (den) GPU(s) 1208 und dem(den) Datenspeicher(n) 1216 eine schnelle, effiziente Plattform für autonome Fahrzeuge der Stufe 3-5 bieten.

[0114] In mindestens einer Ausführungsform können Computer-Vision-Algorithmen auf CPUs ausgeführt werden, die mit einer hochrangigen Programmiersprache, wie beispielsweise der Programmiersprache C, konfiguriert sein können, um eine breite Vielfalt von Verarbeitungsalgorithmen mit einer weiten Vielfalt von visuellen Daten auszuführen. Jedoch sind CPUs in mindestens einer Ausführungsform häufig nicht in der Lage, die Leistungsanforderungen vieler Bildverarbeitungsanwendungen zu erfüllen, wie beispielsweise jene bezogen auf Ausführungszeit und Leistungsverbrauch. In mindestens einer Ausführungsform sind viele CPUs nicht in der Lage, komplexe Objekterkennungsalgorithmen in Echtzeit auszuführen, die für ADAS-Anwendungen im Fahrzeug und für praktische autonome Fahrzeuge der Stufe 3-5 verwendet werden.

[0115] Hier beschriebene Ausführungsformen ermöglichen, dass mehrere neuronale Netzwerke gleichzeitig und/oder sequenziell verwendet und die Ergebnisse zusammen kombiniert werden können, um eine autonome Fahrfunktionalität der Stufe 3-5 zu ermöglichen. Beispielsweise kann in mindestens einer Ausführungsform ein CNN, das auf dem DLA oder einer diskreten GPU (z. B. der GPU(s) 1020) ausgeführt wird, eine Text- und Worterkennung umfassen, die einem Supercomputer ermöglichen, Verkehrszeichen zu lesen und zu verstehen, einschließlich Zeichen, für die das neuronale Netzwerk nicht speziell trainiert wurde. In mindestens einer Ausführungsform kann der DLA ferner ein neuronales Netzwerk umfassen, das in der Lage ist, ein Zeichen zu identifizieren, zu interpretieren und semantisch zu verstehen und dieses semantische Verständnis an die auf einem CPU-Komplex laufenden Wegplanungsmodule weiterzugeben.

[0116] In mindestens einer Ausführungsform können mehrere neuronale Netzwerke gleichzeitig ausgeführt werden, wie es für das Fahren auf Stufe 3, 4 oder 5 erforderlich ist. Beispielsweise kann in mindestens einer Ausführungsform ein Warnzeichen bestehend aus „Vorsicht: Blinklichter zeigen eisige Zustände an“ zusammen mit einem elektrischen Licht von mehreren neuronalen Netzwerken unabhängig oder gemeinsam interpretiert werden. In mindestens einer Ausführungsform kann ein Zeichen selbst durch ein erstes eingesetztes neuronales Netzwerk (z. B. ein neuronales Netzwerk, das trainiert wurde) als ein Verkehrszeichen identifiziert werden, der Text „Blinklichter zeigen eisige Zustände an“ durch ein zweites eingesetztes neuronales Netzwerk interpretiert werden, das die Wegplanungssoftware des Fahrzeugs (vorzugsweise auf dem CPU-Komplex) darüber informiert, dass bei Erkennung von blinkenden Lichtern eisige Bedingungen existieren. In mindestens einer Ausführungsform kann ein blinkende Licht identifiziert werden, indem ein drittes eingesetztes neuronales Netzwerk über mehrere Einzelbilder betrieben wird, das die Wegplanungssoftware des Fahrzeugs über das Vorhandensein (oder Fehlen) von blinkenden Lichtern informiert. In mindestens einer Ausführungsform können alle drei neuronalen Netzwerke z. B. innerhalb des DLA und/oder auf der(den) GPU(s) 1208 gleichzeitig laufen.

[0117] In mindestens einer Ausführungsform kann ein CNN zur Gesichtserkennung und Fahrzeughalteridentifikation Daten von Kamerasensoren verwenden, um die Anwesenheit eines autorisierten Fahrers und/oder eines Eigentümers des Fahrzeugs 1200 zu identifizieren. In mindestens einer Ausführungsform kann die Always-On-Sensor-Verarbeitungs-Engine verwendet werden, um ein Fahrzeug zu entriegeln, wenn sich ein Eigentümer einer Fahrtür nähert, und Lichter anzuschalten, und um ein Fahrzeug im Sicherheitsmodus zu deaktivieren, wenn ein Eigentümer ein Fahrzeug verlässt. Auf diese Weise bietet(bieten) das(die) SoC(s) 1204 Schutz vor Diebstahl und/oder Fahrzeugraub.

[0118] In mindestens einer Ausführungsform kann ein CNN zur Erkennung und Identifizierung von Rettungsfahrzeugen Daten aus Mikrofonen 1296 verwenden, um Sirenen von Rettungsfahrzeugen zu erkennen und zu identifizieren. In mindestens einer Ausführungsform verwendet(verwenden) das(die) SoC(s) 1204 ein CNN zur Klassifizierung von Umwelt- und Stadtgeräuschen sowie zur Klassifizierung visueller Daten. In mindestens einer Ausführungsform wird ein CNN, das auf einem DLA läuft, trainiert, um die relative Annäherungsgeschwindigkeit des Einsatzfahrzeugs (z. B. durch Verwendung des Doppler-Effekts) zu kennzeichnen. In mindestens einer Ausführungsform kann ein CNN ebenfalls trainiert werden, um Rettungsfahrzeuge zu identi-

zieren, die spezifisch für das lokale Gebiet sind, in dem ein Fahrzeug eingesetzt wird, wie durch einen GNSS-Sensor(en) 1258 identifiziert. In mindestens einer Ausführungsform wird beispielsweise das CNN beim Arbeiten in Europa versuchen, europäische Sirenen zu erkennen, und wenn es in den Vereinigten Staaten ist, wird das CNN versuchen, nur nordamerikanische Sirenen zu identifizieren. In mindestens einer Ausführungsform kann, sobald ein Rettungsfahrzeug erfasst ist, ein Steuerprogramm verwendet werden, um eine Notfallroutine für die Fahrzeugsicherheit auszuführen, ein Fahrzeug zu verlangsamen, an einen Straßenrand zu fahren, ein Fahrzeug zu parken und/oder ein Fahrzeug mit Hilfe eines(von) Ultraschallsensors(en) 1262 im Leerlauf zu halten, bis ein oder mehrere Rettungsfahrzeuge vorbeifahren.

[0119] In mindestens einer Ausführungsform kann ein Fahrzeug 1200 CPU(s) 1218 (z. B. diskrete CPU(s) oder dCPU(s)) umfassen, die über eine Hochgeschwindigkeitsverbindung (z. B. PCIe) mit dem(den) SoC(s) 1204 gekoppelt sein kann(können). In mindestens einer Ausführungsform kann(können) die CPU(s) 1218 beispielsweise einen X86-Prozessor umfassen. Die CPU(s) 1218 kann(können) beispielsweise verwendet werden, um eine Vielfalt von Funktionen auszuführen, einschließlich einer Arbitrierung potenziell inkonsistenter Ergebnisse zwischen ADAS-Sensoren und dem(den) SoC(s) 1204 und/oder einer beispielhaften Überwachung von Status und Gesundheitszustand des(der) Controllers 1236 und/oder eines Infotainment-System-on-Chip („Infotainment-SoC“) 1230.

[0120] In mindestens einer Ausführungsform kann ein Fahrzeug 1200 eine oder mehrere GPU(s) 1020 (z. B. diskrete GPU(s) oder dGPU(s)) umfassen, die über eine Hochgeschwindigkeitsverbindung (z. B. NVIDIA's NVLINK) mit dem(den) SoC(s) 1204 gekoppelt werden kann(können). Die GPU(s) 1020 kann(können) eine zusätzliche Funktionalität künstlicher Intelligenz bereitstellen, wie beispielsweise durch die Ausführung redundanter und/oder unterschiedlicher neuronaler Netzwerke, und (kann)können verwendet werden, um neuronale Netzwerke basierend teilweise auf Eingaben (z. B. Sensordaten) von Sensoren des Fahrzeugs 1200 zu trainieren und/oder zu aktualisieren.

[0121] In mindestens einer Ausführungsform kann das Fahrzeug 1200 ferner eine Netzwerkschnittstelle 1224 umfassen, die eine oder mehrere drahtlose Antennen 1226 umfassen kann (z. B. eine oder mehrere drahtlose Antennen für unterschiedliche Kommunikationsprotokolle, wie beispielsweise eine Mobilfunkantenne, eine Bluetooth-Antenne usw.). In mindestens einer Ausführungsform kann eine Netzwerkschnittstelle 1224 verwendet werden, um eine drahtlose Konnektivität über das Internet mit Cloud (z. B. mit einem oder mehreren Server(n) und/oder anderen Netzwerkvorrichtungen), mit anderen Fahrzeugen und/oder mit Rechenvorrichtungen (z. B. Client-Vorrichtungen von Passagieren) zu ermöglichen. In mindestens einer Ausführungsform kann, um mit anderen Fahrzeugen zu kommunizieren, eine direkte Verbindung zwischen einem Fahrzeug 1200 und einem anderen Fahrzeug und/oder eine indirekte Verbindung (z. B. über Netzwerke und über Internet) aufgebaut werden. In mindestens einer Ausführungsform können direkte Verbindungen über eine Fahrzeug-zu-Fahrzeug-Kommunikationsverbindung bereitgestellt werden. In mindestens einer Ausführungsform stellt eine Fahrzeug-zu-Fahrzeug-Kommunikationsverbindung dem Fahrzeug 1200 Informationen über Fahrzeuge in der Nähe des Fahrzeugs 1200 bereitstellen (z. B. Fahrzeuge vor, an der Seite und/oder hinter dem Fahrzeug 1200). In mindestens einer Ausführungsform kann die vorerwähnte Funktionalität Teil einer kooperativen adaptiven Geschwindigkeitsregelungsfunktion eines Fahrzeugs 1200 sein.

[0122] In mindestens einer Ausführungsform kann eine Netzwerkschnittstelle 1224 ein SoC umfassen, das Modulations- und Demodulationsfunktionalität bereitstellt und einem Controller(n) 1236 eine Kommunikation über drahtlose Netzwerke ermöglicht. In mindestens einer Ausführungsform kann eine Netzwerkschnittstelle 1224 ein Hochfrequenz-Frontend zur Aufwärtskonvertierung von einem Basisband zur Hochfrequenz und zur Abwärtskonvertierung von einer Hochfrequenz in ein Basisband umfassen. In mindestens einer Ausführungsform können die Frequenzumwandlungen durch jedes technisch machbare Verfahren durchgeführt werden. Beispielsweise könnten Frequenzumwandlungen durch wohlbekannte Verfahren und oder durch Verwenden von Superheterodyn-Verfahren durchgeführt werden. In mindestens einer Ausführungsform kann die Radiofrequenz-Frontend-Funktionalität durch einen separaten Chip bereitgestellt werden. In mindestens einer Ausführungsform kann eine Netzwerkschnittstelle drahtlose Funktionen für die Kommunikation über LTE, WCDMA, UMTS, GSM, CDMA2000, Bluetooth, Bluetooth LE, Wi-Fi, Z-Wave, ZigBee, LoRaWAN und/oder andere drahtlose Protokolle umfassen.

[0123] In mindestens einer Ausführungsform kann das Fahrzeug 1200 ferner, ohne darauf beschränkt zu sein, eine Datenspeicherung(en) 1228 umfassen, der(die) auch Off-Chip-Speicher (z. B. außerhalb des(der) SoC(s) 1204)) umfassen kann(können). In mindestens einer Ausführungsform kann(können) der (die) Datenspeicher 1228, ohne darauf beschränkt zu sein, ein oder mehrere Speicherelemente einschließlich RAM, SRAM, dynamischen Direktzugriffspeicher („DRAM“), Video-Direktzugriffspeicher („VRAM“), Flash,

Festplatten und/oder andere Komponenten und/oder Vorrichtungen umfassen, die mindestens ein Datenbit speichern können.

[0124] In mindestens einer Ausführungsform kann das Fahrzeug 1200 ferner einen oder mehrere GNSS-Sensoren 1258 (z. B. GPS- und/oder assistierte GPS-Sensoren) umfassen, um bei der Kartierung, Wahrnehmung, Belegungsgittererzeugung und/oder Wegplanungsfunktionen zu helfen. In mindestens einer Ausführungsform kann eine beliebige Anzahl von GNSS-Sensoren 1258 verwendet werden, einschließlich beispielsweise, und ohne darauf beschränkt zu sein, eines GPS mit einem USB-Verbinder und einer Ethernet-zu-Seriell-(z. B., RS-232)-Brücke.

[0125] In mindestens einer Ausführungsform kann das Fahrzeug 1200 ferner einen RADAR-Sensor(en) 1260 umfassen. Ein oder mehrere RADAR-Sensoren 1260 können von dem Fahrzeug 1200 zur Fahrzeugerkennung mit großer Reichweite selbst bei Dunkelheit und/oder extremen Wetterbedingungen verwendet werden. In mindestens einer Ausführungsform können die funktionalen Sicherheitsstufen des RADAR gleich ASIL B sein. Ein RADAR-Sensor(en) 1260 kann(können) einen CAN-Bus und/oder einen Bus 1202 (z. B., um von RADAR-Sensoren 1260 erzeugte Daten zu übertragen) zur Steuerung und zum Zugriff auf Objektverfolgsdaten und mit Zugriff auf Ethernet zum Zugriff auf Rohdaten verwenden. In mindestens einer Ausführungsform kann eine breite Vielfalt von RADAR-Sensortypen verwendet werden. Beispielsweise, und ohne darauf beschränkt zu sein, kann(können) ein RADAR-Sensor(en) 1260 für den vorderen, hinteren und seitlichen RADAR-Einsatz geeignet sein. In mindestens einer Ausführungsform sind ein oder mehrere Sensor(en) 1260 ein Puls-Doppler-RADAR-Sensor(en).

[0126] In mindestens einer Ausführungsform kann(können) der(die) RADAR-Sensoren 1260 unterschiedliche Konfigurationen umfassen, wie z. B. eine große Reichweite mit engem Sichtfeld, eine kurze Reichweite mit breitem Sichtfeld, eine seitliche Abdeckung mit kurzer Reichweite usw. In mindestens einer Ausführungsform kann das RADAR mit großer Reichweite für eine adaptive Geschwindigkeitsregelungsfunktion verwendet werden. In mindestens einer Ausführungsform können RADAR-Systeme mit großer Reichweite ein breites Sichtfeld bereitstellen, das von zwei oder mehr unabhängigen Scans, wie beispielsweise innerhalb einer Reichweite von 250 m, verwirklicht wird. In mindestens einer Ausführungsform kann(können) der(die) RADAR-Sensor(en) 1260 helfen, zwischen statischen und beweglichen Objekten zu unterscheiden, und kann(können) von einem ADAS-System 1238 zur Notbremsunterstützung und Vorwärtskollisionswarnung verwendet werden. Ein Sensor(en) 1260, der(die) in einem RADAR-System mit großer Reichweite enthalten ist(sind), ohne darauf beschränkt zu sein, kann(können) monostatische multimodale RADAR-Sensoren mit mehreren (z. B. sechs oder mehr) festen RADAR-Antennen und einer Hochgeschwindigkeit-CAN- und Flex-Ray-Schnittstelle umfassen. In mindestens einer Ausführungsform mit sechs Antennen können die zentralen vier Antennen ein fokussiertes Strahlmuster erzeugen, das ausgelegt ist, um Umgebungen des Fahrzeugs 1200 bei höheren Geschwindigkeiten mit minimalen Störungen durch den Verkehr in benachbarten Spuren aufzunehmen. In mindestens einer Ausführungsform können andere zwei Antennen ein Sichtfeld erweitern, so dass es möglich ist, Fahrzeuge, die in eine Spur des Fahrzeugs 1200 eintreten oder diese verlassen, schnell erkannt werden können.

[0127] In mindestens einer Ausführungsform können RADAR-Systeme mit mittlerer Reichweite beispielsweise eine Reichweite von bis zu 160m (vorne) oder 80m (hinten) und ein Sichtfeld von bis zu 42 Grad (vorne) oder 150 Grad (hinten) umfassen. In mindestens einer Ausführungsform können RADAR-Systeme mit kurzer Reichweite, ohne darauf beschränkt zu sein, eine beliebige Anzahl von RADAR-Sensoren 1260 umfassen, die für die Installation an beiden Enden der hinteren Stoßstange ausgestaltet sind. Bei Installation an beiden Enden der hinteren Stoßstange kann, in mindestens einer Ausführungsform, ein derartiges RADAR-Sensorsystem zwei Strahlen erzeugen, die den toten Winkel im Heck und neben einem Fahrzeug ständig überwachen. In mindestens einer Ausführungsform können RADAR-Systeme mit kurzer Reichweite in einem ADAS-System 1238 zur Erkennung des toten Winkels und/oder zur Spurwechselassistentz verwendet werden.

[0128] In mindestens einer Ausführungsform kann das Fahrzeug 1200 ferner einen oder mehrere Ultraschallsensoren 1262 umfassen. Ein Ultraschallsensor(en) 1262, der(die) vorne, hinten und/oder an den Seiten des Fahrzeugs 1200 positioniert werden kann(können), für eine Einparkhilfe und/oder zum Erzeugen und Aktualisieren eines Belegungsgitters verwendet werden. In mindestens einer Ausführungsform kann eine breite Vielfalt von Ultraschallsensoren 1262 verwendet werden, und unterschiedliche Ultraschallsensoren 1262 können für unterschiedliche Erkennungsbereiche (z. B. 2,5 m; 4 m) verwendet werden. In mindestens einer Ausführungsform kann(können) ein Ultraschallsensor(en) 1262 auf den funktionalen Sicherheitsstufen von ASIL B arbeiten.

[0129] In mindestens einer Ausführungsform kann das Fahrzeug 1200 ein oder mehrere LIDAR-Sensoren 1264 umfassen. Ein LIDAR-Sensor(en) 1264 kann(können) zur Objekt- und Fußgängererkennung, Notbremsung, Kollisionsvermeidung und/oder für andere Funktionen verwendet werden. In mindestens einer Ausführungsform kann(können) ein LIDAR-Sensor(en) von der Funktionssicherheitsstufe ASIL B sein. In mindestens einer Ausführungsform kann ein Fahrzeug 1200 mehrere LIDAR-Sensoren 1264 (z. B. zwei, vier, sechs usw.) umfassen, die einen Ethernet-Kanal verwenden können (z. B. um einem Gigabit-Ethernet-Schalter Daten bereitzustellen).

[0130] In mindestens einer Ausführungsform kann(können) ein LIDAR-Sensor(en) 1264 in der Lage sein, eine Liste von Objekten und deren Abstände für ein 360-Grad-Sichtfeld bereitzustellen. Handelsübliche LIDAR-Sensoren 1264 können beispielsweise eine beworbene Reichweite von ungefähr 100m mit einer Genauigkeit von 2 cm - 3 cm und mit Unterstützung für eine 100 Mbit/s Ethernet-Verbindung aufweisen. In mindestens einer Ausführungsform können ein oder mehrere nicht vorstehende LIDAR-Sensoren 1264 verwendet werden. In einer derartigen Ausführungsform kann(können) der(die) LIDAR-Sensor(en) 1264 als eine kleine Vorrichtung implementiert sein, die in einer Vorderseite, einem Heck, einer Seite und/oder Ecken des Fahrzeugs 1200 eingebettet sein kann. In mindestens einer Ausführungsform kann(können) ein LIDAR-Sensor(en) 1264, in einer derartigen Ausführungsform, ein horizontales Sichtfeld von bis zu 120 Grad und ein vertikales Sichtfeld von 35 Grad mit einer Reichweite von 200m sogar für Objekte mit geringer Reflexion bereitstellen. In mindestens einer Ausführungsform kann(können) ein vorderseitig montierter LIDAR-Sensor(en) 1264 für ein horizontales Sichtfeld zwischen 45 Grad und 135 Grad konfiguriert sein.

[0131] In mindestens einer Ausführungsform können LIDAR-Technologien, wie beispielsweise 3D-Flash-LIDAR, ebenfalls verwendet werden. 3D-Flash-LIDAR verwendet einen Laserblitz als eine Übertragungsquelle, um Umgebungen des Fahrzeugs 1200 bis zu ungefähr 200m zu beleuchten. In mindestens einer Ausführungsform umfasst eine Flash-LIDAR-Einheit, ohne darauf beschränkt zu sein, einen Rezeptor, der die Laserpulslaufzeit und das reflektierte Licht auf jedem Pixel erfasst, was wiederum einer Reichweite von einem Fahrzeug 1200 bis zu Objekten entspricht. In mindestens einer Ausführungsform kann Flash-LIDAR ermöglichen, dass mit jedem Laserblitz hochpräzise und verzerrungsfreie Bilder von Umgebungen erzeugt werden. In mindestens einer Ausführungsform können vier Flash-LIDAR-Sensoren eingesetzt werden, einer an jeder Seite des Fahrzeugs 1200. In mindestens einer Ausführungsform umfassen 3D-Flash-LIDAR-Systeme, ohne darauf beschränkt zu sein, eine 3D-Festkörper-LIDAR-Kamera mit starrer Anordnung ohne bewegliche Teile außer einem Lüfter (z. B. eine nicht abtastende LIDAR-Vorrichtung). In mindestens einer Ausführungsform kann(können) eine Flash-LIDAR-Vorrichtung(en) einen Laser der Klasse I (augensicher) mit Pulsen von 5 Nanosekunden pro Einzelbild verwenden und das reflektierte Laserlicht in Form von 3D-Reichweitenpunktwolken und gemeinsam registrierten Intensitätsdaten erfassen.

[0132] In mindestens einer Ausführungsform kann ein Fahrzeug ferner einen oder mehrere IMU-Sensoren 1266 umfassen. In mindestens einer Ausführungsform kann(können) ein IMU-Sensor(en) 1266 in mindestens einer Ausführungsform an einer Mitte der Hinterachse des Fahrzeugs 1200 lokalisiert sein. In mindestens einer Ausführungsform kann(können) ein IMU-Sensor(en) 1266, beispielsweise und ohne darauf beschränkt zu sein, einen Beschleunigungssensor(en), Magnetometer, Gyroskop(e), Magnetkompass(e) und/oder andere Sensortypen umfassen. In mindestens einer Ausführungsform kann(können), wie beispielsweise in neunachsigen Anwendungen, ein IMU-Sensor(en) 1266 Beschleunigungssensoren und Gyroskope umfassen, während in neunachsigen Anwendungen ein IMU-Sensor(en) 1266 Beschleunigungssensoren, Gyroskope und Magnetometer umfassen können.

[0133] In mindestens einer Ausführungsform kann(können) ein IMU-Sensor(en) 1266 als ein miniaturisiertes, leistungsstarkes GPS-gestütztes Trägheitsnavigationssystem (GPS-aided Inertial Navigation System; „GPS/INS“) implementiert sein, das mikroelektromechanische Systeme (micro-electro-mechanical systems; MEMS) von Trägheitssensoren, einen hochempfindlichen GPS-Empfänger und erweiterte Kalman-Filteralgorithmen kombiniert, um Schätzungen von Position, Geschwindigkeitsvektor und Höhe bereitzustellen. In mindestens einer Ausführungsform kann(können) ein IMU-Sensor(en) 1266 dem Fahrzeug 1200 ermöglichen, die Fahrtrichtung abzuschätzen, ohne dass eine Eingabe von einem Magnetsensor erforderlich ist, indem Änderungen im Geschwindigkeitsvektor von PS zu einem IMU-Sensor(en) 1266 direkt beobachtet und korreliert werden. In mindestens einer Ausführungsform kann(können) ein IMU-Sensor(en) 1266 und ein GNSS-Sensor(en) 1258 in einer einzigen integrierten Einheit kombiniert sein.

[0134] In mindestens einer Ausführungsform kann ein Fahrzeug 1200 ein oder mehrere Mikrophone 1296 umfassen, die in und/oder um ein Fahrzeug 1200 angeordnet sind. In mindestens einer Ausführungsform

kann(können) ein Mikrofon(e) 1296 unter anderem zur Erkennung und Identifizierung von Rettungsfahrzeugen verwendet werden.

[0135] In mindestens einer Ausführungsform kann ein Fahrzeug 1200 ferner eine beliebige Anzahl von Kameratypen umfassen, einschließlich einer oder mehreren Stereokameras 1268, einer oder mehreren Weitwinkelkameras 1270, einer oder mehreren Infrarotkameras 1072, einer oder mehreren Rundumsichtkameras 1274, einer oder mehreren Kameras 1298 mit großer und/oder mittlerer Reichweite und/oder andere Kameratypen. In mindestens einer Ausführungsform können Kameras zum Erfassen von Bilddaten über einen gesamten Umfang des Fahrzeugs 1200 verwendet werden. In mindestens einer Ausführungsform können die verwendeten Kameratypen von den Ausführungsformen und Anforderungen an ein Fahrzeug 1200 abhängen. In mindestens einer Ausführungsform kann eine beliebige Kombination von Kameratypen verwendet werden, um die erforderliche Abdeckung rund um ein Fahrzeug 1200 bereitzustellen. In mindestens einer Ausführungsform kann sich die Anzahl der Kameras abhängig von der Ausführungsform unterscheiden. Beispielsweise könnte in mindestens einer Ausführungsform ein Fahrzeug 1200 sechs Kameras, sieben Kameras, zehn Kameras, zwölf Kameras und/oder eine andere Anzahl von Kameras umfassen. Kameras können, beispielsweise und ohne darauf beschränkt zu sein, Gigabit Multimedia Serial Link („GMSL“) und/oder Gigabit Ethernet unterstützen. In mindestens einer Ausführungsform könnte jede Kamera sein, wie hier vorstehend mit Bezug auf **Fig. 12A** und **Fig. 12B** ausführlicher beschrieben.

[0136] In mindestens einer Ausführungsform kann ein Fahrzeug 1200 ferner einen oder mehrere Schwingungssensoren 1242 umfassen. Schwingungssensor(en) 1242 kann(können) die Schwingungen von Komponenten des Fahrzeugs 1200, wie beispielsweise (von) einer Achse(n), messen. Beispielsweise können in mindestens einer Ausführungsform Änderungen in Schwingungen eine Veränderung in der Straßenoberfläche angeben. In mindestens einer Ausführungsform können, wenn zwei oder mehr Schwingungssensoren 1242 verwendet werden, Unterschiede zwischen Schwingungen verwendet werden, um Reibung oder Schlupf der Straßenoberfläche zu bestimmen (z. B., wenn ein Schwingungsunterschied zwischen einer angetriebenen Achse und einer sich frei drehenden Achse besteht).

[0137] In mindestens einer Ausführungsform kann ein Fahrzeug 1200 das ADAS-System 1238 umfassen. Das ADAS-System 1238 kann in einigen Beispielen, ohne darauf beschränkt zu sein, ein SoC umfassen. In mindestens einer Ausführungsform kann ein ADAS-System 1238, ohne darauf beschränkt zu sein, eine beliebige Anzahl und Kombination aus einem autonomen/adaptiven/automatischen Geschwindigkeitsregler-System („ACC“-System), einem kooperativen adaptiven Geschwindigkeitsregler-System („CACC“-System), einem Vorwärts-Zusammenstoßwarnung-System („FCW“-System), einem automatischen Notbremsung-System („AEB“-System), einem Spurverlassungswarnung-System („LDW“-System), einem Spurhalteassistent-System („LKA“-System), einem Tote-Winkel-Warnung-System („BSW“-System), einem hinteren Querverkehrswarnung-System („RCTW“-System), einem Kollisionswarn-System („CW“-System), einem Spurzentrierung-System („LC“-System) und/oder anderen Systemen, Merkmalen und/oder Funktionalität umfassen.

[0138] In mindestens einer Ausführungsform kann ein ACC-System einen oder mehrere RADAR-Sensoren 1260, einen oder mehrere LIDAR-Sensoren 1264 und/oder eine beliebige Anzahl von Kameras verwenden. In mindestens einer Ausführungsform kann ein ACC-System ein longitudinales ACC-System und/oder ein laterales ACC-System umfassen. In mindestens einer Ausführungsform überwacht und steuert ein longitudinales ACC-System einen Abstand zum Fahrzeug unmittelbar vor einem Fahrzeug 1200 und passt die Fahrzeuggeschwindigkeit automatisch an, um einen Sicherheitsabstand zu den vorausfahrenden Fahrzeugen einzuhalten. In mindestens einer Ausführungsform führt ein laterales ACC-System Abstandshaltung durch und empfiehlt einem Fahrzeug 1200, bei Bedarf Spuren zu wechseln. In mindestens einer Ausführungsform steht ein laterales ACC-System im Zusammenhang mit anderen ADAS-Anwendungen, wie beispielsweise LC und CW.

[0139] In mindestens einer Ausführungsform verwendet ein CACC-System Informationen von anderen Fahrzeugen, die über die Netzwerkschnittstelle 1224 und/oder ein oder mehrere drahtlose Antennen 1226 von anderen Fahrzeugen über eine drahtlose Verbindung oder indirekt über eine Netzwerkverbindung (z. B. über Internet) empfangen werden können. In mindestens einer Ausführungsform können direkte Verbindungen durch eine Fahrzeug-zu-Fahrzeug (vehicle-to-vehicle; „V2V“-) Kommunikationsverbindung bereitgestellt werden, während indirekte Verbindungen durch eine Infrastruktur-zu-Fahrzeug (infrastructure-to-vehicle; „I2V“-) Kommunikationsverbindung bereitgestellt werden können. Im Allgemeinen stellt das V2V-Kommunikationskonzept Informationen über die unmittelbar vorausfahrenden Fahrzeuge bereit (z. B. Fahrzeuge unmittelbar vor und in der gleichen Spur wie ein Fahrzeug 1200), während das I2V-Kommunikationskonzept Informationen über weiter entfernten Verkehr liefern kann. In mindestens einer Ausführungsform kann ein CACC-

System eine oder beide der I2V- und V2V-Informationsquellen umfassen. In mindestens einer Ausführungsform kann bei gegebener Information der vor einem Fahrzeug 1200 fahrenden Fahrzeuge ein CACC-System zuverlässiger sein und weist das Potenzial auf, die Gleichmäßigkeit des Verkehrsflusses zu verbessern und Stau auf einer Straße zu verringern.

[0140] In mindestens einer Ausführungsform ist das FCW-System ausgestaltet, um einen Fahrer vor einer Gefahr zu warnen, so dass ein Fahrer Korrekturmaßnahmen ergreifen kann. In mindestens einer Ausführungsform verwendet ein FCW-System eine nach vorne gerichtete Kamera und/oder einen oder mehrere RADAR-Sensoren 1260, die mit einem dedizierten Prozessor, DSP, FPGA und/oder ASIC gekoppelt sind, der elektrisch mit einer Fahrerrückkopplung gekoppelt ist, wie beispielsweise einer Anzeige, einem Lautsprecher und/oder einer Schwingungskomponente. In mindestens einer Ausführungsform kann ein FCW-System eine Warnung, wie beispielweise in Form eines Geräusches, einer optischer Warnung, einer Schwingung und/oder einem schnellen Bremsimpuls, bereitstellen.

[0141] In mindestens einer Ausführungsform erkennt ein AEB-System eine drohende Vorwärtskollision mit einem anderen Fahrzeug oder einem anderen Objekt und kann die Bremsen automatisch betätigen, wenn ein Fahrer innerhalb einer bestimmten Zeit oder Abstandsparameter keine Korrekturmaßnahmen ergreift. In mindestens einer Ausführungsform kann ein AEB-System eine oder mehrere nach vorne gerichtete Kameras und/oder einen oder mehrere Radarsensoren 1260 verwenden, die mit einem dedizierten Prozessor, DSP, FPGA und/oder ASIC gekoppelt sind. In mindestens einer Ausführungsform wird, wenn ein AEB-System eine Gefahr erkennt, ein AEB-System zunächst einen Fahrer warnen, um Korrekturmaßnahmen zur Vermeidung einer Kollision zu ergreifen, und, wenn ein Fahrer keine Korrekturmaßnahmen ergreift, kann ein AEB-System automatisch Bremsen in einem Bemühen betätigen, eine Auswirkung einer vorhergesagten Kollision zu verhindern oder mindestens abzumildern. In mindestens einer Ausführungsform kann ein AEB-System Techniken, wie beispielsweise dynamische Bremsunterstützung und/oder Bremsen bei drohender Kollision, umfassen.

[0142] In mindestens einer Ausführungsform stellt ein LDW-System visuelle, akustische und/oder taktile Warnungen bereit, wie beispielsweise Lenkrad- oder Sitzschwingungen, um einen Fahrer zu warnen, wenn ein Fahrzeug 1200 Fahrbahnmarkierungen überquert. In mindestens einer Ausführungsform wird ein LDW-System nicht aktiviert, wenn ein Fahrer ein absichtliches Verlassen der Spur durch Aktivieren eines Blinkers angibt. In mindestens einer Ausführungsform kann ein LDW-System nach vorne/seitlich gerichtete Kameras verwenden, die mit einem dedizierten Prozessor, DSP, FPGA und/oder ASIC gekoppelt sind, das elektrisch mit einer Fahrerrückkopplung gekoppelt ist, wie beispielsweise einer Anzeige, einem Lautsprecher und/oder einer Schwingungskomponente. In mindestens einer Ausführungsform ist ein LKA-System eine Variante eines LDW-Systems. Ein LKA-System stellt Lenkeingabe oder Bremsen bereit, um ein Fahrzeug 1200 zu korrigieren, wenn das Fahrzeug 1200 beginnt, seine Spur zu verlassen.

[0143] In mindestens einer Ausführungsform erkennt ein BSW-System Fahrzeuge im toten Winkel eines Fahrzeugs und warnt einen Fahrer davor. In mindestens einer Ausführungsform kann ein BSW-System einen visuellen, akustischen und/oder taktilen Alarm bereitstellen, um anzugeben, dass das Zusammenführen oder Wechseln von Spuren unsicher ist. In mindestens einer Ausführungsform kann ein BSW-System eine zusätzliche Warnung ausgeben, wenn ein Fahrer einen Blinker verwendet. In mindestens einer Ausführungsform kann ein BSW-System eine oder mehrere rückseitig gerichtete Kameras und/oder einen oder mehrere RADAR-Sensoren 1260 verwenden, die mit einem dedizierten Prozessor, DSP, FPGA und/oder ASIC gekoppelt sind, der(das) elektrisch mit einer Fahrerrückkopplung gekoppelt ist, wie beispielsweise einer Anzeige, einem Lautsprecher und/oder einer Schwingungskomponente.

[0144] In mindestens einer Ausführungsform kann ein RCTW-System eine visuelle, akustische und/oder taktile Benachrichtigung bereitstellen, wenn ein Objekt außerhalb des Rückfahrkamerabereichs erkannt wird, wenn ein Fahrzeug 1200 rückwärtsfährt. In mindestens einer Ausführungsform umfasst ein RCTW-System ein AEB-System, um sicherzustellen, dass die Fahrzeugbremsen betätigt werden, um einen Unfall zu vermeiden. In mindestens einer Ausführungsform kann ein RCTW-System einen oder mehrere nach hinten gerichtete RADAR-Sensoren 1260 verwenden, der(die) mit einem dedizierten Prozessor, DSP, FPGA und/oder ASIC gekoppelt ist(sind), der elektrisch mit einer Fahrerrückkopplung, wie beispielsweise einer Anzeige, einem Lautsprecher und/oder einer Schwingungskomponente, gekoppelt ist.

[0145] In mindestens einer Ausführungsform können herkömmliche ADAS-Systeme für falsch positive Ergebnisse anfällig sein, die für einen Fahrer lästig und störend sein können, jedoch typischerweise nicht katastrophal sind, weil ADAS-Systeme einen Fahrer warnen und dem Fahrer ermöglichen, zu entscheiden,

ob eine Sicherheitsbedingung tatsächlich vorliegt, und entsprechend zu handeln. In mindestens einer Ausführungsform entscheidet das Fahrzeug 1200 selbst im Falle von widersprüchlichen Ergebnissen, ob ein Ergebnis von einem Primärcomputer oder einem Sekundärcomputer (z. B. einem ersten Controller oder einem zweiten Controller 1236) zu beachten ist. Beispielsweise kann in mindestens einer Ausführungsform das ADAS-System 1238 ein Backup- und/oder ein Sekundärcomputer sein, um einem Rationalitätsmodul eines Backup-Computers Wahrnehmungsinformationen zur Verfügung zu stellen. In mindestens einer Ausführungsform kann ein Rationalitätsmonitor des Backup-Computers eine redundante, diverse Software auf Hardwarekomponenten ausführen, um Fehler in der Wahrnehmung und dynamischen Fahraufgaben zu erkennen. In mindestens einer Ausführungsform können Ausgaben von einem ADAS-System 1238 einer Supervisor-MCU bereitgestellt werden. In mindestens einer Ausführungsform bestimmt, wenn Ausgaben von einem Primärcomputer und einem Sekundärcomputer im Widerspruch stehen, die Supervisor-MCU, wie der Konflikt zu lösen ist, um einen sicheren Betrieb zu gewährleisten.

[0146] In mindestens einer Ausführungsform kann ein Primärcomputer konfiguriert sein, um die Supervisor-MCU mit einem Vertrauenswert bereitzustellen, der das Vertrauen eines Primärcomputers in ein gewähltes Ergebnis anzeigt. In mindestens einer Ausführungsform kann, wobei der Vertrauenswert einen Schwellenwert überschreitet, eine Supervisor-MCU einer Richtung eines Primärcomputers ungeachtet dessen folgen, ob ein Sekundärcomputer ein widersprüchliches oder inkonsistentes Ergebnis liefert. In mindestens einer Ausführungsform kann, wobei ein Vertrauenswert einen Schwellenwert nicht erreicht und wobei Primär- und Sekundärcomputer unterschiedliche Ergebnisse angeben (z. B. einen Konflikt), kann eine Supervisor-MCU zwischen Computern vermitteln, um ein angemessene Ergebnis zu bestimmen.

[0147] In mindestens einer Ausführungsform kann eine Supervisor-MCU konfiguriert sein, um neuronale Netzwerke zu betreiben, die trainiert und konfiguriert sind, um basierend teilweise auf den Ergebnissen von einem Primärcomputers und einem Sekundärcomputer Bedingungen zu bestimmen, unter denen ein Sekundärcomputer Fehlalarme liefert. In mindestens einer Ausführungsform kann(können) ein neuronales Netzwerk(e) in einer Supervisor-MCU lernen, wenn einer Ausgabe des Sekundärcomputers vertraut werden kann und wenn nicht. Wenn der Sekundärcomputer in mindestens einer Ausführungsform beispielsweise ein RADAR-basiertes FCW-System ist, kann(können) ein neuronales Netzwerk(e) in der Supervisor-MCU lernen, wenn ein FCW-System metallische Objekte identifiziert, die in Wirklichkeit keine Gefahren sind, wie beispielsweise ein Drainagerost oder eine Schachtabdeckung, die einen Alarm auslöst. In mindestens einer Ausführungsform kann, wenn ein Sekundärcomputer ein kamerabasiertes LDW-System ist, ein neuronales Netzwerk in der Supervisor-MCU lernen, das LDW zu übersteuern, wenn Radfahrer oder Fußgänger anwesend sind und ein Verlassen der Spur tatsächlich das sicherste Manöver ist. In mindestens einer Ausführungsform kann eine Supervisor-MCU mindestens eines von einer DLA oder einer GPU umfassen, die zum Ausführen eines neuronalen Netzwerks(e) mit zugehörigem Speicher geeignet ist. In mindestens einer Ausführungsform kann eine Supervisor-MCU eine Komponente umfassen und/oder als eine Komponente des(der) SoC(s) 1204 enthalten sein.

[0148] In mindestens einer Ausführungsform kann das ADAS-System 1238 einen Sekundärcomputer umfassen, der eine ADAS-Funktionalität unter Verwendung herkömmlicher Regeln der Computer-Vision ausführt. In mindestens einer Ausführungsform kann ein Sekundärcomputer klassische Computer-Vision-Regeln (wenn-dann) verwenden und das Vorhandensein eines oder mehrerer neuronaler Netzwerke in der Supervisor-MCU kann Zuverlässigkeit, Sicherheit und Leistung verbessern. Beispielsweise macht in mindestens einer Ausführungsform die vielfältige Implementierung und absichtliche Nicht-Identität das Gesamtsystem insbesondere gegenüber Fehlern fehlertoleranter, die durch eine Funktionalität von Software (oder einer Software-Hardware-Schnittstelle) verursacht werden. Wenn es beispielsweise in mindestens einer Ausführungsform einen Software-Bug oder Fehler in einer auf dem Primärcomputer laufenden Software gibt und ein nicht-identischer Softwarecode, der auf einem Sekundärcomputer läuft, ein gleiches Gesamtergebnis liefert, dann kann eine Supervisor-MCU mehr Vertrauen aufweisen, dass ein Gesamtergebnis korrekt ist und ein Bug in der Software oder Hardware auf einem Primärcomputer keinen wesentlichen Fehler verursacht.

[0149] In mindestens einer Ausführungsform kann eine Ausgabe eines ADAS-Systems 1238 in einen Wahrnehmungsblock eines Primärcomputers und/oder in einen dynamischen Fahraufgabenblock eines Primärcomputers eingespeist werden. Wenn in mindestens einer Ausführungsform ein ADAS-System 1238 beispielsweise eine Vorwärts-Zusammenstoß-Warnung aufgrund eines unmittelbar vorausgehenden Objekts angibt, kann ein Wahrnehmungsblock diese Information beim Identifizieren von Objekten verwenden. In mindestens einer Ausführungsform kann ein Sekundärcomputer sein eigenes neuronales Netzwerk aufweisen, das trainiert wird und somit das Risiko von falschen Positiven verringert, wie hier beschrieben.

[0150] In mindestens einer Ausführungsform kann ein Fahrzeug 1200 ferner ein Infotainment-SoC 1230 umfassen (z. B. ein Infotainmentsystem in einem Fahrzeug (in-vehicle infotainment; IVI)). Obwohl als ein SoC in mindestens einer Ausführungsform veranschaulicht und beschrieben, kann das Infotainmentsystem 1230, in mindestens einer Ausführungsform, möglicherweise kein SoC sein und kann, ohne darauf beschränkt zu sein, zwei oder mehr diskrete Komponenten umfassen. In mindestens einer Ausführungsform kann ein Infotainment-SoC 1230, ohne darauf beschränkt zu sein, eine Kombination aus Hardware und Software umfassen, die verwendet werden kann, um Audio (z. B. Musik, einen persönlichen digitalen Assistenten, Navigationsanweisungen, Nachrichten, Radio usw.), Video (z. B. TV, Filme, Streaming usw.), Telefon (z. B. Freisprechfunktion), Netzwerkkonnektivität (z. B. LTE, WiFi usw.) und/oder Informationsdienste (z. B. Navigationssysteme, Heck-Einpark-Assistenz, ein Funkdatensystem, fahrzeugbezogene Informationen wie beispielsweise Kraftstoffstand, zurückgelegte Gesamtstrecke, Bremskraftstoffstand, Ölstand, Tür auf/zu, Luftfilterinformation usw.) einem Fahrzeug 1200 bereitzustellen. Beispielsweise könnte ein Infotainment-SoC 1230 Radios, Diskettenspieler, Navigationssysteme, Videoplayer, USB- und Bluetooth-Konnektivität, Carputer, In-Car-Entertainment, WiFi, Lenkrad-Audiosteuerungen, Freisprechfunktion, eine Head-Up-Display („HUD“), eine HMI-Anzeige 1234, ein Telematikgerät, ein Bedienfeld (z. B. zur Steuerung und/oder Wechselwirkung mit unterschiedlichen Komponenten, Merkmalen und/oder Systemen) und/oder andere Komponenten umfassen. In mindestens einer Ausführungsform kann ein Infotainment-SoC 1230 ferner verwendet werden, um Informationen (z. B. visuell und/oder akustisch) einem Benutzer(n) eines Fahrzeugs bereitzustellen, wie beispielsweise Informationen von einem ADAS-System 1238, autonome Fahrinformationen, wie beispielsweise geplante Fahrzeugmanöver, Trajektorien, umgebende Umweltinformationen (z. B. Kreuzungsinformation, Fahrzeuginformation, Straßeninformation usw.) und/oder andere Informationen bereitzustellen.

[0151] In mindestens einer Ausführungsform kann ein Infotainment-SoC 1230 jede Menge und Typ an GPU-Funktionalität umfassen. In mindestens einer Ausführungsform kann ein Infotainment-SoC 1230 über einen Bus 1202 (z. B. CAN-Bus, Ethernet usw.) mit anderen Vorrichtungen, Systemen und/oder Komponenten eines Fahrzeugs 1200 kommunizieren. In mindestens einer Ausführungsform kann ein Infotainment-SoC 1230 mit einer Supervisor-MCU gekoppelt sein, so dass eine GPU des Infotainmentsystems einige Selbstfahrfunktionen in einem Fall übernehmen kann, in dem der(die) primäre(n) Controller 1236 (z. B. die primären und/oder Backup-Computer des Fahrzeugs 1200) versagt(versagen). In mindestens einer Ausführungsform kann ein Infotainment-SoC 1230 ein Fahrzeug 1200 in einen Chauffeur-zu-sicherem-Haltmodus versetzen, wie hier beschrieben.

[0152] In mindestens einer Ausführungsform kann ein Fahrzeug 1200 ferner ein Kombiinstrument 1232 umfassen (z. B. ein digitales Armaturenbrett, ein elektronisches Kombiinstrument, eine digitale Instrumententafel usw.). Ein Kombiinstrument 1232 kann, ohne darauf beschränkt zu sein, einen Controller und/oder einen Supercomputer (z. B. einen diskreten Controller oder einen Supercomputer) umfassen. In mindestens einer Ausführungsform kann ein Kombiinstrument 1232, ohne darauf beschränkt zu sein, eine beliebige Anzahl und Kombination eines Satzes von Instrumenten umfassen, wie beispielsweise einen Geschwindigkeitsmesser, einen Kraftstoffstand, einen Öldruck, einen Drehzahlmesser, einen Kilometerzähler, einen Blinker, eine Schaltpositionsanzeige, eine oder mehrere Sicherheitsgurtwarnleuchten, eine oder mehrere Feststellbremswarnleuchten, eine oder mehrere Motorstörungsleuchten, zusätzliches Rückhaltesystem(z. B. Airbag)-Informationen, Beleuchtungssteuerungen, Sicherheitssystemsteuerungen, Navigationsinformationen usw. In einigen Beispielen können Informationen zwischen einem Infotainment-SoC 1230 und Kombiinstrument 1232 angezeigt und/oder gemeinsam benutzt werden. In mindestens einer Ausführungsform kann ein Kombiinstrument 1232 als Teil eines Infotainment-SoC 1230 integriert sein oder umgekehrt.

[0153] Eine Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915 in dem System von **Fig. 12C** zum Inferenzieren oder Voraussagen von Operationen basierend mindestens teilweise auf Gewichtungparameter verwendet werden, die unter Verwendung von neuronalen Netzwerktrainingsoperationen, neuronalen Netzwerkfunktionen und/oder Architekturen, oder hier beschriebenen Anwendungsfällen eines neuronalen Netzwerks berechnet wurden.

[0154] **Fig. 12D** ist ein Systemdiagramm 1276 zur Kommunikation zwischen einem oder mehreren Cloud-basierten Servern und einem autonomen Fahrzeug 1200 von **Fig. 12A** gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann ein System 1276, ohne darauf beschränkt zu sein, einen oder mehrere Server 1278, ein oder mehrere Netzwerke 1290 sowie eine beliebige Anzahl und Typ von Fahrzeugen, einschließlich eines Fahrzeugs 1200, umfassen. In mindestens einer Ausführungsform

kann(können) ein Server 1278 mehrere GPUs 1284(A)-1284(H) (hier zusammenfassend als GPUs 1284 bezeichnet), PCIe-Schalter 1282(A)-1282(H) (hier zusammenfassend als PCIe-Schalter 1282 bezeichnet) und/oder CPUs 1280(A)-1280(B) (hier zusammenfassend als CPUs 1280 bezeichnet) umfassen. GPUs 1284, CPUs 1280 und PCIe-Schalter 1282 können mit Hochgeschwindigkeits-Zwischenverbindungen, wie beispielsweise und ohne darauf beschränkt zu sein, NVLink-Schnittstellen 1288, die von NVIDIA entwickelt wurden, und/oder PCIe-Anschlüssen 1086 verbunden sein. In mindestens einer Ausführungsform sind GPUs 1284 über einen NVLink und/oder NVSwitch-SoC verbunden und GPUs 1284 und PCIe-Schalter 1282 sind über PCIe-Zwischenverbindungen verbunden. In mindestens einer Ausführungsform, obwohl acht GPUs 1284, zwei CPUs 1280 und vier PCIe-Schalter 1282 veranschaulicht sind, ist dies nicht bestimmt, einschränkend zu sein. In mindestens einer Ausführungsform kann jeder (der) Server 1278, ohne darauf beschränkt zu sein, eine beliebige Anzahl von GPUs 1284, CPUs 1280 und/oder PCIe-Schaltern 1282 in beliebiger Kombination umfassen. Beispielsweise könnten in mindestens einer Ausführungsform ein oder mehrere Server 1278 jeweils acht, sechzehn, zweiunddreißig und/oder mehr GPUs 1284 umfassen.

[0155] In mindestens einer Ausführungsform kann(können) ein Server 1278 über ein Netzwerk(e) 1290 und von Fahrzeugen Bilddaten empfangen, die Bilder repräsentieren, die unerwartete oder veränderte Straßenbedingungen zeigen, wie beispielsweise kürzlich begonnene Straßenarbeiten. In mindestens einer Ausführungsform kann(können) ein Server 1278 über ein Netzwerk(e) 1290 und an Fahrzeuge neuronale Netzwerke 1292, aktualisierte neuronale Netzwerke 1292 und/oder Karteninformation 1294 übertragen, einschließlich, ohne darauf beschränkt zu sein, Informationen hinsichtlich Verkehrs- und Straßenzustände. In mindestens einer Ausführungsform kann(können) Aktualisierungen der Karteninformationen 1294 Aktualisierungen für die HD-Karte 1222 umfassen, wie beispielsweise Informationen über Baustellen, Schlaglöcher, Umwege, Überschwemmungen und/oder andere Hindernisse umfassen. In mindestens einer Ausführungsform könnten sich neuronale Netzwerke 1292, aktualisierte neuronale Netzwerke und/oder Karteninformationen 1294 aus einem neuen Training und/oder aus Erfahrungen, die durch Daten aus einer beliebigen Anzahl von Fahrzeugen in der Umgebung repräsentiert werden, und/oder basierend auf einem in einem Rechenzentrum durchgeführten Training (z. B. unter Verwendung des(der) Servers 1278 und/oder eines anderen/anderer Servers) ergeben haben.

[0156] In mindestens einer Ausführungsform kann(können) ein Server 1278 verwendet werden, um Modelle für maschinelles Lernen (z. B. neuronale Netzwerke) basierend mindestens teilweise auf Trainingsdaten zu trainieren. Trainingsdaten können durch Fahrzeuge erzeugt werden und/oder können in einer Simulation (z. B. unter Verwendung einer Spielmaschine) erzeugt werden. In mindestens einer Ausführungsform werden beliebige viele Trainingsdaten mit einer Kennzeichnung versehen (z. B., wobei ein neuronales Netzwerk von überwachtem Lernen profitiert) und/oder durchläuft eine andere Vorverarbeitung. In mindestens einer Ausführungsform werden beliebig viele Trainingsdaten nicht mit einer Kennzeichnung versehen und/oder vorverarbeitet (z. B., wenn das neuronale Netzwerk kein überwachtes Lernen erfordert). In mindestens einer Ausführungsform können, sobald Modelle für maschinelles Lernen trainiert sind, Modelle für maschinelles Lernen von Fahrzeugen verwendet werden (z. B. an Fahrzeuge über ein Netzwerk(e) 1290 übertragen), und/oder Modelle für maschinelles Lernen können von einem Server(n) 1278 zur Fernüberwachung von Fahrzeugen verwendet werden.

[0157] In mindestens einer Ausführungsform kann(können) ein Server 1278 Daten von Fahrzeugen empfangen und Daten auf aktuelle neuronale Echtzeit-Netzwerke für intelligente Echtzeit-Inferenzierung anwenden. In mindestens einer Ausführungsform kann(können) ein Server 1278 Deep-Learning-Supercomputer und/oder von einer oder mehreren GPUs 1284 angetriebene dedizierte KI-Computer umfassen, wie beispielsweise ein DGX und DGX-Station-Maschinen, die von NVIDIA entwickelt wurden. In mindestens einer Ausführungsform kann(können) ein Server 1278 jedoch eine Deep-Learning-Infrastruktur umfassen, die CPU-angetriebene Rechenzentren verwendet.

[0158] In mindestens einer Ausführungsform kann eine Deep-Learning-Infrastruktur von einem Server(n) 1278 in der Lage sein, schnell und in Echtzeit zu inferenzieren, und kann diese Fähigkeit nutzen, um die Gesundheit der Prozessoren, der Software und/oder der zugehörigen Hardware in einem Fahrzeug 1200 zu bewerten und zu überprüfen. Beispielsweise kann in mindestens einer Ausführungsform eine Deep-Learning-Infrastruktur periodische Aktualisierungen von einem Fahrzeug 1200 erhalten, wie beispielsweise eine Folge von Bildern und/oder Objekten, die ein Fahrzeug 1200 in dieser Folge von Bildern lokalisiert hat (z. B. durch Computer-Vision und/oder andere Techniken des maschinellen Lernens zur Klassifizierung von Lernobjekten). In mindestens einer Ausführungsform kann die Deep-Learning-Infrastruktur ihr eigenes neuronales Netzwerk betreiben, um Objekte zu kennzeichnen und sie mit den von einem Fahrzeug 1200 identifizierten Objekten zu vergleichen, und wenn Ergebnisse nicht übereinstimmen und eine Deep-Learning-Infrastruktur

zu dem Schluss kommt, dass KI in einem Fahrzeug 1200 nicht funktioniert, dann kann(können) ein Server 1278 ein Signal an ein Fahrzeug 1200 senden, das einen ausfallsicheren Computer eines Fahrzeugs 1200 anweist, Steuerung zu übernehmen, Passagiere zu benachrichtigen und ein sicheres Parkmanöver durchzuführen.

[0159] In mindestens einer Ausführungsform kann(können) ein Server 1278 eine oder mehrere GPU(s) 1284 und einen oder mehrere programmierbare Inferenzbeschleuniger (z. B. NVIDIA's TensorRT 3) umfassen. In mindestens einer Ausführungsform kann eine Kombination aus GPU-angetriebenen Servern und Inferenzbeschleunigung Echtzeit-Reaktionsfähigkeit möglich machen. In mindestens einer Ausführungsform, wie beispielsweise wo Leistung weniger kritisch ist, können Server, die von CPUs, FPGAs und anderen Prozessoren betrieben werden, zur Inferenzierung verwendet werden. In mindestens einer Ausführungsform werden eine oder mehrere Hardwarestrukturen 915 verwendet, um eine oder mehrere Ausführungsformen durchzuführen. Einzelheiten hinsichtlich einer Hardwarestruktur(en) 915 werden hier in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt.

COMPUTERSYSTEME

[0160] **Fig. 13** ist ein Blockdiagramm, das ein beispielhaftes Computersystem veranschaulicht, welches ein System mit miteinander verbundenen Vorrichtungen und Komponenten, ein System auf einem Chip (SOC) oder eine Kombination davon sein kann, das mit einem Prozessor gebildet ist, der Ausführungseinheiten zum Ausführen einer Anweisung umfassen kann, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann ein Computersystem 1300, ohne darauf beschränkt zu sein, eine Komponente umfassen, wie beispielsweise einen Prozessor 1302, um Ausführungseinheiten mit Logik zum Ausführen von Algorithmen für Prozessdaten in Übereinstimmung mit der vorliegenden Offenbarung, wie beispielsweise der hier beschriebenen Ausführungsformen, zu benutzen. In mindestens einer Ausführungsform kann ein Computersystem 1300 Prozessoren, wie beispielsweise die PENTIUM®-Prozessorfamilie, Xenon™, Itanium®, XScale™ und/oder StrongARM™ Intel® Core™ oder Intel® Nervana™ Mikroprozessoren, umfassen, die von der Intel Corporation in Santa Clara, Kalifornien, erhältlich sind, obwohl auch andere Systeme (einschließlich PCs mit anderen Mikroprozessoren, Engineering-Arbeitsplätze, Set-Top-Boxen und dergleichen) verwendet werden können. In mindestens einer Ausführungsform kann ein Computersystem 1300 eine Version des Betriebssystems WINDOWS ausführen, das bei der Microsoft Corporation in Redmond, Washington erhältlich ist, obwohl auch andere Betriebssysteme (z.B. UNIX und Linux), eingebettete Software und/oder graphische Benutzeroberflächen verwendet werden können.

[0161] Ausführungsformen können in anderen Geräten wie beispielsweise Handheld-Geräten und Eingebettete-Anwendungen verwendet werden. Einige Beispiele für tragbare bzw. Handheld-Geräte umfassen Mobiltelefone, Internet-Protokoll-Geräte, Digitalkameras, persönliche digitale Assistenten („PDAs“) und Handheld-PCs. In mindestens einer Ausführungsform können eingebettete Anwendungen einen Mikrocontroller, einen digitalen Signalprozessor („DSP“), ein System auf einem Chip, Netzwerkcomputer („NetPCs“), Set-Top-Boxen, Netzwerk-Hubs, Wide Area Network („WAN“) Schalter oder jedes andere System umfassen, das eine oder mehrere Anweisungen ausführen kann, gemäß mindestens einer Ausführungsform.

[0162] In mindestens einer Ausführungsform kann ein Computersystem 1300, ohne darauf beschränkt zu sein, einen Prozessor 1302 umfassen, der, ohne darauf beschränkt zu sein, eine oder mehrere Ausführungseinheiten 1308 umfassen kann, um in Übereinstimmung mit hier beschriebenen Techniken ein Training eines maschinell lernenden Modells und/oder Inferenzieren durchzuführen. In mindestens einer Ausführungsform ist ein System 13 ein Einzelprozessor-Desktop- oder Serversystem, wobei in einer anderen Ausführungsform ein System 13 jedoch ein Multiprozessorsystem sein kann. In mindestens einer Ausführungsform kann ein Prozessor 1302, ohne darauf beschränkt zu sein, einen Mikroprozessor mit komplexem Befehlssatz-Computer („CISC“), einen Mikroprozessor mit reduziertem Befehlssatz („RISC“), einen Mikroprozessor mit sehr langem Befehlswort („VLIW“), einen Prozessor, der eine Kombination von Befehlssätzen implementiert, oder jede andere Prozessorvorrichtung, wie beispielsweise einen digitalen Signalprozessor, umfassen. In mindestens einer Ausführungsform kann ein Prozessor 1302 mit einem Prozessorbus 1310 gekoppelt sein, der Datensignale zwischen einem Prozessor 1302 und anderen Komponenten in einem Computersystem 1300 übertragen kann.

[0163] In mindestens einer Ausführungsform kann ein Prozessor 1302, ohne darauf beschränkt zu sein, einen internen Level-1 („L1“)-Cache-Speicher („Cache“) 1304 umfassen. In mindestens einer Ausführungsform kann ein Prozessor 1302 einen einzelnen internen Cache oder mehrere Ebenen eines internen Caches aufweisen. In mindestens einer Ausführungsform kann sich ein Cache-Speicher extern zu einem Prozessor

1302 befinden. Andere Ausführungsformen können je nach Implementierung und Bedarf auch eine Kombination aus internen und externen Caches umfassen. In mindestens einer Ausführungsform kann eine Registerdatei 1306 verschiedene Arten von Daten in verschiedenen Registern speichern, einschließlich, jedoch nicht beschränkt auf, Ganzzahlregister, Gleitkommaregister, Zustandsregister und einem Befehlszeigerregister.

[0164] In mindestens einer Ausführungsform befindet sich eine Ausführungseinheit 1308, einschließlich, jedoch ohne darauf beschränkt zu sein, Logik zum Ausführen von Ganzzahl- und Gleitkommaoperationen ebenfalls in dem Prozessor 1302. Ein Prozessor 1302 kann ebenfalls einen Mikrocode („ucode“) Nur-Lese-Speicher („ROM“) umfassen, der Mikrocode für bestimmte Makroanweisungen speichert. In mindestens einer Ausführungsform kann eine Ausführungseinheit 1308 Logik zur Handhabung eines gepackten Befehlssatzes 1309 umfassen. In mindestens einer Ausführungsform können, indem ein gepackter Befehlssatz 1309 in einen Befehlssatz eines Allzweck-Prozessors 1302 aufgenommen wird, zusammen mit zugehörigen Schaltungen zur Ausführung von Anweisungen Operationen, die von vielen Multimedia-Anwendungen verwendet werden, unter Verwendung von gepackten Daten in einem Allzweck-Prozessor 1302 durchgeführt werden. In einer oder mehreren Ausführungsformen können viele Multimedia-Anwendungen beschleunigt und effizienter ausgeführt werden, indem die volle Breite eines Datenbusses eines Prozessors für die Ausführung von Operationen auf gepackten Daten verwendet wird, welches die Notwendigkeit eliminieren kann, kleinere Dateneinheiten über einen Datenbus eines Prozessors zu übertragen, um eine oder mehrere Operationen jeweils ein Datenelement auf einmal durchzuführen.

[0165] In mindestens einer Ausführungsform kann eine Ausführungseinheit 1308 auch in Mikrocontrollern, eingebetteten Prozessoren, Graphikvorrichtungen, DSPs und anderen Arten von Logikschaltungen verwendet werden. In mindestens einer Ausführungsform kann ein Computersystem 1300, ohne darauf beschränkt zu sein, einen Speicher 1320 umfassen. In mindestens einer Ausführungsform kann ein Speicher 1320 als eine Dynamic Random Access Memory („DRAM“)-Vorrichtung, als eine Static Random Access Memory („SRAM“)-Vorrichtung, als eine Flash-Speichervorrichtung oder andere Speichervorrichtung implementiert sein. Ein Speicher 1320 kann eine oder mehrere Anweisungen 1319 und/oder Daten 1321 speichern, die durch Datensignale repräsentiert werden, die von einem Prozessor 1302 ausgeführt werden können.

[0166] In mindestens einer Ausführungsform kann ein Systemlogikchip mit einem Prozessorbus 1310 und einem Speicher 1320 gekoppelt sein. In mindestens einer Ausführungsform kann der Systemlogikchip, ohne darauf beschränkt zu sein, einen Speichercontroller-Hub („MCH“) 1316 umfassen, und ein Prozessor 1302 kann über einen Prozessorbus 1310 mit einem MCH 1316 kommunizieren. In mindestens einer Ausführungsform kann ein MCH 1316 einen Speicherpfad 1318 mit hoher Bandbreite zu einem Speicher 1320 zur Befehls- und Datenspeicherung sowie zur Speicherung von Graphikbefehlen, Daten und Texturen bereitstellen. In mindestens einer Ausführungsform kann ein MCH 1316 Datensignale zwischen einem Prozessor 1302, einem Speicher 1320 und anderen Komponenten in einem Computersystem 1300 leiten und Datensignale zwischen einem Prozessorbus 1310, einem Speicher 1320 und einer System-E/A 1322 überbrücken. In mindestens einer Ausführungsform kann ein Systemlogikchip einen Graphikport zur Kopplung an einen Graphik-Controller bereitstellen. In mindestens einer Ausführungsform kann ein MCH 1316 über einen Speicherpfad 1318 mit hoher Bandbreite mit einem Speicher 1320 gekoppelt sein, und eine Graphik-/Video-Karte 1318 kann über eine Accelerated Graphics Port („AGP“) Zwischenverbindung 1316 mit einem MCH 1316 gekoppelt sein.

[0167] In mindestens einer Ausführungsform kann ein Computersystem 1300 die System-E/A-Schnittstelle 1322 als einen proprietären Hub-Schnittstellenbus verwenden, um einen MCH 1316 mit einem E/A-Controller-Hub („ICH“) 1330 zu verbinden. In mindestens einer Ausführungsform kann ein ICH 1330 über einen lokalen E/A-Bus direkte Verbindungen zu einigen E/A-Vorrichtungen bereitstellen. In mindestens einer Ausführungsform kann ein lokaler E/A-Bus, ohne darauf beschränkt zu sein, einen Hochgeschwindigkeits-E/A-Bus zum Verbinden von Peripheriegeräten mit einem Speicher 1320, einem Chipsatz und einem Prozessor 1302 umfassen. Beispiele können, ohne darauf beschränkt zu sein, einen Audio-Controller 1329, einen Firmware-Hub („Flash-BIOS“) 1328, einen drahtlosen Transceiver 1326, eine Datenspeicherung 1324, einen Legacy-E/A-Controller 1323 mit Benutzereingabe und Tastaturschnittstellen, einen seriellen Erweiterungsport 1327, wie beispielsweise einen Universal Serial Bus („USB“) Port, und ein Netzwerk-Controller 1334 sein. Eine Datenspeicherung 1324 kann ein Festplattenlaufwerk, ein Diskettenlaufwerk, eine CD-ROM-Vorrichtung, eine Flash-Speicher-Vorrichtung oder eine andere Massenspeichervorrichtung umfassen.

[0168] In mindestens einer Ausführungsform veranschaulicht **Fig. 13** ein System, welches miteinander verbundene Hardware-Vorrichtungen oder „Chips“ umfasst, während in anderen Ausführungsformen **Fig. 13** ein beispielhaftes System-on-Chip („SoC“) darstellen kann. In mindestens einer Ausführungsform können in

Fig. 13 veranschaulichte Vorrichtungen mit proprietären Zwischenverbindungen, standardisierten Zwischenverbindungen (z.B. PCIe) oder einer Kombination derselben verbunden sein. In mindestens einer Ausführungsform sind eine oder mehrere Komponenten eines Systems 1300 unter Verwendung von Compute Express Link (CXL)-Zwischenverbindungen verbunden.

[0169] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform kann in dem System von **Fig. 13** Inferenzierungs- und/oder Trainingslogik 915 zum Inferenzieren oder Vorhersagen von Operationen verwendet werden, die mindestens teilweise auf Gewichtungsparemtern basieren, die unter Verwendung von Trainingsoperationen für neuronale Netzwerke, Funktionen und/oder Architekturen neuronaler Netzwerke oder hier beschriebenen Anwendungsfällen neuronaler Netzwerke berechnet wurden.

[0170] **Fig. 14** ist ein Blockdiagramm, das eine elektronische Vorrichtung 1400 zur Verwendung eines Prozessors 1410 darstellt, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann die elektronische Vorrichtung 1400 beispielsweise und ohne darauf beschränkt zu sein ein Notebook, ein turmförmig aufgebauter bzw. Tower-Server, ein Rahmengestell- bzw. Rack-Server, ein Einschub- bzw. Blade-Server, ein tragbarer Rechner bzw. Laptop, ein Tischcomputer bzw. Desktop, ein Tablett, ein mobiles Gerät, ein Telefon, ein eingebetteter bzw. Embedded-Computer oder jede andere geeignete elektronische Vorrichtung sein.

[0171] In mindestens einer Ausführungsform kann die elektronische Vorrichtung 1400, ohne darauf beschränkt zu sein, einen Prozessor 1410 umfassen, der kommunikativ mit einer beliebigen geeigneten Anzahl oder Art von Komponenten, Peripheriegeräten, Modulen oder Vorrichtungen gekoppelt ist. In mindestens einer Ausführungsform ist ein Prozessor 1410 über einen Bus oder eine Schnittstelle, wie beispielsweise einen I²C-Bus, einen System Management Bus („SMBus“), einen Low Pin Count (LPC)-Bus, eine serielle Peripherieschnittstelle („SPI“), einen High Definition Audio („HDA“)-Bus, einen Serial Advance Technology Attachment („SATA“)-Bus, einen Universal Serial Bus („USB“) (Versionen 1, 2, 3 usw.) oder einen Universal Asynchronous Receiver/Transmitter („UART“)-Bus gekoppelt. In mindestens einer Ausführungsform veranschaulicht **Fig. 14** ein System, welches miteinander verbundene Hardware-Vorrichtungen oder „Chips“ umfasst, während in anderen Ausführungsformen **Fig. 14** ein beispielhaftes System auf einem Chip („SoC“) darstellen kann. In mindestens einer Ausführungsform können in **Fig. 14** veranschaulichte Vorrichtungen mit proprietären Verbindungen, standardisierten Zwischenverbindungen (z.B. PCIe) oder einer Kombination davon verbunden sein. In mindestens einer Ausführungsform sind eine oder mehrere Komponenten von **Fig. 14** unter Verwendung von Compute Express Link (CXL)-Zwischenverbindungen miteinander verbunden.

[0172] In mindestens einer Ausführungsform kann **Fig. 14** eine Anzeige 1424, einen berührungsempfindlichen Bildschirm bzw. Touchscreen 1425, ein berührungsempfindliches Feld bzw. Touchpad 1430, eine Nahfeld-Kommunikationseinheit (Near Field Communications Unit; „NFC“) 1445, einen Sensor-Hub 1440, einen Thermosensor 1446, einen Express-Chipsatz („EC“) 1435, ein Trusted Platform Module („TPM“) 1438, BIOS/Firmware/Flash-Speicher („BIOS, FW-Flash“) 1422, einen DSP 1460, ein Laufwerk („SSD oder HDD“) 1412, wie beispielsweise eine Solid State Disk („SSD“) oder eine Festplatte („HDD“), eine Wireless Local Area Network Unit („WLAN“) 1450, eine Bluetooth-Einheit 1452, eine Wireless Wide Area Network Unit („WWAN“) 1456, ein Global Positioning System (GPS) 1455, eine Kamera („USB 3.0 Camera“) 1454, wie beispielsweise eine USB 3.0-Kamera, oder eine Low Power Double Data Rate („LPDDR“)-Speichereinheit („LPDDR3“) 1415, die beispielsweise in einem LPDDR3-Standard implementiert ist, umfassen. Diese Komponenten können jeweils in beliebig geeigneter Art und Weise implementiert sein.

[0173] In mindestens einer Ausführungsform können andere Komponenten mit einem Prozessor 1410 durch oben erläuterte Komponenten kommunikativ gekoppelt sein. In mindestens einer Ausführungsform können ein Beschleunigungssensor 1441, ein Umgebungslichtsensor (Ambient Light Sensor; „ALS“) 1442, ein Kompass 1443 und ein Gyroskop 1444 kommunikativ mit einem Sensor-Hub 1440 gekoppelt sein. In mindestens einer Ausführungsform können ein Thermosensor 1439, ein Lüfter 1437, eine Tastatur 1436 und ein Touchpad 1430 kommunikativ mit einem EC 1435 gekoppelt sein. In mindestens einer Ausführungsform können Lautsprecher 1463, Kopfhörer 1464 und ein Mikrofon („mic“) 1465 kommunikativ mit einer Audioeinheit („Audio-Codec und Class D-Amp“) 1462 gekoppelt sein, die wiederum kommunikativ mit einem DSP 1460 gekoppelt sein kann. In mindestens einer Ausführungsform kann eine Audioeinheit 1462, beispielsweise und ohne darauf beschränkt zu sein, einen Audio-Codierer/Decodierer („Codec“) und einen Klasse-D-Verstärker umfassen. In mindestens einer Ausführungsform kann eine SIM-Karte („SIM“) 1457 kommunikativ mit einer

WWAN-Einheit 1456 gekoppelt sein. In mindestens einer Ausführungsform können Komponenten, wie beispielsweise eine WLAN-Einheit 1450 und eine Bluetooth-Einheit 1452 sowie auch eine WWAN-Einheit 1456, in einem Formfaktor der nächsten Generation (Next Generation Form Faktor; „NGFF“) implementiert sein.

[0174] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915 in einem System von **Fig. 14** zum Inferenzieren oder Vorhersagen von Operationen verwendet werden, die mindestens teilweise auf Gewichtungsparametern basieren, die unter Verwendung von Trainingsoperationen für neuronale Netzwerke, Funktionen und/oder Architekturen neuronaler Netzwerke oder hier beschriebenen Anwendungsfällen neuronaler Netzwerke berechnet wurden.

[0175] **Fig. 15** veranschaulicht ein Computersystem 1500 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist ein Computersystem 1500 konfiguriert, um verschiedene Prozesse und Verfahren zu implementieren, die überall in dieser Offenbarung beschrieben sind.

[0176] In mindestens einer Ausführungsform umfasst ein Computersystem 1500, ohne darauf beschränkt zu sein, mindestens eine zentrale Verarbeitungseinheit („CPU“) 1502, die mit einem Kommunikationsbus 1510 verbunden ist, der unter Verwendung eines beliebigen geeigneten Protokolls, wie beispielsweise PCI („Peripheral Component Zwischenverbindung“), Peripheral Component Zwischenverbindung Express („PCI-Express“), AGP („Accelerated Graphics Port“), HyperTransport oder eines anderen Busses oder einem oder mehreren Punkt-zu-Punkt-Kommunikationsprotokoll(en) implementiert ist. In mindestens einer Ausführungsform umfasst ein Computersystem 1500, ohne darauf beschränkt zu sein, einen Hauptspeicher 1504 und eine Steuerlogik (die z.B. als Hardware, Software oder eine Kombination davon implementiert ist), und Daten werden in einem Hauptspeicher 1504 gespeichert, der die Form eines Direktzugriffsspeichers („RAM“) annehmen kann. In mindestens einer Ausführungsform stellt ein Netzwerkschnittstellen-Teilsystem („Netzwerkschnittstelle“) 1522 eine Schnittstelle zu anderen Rechenvorrichtungen und Netzwerken zum Empfangen von Daten von und Übertragen von Daten an andere Systeme von einem Computersystem 1500 bereit.

[0177] In mindestens einer Ausführungsform umfasst ein Computersystem 1500, ohne darauf beschränkt zu sein, Eingabevorrichtungen 1508, ein Parallelverarbeitungssystem 1512 und Anzeigevorrichtungen 1506, die unter Verwendung einer herkömmlichen Kathodenstrahlröhre („CRT“), Flüssigkristallanzeigen („LCD“), Leuchtdioden („LED“), Plasmaanzeigen oder anderer geeigneter Anzeigetechnologien implementiert sein können. In mindestens einer Ausführungsform wird eine Benutzereingabe von Eingabegeräten 1528, wie beispielsweise einer Tastatur, einer Maus, einem Touchpad, einem Mikrofon und mehr empfangen. In mindestens einer Ausführungsform kann jedes von vorhergehenden Modulen auf einer einzigen Halbleiterplattform gelegen sein, um ein Verarbeitungssystem zu bilden.

[0178] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915 in einem System von **Fig. 15** zum Inferenzieren oder Vorhersagen von Operationen verwendet werden, die mindestens teilweise auf Gewichtungsparametern basieren, die unter Verwendung von Trainingsoperationen für neuronale Netzwerke, Funktionen und/oder Architekturen neuronaler Netzwerke oder hier beschriebenen Anwendungsfällen neuronaler Netzwerke berechnet wurden.

[0179] **Fig. 16** veranschaulicht ein Computersystem 1600 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform umfasst ein Computersystem 1600, ohne darauf beschränkt zu sein, einen Computer 1610 und einen USB-Stick 1620. In mindestens einer Ausführungsform kann der Computer 1610, ohne darauf beschränkt zu sein, eine beliebige Anzahl und Art eines(von) Prozessors(en) (nicht gezeigt) und einen Speicher (nicht gezeigt) umfassen. In mindestens einer Ausführungsform umfasst der Computer 1610, ohne darauf beschränkt zu sein, einen Server, eine Cloud-Instanz, einen Laptop und einen Desktop-Computer.

[0180] In mindestens einer Ausführungsform umfasst ein USB-Stick 1620, ohne darauf beschränkt zu sein, eine Verarbeitungseinheit 1630, eine USB-Schnittstelle 1640 und eine USB-Schnittstellenlogik 1650. In mindestens einer Ausführungsform kann eine Verarbeitungseinheit 1630 ein beliebiges Anweisungsausführungssystem, Einrichtung oder Vorrichtung sein, das/die in der Lage ist, Anweisungen auszuführen. In mindestens

einer Ausführungsform kann eine Verarbeitungseinheit 1630, ohne darauf beschränkt zu sein, eine beliebige Anzahl und Art von Verarbeitungskernen umfassen (nicht gezeigt). In mindestens einer Ausführungsform umfasst ein Verarbeitungskern 1630 eine anwendungsspezifische integrierte Schaltung („ASIC“), die optimiert ist, um jede Menge und Art von Operationen im Zusammenhang mit maschinellem Lernen durchzuführen. Beispielsweise ist in mindestens einer Ausführungsform eine Verarbeitungseinheit 1630 eine Tensor-Verarbeitungseinheit („TPC“), die für die Durchführung von Inferenzoperationen des maschinellen Lernens optimiert ist. In mindestens einer Ausführungsform ist eine Verarbeitungseinheit 1630 eine Visionsverarbeitungseinheit („VPU“), die optimiert ist, um Machine-Vision- und Machine-Learning-Inferenzoperationen durchzuführen.

[0181] In mindestens einer Ausführungsform kann eine USB-Schnittstelle 1640 jede beliebige Art von USB-Verbinde oder USB-Buchse sein. Beispielsweise ist in mindestens einer Ausführungsform eine USB-Schnittstelle 1640 eine USB 3.0 Typ-C Buchse für Daten und Leistung. In mindestens einer Ausführungsform ist eine USB-Schnittstelle 1640 ein USB 3.0 Typ-A-Verbinde. In mindestens einer Ausführungsform kann eine USB-Schnittstellenlogik 1650 jede beliebige Menge und Art von Logik umfassen, die einer Verarbeitungseinheit 1630 ermöglicht, über einen USB-Verbinde 1640 mit Vorrichtungen (z.B. einem Computer 1610) zu kommunizieren.

[0182] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform kann in dem System von **Fig. 16** die Inferenzierungs- und/oder Trainingslogik 915 zum Inferenzieren oder Vorhersagen von Operationen verwendet werden, die mindestens teilweise auf Gewichtungparametern basieren, die unter Verwendung von Trainingsoperationen für neuronale Netzwerke, Funktionen und/oder Architekturen neuronaler Netzwerke oder hier beschriebenen Anwendungsfällen neuronaler Netzwerke berechnet wurden.

[0183] **Fig. 17** veranschaulicht beispielhafte integrierte Schaltungen und zugehörige Graphikprozessoren, die unter Verwendung eines oder mehrerer IP-Kerne gemäß verschiedenen hier beschriebenen Ausführungsformen hergestellt werden können. Zusätzlich zu dem, was veranschaulicht ist, können andere Logik und Schaltungen in mindestens einer Ausführungsform enthalten sein, einschließlich zusätzlicher Grafikprozessoren/-kerne, Peripherieschnittstellen-Controller oder Allzweck-Prozessorkerne.

[0184] **Fig. 17** ist ein Blockdiagramm, das beispielhaft eine integrierte Schaltung 1700 eines Systems auf einem Chip (SOC) veranschaulicht, die unter Verwendung einer oder mehrere IP-Kerne hergestellt werden kann, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform umfasst eine integrierte Schaltung 1700 einen oder mehrere Anwendungsprozessoren 1705 (e.g., CPUs), mindestens einen Graphikprozessor 1710, und kann zusätzlich einen Bildprozessor 1715 und/oder einen Videoprozessor 1720 umfassen, von denen jeder ein modularer IP-Kern sein kann. In mindestens einer Ausführungsform umfasst eine integrierte Schaltung 1700 eine Peripherie- oder Buslogik, die einen USB-Controller 1725, einen UART-Controller 1730, einen SPI/SDIO-Controller 1735, und einen I2S/I2C-Controller 1740 umfasst. In mindestens einer Ausführungsform kann eine integrierte Schaltung 1700 eine Anzeigevorrichtung 1745 enthalten, die mit einem oder mehreren HDMI-Controllern (High-Definition Multimedia Interface) 1750 und einer MIPI-Anzeigeschnittstelle 1755 (Mobile Industry Processor Interface) verbunden ist. In mindestens einer Ausführungsform kann ein Speicher durch ein Flash-Speicher-Subsystem 1760 bereitgestellt werden, das einen Flash-Speicher und einen Flash-Speicher-Controller enthält. In mindestens einer Ausführungsform kann eine Speicherschnittstelle über einen Speicher-Controller 1765 für einen Zugriff auf SDRAM- oder SRAM-Speichervorrichtungen bereitgestellt werden. In mindestens einer Ausführungsform umfassen einige integrierte Schaltungen zusätzlich eine eingebettete Sicherheits-Engine 1770.

[0185] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915 in einer integrierten Schaltung zum Inferenzieren oder Voraussagen von Operationen verwendet werden, die mindestens teilweise auf Gewichtungparameter basieren, die unter Verwendung von neuronalen Netzwerktrainingsoperationen, neuronalen Netzwerkfunktionen und/oder Architekturen, oder hier beschriebenen Anwendungsfällen eines neuronalen Netzwerks berechnet wurden.

[0186] Fig. 18A veranschaulicht eine beispielhafte Architektur, in der mehrere GPUs 1810-1813 kommunikativ mit mehreren Mehrkern-Prozessoren 1805-1806 über Hochgeschwindigkeitsverbindungen 1840-1843 (z.B. Busse, Punkt-zu-Punkt-Zwischenverbindungen usw.) gekoppelt ist. In einer Ausführungsform unterstützen die Hochgeschwindigkeitsverbindungen 1840-1843 einen Kommunikationsdurchsatz von 4GB/s, 30GB/s, 80GB/s oder höher. Verschiedene Zwischenverbindungsprotokolle können verwendet werden, einschließlich, jedoch nicht beschränkt auf, PCIe 4.0 oder 5.0 und NVLink 2.0.

[0187] Außerdem sind in einer Ausführungsform zwei oder mehr GPUs 1810-1813 über Hochgeschwindigkeitsverbindungen 1829-1830 miteinander verbunden, die unter Verwendung von gleicher oder unterschiedlicher Protokollen/Verbindungen als diejenigen implementiert sein können, die für die Hochgeschwindigkeitsverbindungen 1840-1843 verwendet werden. Auf ähnliche Weise können zwei oder mehr Mehrkern-Prozessoren 1805-1806 über eine Hochgeschwindigkeitsverbindung 1828 verbunden sein, welche symmetrische Mehrkern-Prozessor-(SMP)-Busse sein können, die mit 20GB/s, 30GB/s, 120 GB/s oder mehr arbeiten. Alternativ kann die gesamte Kommunikation zwischen in **Fig. 18A** gezeigten, verschiedenen Systemkomponenten unter Verwendung derselben Protokolle/Verbindungen (z.B. über eine gemeinsame Zwischenverbindungsarchitektur) ausgeführt werden.

[0188] In einer Ausführungsform ist jeder Mehrkern-Prozessor 1805-1806 mit einem Prozessorspeicher 1801-1802 über Speicher-Zwischenverbindungen 1826-1827 jeweils gekoppelt, und jede GPU 1810-1813 ist jeweils über GPU-Speicher-Zwischenverbindungen 1850-1853 kommunikativ mit dem GPU-Speicher 1820-1823 gekoppelt. Speicher-Zwischenverbindungen 1826-1827 und 1850-1853 können dieselben oder unterschiedliche Speicherzugriffstechnologien nutzen. Beispielsweise und nicht beschränkend können die Prozessorspeicher 1801-1802 und GPU-Speicher 1820-1823 flüchtige Speicher wie Dynamic Random Access-Speicher (DRAMs) (einschließlich gestapelter DRAMs), Graphik-DDR-SDRAM (GDDR) (z.B. GDDR5, GDDR6) oder Speicher mit hoher Bandbreite (High Bandwidth Memory; „HBM“) und/oder nichtflüchtige Speicher wie 3D XPoint oder Nano-Ram sein. In einer Ausführungsform kann ein Teil der Prozessorspeicher 1801-1802 flüchtiger Speicher und ein anderer Teil nichtflüchtiger Speicher sein (z.B. unter Verwendung einer zweistufigen Speicher (2LM)-Hierarchie).

[0189] Wie nachstehend beschrieben können, obwohl verschiedene Prozessoren 1805-1806 und GPUs 1810-1813 physikalisch mit jeweils einem bestimmten Speicher 1801-1802, 1820 gekoppelt sein, eine vereinheitlichte Speicherarchitektur kann implementiert sein, in welcher ein gleicher virtueller Systemadressraum (auch als „effektiver Adressraum“ bezeichnet) auf verschiedene physikalische Speicher verteilt ist. Beispielsweise können Prozessorspeicher 1801-1802 jeweils 64GB Systemspeicheradressraum und GPU-Speicher 1820-1823 jeweils 32GB Systemspeicheradressraum umfassen, welches insgesamt zu 256GB adressierbarem Speicher resultiert.

[0190] Fig. 18B veranschaulicht zusätzliche Einzelheiten für eine Zwischenverbindung zwischen einem Mehrkern-Prozessor 1807 und einem Graphikbeschleunigungsmodul 1846 gemäß einer beispielhaften Ausführungsform. Das Graphikbeschleunigungsmodul 1846 kann ein oder mehrere GPU-Chips umfassen, die auf einer Verdrahtungskarte integriert sind, welche über eine Hochgeschwindigkeitsverbindung 1840 (z.B. ein PCI-Bus, NVLink usw.) mit dem Prozessor 1807 gekoppelt ist. Alternativ kann das Graphikbeschleunigungsmodul 1846 auf einem gleichen Gehäuse oder Chip als Prozessor 1807 integriert sein.

[0191] In mindestens einer Ausführungsform umfasst der veranschaulichte Prozessor 1807 mehrere Kernen 1860A-1860D jeweils mit einem Translation-Lookaside-Puffer („TLB“) 1861A-1861D und einem oder mehrere Caches 1862A-1862D umfasst. In mindestens einer Ausführungsform können Kerne 1860A-1860D verschiedene andere Komponenten zur Ausführung von Anweisungen und zur Verarbeitung von Daten umfassen, welche nicht veranschaulicht sind. Caches 1862A-1862D können Level-1 (L1)- und Level-2 (L2)-Caches umfassen. Außerdem können ein oder mehrere gemeinsam genutzte Caches 1856 in Caches 1862A-1862D umfasst sein und von Sätzen von Kernen 1860A-1860D gemeinsam genutzt werden. Eine Ausführungsform des Prozessors 1807 umfasst beispielsweise 24 Kerne jeweils mit seinem eigenen L1-Cache, zwölf gemeinsam genutzten L2-Caches und zwölf gemeinsam genutzten L3-Caches. In dieser Ausführungsform werden ein oder mehrere L2- und L3-Caches von zwei benachbarten Kernen gemeinsam genutzt. In mindestens einer Ausführungsform verbinden sich der Prozessor 1807 und das Graphikbeschleunigungsmodul 1846 mit dem Systemspeicher 1814, welcher die Prozessorspeicher 1801-1802 von **Fig. 18A** umfassen kann.

[0192] Kohärenz wird für in verschiedenen Caches 1862A-1862D, 1856 und dem Systemspeicher 1814 gespeicherten Daten und Anweisungen über Inter-Kern-Kommunikation über einen Kohärenzbus 1864 aufrechterhalten. Beispielsweise kann jedem Cache eine Cache-Kohärenz-Logik/Schaltung zugeordnet sein,

um als Antwort auf erfasste Lese- oder Schreiboperationen auf bestimmte Cache-Zeilen mit einem Kohärenzbus 1864 zu kommunizieren. In mindestens einer Ausführungsform ist ein Cache-Beobachtungsprotokoll über den Kohärenzbus 1864 implementiert, um Cache-Zugriffe zu beobachten.

[0193] In einer Ausführungsform koppelt eine Proxy-Schaltung (PROXY) 1825 das Graphikbeschleunigungsmodul 1846 kommunikativ mit dem Kohärenzbus 1864, so dass das Graphikbeschleunigungsmodul 1846 als ein Gleichgestelltes der Kerne 1860A-1860D an einem Cache-Kohärenzprotokoll teilnehmen kann. Insbesondere bietet eine Schnittstelle (INTF) 1835 Konnektivität zur Proxy-Schaltung 1825 über die Hochgeschwindigkeitsverbindung 1840 und eine Schnittstelle (INTF) 1837 verbindet das Graphikbeschleunigungsmodul 1846 mit der Hochgeschwindigkeitsverbindung 1840.

[0194] In einer Implementierung stellt eine Beschleuniger-Integrationsschaltung 1836 Cache-Verwaltungs-, Speicherzugriffs-, Kontextverwaltungs- und Unterbrechungsverwaltungsdienste im Namen mehrerer Graphikverarbeitungs-Engines (1831, 1832, N) des Graphikbeschleunigungsmoduls 1846 bereit. In mindestens einer Ausführungsform können die Graphikverarbeitungs-Engines (1831, 1832, N) jeweils eine separate Graphikverarbeitungseinheit (GPU) umfassen. Alternativ können die Graphikverarbeitungs-Engines (1831, 1832, N) verschiedene Arten von Graphikverarbeitungs-Engines innerhalb einer GPU umfassen, wie beispielsweise Graphikausführungseinheiten, Medienverarbeitungs-Engines (z.B. Video-Codierer/Decodierer), Abtaster und Blit-Engines. In mindestens einer Ausführungsform kann das Graphikbeschleunigungsmodul 1846 eine GPU mit mehreren Graphikverarbeitungs-Engines (1831, 1832, N) sein, oder die Graphikverarbeitungs-Engines (1831, 1832, N) können individuelle GPUs sein, die in bzw. auf einem gemeinsamen Gehäuse, einer gemeinsamen Verdrahtungskarte oder einem gemeinsamen Chip integriert sind.

[0195] In einer Ausführungsform umfasst die Beschleuniger-Integrationsschaltung 1836 eine Speicherverwaltungseinheit (MMU) 1839 zum Durchführen verschiedener Speicherverwaltungsfunktionen, wie beispielsweise Speicherübersetzungen von virtuell zu physikalisch (auch als Speicherübersetzungen von effektiv zu real bezeichnet) und Speicherzugriffsprotokollen zum Zugriff auf den Systemspeicher 1814. Die MMU 1839 kann auch einen Translation-Lookaside-Puffer (TLB) (nicht gezeigt) zum Zwischenspeichern von virtuellen/effektiven zu physikalischen/realen Adressübersetzungen umfassen. In einer Implementierung speichert ein Cache 1838 Befehle und Daten für effizienten Zugriff durch Graphikverarbeitungs-Engines (1831-1832, N). In einer Ausführungsform werden die im Cache 1838 und Graphikspeichern (1833-1834, M) gespeicherten Daten mit Kerncaches (1862A-1862D, 1856) und dem Systemspeicher 1814 kohärent gehalten. Wie erwähnt, kann dies über eine Proxy-Schaltung 1825 im Namen des Caches 1838 und der Speicher (1833-1834, M) erreicht werden (z.B. Senden von Aktualisierungen an den Cache 1838 mit Bezug auf Modifikationen/Zugriffe auf Cache-Zeilen auf Prozessor-Caches 1862A-1862D, 1856 und Empfangen von Aktualisierungen von dem Cache 1838) erreicht werden.

[0196] Ein Satz von Registern 1845 speichert Kontextdaten für Threads, die von den Graphikverarbeitungs-Engines (1831-1832, N) ausgeführt werden, und eine Kontextverwaltungsschaltung 1848 verwaltet Thread-Kontexte. Beispielsweise kann die Kontextverwaltungsschaltung 1848 Speicherungs- und Wiederherstellungs-Operationen durchführen, um Kontexte verschiedener Threads während Kontextwechseln zu speichern und wiederherzustellen (z.B., wenn ein erster Thread gesichert wird und ein zweiter Thread gespeichert wird, so dass ein zweiter Thread von einer Graphikverarbeitungs-Engine ausgeführt werden kann). Beispielsweise kann eine Kontextverwaltungsschaltung 1848 bei einem Kontextwechsel aktuelle Registerwerte in einen bestimmten Bereich in dem Speicher speichern (der z.B. durch einen Kontextzeiger identifiziert wird). Bei der Rückkehr zu einem Kontext kann sie dann die Registerwerte wiederherstellen. In einer Ausführungsform empfängt und verarbeitet eine Unterbrechungsverwaltungsschaltung (INTRPT MGMT) 1847 von Systemvorrichtungen empfangene Unterbrechungen.

[0197] In einer Implementierung werden virtuelle/effektive Adressen von einer Graphikverarbeitungs-Engine 1831 in reale/physikalische Adressen in dem Systemspeicher 1814 durch die MMU 1839 übersetzt. Eine Ausführungsform der Beschleunigerintegrationsschaltung 1836 unterstützt mehrere (z.B. 4, 8, 16) Graphikbeschleunigermodule 1846 und/oder andere Beschleunigungsvorrichtungen. Das Graphikbeschleunigermodul 1846 kann einer einzelnen Anwendung zugeordnet sein, die auf einem Prozessor 1807 ausgeführt wird, oder kann zwischen mehreren Anwendungen gemeinsam genutzt werden. In einer Ausführungsform wird eine virtualisierte Graphikausführungsumgebung präsentiert, in welcher Ressourcen der Graphikverarbeitungs-Engines (1831-1832, N) von mehreren Anwendungen oder virtuellen Maschinen (VMs) gemeinsam genutzt werden. In mindestens einer Ausführungsform können Ressourcen in „Scheiben“ bzw. „Slices“ unterteilt sein, die basierend auf Verarbeitungsanforderungen und Prioritäten, die mit VMs und/oder Anwendungen verbunden sind, verschiedenen VMs und/oder Anwendungen zugeteilt werden.

[0198] In mindestens einer Ausführungsform fungiert die Beschleuniger-Integrationsschaltung 1836 als eine Brücke zu einem System für das Graphikbeschleunigungsmodul 1846 und stellt Adressübersetzungs- und Systemspeicher-Cache-Dienste bereit. Außerdem kann die Beschleuniger-Integrationsschaltung 1836 Virtualisierungseinrichtungen für einen Host-Prozessor bereitstellen, um die Virtualisierung von Graphikverarbeitungs-Engines 1831-1832, Interrupts und Speicherverwaltung zu verwalten.

[0199] Weil Hardwareressourcen der Graphikverarbeitungs-Engines (1831-1832, N) explizit auf einen realen Adressraum abgebildet werden, der von dem Host-Prozessor 1807 gesehen wird, kann jeder Host-Prozessor diese Ressourcen direkt unter Verwendung eines effektiven Adresswerts direkt adressieren. In mindestens einer Ausführungsform ist eine Funktion der Beschleunigerintegrationsschaltung 1836 die physikalische Trennung der Graphikverarbeitungs-Engines (1831-1832, N), so dass sie einem System als unabhängige Einheiten erscheinen.

[0200] In mindestens einer Ausführungsform sind ein oder mehrere Graphikspeicher (1833-1834, N) mit jeder der Graphikverarbeitungs-Engines (1831-1832, N) gekoppelt. Graphikspeicher (1833-1834, N) speichern Anweisungen und Daten, die von jeder der Graphikverarbeitungs-Engines (1831-1832, N) verarbeitet werden. Graphikspeicher (1833-1834, N) können flüchtige Speicher, wie beispielsweise DRAMs (einschließlich gestapelter DRAMs), GDDR-Speicher (z.B. GDDR5, GDDR6) oder HBM sein, und/oder können nicht-flüchtige Speicher wie beispielsweise 3D XPoint oder Nano-Ram sein.

[0201] In einer Ausführungsform werden, um den Datenverkehr über die Hochgeschwindigkeitsverbindung 1840 zu verringern, Biasing-Verfahren verwendet, um sicherzustellen, dass Daten, die in den Graphikspeichern (1833-1834, N) gespeichert sind, Daten sind, welche am häufigsten von den Graphikverarbeitungs-Engines (1831-1832, N) verwendet werden und vorzugsweise nicht (wenigstens nicht häufig) von den Kernen 1860A-1860D verwendet werden. Auf ähnliche Weise versucht ein Biasing-Mechanismus, die von den Kernen (und vorzugsweise nicht von den Graphikverarbeitungs-Engines (1831-1832, N)) benötigten Daten in den Caches 1862A-1862D, 1856 und dem Systemspeicher 1814 zu halten.

[0202] **Fig. 18C** veranschaulicht eine weitere beispielhafte Ausführungsform, bei welcher die Beschleunigerintegrationsschaltung 1836 in dem Prozessor 1807 integriert ist. In dieser Ausführungsform kommunizieren Graphikverarbeitungs-Engines (1831-1832, N) direkt über die Hochgeschwindigkeitsverbindung 1840 mit der Beschleunigerintegrationsschaltung 1836 über die Schnittstelle 1837 und Schnittstelle 1835 (die wiederum eine beliebige Form von Bus- oder Schnittstellenprotokoll verwenden kann). Die Beschleuniger-Integrationsschaltung 1836 kann ähnliche Operationen wie diejenigen durchführen, die in Bezug auf **Fig. 18B** beschrieben sind, jedoch potenziell mit einem höheren Durchsatz, da sie in unmittelbarer Nähe zum Kohärenzbus 1864 und Caches 1862A-1862D, 1856 liegt. Eine Ausführungsform unterstützt unterschiedliche Programmiermodelle, die ein dediziertes Prozessprogrammiermodell (keine Virtualisierung des Graphikbeschleunigungsmoduls) und gemeinsam genutzte Programmiermodelle (mit Virtualisierung) umfassen, welche Programmiermodelle umfassen können, die von der Beschleunigerintegrationsschaltung 1836 gesteuert werden, und Programmiermodelle, welche von dem Graphikbeschleunigungsmodul 1846 gesteuert werden.

[0203] In mindestens einer Ausführungsform sind die Graphikverarbeitungs-Engines (1831-1832, N) für eine einzelne Anwendung oder einen einzelnen Prozess unter einem einzigen Betriebssystem dediziert. In mindestens einer Ausführungsform kann eine einzelne Anwendung andere Anwendungsanforderungen an die Graphikverarbeitungs-Engines (1831-1832, N) leiten, um eine Virtualisierung innerhalb einer VM/Partition bereitstellen.

[0204] In mindestens einer Ausführungsform können die Graphikverarbeitungs-Engines (1831-1832, N) von mehreren VM-/Anwendungs-Partitionen gemeinsam genutzt werden. In mindestens einer Ausführungsform können gemeinsam genutzte Modelle einen Systemhypervisor verwenden, um Graphikverarbeitungs-Engines (1831-1832, N) zu virtualisieren, um einen Zugriff durch jedes Betriebssystem zu ermöglichen. Für Systeme mit einer einzigen Partition ohne einen Hypervisor sind die Graphikverarbeitungs-Engines (1831-1832, N) im Besitz eines Betriebssystems. In mindestens einer Ausführungsform kann ein Betriebssystem die Graphikverarbeitungs-Engines (1831-1832, N) virtualisieren, um einen Zugriff auf jeden Prozess oder jede Anwendung zu ermöglichen.

[0205] In mindestens einer Ausführungsform wählt das Graphikbeschleunigungsmodul 1846 oder eine einzelne Graphikverarbeitungs-Engine (1831-1832, N) ein Prozesselement unter Verwendung eines Prozess-Handle aus. In einer Ausführungsform werden Prozesselemente in dem Systemspeicher 1814 gespeichert und sind unter Verwendung der hier beschriebenen Techniken zur Übersetzung von effektiven Adressen in

echte Adressen adressierbar. In mindestens einer Ausführungsform kann ein Prozesshandle ein implementierungsspezifischer Wert sein, der einem Hostprozess bei der Registrierung seines Kontextes mit der Graphikverarbeitungs-Engine (1831-1832, N) bereitgestellt wird (d.h. Aufruf einer Systemsoftware, um ein Prozesselement zu einer verknüpften Liste von Prozesselementen hinzuzufügen). In mindestens einer Ausführungsform können untere 16 Bits eines Prozess-Handle ein Versatz des Prozesselements innerhalb einer verknüpften Liste von Prozesselementen sein.

[0206] Fig. 18D veranschaulicht ein beispielhaftes Beschleunigerintegrations-Slice 1890. Wie hier verwendet, umfasst ein „Slice“ einen bestimmten Teil der Verarbeitungsressourcen der Beschleunigerintegrations-schaltung 1836. Ein anwendungseffektiver Adressraum 1882 in dem Systemspeicher 1814 speichert Prozesselemente 1883. In einer Ausführungsform werden Prozesselemente 1883 als Antwort auf GPU-Aufrufe 1881 aus den auf dem Prozessor 1807 ausgeführten Anwendungen 1880 gespeichert. Ein Prozesselement 1883 enthält den Prozesszustand für die entsprechende Anwendung 1880. Ein in dem Prozesselement 1883 enthaltener Arbeitsdeskriptor (WD) 1884 kann ein einzelner von einer Anwendung angeforderter Auftrag bzw. Job sein oder einen Zeiger auf eine Warteschlange von Aufträgen enthalten. In mindestens einer Ausführungsform ist der WD 1884 ein Zeiger auf eine Auftragsanforderungswarteschlange in dem wirksamem Adressraum 1882 einer Anwendung.

[0207] Ein Graphikbeschleunigungsmodul 1846 und/oder einzelne Graphikverarbeitungs-Engines (1831-1832, N) können von allen oder einer Teilmenge von Prozessen in einem System gemeinsam genutzt werden. In mindestens einer Ausführungsform kann eine Infrastruktur zum Einrichten des Prozesszustands und zum Senden einer WD 1884 an ein Graphikbeschleunigungsmodul 1846 zum Starten eines Auftrags in einer virtualisierten Umgebung enthalten sein.

[0208] In mindestens einer Ausführungsform ist ein dediziertes Prozessprogrammierungsmodell implementierungsspezifisch. In diesem Modell gehört einem einzelnen Prozess das Graphikbeschleunigungsmodul 1846 oder eine einzelne Graphikverarbeitungs-Engine 1831. Weil das Graphikbeschleunigungsmodul 1846 einem einzelnen Prozess gehört, initialisiert ein Hypervisor die Beschleunigerintegrationsschaltung 1836 für eine besitzende Partition und ein Betriebssystem die Beschleunigerintegrationsschaltung 1836 für einen besitzenden Prozess, wenn das Graphikbeschleunigungsmodul 1846 zugewiesen wird.

[0209] Im Betrieb holt eine WD-Holeinheit 1891 in dem Beschleunigerintegrations-Slice 1890 den nächsten WD 1884, welcher eine Angabe der von einer oder mehreren Graphikverarbeitungs-Engines des Graphikbeschleunigungsmoduls 1846 zu verrichtenden Arbeiten umfasst. Daten aus dem WD 1884 können in Registern 1845 gespeichert und von der MMU 1839, der Unterbrechungsverwaltungsschaltung 1847 und/oder der Kontextverwaltungsschaltung (CONTEXT MGMT) 1848, wie veranschaulicht, verwendet werden. Eine Ausführungsform der MMU 1839 beinhaltet beispielsweise eine Segment-/Seiten-Durchlauf-Schaltungsanordnung zum Zugreifen auf Segment-/Seitentabellen 1886 innerhalb eines virtuellen Adressraums 1885 des Betriebssystems (OS). Die Unterbrechungsverwaltungsschaltung 1847 kann Unterbrechungsereignisse 1892 verarbeiten, die von dem Graphikbeschleunigungsmodul 1846 empfangen werden. Bei der Durchführung von Graphikoperationen wird eine von einer Graphikverarbeitungs-Engine (1831-1832, N) erzeugte effektive Adresse 1893 von der MMU 1839 in eine reale Adresse übersetzt.

[0210] In einer Ausführungsform wird ein und derselbe Satz von Registern 1845 für jede Graphikverarbeitungs-Engine (1831-1832, N) und/oder das Graphikbeschleunigungs-Modul 1846 dupliziert und kann von einem Hypervisor oder Betriebssystem initialisiert werden. Jedes dieser duplizierten Register kann in einem Beschleunigerintegrations-Slice 1890 umfasst sein. Beispielhafte Register, die von einem Hypervisor initialisiert werden können, werden in Tabelle 1 gezeigt.

Tabelle 1 - Hypervisor-initialisierte Register

1	Slice-Steuerregister
2	Zeiger auf reale Adresse (RA) des Bereichs geplanter Prozesse
3	Register für die Übersteuerung von Berechtigungsmasken
4	Versatz Unterbrechungsvektortabelleneintrag
5	Grenze Unterbrechungsvektortabelleneintrag
6	Zustandsregister
7	ID der logischen Partition

8	Zeiger auf reale Adresse (RA) des Hypervisor-Beschleunigerauslastungseintrags
9	Speicherbeschreibungsregister

[0211] Beispielhafte Register, die von einem Betriebssystem initialisiert werden können, werden in Tabelle 2 gezeigt.

Tabelle 2 - Betriebssystem-initialisierte Register

1	Prozess- und Thread-Identifikation
2	Zeiger auf effektive Adresse (EA) des Kontext-Sichern/Wiederherstellens
3	Zeiger auf virtuelle Adresse (VA) des Beschleunigerauslastungseintrags
4	Zeiger auf virtuelle Adresse (VA) der Speichersegmenttabelle
5	Berechtigungsmaske
6	Arbeitsdeskriptor

[0212] In einer Ausführungsform ist jeder WD 1884 spezifisch für ein bestimmtes Graphikbeschleunigungsmodul 1846 und/oder Graphikverarbeitungs-Engines (1831-1832, N). Er enthält alle Informationen, die von einer Graphikverarbeitungs-Engine (1831-1832, N) zum Verrichten von Arbeit benötigt werden, oder er kann ein Zeiger auf einen Speicherort sein, bei dem eine Anwendung eine Befehlswarteschlange für abzuschließende Arbeiten eingerichtet hat.

[0213] **Fig. 18E** veranschaulicht zusätzliche Einzelheiten für eine beispielhafte Ausführungsform eines gemeinsam genutzten Modells. Diese Ausführungsform umfasst einen Hypervisor-Real-Adressraum 1898, in welchem eine Prozesselementliste 1899 gespeichert ist. Der Hypervisor-Real-Adressraum 1898 ist über einen Hypervisor 1896 zugänglich, welcher Graphikbeschleunigungsmodul-Engines für das Betriebssystem 1895 virtualisiert.

[0214] In mindestens einer Ausführungsform ermöglichen gemeinsam genutzte Programmiermodelle für alle oder eine Teilmenge von Prozessen in einem System die Verwendung eines Graphikbeschleunigungsmoduls 1846. Es gibt zwei Programmiermodelle, bei denen das Graphikbeschleunigungsmodul 1846 von mehreren Prozessen und Partitionen gemeinsam genutzt wird: zeit-sliced gemeinsam genutzt und graphikorientiert gemeinsam genutzt.

[0215] In diesem Modell besitzt der Systemhypervisor 1896 das Graphikbeschleunigungsmodul 1846 und stellt seine Funktion allen Betriebssystemen 1895 zur Verfügung. Damit ein Graphikbeschleunigungsmodul 1846 eine Virtualisierung durch den Systemhypervisor 1896 unterstützen kann, kann das Graphikbeschleunigungsmodul 1846 das folgende beachten: (1) Eine Auftragsanforderung einer Anwendung muss autonom sein (d.h. der Zustand muss zwischen Aufträgen nicht aufrechterhalten werden) oder das Graphikbeschleunigungsmodul 1846 muss einen Mechanismus zum Speichern und Wiederherstellen von Kontext bereitstellen; (2) die Auftragsanforderung einer Anwendung wird durch das Graphikbeschleunigungsmodul 1846 garantiert, um einen bestimmten Zeitraum abzuschließen einschließlich etwaiger Übersetzungsfehler, oder das Graphikbeschleunigungsmodul 1846 bietet eine Möglichkeit, der Verarbeitung eines Auftrags zuvorzukommen, (3) Dem Graphikbeschleunigungsmodul 1846 muss Fairness zwischen Prozessen garantiert werden, wenn es in einem gezielten, gemeinsam genutzten Programmiermodell arbeitet.

[0216] In mindestens einer Ausführungsform muss die Anwendung 1880 einen Systemaufruf des Betriebssystems 1895 mit einem Typ des Graphikbeschleunigungsmoduls 1846, einem Arbeitsdeskriptor (WD), einem Wert eines Berechtigungsmaskenregisters (AMR) und einem Zeiger auf einen Kontext-Speicher-/Wiederherstellungsbereich (CSRP) durchführen. In mindestens einer Ausführungsform beschreibt der Typ des Graphikbeschleunigungsmoduls 1846 eine Soll-Beschleunigungsfunktion für einen Systemaufruf. In mindestens einer Ausführungsform kann der Typ des Graphikbeschleunigungsmoduls 1846 ein systemspezifischer Wert sein. In mindestens einer Ausführungsform ist der WD speziell für das Graphikbeschleunigungsmodul 1846 formatiert und kann in einer Form eines Befehls des Graphikbeschleunigungsmoduls 1846, eines effektiven Adresszeigers auf eine benutzerdefinierte Struktur, eines effektiven Adresszeigers auf eine Befehlswarteschlange oder einer beliebigen anderen Datenstruktur zur Beschreibung der von dem Graphikbeschleunigungsmodul 1846 auszuführenden Arbeiten vorliegen. In einer Ausführungsform ist ein AMR-Wert ein AMR--

Zustand, der für einen aktuellen Prozess zu verwenden ist. In mindestens einer Ausführungsform ist ein an ein Betriebssystem übergebener Wert vergleichbar mit einer Anwendung, die ein AMR einstellt. Falls die Implementierungen der Beschleunigerintegrationsschaltung 1836 und des Graphikbeschleunigungsmoduls 1846 ein Benutzerberechtigungsmasken-Übersteuerungsregister (User Authority Mask Override Register; „UAMOR“) nicht unterstützen, kann ein Betriebssystem einen aktuellen UAMOR-Wert auf einen AMR-Wert anwenden, bevor es einen AMR in einem Hypervisor-Aufruf übergibt. Der Hypervisor 1896 kann optional einen aktuellen Berechtigungsmasken-Übersteuerungsregister (Authority Mask Override Register; AMOR)--Wert anwenden, bevor er ein AMR in das Prozesselement 1883 platziert. In mindestens einer Ausführungsform ist CSRP eines von Registern 1845, die eine effektive Adresse eines Bereichs im Adressraum 1882 einer Anwendung für das Graphikbeschleunigungsmodul 1846 zum Speichern und Wiederherstellen des Kontextzustands enthalten. Dieser Zeiger ist optional, falls zwischen Aufträgen kein Zustand gespeichert werden muss oder falls einem Auftrag zuvorgekommen wird. In mindestens einer Ausführungsform kann der Kontext-Speicherungs-/Wiederherstellungs-Bereich als fester Systemspeicher dienen.

[0217] Bei Empfangen eines Systemaufrufs kann das Betriebssystem 1895 verifizieren, dass die Anwendung 1880 registriert ist und die Berechtigung zur Verwendung des Graphikbeschleunigungsmoduls 1846 erhalten hat. Das Betriebssystem 1895 ruft dann den Hypervisor 1896 mit den in Tabelle 3 gezeigten Informationen auf.

Tabelle 3 - Parameter für Hypervisor-Aufruf durch Betriebssystem

1	Ein Arbeitsdeskriptor (WD)
2	Ein Berechtigungsmaskenregister (AMR)-Wert (potenziell maskiert) .
3	Ein Zeiger auf eine effektive Adresse (EA) des Kontext-Speicherungs-/Wiederherstellungs-Bereichs (CSRP)
4	Eine Prozess-ID (PID) und eine optionale Thread-ID (TID) .
5	Ein Zeiger auf eine virtuelle Adresse (VA) des Beschleunigerauslastungseintrags (AURP)
6	Zeiger auf virtuelle Adresse der Speichersegmenttabelle (SSTP)
7	Eine logische Unterbrechungsdienstnummer (LISN)

[0218] Bei Empfangen eines Hypervisoraufrufs verifiziert der Hypervisor 1896, dass das Betriebssystem 1895 registriert hat und berechtigt ist, das Graphikbeschleunigungsmodul 1846 zu verwenden. Der Hypervisor 1896 stellt dann das Prozesselement 1883 in eine verknüpfte Liste von Prozesselementen für einen Typ eines entsprechenden Graphikbeschleunigungsmodul 1846 ein. Ein Prozesselement kann in Tabelle 4 gezeigte Informationen umfassen.

Tabelle 4 - Informationen zu Prozesselementen

1	Ein Arbeitsdeskriptor (WD)
2	Ein Berechtigungsmaskenregister (AMR)-Wert (potenziell maskiert)
3	Ein Zeiger auf eine effektive Adresse (EA) des Kontext-Speicherungs-/Wiederherstellungs-Bereichs (CSRP)
4	Eine Prozess-ID (PID) und eine optionale Thread-ID (TID)
5	Ein Zeiger auf eine virtuelle Adresse (VA) des Beschleunigerauslastungseintrags (AURP)
6	Zeiger auf virtuelle Adresse der Speichersegmenttabelle (SSTP)
7	Eine logische Unterbrechungsdienstnummer (LISN)
8	Unterbrechungsvektortabelle, abgeleitet aus Hypervisor-Aufrufparametern
9	Ein Zustandsregister (SR)-Wert
10	Eine logische Partitions-ID (LPID)
11	Ein Zeiger auf eine reale Adresse (RA) des Hypervisor-Beschleunigerauslastungseintrags
12	Speicherdeskriptorregister (SDR)

[0219] In mindestens einer Ausführungsform initialisiert der Hypervisor mehrere Register 1845 des Beschleunigerintegrations-Slice 1890.

[0220] Wie in **Fig. 18F** veranschaulicht, wird in mindestens einer Ausführungsform ein vereinigter Speicher verwendet, der über einen gemeinsamen virtuellen Speicheradressraum adressierbar ist, der für den Zugriff auf physikalische Prozessorspeicher 1801-1802 und GPU-Speicher 1820-1823 verwendet wird. In dieser Implementierung verwenden Operationen, die auf GPUs 1810-1813 ausgeführt werden, ein und denselben virtuellen/effektiven Speicheradressraum, um auf die Prozessorspeicher 1801-1802 zuzugreifen und umgekehrt, wodurch die Programmierbarkeit vereinfacht wird. In einer Ausführungsform ist ein erster Abschnitt eines virtuellen/effektiven Adressraums dem Prozessorspeicher 1801, ein zweiter Abschnitt dem zweiten Prozessorspeicher 1802, ein dritter Abschnitt dem GPU-Speicher 1812 zugeteilt, und so weiter. In mindestens einer Ausführungsform wird dadurch ein gesamter virtueller/effektiver Speicherraum (manchmal auch als ein effektiver Adressraum bezeichnet) über jeden der Prozessorspeicher 1801-1802 und GPU-Speicher 1820-1823 hinweg verteilt, so dass jeder Prozessor oder jede GPU auf jeden physikalischen Speicher zugreifen kann, wobei eine virtuelle Adresse auf diesen Speicher abgebildet wird.

[0221] In einer Ausführungsform stellen die Bias/Kohärenzverwaltungs-Schaltungen 1894A-1894E innerhalb einer oder mehrerer MMUs 1839A-1839E die Cache-Kohärenz zwischen Caches eines oder mehrerer Host-Prozessoren (z.B. 1805) und den GPUs 1810-1813 sicher und implementieren Biasing-Techniken, die physikalische Speicher angeben, in welchen bestimmte Arten von Daten gespeichert werden sollten. Während in **Fig. 18F** mehrere Instanzen der Bias-/Kohärenzverwaltungsschaltungen 1894A-1894E veranschaulicht sind, können die Bias-/Kohärenzschaltungen innerhalb einer MMU eines oder mehrerer Hostprozessoren 1805 und/oder innerhalb der Beschleunigerintegrationsschaltung 1836 implementiert sein.

[0222] Eine Ausführungsform ermöglicht GPU-gebunden Speicher 1820-1823 als Teil des Systemspeichers abgebildet zu werden und unter Verwendung einer gemeinsam genutzten Technologie virtuellen Speichers (Shared Virtual Memory; SVM) darauf zugegriffen zu werden, ohne jedoch Leistungseinbußen im Zusammenhang mit voller Systemcache-Kohärenz zu erleiden. In mindestens einer Ausführungsform bietet eine Möglichkeit, auf den an die GPU-gebundenen Speicher 1820-1823 als Systemspeicher ohne lästigen Cache-kohärenz-Overhead zuzugreifen, eine vorteilhafte Betriebsumgebung für GPU-Entlastung. Diese Anordnung ermöglicht einer Software des Host-Prozessors 1805 Operanden einzurichten und ohne den Aufwand traditioneller E/A-DMA-Datenkopien auf Rechenergebnisse zuzugreifen. Derartige traditionelle Kopien beinhalten Treiberaufrufe, Unterbrechungen bzw. Interrupts und Memory Mapped I/O (MMIO)-Zugriffe, die im Vergleich zu einfachen Speicherzugriffen alle ineffizient sind. In mindestens einer Ausführungsform kann eine Fähigkeit, auf den an die GPU gebundenen Speicher 1820-1823 ohne Cache-Kohärenz-Overheads zuzugreifen, entscheidend für die Ausführungszeit einer entlasteten Berechnung sein. In Fällen mit wesentlichem Streaming-Speicherschreibe-Datenaufkommen kann beispielsweise der Cache-Kohärenz-Overhead eine effektive Schreibbandbreite, die von einer GPU 1810-1813 gesehen wird, beträchtlich verringern. In mindestens einer Ausführungsform können die Effizienz der Operandeneinrichtung, die Effizienz des Zugriffs auf Ergebnisse und die Effizienz der Berechnung durch die GPU eine Rolle bei der Bestimmung der Effektivität einer GPU-Entlastung spielen.

[0223] In mindestens einer Ausführungsform wird die Auswahl von GPU-Bias und Host-Prozessor-Bias durch eine Bias-Tracker-Datenstruktur gesteuert. Beispielsweise kann eine Bias-Tabelle verwendet werden, die eine Seitengranularstruktur sein kann (d.h. auf eine Granularität einer Speicherseite gesteuert wird), die 1 oder 2 Bit pro GPU-gebundener Speicherseite enthält. In mindestens einer Ausführungsform kann eine Bias-Tabelle in einem gestohlenen Speicherbereich eines oder mehrerer GPU-gebundener Speicher 1820-1823 mit oder ohne einen Bias-Cache in der GPU 1810-1813 implementiert sein (z.B. um häufig/kürzlich verwendete Einträge einer Bias-Tabelle zwischenspeichern). Alternativ kann eine gesamte Bias-Tabelle innerhalb einer GPU aufrechterhalten werden.

[0224] In mindestens einer Ausführungsform wird auf einen Bias-Tabelleneintrag zugegriffen, der jedem Zugriff auf den GPU-gebundenen Speicher 1820-1823 zugeordnet ist, bevor tatsächlich auf einen GPU-Speicher zugegriffen wird, welcher die folgenden Operationen veranlasst. Zuerst werden lokale Anfragen von der GPU 1810, die ihre Seite in GPU-Bias finden, direkt an einen entsprechenden GPU-Speicher 1820-1823 weitergeleitet. Lokale Anfragen von einer GPU, die ihre Seite in Host-Bias findet, an den Prozessor 1805 weitergeleitet (z.B. über eine Hochgeschwindigkeitsverbindung, wie obenstehend erläutert). In einer Ausführungsform beenden Anfragen von dem Prozessor 1805, die eine angeforderte Seite in Hostprozessor-Bias finden, eine Anfrage wie ein normales Lesen des Speichers. Alternativ können Anfragen, die an eine Seite mit GPU-Bias gerichtet sind, an die GPU 1810-1813 weitergeleitet werden. In mindestens einer Ausführungsform kann

eine GPU dann eine Seite in einen Hostprozessor-Bias überführen, wenn sie derzeit keine Seite verwendet. In mindestens einer Ausführungsform kann ein Bias-Zustand einer Seite entweder durch einen softwarebasierten Mechanismus, einen hardwaregestützten softwarebasierten Mechanismus oder, für eine begrenzte Anzahl von Fällen, einen rein hardwarebasierten Mechanismus geändert werden.

[0225] Ein Mechanismus zum Ändern des Bias-Zustands benutzt einen API-Aufruf (z.B. OpenCL), welcher wiederum den Gerätetreiber einer GPU aufruft, welcher wiederum eine Nachricht an eine GPU sendet (oder einen Befehlsdeskriptor in eine Warteschlange stellt), der sie anweist, einen Bias-Zustand zu ändern und, für einige Übergänge, eine Cache-Leerungs-Operation in einem Host durchzuführen. In mindestens einer Ausführungsform wird eine Cache-Leerungsoperation für einen Übergang von Bias des Host-Prozessors 1805 zur Bias der GPU verwendet, nicht jedoch für einen entgegengesetzten Übergang.

[0226] In einer Ausführungsform wird Cache-Kohärenz durch vorübergehendes Rendern von Seiten mit GPU-Bias, die von dem Hostprozessor 1805 nicht zwischengespeichert werden können, aufrechterhalten. Um auf diese Seiten zuzugreifen, kann der Prozessor 1805 Zugriff von der GPU 1810 anfordern, welche den Zugriff sofort gewähren kann oder auch nicht. Um die Kommunikation zwischen Prozessor 1805 und GPU 1810-1813 zu verringern, ist es somit vorteilhaft, sicherzustellen, dass Seiten mit GPU-Bias diejenigen sind, die von einer GPU, jedoch nicht von dem Hostprozessor 1805 benötigt werden und umgekehrt.

[0227] Eine Hardwarestruktur(en) 915 wird(en) verwendet, um eine oder mehrere Ausführungsformen auszuführen. Einzelheiten hinsichtlich der Hardwarestruktur(en) 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt.

[0228] **Fig. 19A-19B** veranschaulichen beispielhafte integrierte Schaltungen und zugeordnete Graphikprozessoren, die unter Verwendung eines oder mehrerer IP-Kerne hergestellt sein können, gemäß verschiedenen der hier beschriebenen Ausführungsformen. Zusätzlich zu dem, was veranschaulicht ist, können andere Logik und Schaltungen in mindestens einer Ausführungsform umfasst sein, einschließlich zusätzlicher Graphikprozessoren/Kerne, peripherer Schnittstellen-Controllers oder universeller Prozessorkerne.

[0229] **Fig. 19A-19B** sind Blockdiagramme, die beispielhafte Graphikprozessoren zur Verwendung innerhalb eines SoC veranschaulichen, gemäß hier beschriebenen Ausführungsformen. **Fig. 19A** veranschaulicht einen beispielhaften Graphikprozessor 1910 eines Systems auf einer integrierten Chip-Schaltung, das unter Verwendung eines oder mehrerer IP-Kerne hergestellt sein kann, gemäß mindestens einer Ausführungsform. **Fig. 19B** veranschaulicht einen zusätzlichen beispielhaften Graphikprozessor 1940 eines Systems auf einer integrierten Chip-Schaltung, der unter Verwendung eines oder mehrerer IP-Kerne hergestellt sein kann, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist der Graphikprozessor 1910 von **Fig. 19A** ein Graphikprozessorkern mit niedrigem Stromverbrauch. In mindestens einer Ausführungsform ist der Graphikprozessor 1940 von **Fig. 19B** ein leistungsfähigerer Graphikprozessorkern. In mindestens einer Ausführungsform kann jeder der Graphikprozessoren 1910, 1940 eine Variante des Graphikprozessors 1610 von **Fig. 17** sein.

[0230] In mindestens einer Ausführungsform umfasst ein Graphikprozessor 1910 einen Vertex-Prozessor 1905 und einen oder mehrere Fragment-Prozessoren 1915A-1915N (z.B. 1915A, 1915B, 1915C, 1915D bis 1915N-1 und 1915N). In mindestens einer Ausführungsform kann ein Graphikprozessor 1910 verschiedene Shader-Programme über eine separate Logik ausführen, so dass ein Vertex-Prozessor 1905 für die Ausführung von Operationen für Vertex-Shader-Programme optimiert ist, während ein oder mehrere Fragment-Prozessoren 1915A-1915N Fragment(beispielsweise Pixel)-Shading-Operationen für Fragment- oder Pixel-Shader-Programme ausführen. In mindestens einer Ausführungsform führt ein Vertex-Prozessor 1905 eine Vertex-Verarbeitungsphase einer 3D-Graphikpipeline durch und erzeugt Primitive und Vertex-Daten. In mindestens einer Ausführungsform verwenden der(die) Fragment-Prozessor(en) 1915A-1915N Primitive- und Vertex-Daten, die von einem Vertex-Prozessor 1905 erzeugt wurden, um einen Frame- bzw. Einzelbildpuffer zu erzeugen, der auf einer Anzeigevorrichtung angezeigt wird. In mindestens einer Ausführungsform sind der bzw. die Fragment-Prozessor(en) 1915A-1915N optimiert, um Fragment-Shader-Programme auszuführen, wie sie in einer OpenGL-API vorgesehen sind, welche verwendet werden können, um ähnliche Operationen wie ein Pixel-Shader-Programm durchzuführen, wie sie in einer Direct 3D-API vorgesehen sind.

[0231] In mindestens einer Ausführungsform umfasst ein Graphikprozessor 1910 zusätzlich eine oder mehrere Speicherverwaltungseinheiten (MMUs) 1920A-1920B, Cache(s) 1925A-1925B und Schaltungszwischenverbindung(en) 1930A-1930B. In mindestens einer Ausführungsform stellen eine oder mehrere MMU(s) 1920A-1920B ein Adressmapping von virtuell nach physikalisch für einen Graphikprozessor 1910 bereit, ein-

schließlich für einen Vertex-Prozessor 1905 und/oder des/der Fragment-Prozessoren 1915A-1915N, die auf Vertex- oder Bild-/Texturdaten Bezug nehmen können, die in Speicher gespeichert sind, zusätzlich zu Vertex- oder Bild-/Texturdaten, die in einem oder mehreren Cache(s) 1925A-1925B gespeichert sind. In mindestens einer Ausführungsform können eine oder mehrere MMU(s) 1920A-1920B mit anderen MMU(s) innerhalb des Systems synchronisiert werden, einschließlich einer oder mehrerer MMU(s), die einem oder mehreren Anwendungsprozessor(en) 1705, Bildprozessoren 1715 und/oder Videoprozessoren 1720 von **Fig. 17** zugeordnet sind, so dass jeder Prozessor 1705-1720 an einem gemeinsam genutzten oder vereinigten virtuellen Speichersystem partizipieren kann. In mindestens einer Ausführungsform ermöglichen eine oder mehrere Schaltungs-Zwischenverbindung(en) 1930A-1930B einem Graphikprozessor 1910, sich mit anderen IP-Kernen innerhalb des SoC entweder über einen internen Bus des SoC oder über eine direkte Verbindung schnittstellenmäßig zu verbinden.

[0232] In mindestens einer Ausführungsform umfasst ein Graphikprozessor 1940 ein oder mehrere MMU(s), Caches 1925A-1925B und Schaltungszwischenverbindungen 1930A-1930B eines Graphikprozessors 1910 von **Fig. 19A**. In mindestens einer Ausführungsform umfasst ein Graphikprozessor 1940 einen oder mehrere Shader-Kerne 1955A-1955N (z.B. 1955A, 1955B, 1955C, 1955D, 1955E, 1955F bis 1955N-1 und 1955N), welche eine vereinheitlichte Shader-Kern-Architektur bereitstellen, bei der ein einzelner Kern oder Typ von Kern alle Arten von programmierbarem Shader-Code ausführen kann, einschließlich Shader-Programmcode zur Implementierung von Vertex-Shadern, Fragment-Shadern und/oder Rechen-Shadern. In mindestens einer Ausführungsform kann eine Anzahl von Shader-Kernen variieren. In mindestens einer Ausführungsform umfasst ein Graphikprozessor 1940 einen Inter-Core-Task-Manager 1945, der als ein Thread-Dispatcher fungiert, um Ausführung-Threads an einen oder mehrere Shader-Kerne 1955A-1955N zu senden, und eine Kachelungs-Einheit 1958, um Kachelungs-Operationen für kachelbasiertes Rendering zu beschleunigen, bei dem Rendering-Operationen für eine Szene im Bildraum unterteilt sind, um beispielsweise lokale räumliche Kohärenz innerhalb einer Szene auszunutzen oder die Nutzung interner Caches zu optimieren.

[0233] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915 bei der integrierten Schaltung 19A und/oder 19B verwendet werden, um Operationen zu inferenzieren oder vorherzusagen, die mindestens teilweise auf Gewichtungsparametern basieren, die unter Verwendung von Trainingsoperationen für neuronale Netzwerke, Funktionen und/oder Architekturen neuronaler Netzwerke oder hier beschriebenen Anwendungsfällen neuronaler Netzwerke berechnet wurden.

[0234] **Fig. 20A-20B** veranschaulichen eine zusätzliche beispielhafte Graphikprozessorlogik gemäß hier beschriebenen Ausführungsformen. **Fig. 20A** veranschaulicht einen Graphikkern 2000, der in mindestens einer Ausführungsform in einem Graphikprozessor 1710 von **Fig. 17** umfasst sein kann und in mindestens einer Ausführungsform ein vereinheitlichter Shader-Kern 1955A-1955N wie in **Fig. 19B** in mindestens einer Ausführungsform sein kann. **Fig. 20B** veranschaulicht eine hochparallele universelle Graphikverarbeitungseinheit („GPGPU“) 2030, die für den Einsatz auf einem Multi-Chip-Modul in mindestens einer Ausführungsform geeignet ist.

[0235] In einer Ausführungsform umfasst ein Graphikkern 2000 einen gemeinsam genutzten Anweisungs-Cache 2002, eine Textureinheit 2018 und einen Cache/gemeinsam genutzten Speicher 2020, die für Ausführungsressourcen innerhalb des Graphikkerns 2000 gemeinsam sind. In mindestens einer Ausführungsform kann der Graphikkern 2000 mehrere Slices 2001A-2001N oder Partitionen für jeden Kern umfassen, und ein Graphikprozessor kann mehrere Instanzen des Graphikkerns 2000 umfassen. Slices 2001A-2001N können eine Unterstützungslogik umfassen, die einen lokalen Anweisungs-Cache 2004A-2004N, einen Thread-Planer 2006A-2006N, einen Thread-Dispatchers 2008A-2008N und einen Satz von Registern 2010A-2010N umfasst. In einer Ausführungsform können Slices 2001A-2001N einen Satz zusätzlicher Funktionseinheiten (AFUs 2012A-2012N), Gleitkommaeinheiten (FPU 2016A-2016N), Ganzzahl-Rechenlogikeinheiten (ALUs 2016A-2016N), Adressberechnungseinheiten (ACU 2015A-2015N), doppelt genaue Gleitkommaeinheiten (DPFPU 2015A-2015N) und Matrixverarbeitungseinheiten (MPU 2017A-2017N) umfassen.

[0236] In einer Ausführungsform können FPUs 2014A-2014N einfach genaue (32-Bit) und halb genaue (16-Bit) Gleitkomma-Operationen durchführen, während DPFPU 2015A-2015N doppelt genaue (64-Bit) Gleitkomma-Operationen durchführen. In mindestens einer Ausführungsform können die ALUs 2016A-2016N Ganzzahl-Operationen mit variabler Genauigkeit und 8-Bit-, 16-Bit- und 32-Bit-Genauigkeit durchführen und für Operationen mit gemischter Genauigkeit konfiguriert sein. In einer Ausführungsform kön-

nen MPUs 2017A-2017N auch für Matrix-Operationen mit gemischter Genauigkeit konfiguriert sein, einschließlich halb genauer Gleitkomma- und 8-Bit Ganzzahl-Operationen. In einer Ausführungsform können MPUs 2017-2017N eine Vielzahl von Matrix-Operationen durchführen, um die Frameworks von Anwendungen für maschinelles Lernen zu beschleunigen, einschließlich des Ermöglichens von Unterstützung für beschleunigte allgemeine bzw. generelle Matrix-zu-Matrix Multiplikation (GEMM). In mindestens einer Ausführungsform können die AFUs 2012A-2012N zusätzliche logische Operationen durchführen, die nicht von Gleitkomma- oder Integer-Einheiten unterstützt werden, die trigonometrische Operationen (z.B. Sinus, Cosinus usw.) umfassen.

[0237] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915 in einem Graphikkern 2000 zum Inferenzieren oder Vorhersagen von Operationen verwendet werden, die mindestens teilweise auf Gewichtungsparemtern basieren, die unter Verwendung von Trainingsoperationen für neuronale Netzwerke, Funktionen und/oder Architekturen neuronaler Netzwerke oder hier beschriebenen Anwendungsfällen neuronaler Netzwerke berechnet wurden.

[0238] **Fig. 20B** veranschaulicht eine Allzweck-Verarbeitungseinheit (GPGPU) 2030, die konfiguriert sein kann, um hochparallele Rechenoperationen zu ermöglichen, die von einer Anordnung von Graphikverarbeitungseinheiten durchzuführen sind, in mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann eine GPGPU 2030 direkt mit anderen Instanzen einer GPGPU 2030 verknüpft sein, um einen Multi-GPU-Cluster zu erstellen, um die Trainingsgeschwindigkeit für tiefe neuronale Netzwerke zu verbessern. In mindestens einer Ausführungsform umfasst eine GPGPU 2030 eine Host-Schnittstelle 2032, um eine Verbindung mit einem Host-Prozessor zu ermöglichen. In mindestens einer Ausführungsform ist die Host-Schnittstelle 2032 eine PCI-Express-Schnittstelle. In mindestens einer Ausführungsform kann die Host-Schnittstelle 2032 eine herstellerspezifische Kommunikationsschnittstelle oder Kommunikationsstruktur sein. In mindestens einer Ausführungsform empfängt eine GPGPU 2030 Befehle von einem Host-Prozessor und verwendet einen globalen Planer 2034, um die diesen Befehlen zugeordneten Ausführung-Threads an einen Satz von Rechenclustern 2036A-2036H zu verteilen. In mindestens einer Ausführungsform nutzen die Rechencluster 2036A-2036H einen Cache-Speicher 2038 gemeinsam. In mindestens einer Ausführungsform kann ein Cache-Speicher 2038 als ein übergeordneter Cache für Cache-Speicher innerhalb der Rechencluster 2036A-2036H dienen.

[0239] In mindestens einer Ausführungsform umfasst eine GPGPU 2030 Speicher 2044A-2044B, der mit Rechenclustern 2036A-2036H über einen Satz von Speicher-Controllern 2042A-2042B gekoppelt ist. In mindestens einer Ausführungsform kann der Speicher 2044A-2044B verschiedene Arten von Speichervorrichtungen umfassen, einschließlich eines dynamischen Direktzugriffsspeichers (DRAM) oder eines Graphik-Direktzugriffsspeichers, wie beispielsweise einen synchronen Graphik-Direktzugriffsspeicher (SGRAM), der einen Graphikspeicher mit doppelter Datenrate (GDDR) umfasst.

[0240] In mindestens einer Ausführungsform umfassen Rechencluster 2036A-2036H jeweils einen Satz von Graphikkernen, wie beispielsweise einen Graphikkern 2000 von **Fig. 20A**, der mehrere Typen von Ganzzahl- und Gleitkomma-Logikeinheiten umfassen kann, die Rechenoperationen mit einer Reihe von Genauigkeitsstufen durchführen können, die auch für Berechnungen zum maschinellen Lernen geeignet sind.

[0241] Beispielsweise kann in mindestens einer Ausführungsform mindestens eine Teilmenge von Gleitkommaeinheiten in jedem der Rechencluster 2036A-2036H konfiguriert sein, um 16-Bit- oder 32-Bit-Gleitkommaoperationen durchzuführen, während eine andere Teilmenge der Gleitkommaeinheiten konfiguriert sein kann, 64-Bit-Gleitkommaoperationen durchzuführen.

[0242] In mindestens einer Ausführungsform können mehrere Instanzen einer GPGPU 2030 konfiguriert sein, um als Rechencluster zu arbeiten. In mindestens einer Ausführungsform variiert die Kommunikation, die von den Rechenclustern 2036A-2036H für die Synchronisation und den Datenaustausch verwendet wird, je nach Ausführungsform. In mindestens einer Ausführungsform kommunizieren mehrere Instanzen einer GPGPU 2030 über eine Host-Schnittstelle 2032. In mindestens einer Ausführungsform umfasst eine GPGPU 2030 einen E/A-Hub 2039, der eine GPGPU 2030 mit einer GPU-Verbindung 2040 koppelt, die eine direkte Verbindung zu anderen Instanzen einer GPGPU 2030 ermöglicht. In mindestens einer Ausführungsform ist eine GPU-Verbindung 2040 mit einer dedizierten GPU-zu-GPU-Brücke gekoppelt, die Kommunikation und Synchronisation zwischen mehreren Instanzen einer GPGPU 2030 ermöglicht. In mindestens einer Ausführungsform

rungsform ist eine GPU-Verbindung 2040 mit einer Hochgeschwindigkeitsverbindung gekoppelt, um Daten an andere GPGPUs oder Parallelprozessoren zu senden und zu empfangen. In mindestens einer Ausführungsform befinden sich mehrere Instanzen der GPGPU 2030 in separaten Datenverarbeitungssystemen und kommunizieren über eine Netzwerkvorrichtung, die über die Host-Schnittstelle 2032 zugänglich ist. In mindestens einer Ausführungsform kann die GPU-Verbindung 2040 konfiguriert sein, eine Verbindung zu einem Host-Prozessor zusätzlich oder alternativ zu der Host-Schnittstelle 2032 zu ermöglichen.

[0243] In mindestens einer Ausführungsform kann eine GPGPU 2030 konfiguriert sein, um neuronale Netzwerke zu trainieren. In mindestens einer Ausführungsform kann eine GPGPU 2030 innerhalb einer Inferenzplattform verwendet werden. In mindestens einer Ausführungsform, bei welcher eine GPGPU 2030 zur Inferenzierung verwendet wird, kann eine GPGPU weniger Rechencluster 2036A-2036H umfassen als dann, wenn eine GPGPU 2030 zum Training eines neuronalen Netzwerks verwendet wird. In mindestens einer Ausführungsform kann sich die dem Speicher 2044A-2044B zugeordnete Speichertechnologie zwischen Inferenzierungs- und Trainingskonfigurationen unterscheiden, wobei Speichertechnologien mit höherer Bandbreite für Trainingskonfigurationen vorgesehen sind. In mindestens einer Ausführungsform kann die Inferenzkonfiguration einer GPGPU 2030 inferenzspezifische Anweisungen unterstützen. Beispielsweise kann in mindestens einer Ausführungsform eine Inferenzkonfiguration eine oder mehrere 8-Bit-Ganzzahl-Skalarprodukt-Anweisungen unterstützen, die während Inferenzoperationen für bereitgestellte neuronale Netzwerke verwendet werden können.

[0244] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915 in einer GPGPU 2030 zum Inferenzieren oder Vorhersagen von Operationen verwendet werden, die mindestens teilweise auf Gewichtungsparametern basieren, die unter Verwendung von Trainingsoperationen für neuronale Netzwerke, Funktionen und/oder Architekturen neuronaler Netzwerke oder hier beschriebenen Anwendungsfällen neuronaler Netzwerke berechnet wurden.

[0245] **Fig. 21** ist ein Blockdiagramm, das ein Rechensystem 2100 gemäß mindestens einer Ausführungsform darstellt. In mindestens einer Ausführungsform umfasst ein Rechensystem 2100 ein Verarbeitungsteilsystem 2101 mit einem oder mehreren Prozessor(en) 2102 und einen Systemspeicher 2104, der über einen Zwischenverbindungs Pfad kommuniziert, der einen Speicher-Hub 2105 umfassen kann. In mindestens einer Ausführungsform kann ein Speicher-Hub 2105 eine separate Komponente innerhalb einer Chipsatz-Komponente oder in einen oder mehrere Prozessor(en) 2102 integriert sein. In mindestens einer Ausführungsform ist ein Speicher-Hub 2105 über eine Kommunikationsverbindung 2106 mit einem E/A-Teilsystem 2111 gekoppelt. In mindestens einer Ausführungsform umfasst ein E/A-Teilsystem 2111 einen E/A-Hub 2107, der einem Rechensystem 2100 ermöglichen kann, Eingaben von einer oder mehreren Eingabevorrichtung(en) 2108 zu empfangen. In mindestens einer Ausführungsform kann ein E/A-Hub 2107 einem Anzeigen-Controller, der in einem oder mehreren Prozessor(en) 2102 umfasst sein kann, ermöglichen, Ausgaben an eine oder mehrere Anzeigevorrichtung(en) 2110A bereitzustellen. In mindestens einer Ausführungsform können eine oder mehrere mit dem E/A-Hub 2107 gekoppelte Anzeigevorrichtung(en) 2110A eine lokale, interne oder eingebettete Anzeigevorrichtung umfassen.

[0246] In mindestens einer Ausführungsform umfasst ein Verarbeitungsteilsystem 2101 einen oder mehrere Parallelprozessoren 2112, die über einen Bus oder eine andere Kommunikationsverbindung 2113 mit einem Speicher-Hub 2105 gekoppelt sind. In mindestens einer Ausführungsform kann eine Kommunikationsverbindung 2113 eine von einer beliebigen Anzahl von auf Standards basierenden Kommunikation-Verbindungstechnologien oder - Verbindungsprotokollen sein, wie beispielsweise, jedoch nicht beschränkt auf, PCI Express, oder kann eine herstellerspezifische Kommunikationsschnittstelle oder Kommunikationsstruktur sein. In mindestens einer Ausführungsform bilden ein oder mehrere Parallelprozessoren 2112 ein rechnerisch fokussiertes Parallel- oder Vektor-Verarbeitungssystem, das eine große Anzahl von Verarbeitungskernen und/oder Verarbeitungsklustern umfassen kann, wie beispielsweise einen Prozessor mit vielen integrierten Kernen (Many Integrated Core; MIC). In mindestens einer Ausführungsform bilden ein oder mehrere Parallelprozessoren 2112 ein Graphikverarbeitungs-Teilsystem, das Pixel an eine oder mehrere Anzeigevorrichtungen 2110A ausgeben kann, die über einen E/A-Hub 2107 gekoppelt sind. In mindestens einer Ausführungsform können ein oder mehrere Parallelprozessoren 2112 auch einen Anzeigen-Controller und eine Anzeigeschnittstelle (nicht gezeigt) umfassen, um eine direkte Verbindung zu einer oder mehreren Anzeigevorrichtung(en) 2110B zu ermöglichen.

[0247] In mindestens einer Ausführungsform kann sich eine Systemspeichereinheit 2116 mit einem E/A-Hub 2107 verbinden, um einen Speichermechanismus für ein Rechensystem 2100 bereitzustellen. In mindestens einer Ausführungsform kann ein E/A-Schalter 2116 verwendet werden, um einen Schnittstellenmechanismus bereitzustellen, um Verbindungen zwischen einem E/A-Hub 2107 und anderen Komponenten, wie beispielsweise einem Netzwerkadapter 2118 und/oder einem drahtlosen Netzwerkadapter 2119, der in einer Plattform integriert sein kann, und verschiedenen anderen Vorrichtungen, die über eine oder mehrere Add-In-Vorrichtungen 2112 hinzugefügt sein können, zu ermöglichen. In mindestens einer Ausführungsform kann ein Netzwerkadapter 2118 ein Ethernet-Adapter oder ein anderer verdrahteter Netzwerkadapter sein. In mindestens einer Ausführungsform kann ein drahtloser Netzwerkadapter 2119 ein oder mehrere Wi-Fi, Bluetooth-, Nahfeldkommunikation(NFC)- oder andere Netzwerkvorrichtungen umfassen, die ein oder mehrere drahtlose Funkeinrichtungen umfassen.

[0248] In mindestens einer Ausführungsform kann ein Rechensystem 2100 andere Komponenten umfassen, die nicht explizit gezeigt sind, einschließlich USB- oder andere Portverbindungen, optische Speicherlaufwerke, Videoaufnahmevorrichtungen und dergleichen, die auch mit einem E/A-Hub 2107 verbunden sein können. In mindestens einer Ausführungsform können Kommunikationspfade, die verschiedene Komponenten in **Fig. 21** miteinander verbinden, unter Verwendung beliebiger geeigneter Protokolle, wie z. B. PCI-(Peripheral Component Zwischenverbindung)-basierter Protokolle (z. B. PCI-Express), oder anderer Bus- oder Punkt-zu-Punkt-Kommunikationsschnittstellen und/oder - Protokolle(n), wie z.B. NV-Link-Hochgeschwindigkeitsverbindungen oder Zwischenverbindungsprotokolle, implementiert sein.

[0249] In mindestens einer Ausführungsform umfassen ein oder mehrere Parallelprozessoren 2112 für Graphik- und Videoverarbeitung optimierte Schaltungen, die beispielsweise Videoausgabeschaltungen umfassen und eine Graphikverarbeitungseinheit (GPU) bilden. In mindestens einer Ausführungsform umfassen ein oder mehrere Parallelprozessoren 2112 für allgemeine Verarbeitung optimierte Schaltungen. In mindestens einer Ausführungsform können Komponenten eines Computersystems 2100 mit einem oder mehreren anderen Systemelementen auf einer einzigen integrierten Schaltung integriert sein. Beispielsweise können in mindestens einer Ausführungsform ein oder mehrere Parallelprozessoren 2112, der Speicher-Hub 2105, der bzw. die Prozessor(en) 2102 und der E/A-Hub 2107 in eine System-on-Chip (SoC)-integrierte Schaltung integriert sein. In mindestens einer Ausführungsform können Komponenten eines Computersystems 2100 in ein einziges Gehäuse integriert sein, um eine System-in-Package(SIP)-Konfiguration zu bilden. In mindestens einer Ausführungsform kann mindestens ein Teil von Komponenten eines Computersystems 2100 in ein Multi-Chip-Modul (MCM) integriert sein, welches mit anderen Multi-Chip-Modulen zu einem modularen Computersystem verbunden sein kann.

[0250] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915 in einem System 2100 zum Inferenzieren oder Vorhersagen von Operationen verwendet werden, die mindestens teilweise auf Gewichtungsparametern basieren, die unter Verwendung von Trainingsoperationen für neuronale Netzwerke, Funktionen und/oder Architekturen neuronaler Netzwerke oder hier beschriebenen Anwendungsfällen neuronaler Netzwerke berechnet wurden.

PROZESSOREN

[0251] **Fig. 22A** veranschaulicht einen Parallelprozessor 2200 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform können verschiedene Komponenten eines Parallelprozessors 2200 unter Verwendung einer oder mehrerer integrierter Schaltungsvorrichtungen, wie beispielsweise programmierbare Prozessoren, anwendungsspezifische integrierte Schaltungen (ASICs) oder feldprogrammierbare Gate-Arrays (FPGA) implementiert sein. In mindestens einer Ausführungsform ist ein veranschaulichter Parallelprozessor 2200 eine Variante eines oder mehrerer in **Fig. 21** gezeigter Parallelprozessoren 2112 gemäß einer beispielhaften Ausführungsform.

[0252] In mindestens einer Ausführungsform umfasst ein Parallelprozessor 2200 eine Parallelverarbeitungseinheit 2202. In mindestens einer Ausführungsform umfasst eine Parallelverarbeitungseinheit 2202 eine E/A-Einheit 2204, die eine Kommunikation mit anderen Vorrichtungen ermöglicht, einschließlich anderer Instanzen einer Parallelverarbeitungseinheit 2202. In mindestens einer Ausführungsform kann die E/A-Einheit 2204 direkt mit anderen Vorrichtungen verbunden sein. In mindestens einer Ausführungsform verbindet sich die E/A-Einheit 2204 mit anderen Vorrichtungen unter Verwendung eines Hubs oder einer Schalter--

Schnittstelle, wie beispielsweise dem Speicher-Hub 2105. In mindestens einer Ausführungsform bilden Verbindungen zwischen einem Speicher-Hub 2105 und einer E/A-Einheit 2204 eine Kommunikationsverbindung 2113. In mindestens einer Ausführungsform verbindet sich eine E/A-Einheit 2204 mit einer Host-Schnittstelle 2206 und einer Speicherquerschienen 2216, wobei die Host-Schnittstelle 2206 Befehle empfängt, die zur Durchführung von Verarbeitungsoperationen bestimmt sind, und eine Speicherquerschienen 2216 Befehle empfängt, die zur Durchführung von Speicheroperationen bestimmt sind.

[0253] In mindestens einer Ausführungsform kann dann, wenn eine Host-Schnittstelle 2206 über eine E/A-Einheit 2204 einen Befehlspuffer empfängt, eine Host-Schnittstelle 2206 Arbeitsoperationen anweisen, um diese Befehle an einem Frontend 2208 auszuführen. In mindestens einer Ausführungsform wird ein Frontend 2208 mit einem Planer 2210 gekoppelt, der konfiguriert ist, um Befehle oder andere Arbeitselemente an eine Verarbeitungsclusteranordnung 2212 zu verteilen. In mindestens einer Ausführungsform stellt ein Planer 2210 sicher, dass die Clusteranordnung 2212 richtig konfiguriert ist und sich in einem gültigen Zustand befindet, bevor Aufgaben an eine Verarbeitungsclusteranordnung 2212 verteilt werden. In mindestens einer Ausführungsform ist ein Planer 2210 über Firmware-Logik implementiert, die auf einem Mikrocontroller ausgeführt wird. In mindestens einer Ausführungsform ist ein von dem Mikrocontroller implementierter Planer 2210 konfigurierbar, um komplexe Planungs- und Arbeitsverteilungsoperationen mit grober und feiner Granularität durchzuführen, was eine schnelle Vorwegnahme und Kontextumschaltung von Threads ermöglicht, die auf einer Verarbeitungsanordnung 2212 ausgeführt werden. In mindestens einer Ausführungsform kann die Host-Software Arbeitslasten für die Planung auf einer Verarbeitungsanordnung 2212 über eine von mehreren Graphikverarbeitungstürklings nachweisen. In mindestens einer Ausführungsform können dann Arbeitslasten über eine Verarbeitungsanordnung 2212 hinweg durch die Logik eines Planers 2210 innerhalb eines Mikrocontrollers einschließlich eines Planers 2210 automatisch verteilt werden.

[0254] In mindestens einer Ausführungsform kann eine Verarbeitungsclusteranordnung 2212 bis zu „N“-Verarbeitungscluster (z. B. Cluster 2214A, Cluster 2214B, bis Cluster 2214N) umfassen. In mindestens einer Ausführungsform kann jeder Cluster 2214A-2214N einer Verarbeitungsclusteranordnung 2212 eine große Anzahl gleichzeitiger Threads ausführen. In mindestens einer Ausführungsform kann ein Planer 2210 den Clustern 2214A-2214N der Verarbeitungsclusteranordnung 2212 unter Verwendung verschiedener Planungs- und/oder Arbeitsverteilungsalgorithmen Arbeit zuweisen, welche je nach Arbeitsbelastung, die für jede Art von Programm oder Berechnung aufkommt, variieren kann. In mindestens einer Ausführungsform kann die Planung von einem Planer 2210 dynamisch gehandhabt werden, oder kann teilweise durch Compilerlogik während der Kompilierung der Programmlogik, die zur Ausführung durch die Verarbeitung der Clusteranordnung 2212 konfiguriert ist, unterstützt werden. In mindestens einer Ausführungsform können verschiedene Cluster 2214A-2214N der Verarbeitungsclusteranordnung 2212 zur Verarbeitung verschiedener Arten von Programmen oder zur Durchführung verschiedener Arten von Berechnungen zugeteilt sein.

[0255] In mindestens einer Ausführungsform kann eine Verarbeitungsclusteranordnung 2212 konfiguriert sein, um verschiedene Arten von Parallelverarbeitungsoperationen durchzuführen. In mindestens einer Ausführungsform ist eine Verarbeitungsclusteranordnung 2212 konfiguriert, um parallele Allzweck-Rechenoperationen durchzuführen. Beispielsweise kann in mindestens einer Ausführungsform eine Verarbeitungsclusteranordnung 2212 Logik umfassen, um Verarbeitungsaufgaben einschließlich eines Filterns von Video- und/oder Audiodaten, Durchführens von Modellierungsoperationen, einschließlich physikalischer Operationen, und Durchführens von Datentransformationen auszuführen.

[0256] In mindestens einer Ausführungsform ist eine Verarbeitungsclusteranordnung 2212 konfiguriert, um parallele Graphikverarbeitungsoperationen durchzuführen. In mindestens einer Ausführungsform kann eine Verarbeitungsclusteranordnung 2212 zusätzliche Logik umfassen, um die Ausführung solcher Graphikverarbeitungsoperationen zu unterstützen, einschließlich, jedoch nicht beschränkt auf, Texturabtastlogik zur Durchführung von Texturoperationen, sowie einer Tessellierungslogik und anderer Vertex-Verarbeitungslogik. In mindestens einer Ausführungsform kann eine Verarbeitungsclusteranordnung 2212 konfiguriert sein, um grafikverarbeitungsbezogene Shader-Programme auszuführen, wie beispielsweise Vertex-Shader, Tessellation-Shader, Geometrie-Shader und Pixel-Shader. In mindestens einer Ausführungsform kann eine Parallelverarbeitungseinheit 2202 Daten aus einem Systemspeicher über eine E/A-Einheit 2204 zur Verarbeitung übertragen. In mindestens einer Ausführungsform können während der Verarbeitung übertragene Daten in einem On-Chip-Speicher (z. B. Speicher des Parallelprozessors 2220) während der Verarbeitung gespeichert und anschließend in den Systemspeicher zurückgeschrieben werden.

[0257] In mindestens einer Ausführungsform kann dann, wenn eine Parallelverarbeitungseinheit 2202 zur Durchführung von Graphikverarbeitung verwendet wird, ein Planer 2210 konfiguriert sein, um eine Verarbei-

tungsarbeitslast in etwa gleich große Aufgaben aufzuteilen, um die Verteilung von Graphikverarbeitungsoperationen auf mehrere Cluster 2214A-2214N der Verarbeitungsclusteranordnung 2212 besser zu ermöglichen. In mindestens einer Ausführungsform können Abschnitte der Verarbeitungsclusteranordnung 2212 konfiguriert sein, um verschiedene Arten von Verarbeitung durchzuführen. Beispielsweise kann in mindestens einer Ausführungsform ein erster Abschnitt konfiguriert sein, um Vertex-Shading und Topologieerzeugung durchzuführen, kann ein zweiter Abschnitt konfiguriert sein, um Tessellation und Geometrie-Shading durchzuführen, und ein dritter Abschnitt kann konfiguriert sein, um Pixel-Shading oder andere Bildschirmraumoperationen durchzuführen, um ein gerendertes Bild für die Anzeige zu erzeugen. In mindestens einer Ausführungsform können Zwischendaten, die von einem oder mehreren der Cluster 2214A-2214N erzeugt wurden, in Puffern gespeichert werden, so dass Zwischendaten zwischen Clustern 2214A-2214N zur weiteren Verarbeitung übertragen werden können.

[0258] In mindestens einer Ausführungsform kann eine Verarbeitungsclusteranordnung 2212 Verarbeitungsaufgaben empfangen, die über einen Planer 2210 auszuführen sind, der von einem Frontend 2208 Befehle empfängt, die Verarbeitungsaufgaben definieren. In mindestens einer Ausführungsform können Verarbeitungsaufgaben Indizes von zu verarbeitenden Daten umfassen, z. B. Oberflächen-(Patch)-Daten, Stammfunktionsdaten, Vertex-Daten und/oder Pixeldaten, sowie Zustandsparameter und Befehle, die definieren, wie Daten zu verarbeiten sind (z. B. welches Programm auszuführen ist). In mindestens einer Ausführungsform kann ein Planer 2210 konfiguriert sein, um Indizes zu holen, die Aufgaben entsprechen, oder kann Indizes von dem Frontend 2208 empfangen. In mindestens einer Ausführungsform kann ein Frontend 2208 konfiguriert sein, um sicherzustellen, dass eine Verarbeitungsclusteranordnung 2212 in einem gültigen Zustand konfiguriert ist, bevor eine durch eingehende Befehlspuffer (z. B. Stapelpuffer, Schiebepuffer, usw.) spezifizierte Arbeitslast initiiert wird.

[0259] In mindestens einer Ausführungsform kann jede von einer oder mehreren Instanzen einer Parallelverarbeitungseinheit 2202 mit dem Parallelprozessorspeicher 2222 gekoppelt sein. In mindestens einer Ausführungsform kann auf den Parallelprozessorspeicher 2222 über eine Speicherquerschienen 2216 zugegriffen werden, die Speicheranforderungen von einer Verarbeitungsclusteranordnung 2212 sowie einer E/A-Einheit 2204 empfangen kann. In mindestens einer Ausführungsform kann eine Speicherquerschienen 2216 über eine Speicherschnittstelle 2218 auf den Parallelprozessorspeicher 2222 zugreifen. In mindestens einer Ausführungsform kann die Speicherschnittstelle 2218 mehrere Partitionierungseinheiten (z. B. Partitionierungseinheit 2220A, Partitionierungseinheit 2222B bis Partitionierungseinheit 2220N) umfassen, die jeweils mit einem Abschnitt (z. B. der Speichereinheit) des Parallelprozessorspeichers 2222 koppeln können. In mindestens einer Ausführungsform ist eine Anzahl von Partitionierungseinheiten 2220A-2220N konfiguriert, um gleich einer Anzahl von Speichereinheiten zu sein, so dass eine erste Partitionierungseinheit 2220A eine entsprechende erste Speichereinheit 2224A aufweist, eine zweite Partitionierungseinheit 2222B eine entsprechende Speichereinheit 2224B aufweist und eine N-te Partitionierungseinheit 2220N eine entsprechende N-te Speichereinheit 2224N aufweist. In mindestens einer Ausführungsform kann eine Anzahl von Partitionierungseinheiten 2220A-2220N nicht gleich einer Anzahl von Speichervorrichtungen sein.

[0260] In mindestens einer Ausführungsform können die Speichereinheiten 2224A-2224N verschiedene Arten von Speichervorrichtungen umfassen, einschließlich dynamischen Direktzugriffsspeichers (DRAM) oder Graphik-Direktzugriffsspeichers, wie beispielsweise synchronen Graphik-Direktzugriffsspeichers (SGRAM), einschließlich Graphikspeichers mit doppelter Datenrate (GDDR). In mindestens einer Ausführungsform können die Speichereinheiten 2224A-2224N auch 3D-Stapelspeicher umfassen, einschließlich, jedoch nicht beschränkt auf, Speicher mit hoher Bandbreite (HBM). In mindestens einer Ausführungsform können Renderziele, wie beispielsweise Einzelbildpuffer bzw. Frame-Puffer oder Texturkarten, über Speichereinheiten 2224A-2224N hinweg gespeichert sein, so dass die Partitionseinheiten 2220A-2220N Abschnitte jedes Renderziels parallel schreiben können, um verfügbare Bandbreite des Parallelprozessorspeichers 2222 effizient zu nutzen. In mindestens einer Ausführungsform kann eine lokale Instanz des Parallelprozessorspeichers 2222 zugunsten eines einheitlichen Speicherdesigns, das Systemspeicher in Verbindung mit lokalem Cache-Speicher nutzt, ausgeschlossen sein.

[0261] In mindestens einer Ausführungsform kann jeder beliebige der Cluster 2214A-2214N der Verarbeitungsclusteranordnung 2212 Daten verarbeiten, die in eine beliebige der Speichereinheiten 2224A-2224N in dem Parallelprozessorspeicher 2222 geschrieben werden. In mindestens einer Ausführungsform kann eine Speicherquerschienen 2216 konfiguriert sein, um eine Ausgabe jedes Clusters 2214A-2214N an eine beliebige Partitionseinheit 2212A-2212N oder an einen anderen Cluster 2214A-2214N zu übertragen, der zusätzliche Verarbeitungsoperationen an einer Ausgabe durchführen kann. In mindestens einer Ausführungsform kann jeder Cluster 2214A-2214N mit einer Speicherschnittstelle 2218 über die Speicherquerschienen 2216

kommunizieren, um aus verschiedenen externen Speichervorrichtungen zu lesen oder in diese zu schreiben. In mindestens einer Ausführungsform weist eine Speicherquerschleife 2216 zur Kommunikation mit der E/A-Einheit 2204 eine Verbindung zu einer Speicherschnittstelle 2218 auf, sowie eine Verbindung zu einer lokalen Instanz eines Parallelprozessorspeichers 2222, so dass Verarbeitungseinheiten innerhalb verschiedener Verarbeitungscluster 2214A-2214N mit Systemspeicher oder anderem Speicher kommunizieren können, der nicht lokal zu einer Parallelverarbeitungseinheit 2202 ist. In mindestens einer Ausführungsform kann eine Speicherquerschleife 2216 virtuelle Kanäle verwenden, um Ströme von Datenaufkommen zwischen den Clustern 2214A-2214N und den Partitionierungseinheiten 2220A-2220N zu trennen.

[0262] In mindestens einer Ausführungsform können mehrere Instanzen einer Parallelverarbeitungseinheit 2202 auf einer einzigen Erweiterungskarte bereitgestellt sein oder können mehrere Erweiterungskarten miteinander verbunden sein. In mindestens einer Ausführungsform können verschiedene Instanzen einer Parallelverarbeitungseinheit 2202 konfiguriert sein, auch dann miteinander zu arbeiten, wenn verschiedene Instanzen eine unterschiedliche Anzahl von Verarbeitungskernen, unterschiedliche Mengen an lokalem Parallelprozessorspeicher und/oder andere Konfigurationsunterschiede aufweisen. Beispielsweise können in mindestens einer Ausführungsform einige Instanzen einer Parallelverarbeitungseinheit 2202 relativ zu anderen Instanzen Gleitkommaeinheiten mit höherer Genauigkeit umfassen. In mindestens einer Ausführungsform können Systeme, die eine oder mehrere Instanzen einer Parallelverarbeitungseinheit 2202 oder Parallelprozessors 2200 umfassen, in einer Mehrzahl von Konfigurationen und Formfaktoren implementiert sein, einschließlich, jedoch nicht beschränkt auf, Desktop-, Laptop- oder Handheld-Personal Computer, Server, Workstations, Spielkonsolen und/oder eingebettete Systeme.

[0263] **Fig. 22B** ist ein Blockdiagramm einer Partitionierungseinheit 2220 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist eine Partitionierungseinheit 2220 eine Instanz einer der Partitionierungseinheiten 2220A-2220N aus **Fig. 22**. In mindestens einer Ausführungsform umfasst die Partitionierungseinheit 2220 einen L2-Cache 2221, ein Frame-Buffer-Interface bzw. eine Einzelbildpufferschnittstelle 2225 und eine ROP 2226 (Rasteroperationseinheit). Der L2-Cache 2221 ist ein Lese-/Schreib-Cache, der konfiguriert ist, um Lade- und Speicheroperationen durchzuführen, die von einer Speicherquerschleife 2216 und ROP 2226 empfangen wurden. In mindestens einer Ausführungsform werden Lesefehler und dringende Rückschreibanforderungen von dem L2-Cache 2221 an die Einzelbildpufferschnittstelle 2225 zur Verarbeitung ausgegeben. In mindestens einer Ausführungsform können Aktualisierungen ebenfalls über eine Einzelbildpufferschnittstelle 2225 zur Verarbeitung auch an einen Einzelbildpuffer gesendet werden. In mindestens einer Ausführungsform ist eine Einzelbildpufferschnittstelle 2225 mit einer der Speichereinheiten in dem Parallelprozessorspeicher, wie beispielsweise den Speichereinheiten 2224A-2224N aus **Fig. 22** (z. B. innerhalb eines Parallelprozessorspeichers 2220) schnittstellenmäßig verbunden.

[0264] In mindestens einer Ausführungsform ist eine ROP 2226 eine Verarbeitungseinheit, die Rasteroperationen wie Schablone, Z-Test, Blending und dergleichen durchführt. In mindestens einer Ausführungsform gibt eine ROP 2226 dann verarbeitete Graphikdaten aus, die in einem Graphikspeicher gespeichert sind. In mindestens einer Ausführungsform umfasst eine ROP 2226 eine Kompressionslogik zum Komprimieren von Tiefen- oder Farbdaten, die in den Speicher geschrieben werden, und zum Dekomprimieren von Tiefen- oder Farbdaten, die aus dem Speicher gelesen werden. In mindestens einer Ausführungsform kann die Kompressionslogik eine verlustfreie Kompressionslogik sein, die einen oder mehrere von mehreren Kompressionsalgorithmen verwendet. Die Art der Komprimierung, die von einer ROP 2226 durchgeführt wird, kann basierend auf statistischen Merkmalen der zu komprimierenden Daten variieren. Beispielsweise wird in mindestens einer Ausführungsform eine Delta-Farbkompression auf Tiefen- und Farbdaten auf kachelweiser Grundlage durchgeführt.

[0265] In mindestens einer Ausführungsform ist eine ROP 2226 in jedem Verarbeitungscluster (z. B. Cluster 2214A-2214N aus **Fig. 22**) anstelle von in der Partitionierungseinheit 2220 umfasst. In mindestens einer Ausführungsform werden Lese- und Schreibanforderungen für Pixeldaten über eine Speicherquerschleife 2216 anstelle von Pixelfragmentdaten übertragen. In mindestens einer Ausführungsform können verarbeitete Graphikdaten auf einer Anzeigevorrichtung, wie beispielsweise einer oder mehreren Anzeigevorrichtung(en) 2110 aus **Fig. 21** angezeigt werden, zur weiteren Verarbeitung durch den/die Prozessor(en) 2102 weitergeleitet werden, oder zur weiteren Verarbeitung durch eine der Verarbeitungsentitäten innerhalb des Parallelprozessors 2200 aus **Fig. 22** weitergeleitet werden.

[0266] **Fig. 22C** ist ein Blockdiagramm eines Verarbeitungsclusters 2214 innerhalb einer Parallelverarbeitungseinheit gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist ein Verarbeitungscluster eine Instanz eines der Verarbeitungscluster 2214A-2214N aus **Fig. 22**. In mindestens einer

Ausführungsform kann ein Verarbeitungscluster 2214 konfiguriert sein, um viele Threads parallel auszuführen, wobei sich der Begriff „Thread“ auf eine Instanz eines bestimmten Programms bezieht, das auf einem bestimmten Satz von Eingangsdaten ausgeführt wird. In mindestens einer Ausführungsform werden Einzelanweisungs-Mehrfachdaten ((Single Instruction, Multiple Data; SIMD)-Befehlsausgabetechniken verwendet, die parallele Ausführung einer großen Anzahl von Threads zu unterstützen, ohne mehrere unabhängige Befehlseinheiten bereitzustellen. In mindestens einer Ausführungsform werden Einzelanweisungs-Mehrfachthread (Single Instruction, Multiple Thread; SIMT)-Techniken verwendet, die parallele Ausführung einer großen Anzahl von im Allgemeinen synchronisierten Threads zu unterstützen, wobei eine gemeinsame Befehlseinheit verwendet wird, die konfiguriert ist, um Anweisungen an eine Anzahl von Verarbeitungs-Engines innerhalb jedes Verarbeitungsclusters auszugeben.

[0267] In mindestens einer Ausführungsform kann der Betrieb eines Verarbeitungsclusters 2214 über einen Pipeline-Verwalter 2232 gesteuert werden, der Verarbeitungsaufgaben an SIMT-Parallelprozessoren verteilt. In mindestens einer Ausführungsform empfängt ein Pipeline-Verwalter 2232 Anweisungen von einem Planer 2210 von **Fig. 22** und verwaltet die Ausführung dieser Anweisungen über einen Graphik-Multiprozessor 2234 und/oder eine Textureinheit 2236. In mindestens einer Ausführungsform ist der Graphik-Multiprozessor 2234 eine beispielhafte Instanz eines SIMT-Parallelprozessors. In mindestens einer Ausführungsform können jedoch verschiedene Arten von SIMT-Parallelprozessoren unterschiedlicher Architekturen innerhalb eines Verarbeitungsclusters 2214 umfasst sein. In mindestens einer Ausführungsform können eine oder mehrere Instanzen eines Graphik-Multiprozessors 2234 in einem Verarbeitungscluster 2214 umfasst sein. In mindestens einer Ausführungsform kann ein Graphik-Multiprozessor 2234 Daten verarbeiten und kann eine Datenquerschiene 2240 verwendet, um verarbeitete Daten an eines von mehreren möglichen Zielen, einschließlich anderer Shader-Einheiten, zu verteilen. In mindestens einer Ausführungsform kann ein Pipeline-Verwalter 2232 die Verteilung der verarbeiteten Daten erleichtern, indem er Ziele für verarbeitete Daten spezifiziert, die über eine Datenquerschiene 2240 zu verteilen sind.

[0268] In mindestens einer Ausführungsform kann jeder Graphik-Multiprozessor 2234 innerhalb des Verarbeitungsclusters 2214 einen identischen Satz funktionaler Ausführungslogik (z. B. Arithmetik-Logikeinheiten, Lade-Speicher-Einheiten usw.) umfassen. In mindestens einer Ausführungsform kann funktionale Ausführungslogik nach Art des Pipeline-Prinzips konfiguriert sein, in welcher neue Anweisungen ausgegeben werden können, bevor frühere Anweisungen beendet sind. In mindestens einer Ausführungsform unterstützt die funktionale Ausführungslogik mehrere Operationen, einschließlich Ganzzahl- und Gleitkommaarithmetik, Vergleichsoperationen, boolesche Operationen, Bit-Shifting und Berechnung verschiedener algebraischer Funktionen. In mindestens einer Ausführungsform kann die gleiche Hardware der funktionalen Einheit wirksam genutzt werden, um verschiedene Operationen durchzuführen, und jede beliebige Kombination von funktionalen Einheiten kann vorhanden sein.

[0269] In mindestens einer Ausführungsform bilden an einen Verarbeitungscluster 2214 übertragene Anweisungen einen Thread. In mindestens einer Ausführungsform ist ein Satz von Threads, die über einen Satz von Parallelverarbeitungs-Engines ausgeführt werden, eine Thread-Gruppe. In mindestens einer Ausführungsform führt die Thread-Gruppe ein Programm an verschiedenen Eingangsdaten aus. In mindestens einer Ausführungsform kann jeder Thread innerhalb einer Thread-Gruppe einer unterschiedlichen Verarbeitungs-Engine innerhalb eines Graphik-Multiprozessors 2234 zugeordnet sein. In mindestens einer Ausführungsform kann eine Thread-Gruppe weniger Threads umfassen als eine Anzahl von Verarbeitungs-Engines innerhalb eines Graphik-Multiprozessors 2234. In mindestens einer Ausführungsform kann, wenn eine Thread-Gruppe weniger Threads als eine Anzahl von Verarbeitungs-Engines enthält, eine oder mehrere der Verarbeitungs-Engines während der Zyklen, in welchen diese Thread-Gruppe verarbeitet wird, im Leerlauf sein. In mindestens einer Ausführungsform kann eine Thread-Gruppe mehr Threads als eine Anzahl von Verarbeitungs-Engines innerhalb eines Graphik-Multiprozessors 2234 umfassen. In mindestens einer Ausführungsform kann, wenn eine Thread-Gruppe mehr Threads als eine Anzahl von Verarbeitungs-Engines innerhalb eines Graphik-Multiprozessors 2234 umfasst, die Verarbeitung über aufeinanderfolgende Taktzyklen durchgeführt werden. In mindestens einer Ausführungsform können mehrere Thread-Gruppen nebenläufig auf einem Graphik-Multiprozessor 2234 ausgeführt werden.

[0270] In mindestens einer Ausführungsform umfasst ein Graphik-Multiprozessor 2234 einen internen Cache-Speicher, um Lade- und Speicheroperationen durchzuführen. In mindestens einer Ausführungsform kann ein Graphik-Multiprozessor 2234 auf einen internen Cache verzichten und einen Cache-Speicher (z. B. ein L1-Cache 2248) innerhalb eines Verarbeitungsclusters 2214 verwenden. In mindestens einer Ausführungsform weist jeder Graphik-Multiprozessor 2234 auch Zugriff auf L2-Caches innerhalb von Partitionseinheiten auf (z. B. den Partitionseinheiten 2220A-2220N von **Fig. 22**), die unter allen Verarbeitungsclustern

2214 gemeinsam genutzt werden und zum Übertragen von Daten zwischen Threads verwendet werden können. In mindestens einer Ausführungsform kann ein Graphik-Multiprozessor 2234 auch auf einen globalen Speicher außerhalb des Chips zugreifen, der einen oder mehrere lokale Parallelprozessorspeicher und/oder Systemspeicher umfassen kann. In mindestens einer Ausführungsform kann jeder Speicher, der außerhalb einer Parallelverarbeitungseinheit 2202 liegt, als globaler Speicher verwendet werden. In mindestens einer Ausführungsform umfasst ein Verarbeitungscluster 2214 mehrere Instanzen des Graphik-Multiprozessors 2234, der gemeinsame Anweisungen und Daten austauschen kann, welche in dem LI-Cache 2248 gespeichert sein können.

[0271] In mindestens einer Ausführungsform kann jeder Verarbeitungscluster 2214 eine MMU 2245 (Speicherverwaltungseinheit) umfassen, die konfiguriert ist, um virtuelle Adressen auf physikalische Adressen abzubilden. In mindestens einer Ausführungsform können sich eine oder mehrere Instanzen einer MMU 2245 innerhalb einer Speicherschnittstelle 2218 von **Fig. 22** befinden. In mindestens einer Ausführungsform umfasst eine MMU 2245 einen Satz von Seitentableneinträgen (Page Table Entries; PTEs), die verwendet werden, um eine virtuelle Adresse auf eine physikalische Adresse einer Kachel und optional einen Cache-Zeilenindex abzubilden. In mindestens einer Ausführungsform kann eine MMU 2245 Adressübersetzungs-Lookaside-Puffer (TLB) oder Caches umfassen, die sich in dem Graphik-Multiprozessor 2234 oder in dem LI-Cache 2248 oder in dem Verarbeitungscluster 2214 befinden können. In mindestens einer Ausführungsform wird eine physikalische Adresse verarbeitet, um einen Oberflächendatenzugriff lokal zu verteilen, um eine effiziente Verschachtelung von Anforderungen zwischen Partitionierungseinheiten 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 ein Nichttreffer ist.

[0272] In mindestens einer Ausführungsform kann ein Verarbeitungscluster 2214 derart konfiguriert sein, dass jeder Graphik-Multiprozessor 2234 mit einer Textureinheit 2236 zum Durchführen von Texturmappingoperationen, z. B. einem Bestimmen von Texturprobenpositionen, einem Lesen von Texturdaten und einem Filtern von Texturdaten, gekoppelt ist. In mindestens einer Ausführungsform werden Texturdaten aus einem internen Textur-LI-Cache (nicht gezeigt) oder aus einem LI-Cache innerhalb eines Graphik-Multiprozessors 2234 gelesen und bei Bedarf aus einem L2-Cache, einem lokalen Parallelprozessorspeicher oder einem Systemspeicher geholt. In mindestens einer Ausführungsform gibt jeder Graphik-Multiprozessor 2234 verarbeitete Aufgaben an eine Datenquerschiene 2240 aus, um eine verarbeitete Aufgabe einem anderen Verarbeitungscluster 2214 zur weiteren Verarbeitung bereitzustellen oder um eine verarbeitete Aufgabe über die Speicherquerschiene 2216 in einem L2-Cache, lokalen Parallelprozessorspeicher oder Systemspeicher zu speichern. In mindestens einer Ausführungsform ist eine preROP 2242 (Pre-Raster Operations Unit) bzw. Vorrasteroperationseinheit konfiguriert, um Daten von einem Graphik-Multiprozessor 2234 zu empfangen und Daten an ROP-Einheiten zu leiten, welche mit Partitionierungseinheiten, wie hier beschrieben, angeordnet sein können (z. B. die Partitionierungseinheiten 2220A-2220N von **Fig. 22**). In mindestens einer Ausführungsform kann eine PreROP-Einheit 2242 Optimierungen zur Farbmischung durchführen, Pixelfarbdaten zu organisieren und Adressübersetzungen durchführen.

[0273] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915 in einem Graphikverarbeitungscluster 2214 zum Inferenzieren oder Vorhersagen von Operationen verwendet werden, die mindestens teilweise auf Gewichtungsparemtern basieren, die unter Verwendung von Trainingsoperationen für neuronale Netzwerke, Funktionen und/oder Architekturen neuronaler Netzwerke oder hier beschriebenen Anwendungsfällen neuronaler Netzwerke berechnet wurden.

[0274] **Fig. 22D** zeigt einen Graphik-Multiprozessor 2234 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform koppelt ein Graphik-Multiprozessor 2234 mit einem Pipeline-Verwalter 2232 eines Verarbeitungsclusters 2214. In mindestens einer Ausführungsform weist ein Graphik-Multiprozessor 2234 eine Ausführungspipeline auf, die einen Anweisungs-Cache 2252, eine Anweisungseinheit 2254, eine Adressenzuordnungseinheit 2256, eine Registerdatei 2258, einen oder mehrere Vielzweck-Graphikverarbeitungseinheit GPGPU-Kerne 2262 und eine oder mehrere Lade-/Speicher-Einheiten 2266 umfasst, ohne darauf beschränkt zu sein. Die GPGPU-Kerne 2262 und die Lade-/Speicher-Einheiten 2266 sind mit einem Cache-Speicher 2272 und einem gemeinsam genutzten Speicher 2270 über einen Speicher und eine Cache-Zwischenverbindung 2268 gekoppelt.

[0275] In mindestens einer Ausführungsform empfängt ein Anweisungs-Cache 2252 einen Strom von Anweisungen, die von einem Pipeline-Verwalter 2232 auszuführen sind. In mindestens einer Ausführungsform werden Anweisungen in einem Anweisungs-Cache 2252 zwischengespeichert und zur Ausführung durch eine Anweisungseinheit 2254 bereitgestellt. In mindestens einer Ausführungsform kann eine Anweisungseinheit 2254 Anweisungen als Thread-Gruppen (z. B. Warps) senden, wobei jeder Thread der Thread-Gruppe einer anderen Ausführungseinheit innerhalb einer GPGP 2262 zugeordnet ist. In mindestens einer Ausführungsform kann eine Anweisung auf einen beliebigen eines lokalen, gemeinsam genutzten oder globalen Adressraums zugreifen, indem sie eine Adresse innerhalb eines einheitlichen Adressraums spezifiziert. In mindestens einer Ausführungsform kann eine Adressenzuordnungseinheit 2256 verwendet werden, um Adressen in einem einheitlichen Adressraum in eine eindeutige Speicheradresse zu übersetzen, auf die von Lade-/Speicher-Einheiten 2266 zugegriffen werden kann.

[0276] In mindestens einer Ausführungsform stellt eine Registerdatei 2258 einen Satz von Registern für funktionale Einheiten bzw. Funktionseinheiten des Graphik-Multiprozessors 2234 bereit. In mindestens einer Ausführungsform stellt eine Registerdatei 2258 einen temporären Speicher für Operanden bereit, die mit Datenpfaden von Funktionseinheiten (z. B. GPGPU-Kernen 2262, Lade-/Speicher-Einheiten 2266) eines Graphik-Multiprozessors 2234 verbunden sind. In mindestens einer Ausführungsform ist eine Registerdatei 2258 auf jede der Funktionseinheiten aufgeteilt, so dass jeder Funktionseinheit ein dedizierter Abschnitt einer Registerdatei 2258 zugeordnet ist. In einer Ausführungsform ist eine Registerdatei 2258 zwischen verschiedenen Warps aufgeteilt, die von einem Graphik-Multiprozessor 2234 ausgeführt werden.

[0277] In mindestens einer Ausführungsform können GPGPU-Kerne 2262 jeweils Gleitkommaeinheiten (FPUs) und/oder Ganzzahlarithmetik-Logikeinheiten (ALUs) umfassen, die zum Ausführen von Anweisungen des Graphik-Multiprozessors 2234 verwendet werden. Die GPGPU-Kerne 2262 können in der Architektur ähnlich sein oder sich in der Architektur unterscheiden. In mindestens einer Ausführungsform umfasst ein erster Abschnitt der GPGPU-Kerne 2262 eine FPU einfacher Genauigkeit und eine Ganzzahl-ALU, während ein zweiter Abschnitt der GPGPU-Kerne eine FPU doppelter Genauigkeit umfasst. In mindestens einer Ausführungsform können FPUs den IEEE 754-1208-Standard für Gleitkommaarithmetik implementieren oder eine Gleitkommaarithmetik mit variabler Genauigkeit ermöglichen. In mindestens einer Ausführungsform kann ein Graphik-Multiprozessor 2234 zusätzlich eine oder mehrere Einheiten mit fester Funktion oder Spezialfunktion zum Durchführen bestimmter Funktionen, wie beispielsweise Kopierrechteck- oder Pixelblendoperationen, umfassen. In mindestens einer Ausführungsform kann bzw. können ein oder mehrere GPGPU-Kerne 2262 auch Logik für eine feste oder spezielle Funktion umfassen.

[0278] In mindestens einer Ausführungsform umfassen die GPGPU-Kerne 2262 eine SIMD-Logik, die in der Lage ist, eine einzelne Anweisung auf bzw. für mehrere Datensätze durchzuführen. In mindestens einer Ausführungsform können GPGPU-Kerne 2262 SIMD4-, SIMD8- und SIMD16-Anweisungen physikalisch ausführen und SIMD1-, SIMD2- und SIMD32-Anweisungen logisch ausführen. In mindestens einer Ausführungsform können SIMD-Anweisungen für GPGPU-Cores zur Kompilierungszeit von einem Shader-Compiler generiert oder automatisch bei dem Ausführen von Programmen erzeugt 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. Beispielsweise können in mindestens einer Ausführungsform acht SIMT-Threads, die gleiche oder ähnliche Operationen durchführen, über eine einzige SIMD8-Logikeinheit parallel ausgeführt werden.

[0279] In mindestens einer Ausführungsform ist eine Speicher- und Cache-Zwischenverbindung 2268 ein Zwischenverbindungsnetzwerk, das jede Funktionseinheit eines Graphik-Multiprozessors 2234 mit der Registerdatei 2258 und einem gemeinsamen Speicher 2270 verbindet. In mindestens einer Ausführungsform ist eine Speicher- und Cache-Zwischenverbindung 2268 eine Querschienen-Zwischenverbindung, die einer Lade-/Speicher-Einheit 2266 ermöglicht, Lade- und Speicheroperationen zwischen einem gemeinsamen Speicher 2270 und einer Registerdatei 2258 zu implementieren. In mindestens einer Ausführungsform kann eine Registerdatei 2258 mit der gleichen Frequenz wie die GPGPU-Kerne 2262 arbeiten, so dass die Datenübertragung zwischen den GPGPU-Kernen 2262 und einer Registerdatei 2258 eine sehr geringe Latenzzeit aufweist. In mindestens einer Ausführungsform kann ein gemeinsamer Speicher 2270 verwendet werden, um die Kommunikation zwischen Threads zu ermöglichen, die auf Funktionseinheiten innerhalb eines Graphik-Multiprozessors 2234 ausgeführt werden. In mindestens einer Ausführungsform kann ein Cache-Speicher 2272 beispielsweise als ein Datencache verwendet werden, um Texturdaten zwischenspeichern, die zwischen Funktionseinheiten und einer Textureinheit 2236 ausgetauscht werden. In mindestens einer Ausführungsform kann ein gemeinsam genutzter Speicher 2270 auch als ein programmverwalteter Cache ver-

wendet werden. In mindestens einer Ausführungsform können Threads, die auf GPGPU-Kernen 2262 ausgeführt werden, Daten innerhalb eines gemeinsam genutzten Speichers programmatisch zusätzlich zu automatisch zwischengespeicherten Daten speichern, die innerhalb eines Cache-Speichers 2272 gespeichert sind.

[0280] In mindestens einer Ausführungsform ist ein Parallelprozessor oder eine GPGPU, wie hier beschrieben, kommunikativ mit Host/Prozessor-Kernen gekoppelt, um Graphikoperationen, Maschinenlernoperationen, Musteranalyseoperationen und verschiedene allgemeine GPU (GPGPU)-Funktionen zu beschleunigen. In mindestens einer Ausführungsform kann die GPU kommunikativ mit den Host/Prozessor-Kernen über einen Bus oder eine andere Zwischenverbindung (z. B. eine Hochgeschwindigkeits-Zwischenverbindung wie PCIe oder NVLink) gekoppelt sein. In mindestens einer Ausführungsform kann eine GPU in bzw. auf demselben Gehäuse oder Chip wie die Kerne integriert und kommunikativ mit Kernen über einen internen Prozessorbust/eine interne Zwischenverbindung (z. B. intern in Bezug auf das Gehäuse oder den Chip) gekoppelt sein. In mindestens einer Ausführungsform können Prozessorkerne unabhängig von der Art und Weise, wie eine GPU verbunden ist, einer GPU Arbeit in Form von Sequenzen von Befehlen/Anweisungen zuweisen, die in einem Arbeitsdeskriptor umfasst sind. In mindestens einer Ausführungsform verwendet eine GPU dann dedizierte Schaltkreise/Logik zum effizienten Verarbeiten dieser Befehle/Anweisungen.

[0281] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915 in einem Graphik-Multiprozessor 2234 verwendet werden, um Operationen zu inferenzieren oder vorherzusagen, die mindestens teilweise auf Gewichtungsparametern basieren, die unter Verwendung von Trainingsoperationen für neuronale Netzwerke, Funktionen und/oder Architekturen neuronaler Netzwerke oder hier beschriebenen Anwendungsfällen neuronaler Netzwerke berechnet wurden.

[0282] **Fig. 23** veranschaulicht ein Multi-GPU-Computersystem 2300 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann das Multi-GPU-Computersystem 2300 einen Prozessor 2302 umfassen, der über einen Host-Schnittstellen-Schalter 2304 mit mehreren Allzweck-Graphikverarbeitungseinheiten (GPGPUs) 2306A-D gekoppelt ist. In mindestens einer Ausführungsform ist der Host-Schnittstellen-Schalter 2304 eine PCI-Express-Schaltervorrichtung, die den Prozessor 2302 an einen PCI-Express-Bus koppelt, über welchen der Prozessor 2302 mit den GPGPUs 2306A-D kommunizieren kann. GPGPUs 2306A-D können sich über eine Anzahl von Hochgeschwindigkeits-Punkt-zu-Punkt-GPU-zu-GPU-Verbindungen 2316 zwischenverbinden. In mindestens einer Ausführungsform verbinden sich die P2P-Verbindungen 2316 über eine dedizierte GPU-Verbindung mit jeder der GPGPUs 2306A-D. In mindestens einer Ausführungsform ermöglichen die P2P-GPU-Verbindungen 2316 eine direkte Kommunikation zwischen jeder der GPGPUs 2306A-D, ohne eine Kommunikation über den Host-Schnittstellenbus 2304 zu erfordern, mit welchem der Prozessor 2302 verbunden ist. In mindestens einer Ausführungsform bleibt mit GPU-zu-GPU-Datenaufkommen, das auf die P2P-GPU-Verbindungen 2316 gerichtet ist, der Host-Schnittstellenbus 2304 für Systemspeicherzugriff oder zur Kommunikation mit anderen Instanzen des Multi-GPU-Computersystems 2300, beispielsweise, über eine oder mehrere Netzwerkvorrichtung(en) verfügbar. Während in mindestens einer Ausführungsform die GPGPUs 2306A-D über den Host-Schnittstellen-Schalter 2304 mit dem Prozessor 2302 verbinden, umfasst in mindestens einer Ausführungsform der Prozessor 2302 direkte Unterstützung für die P2P-GPU-Verbindungen 2316 und kann direkt mit den GPGPUs 2306A-D verbunden werden.

[0283] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915 in dem Multi-GPU-Computersystem 2300 verwendet werden, um Operationen zu inferenzieren oder vorherzusagen, die mindestens teilweise auf Gewichtungsparametern basieren, die unter Verwendung von Trainingsoperationen für neuronale Netzwerke, Funktionen und/oder Architekturen neuronaler Netzwerke oder hier beschriebenen Anwendungsfällen neuronaler Netzwerke berechnet wurden.

[0284] **Fig. 24** ist ein Blockdiagramm eines Graphikprozessors 2400 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform umfasst der Graphikprozessor 2400 eine Ringzwischenverbindung 2402, ein Pipeline-Frontend 2404, eine Medien-Engine 2437 und Graphikkern 2480A-2480N. In mindestens einer Ausführungsform koppelt die Ringzwischenverbindung 2402 den Graphikprozessor 2400 an andere Verarbeitungseinheiten, einschließlich anderer Graphikprozessoren oder einen oder mehrere All-

zweck-Prozessorkerne. In mindestens einer Ausführungsform ist der Graphikprozessor 2400 einer von vielen Prozessoren, die innerhalb eines Multicore-Verarbeitungssystems integriert sind.

[0285] In mindestens einer Ausführungsform empfängt der Graphikprozessor 2400 Stapel von Befehlen über die Ringzwischenverbindung 2402. In mindestens einer Ausführungsform werden eingehende Befehle von einem Befehlsstreamer 2403 in dem Pipeline-Frontend 2404 interpretiert. In mindestens einer Ausführungsform umfasst der Graphikprozessor 2400 eine skalierbare Ausführungslogik zur Durchführung von 3D-Geometrieverarbeitung und Medienverarbeitung über den/die Graphikkern(e) 2480A-2480N. In mindestens einer Ausführungsform liefert der Befehlsstreamer 2403 für 3D-Geometrieverarbeitungsbefehle Befehle an die Geometrie-Pipeline 2436. In mindestens einer Ausführungsform liefert der Befehlsstreamer 2403 für mindestens einige Befehle zur Medienverarbeitung Befehle an ein Video-Frontend 2434, das mit einer Medien-Engine 2437 gekoppelt ist. In mindestens einer Ausführungsform umfasst die Medien-Engine 2437 eine Videoqualitäts-Engine (VQE) 2430 zur Video- und Bild-Nachbearbeitung und eine Multiformat-Codier-/Decodier-Engine (MFX) 2433 zur hardwarebeschleunigten Mediendatencodierung und -decodierung. In mindestens einer Ausführungsform erzeugen die Geometrie-Pipeline 2436 und die Medien-Engine 2437 jeweils Ausführungsthreads für Thread-Ausführungsressourcen, die von mindestens einem Graphikkern 2480A bereitgestellt werden.

[0286] In mindestens einer Ausführungsform umfasst der Graphikprozessor 2400 skalierbare Thread-Ausführungsressourcen mit modularen Kernen 2480A-2480N (manchmal als Kern-Slices bezeichnet), die jeweils mehrere Teilkerne 2450A-2450N, 2460A-2460N (manchmal als Kern-Subslices bezeichnet) aufweisen. In mindestens einer Ausführungsform kann der Graphikprozessor 2400 eine beliebige Anzahl von Graphikkernen 2480A bis 2480N aufweisen. In mindestens einer Ausführungsform umfasst der Graphikprozessor 2400 einen Graphikkern 2480A mit mindestens einem ersten Teilkern 2450A und einem zweiten Teilkern 2460A. In mindestens einer Ausführungsform ist der Graphikprozessor 2400 ein stromsparender Prozessor mit einem einzigen Teilkern (z. B. 2450A). In mindestens einer Ausführungsform umfasst der Graphikprozessor 2400 mehrere Graphikkern(e) 2480A-2480N, die jeweils einen Satz von ersten Teilkernen 2450A-2450N und einen Satz von zweiten Teilkernen 2460A-2460N umfassen. In mindestens einer Ausführungsform umfasst jeder Teilkern in den ersten Teilkernen 2450A-2450N mindestens einen ersten Satz von Ausführungseinheiten 2452A-2452N und Medien/Textur-Abtastern 2454A-2454N. In mindestens einer Ausführungsform umfasst jeder Teilkern in zweiten Teilkernen 2460A-2460N mindestens einen zweiten Satz von Ausführungseinheiten 2462A-2462N und Abtastern 2464A-2464N. In mindestens einer Ausführungsform teilt sich jeder Teilkern 2450A-2450N, 2460A-2460N einen Satz gemeinsamer Ressourcen 2470A-2470N. In mindestens einer Ausführungsform umfassen gemeinsam genutzte Ressourcen gemeinsam genutzten Cache-Speicher und Pixeloperationslogik.

[0287] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenzierungs- und/oder Trainingslogik 915 in dem Graphikprozessor 2400 zum Inferenzieren oder Vorhersagen von Operationen verwendet werden, die mindestens teilweise auf Gewichtungsparametern basieren, die unter Verwendung von Trainingsoperationen für neuronale Netzwerke, Funktionen und/oder Architekturen neuronaler Netzwerke oder hier beschriebenen Anwendungsfällen neuronaler Netzwerke berechnet wurden.

[0288] **Fig. 25** ist ein Blockdiagramm, das eine Mikroarchitektur für einen Prozessor 2500 veranschaulicht, der Logikschaltungen zum Ausführen von Anweisungen umfassen kann, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann ein Prozessor 2500 Anweisungen ausführen, die x86-Anweisungen, ARM-Anweisungen, speziellen Anweisungen für anwendungsspezifische integrierte Schaltungen (ASICs) usw. umfassen können. In mindestens einer Ausführungsform kann ein Prozessor 2500 Register zum Speichern gepackter Daten umfassen, wie beispielsweise 64-Bit breite MMXTM-Register in Mikroprozessoren, die mit MMX-Technologie der Intel Corporation aus Santa Clara, Kalifornien, ausgestattet sind. In mindestens einer Ausführungsform können MMX-Register, die sowohl in Ganzzahl- als auch in Gleitkomma-Form verfügbar sind, mit gepackten Datenelementen arbeiten, die Einzelanweisungs-Mehrfachdaten („SIMD“) und Streaming-SIMD-Erweiterungen („SSE“) begleiten. In mindestens einer Ausführungsform können 128-Bit breite XMM-Register, die sich auf SSE2, SSE3, SSE4, AVX oder darüber hinaus beziehen (allgemein als „SSEx“ bezeichnet), solche gepackten Datenoperanden umfassen. In mindestens einer Ausführungsform können Prozessor 2510 Anweisungen ausführen, um maschinelles Lernen oder Algorithmen für tiefes Lernen, Training oder Inferenzierung zu beschleunigen.

[0289] In mindestens einer Ausführungsform umfasst ein Prozessor 2500 ein In-Order-Frontend („Frontend“) 2501, um auszuführende Anweisungen zu holen und Anweisungen vorzubereiten, die später in der Prozess-orphipeline zu verwenden sind. In mindestens einer Ausführungsform kann ein Frontend 2501 mehrere Einheiten umfassen. In mindestens einer Ausführungsform holt ein Anweisungs-Vorabrufers bzw. -Prefetcher 2526 Anweisungen aus dem Speicher und leitet Anweisungen an einen Anweisungsdecoder 2528 weiter, der wiederum Anweisungen decodiert oder interpretiert. Beispielsweise decodiert in mindestens einer Ausführungsform ein Anweisungsdecoder 2528 eine empfangene Anweisung in eine oder mehrere Operationen, genannt „Mikroanweisungen“ oder „Mikrooperationen“ (die auch als „Mikro-Ops“ oder „Uops“ bezeichnet werden), die eine Maschine ausführen kann. In mindestens einer Ausführungsform parst bzw. analysiert der Anweisungsdecoder 2528 eine 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 Trace- bzw. Spur-Cache 2530 decodierte Uops zu programmgeordneten Sequenzen oder Spuren in einer Uop-Warteschlange 2534 zur Ausführung zusammenstellen. In mindestens einer Ausführungsform stellt dann, wenn ein Trace-Cache 2530 auf eine komplexe Anweisung trifft, ein Mikrocode-ROM 2532 die für den Abschluss der Operation erforderlichen Uops bereit.

[0290] In mindestens einer Ausführungsform können einige Anweisungen in eine einzige Mikro-Op umgewandelt werden, während andere mehrere Mikro-Operationen benötigen, um die volle Operation abzuschließen. In mindestens einer Ausführungsform kann dann, wenn mehr als vier Mikro-Ops benötigt werden, um eine Anweisung abzuschließen, ein Anweisungsdecoder 2528 auf den Mikrocode-ROM 2532 zugreifen, um eine Anweisung auszuführen. In mindestens einer Ausführungsform kann eine Anweisung zur Verarbeitung an dem Anweisungsdecoder 2521 in eine kleine Anzahl von Mikro-Ops decodiert werden. In mindestens einer Ausführungsform kann eine Anweisung in dem Mikrocode ROM 2532 gespeichert werden, falls eine Anzahl von Mikro-Operationen zur Durchführung einer Operation erforderlich sein sollte. In mindestens einer Ausführungsform bezieht sich ein Trace-Cache 2530 auf eine programmierbare Eintrittspunkt-Logikanordnung („PLA“), um einen korrekten Mikroanweisungszeiger zum Lesen von Mikrocodesequenzen zu bestimmen, um eine oder mehrere Anweisungen von einem Mikrocode ROM 2532 abzuschließen, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann, nachdem ein Mikrocode-ROM 2532 die Sequenzierung von Mikro-Ops für eine Anweisung abschließt, ein Frontend 2501 einer Maschine das Abholen von Mikro-Ops von einem Trace-Cache 2530 wieder aufnehmen kann.

[0291] In mindestens einer Ausführungsform kann eine Außer-Betrieb bzw. Out-of-Order-Ausführungs-Engine („Out-of-Order-Engine“) 2503 Anweisungen für eine Ausführung vorbereiten. In mindestens einer Ausführungsform weist die Out-of-Order-Ausführungslogik eine Anzahl von Puffern auf, um den Fluss von Anweisungen zu glätten und neu zu ordnen, um die Leistung zu optimieren, während sie eine Pipeline hinuntergehen und zur Ausführung eingeplant werden. Eine Out-of-Order-Ausführungs-Engine 2503 umfasst, ohne darauf beschränkt zu sein, einen Allokator/Register-Umbenener 2540, eine Speicher-Uop-Warteschlange 2542, eine Ganzzahl/Gleitkomma-Uop-Warteschlange 2544, einen Speicher-Planer 2546, einen Schnell-Planer 2502, einen Langsam/Universal-Gleitkomma-Planer („Langsam/Universal-FP-Planer“) 2504 und einen einfachen Gleitkomma-Planer („Einfach-FP-Planer“) 2506. In mindestens einer Ausführungsform werden ein Schnell-Planer 2502, ein Langsam/Universal-Gleitkomma-Planer 2504 und ein Einfach-Gleitkomma-Planer 2506 hier auch kollektiv als „Uop-Planer 2502, 2504, 2506“ bezeichnet. teilt der Allokator/Register-Umbenener 2540 weist Engine- bzw. Maschinenpuffer und Ressourcen zu, die jeder Uop zur Ausführung benötigt. In mindestens einer Ausführungsform benennt der Allokator/Register-Umbenener 2540 Logikregister auf Einträge in einer Registerdatei um. In mindestens einer Ausführungsform teilt der Allokator/Register-Umbenener 2540 auch einen Eintrag für jeden Uop in einer von zwei Uop-Warteschlangen, der Speicher-Uop-Warteschlange 2542 für Speicheroperationen und der Ganzzahl/Gleitkomma-Uop-Warteschlange 2544 für Nichtspeicheroperationen, vor dem Speicher-Planer 2546 und den Uop-Planern 2502, 2504, 2506 zu. In mindestens einer Ausführungsform bestimmen Uop-Planer 2502, 2504, 2506, wann ein Uop ausführbereit ist, basierend auf der Bereitschaft ihrer abhängigen Eingangsregister-Operandenquellen und der Verfügbarkeit von Ausführungsressourcen, die Uops benötigen, um ihre Operation abzuschließen. In mindestens einer Ausführungsform kann der Schnell-Planer 2502 auf jeder Hälfte eines Haupttaktzyklus planen, während ein Langsam/Universal-Gleitkomma-Planer 2504 und ein Einfach-Gleitkomma-Planer 2506 einmal pro Hauptprozessortaktzyklus planen können. In mindestens einer Ausführungsform arbitrieren die Uop-Planer 2502, 2504, 2506 für Sende-Ports, um Uops zur Ausführung einzuplanen.

[0292] In mindestens einer Ausführungsform umfasst ein Ausführungsblock 2511, ohne darauf beschränkt zu sein, ein Ganzzahl-Registerdatei/Umgehungs-Netzwerk 2508, ein Gleitkomma-Registerdatei/Umgehungsnetzwerk („FP-Registerdatei/Umgehungs-netzwerk“) 2510, Adresserzeugungseinheiten („AGUs“) 2512 und

2514, schnelle Arithmetiklogikeinheiten (ALUs) („schnelle ALUs“) 2516 und 2518, eine langsame Arithmetiklogikeinheit („langsame ALU“) 2512, eine Gleitkomma-ALU („FP“) 2522 und eine Gleitkomma-Bewegungseinheit („FP move“) 2524. In mindestens einer Ausführungsform werden ein Ganzzahl-Registerdatei/Umgehungs-Netzwerk 2508 und ein Gleitkomma-Registerdatei/Umgehungs-Netzwerk 2510 hier auch als „Registerdateien 2508, 2510“ bezeichnet. In mindestens einer Ausführungsform werden die AGUSs 2512 und 2514, die schnellen ALUs 2516 und 2518, eine langsame ALU 2512, eine Gleitkomma-ALU 2522 und eine Gleitkomma-Bewegungseinheit 2524 hier ebenfalls als „Ausführungseinheiten 2512, 2514, 2516, 2518, 2512, 2522 und 2524“ bezeichnet. In mindestens einer Ausführungsform kann ein Ausführungsblock 2511, ohne darauf beschränkt zu sein, eine beliebige Anzahl (einschließlich Null) und Art von Registerdateien, Umgehungsnetzwerken, Adresserzeugungseinheiten und Ausführungseinheiten in beliebiger Kombination umfassen.

[0293] In mindestens einer Ausführungsform können Registerdateien 2508, 2510 zwischen Uop-Planern 2502, 2504, 2506 und Ausführungseinheiten 2512, 2514, 2516, 2518, 2512, 2522 und 2524 angeordnet sein. In mindestens einer Ausführungsform führt ein Ganzzahl-Registerdatei/Umgehungs-Netzwerk 2508 Ganzzahl-Operationen aus. In mindestens einer Ausführungsform führt ein Gleitkomma-Registerdatei/Umgehungs-Netzwerk 2510 Gleitkommaoperationen durch. In mindestens einer Ausführungsform kann jedes der Registerdateien 2508, 2510, ohne darauf beschränkt zu sein, ein Umgehungs-Netzwerk umfassen, das gerade abgeschlossene Ergebnisse, die noch nicht in eine Registerdatei geschrieben wurden, umgehen oder an neue abhängige Uops weiterleiten kann. In mindestens einer Ausführungsform können Registerdateien 2508, 2510 Daten miteinander kommunizieren. In mindestens einer Ausführungsform kann das Ganzzahl-Registerdatei/Umgehungs-Netzwerk 2508, ohne darauf beschränkt zu sein, zwei separate Registerdateien, eine Registerdatei für 32 Datenbits niedriger Ordnung und eine zweite Registerdatei für 32 Datenbits hoher Ordnung, umfassen. In mindestens einer Ausführungsform kann ein Gleitkomma-Registerdatei/Umgehungs-Netzwerk 2510, ohne darauf beschränkt zu sein, 128-Bit breite Einträge umfassen, weil Gleitkommaanweisungen typischerweise Operanden von 64 bis 128 Bit Breite aufweisen.

[0294] In mindestens einer Ausführungsform können Ausführungseinheiten 2512, 2514, 2516, 2518, 2512, 2522, 2524 Anweisungen ausführen. In mindestens einer Ausführungsform speichern Registerdateien 2508, 2510 Ganzzahl- und Gleitkomma-Daten-Operandenwerte, die Mikroanweisungen ausführen müssen. In mindestens einer Ausführungsform kann ein Prozessor 2500, ohne darauf beschränkt zu sein, eine beliebige Anzahl und Kombination von Ausführungseinheiten 2512, 2514, 2516, 2518, 2512, 2522, 2524 umfassen. In mindestens einer Ausführungsform können eine Gleitkomma-ALU 2522 und eine Gleitkomma-Bewegungseinheit 2524 Gleitkomma-, MMX-, SIMD-, AVX- und SSE-Operationen oder andere Operationen ausführen, einschließlich spezieller Anweisungen für maschinelles Lernen. In mindestens einer Ausführungsform kann eine Gleitkomma ALU 2522, ohne darauf beschränkt zu sein, einen 64-Bit durch 64-Bit-Gleitkommateiler umfassen, um Divisions-, Quadratwurzel- und Rest-Mikrooperationen auszuführen. In mindestens einer Ausführungsform können Anweisungen, die einen Gleitkommawert umfassen, von Gleitkomma-Hardware gehandhabt werden. In mindestens einer Ausführungsform können ALU-Operationen an schnelle ALUs 2516, 2518 übergeben werden. In mindestens einer Ausführungsform können schnelle ALUS 2516, 2518 schnelle Operationen mit einer effektiven Latenz von einem halben Taktzyklus ausführen. In mindestens einer Ausführungsform gehen die komplexesten Ganzzahl-Operationen auf die langsame ALU 2512 über, da die langsame ALU 2512, ohne darauf beschränkt zu sein, eine Ganzzahl-Ausführungshardware für Operationen mit langer Latenz umfassen kann, wie beispielsweise einen Multiplikator, einen Schieber, Flag-Logik und Zweigverarbeitung. In mindestens einer Ausführungsform können Speicher-Lade-/Speicher-Operationen von AGUS 2512, 2514 ausgeführt werden. In mindestens einer Ausführungsform kann eine schnelle ALU 2516, eine schnelle ALU 2518 und eine langsame ALU 2512 Ganzzahl-Operationen an 64-Bit-Datenoperanden durchführen. In mindestens einer Ausführungsform können eine schnelle ALU 2516, eine schnelle ALU 2518 und eine langsame ALU 2520 implementiert sein, um mehrere Datenbitgrößen einschließlich sechzehn, zweiunddreißig, 128, 326 usw. zu unterstützen. In mindestens einer Ausführungsform können eine Gleitkomma-ALU 2522 und eine Gleitkomma-Bewegungseinheit 2524 implementiert sein, um eine Anzahl von Operanden mit Bits unterschiedlicher Breite zu unterstützen. In mindestens einer Ausführungsform können eine Gleitkomma-ALU 2522 und eine Gleitkomma-Bewegungseinheit 2524 an 128-Bit breite gepackte Datenoperanden in Verbindung mit SIMD- und Multimedia-Anweisungen arbeiten.

[0295] In mindestens einer Ausführungsform versenden Uop-Planer 2502, 2504, 2506 abhängige Operationen, bevor die Ausführung einer Elternlast beendet ist. In mindestens einer Ausführungsform kann, da Uops spekulativ geplant und in einem Prozessor 2500 ausgeführt werden können, ein Prozessor 2500 auch eine Logik zur Handhabung von Speicherfehlzugriffen umfassen. In mindestens einer Ausführungsform kann dann, wenn eine Datenlast in einem Datencache fehlt, abhängige Operationen im Lauf der Pipeline geben,

die einen Planer mit vorübergehend falschen Daten zurückgelassen haben. In mindestens einer Ausführungsform verfolgt ein Wiedergabemechanismus Anweisungen, die falsche Daten verwenden, und führt sie erneut aus. In mindestens einer Ausführungsform müssen abhängige Operationen möglicherweise wiedergegeben werden und unabhängige Operationen können abgeschlossen werden. In mindestens einer Ausführungsform können Planer und ein Wiedergabemechanismus von mindestens einer Ausführungsform eines Prozessors ebenfalls ausgelegt sein, um Anweisungssequenzen für Textkettenvergleichsoperationen abzufangen.

[0296] In mindestens einer Ausführungsform kann sich „Register“ auf On-Board-Prozessor-Speicherplätze beziehen, die als Teil von Anweisungen zum Identifizieren von Operanden verwendet werden können. In mindestens einer Ausführungsform können Register diejenigen sein, die (aus der Sicht eines Programmierers) von außerhalb des Prozessors verwendbar sind. In mindestens einer Ausführungsform sind Register möglicherweise nicht auf einen bestimmten Schaltungstyp beschränkt. Vielmehr kann in mindestens einer Ausführungsform ein Register Daten speichern, Daten bereitstellen und die hier beschriebenen Funktionen ausführen. In mindestens einer Ausführungsform können hier beschriebene Register durch Schaltkreise innerhalb eines Prozessors unter Verwendung einer beliebigen Anzahl verschiedener Techniken implementiert sein, wie beispielsweise dedizierte physikalische Register, dynamisch allokierte physikalische Register, die Registerumbenennung verwenden, Kombinationen von dedizierten und dynamisch zugeordneten physikalischen Registern usw. In mindestens einer Ausführungsform speichern Ganzzahlregister 32-Bit-Ganzzahldaten. Eine Registerdatei mindestens einer Ausführungsform enthält darüber hinaus acht Multimedia-SIMD-Register für gepackte Daten.

[0297] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform können Abschnitte oder die gesamte Inferenzierungs- und/oder Trainingslogik 915 in einem EXE-Block 2511 und anderem Speicher oder Registern integriert sein, die gezeigt oder nicht gezeigt sind. Beispielsweise können in mindestens einer Ausführungsform hier beschriebene Trainings- und/oder Inferenzierungstechniken eine oder mehrere der in dem EXE-Block 2511 veranschaulichten ALUs verwenden. Außerdem können Gewichtungsparemeter in einem On-Chip- oder Off-Chip-Speicher und/oder in Registern (gezeigt oder nicht gezeigt) gespeichert sein, die ALUs eines EXE-Blocks 2511 konfigurieren, um einen oder mehrere Algorithmen maschinellen Lernens, neuronale Netzwerkarchitekturen, Anwendungsfälle oder hier beschriebene Trainingstechniken auszuführen.

[0298] **Fig. 26** veranschaulicht einen tief lernenden bzw. Deep-Learning-Anwendungsprozessor 2600 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform verwendet ein Deep-Learning-Anwendungsprozessor 2600 Anweisungen, die dann, wenn sie durch einen Deep-Learning-Anwendungsprozessor 2600 ausgeführt werden, einen Deep-Learning-Anwendungsprozessor 2600 veranlassen, einige oder alle der in dieser Offenbarung beschriebenen Prozesse und Techniken auszuführen. In mindestens einer Ausführungsform ist ein Deep-Learning-Anwendungsprozessor 2600 eine anwendungsspezifische integrierte Schaltung (ASIC). In mindestens einer Ausführungsform führt der Anwendungsprozessor 2600 Matrix-Multiplikationsoperationen entweder „fest verdrahtet“ in Hardware oder als ein Ergebnis der Durchführung einer oder mehrerer Anweisungen oder beides aus. In mindestens einer Ausführungsform umfasst der Deep-Learning-Anwendungsprozessor 2600, ohne darauf beschränkt zu sein, Verarbeitungscluster 2610 (1)-2610(12), Inter-Chip-Verbindungen („ICLs“) 2612(1)-2612(12), Inter-Chip-Controller („ICCs“) 2630 (1)-2630(2), Speicher mit hoher Bandbreite einer zweiten Generation („HBM2“) 2640(1)-2640(4), Speicher-Controller („Mem Ctrls“) 2642(1)-2642(4), eine physikalische Schicht von Speicher mit hoher Bandbreite („HBM PHY“) 2644(1)-2644(4), eine Verwaltungs-Controller-Zentralverarbeitungseinheit („Verwaltungs-Controller-CPU“) 2650, eine serielle Peripheriegeräte-Schnittstelle, eine integrierte Inter-Schaltung und einen Allzweck-Eingabe-/Ausgabeblock („SPI, I²C, GPIO“) 2660, eine Peripheriekomponenten-Zwischenverbindungs-Express-Controller und einen Direktspeicherzugriffsblock („PCIe-Controller und DMA“) 2670, sowie einen sechzehnkanaligen Peripheriekomponenten-Zwischenverbindungs-Express-Port („PCIe x 16“ Port) 2680.

[0299] In mindestens einer Ausführungsform können Verarbeitungscluster 2610 Deep-Learning-Operationen durchführen, einschließlich Inferenz- oder Vorhersage-Operationen basierend auf Gewichtungsparemetern, die basierend auf einer oder mehreren Trainingstechniken berechnet wurden, einschließlich der hier beschriebenen. In mindestens einer Ausführungsform kann jeder Verarbeitungscluster 2610, ohne darauf beschränkt zu sein, eine beliebige Anzahl und Art von Prozessoren umfassen. In mindestens einer Ausführungsform kann ein Deep-Learning-Anwendungsprozessor 2600 eine beliebige Anzahl und Art von Verarbei-

tungsclustern 2600 umfassen. In mindestens einer Ausführungsform sind Inter-Chip-Verbindungen 2612 bidirektional. In mindestens einer Ausführungsform ermöglichen Inter-Chip-Verbindungen 2612 und Inter-Chip-Controller 2630 mehreren Deep-Learning-Anwendungs-Prozessoren 2600, Informationen auszutauschen, einschließlich von Aktivierungsinformationen, die aus der Ausführung eines oder mehrerer Algorithmen maschinellen Lernens resultieren, die in einem oder mehreren neuronalen Netzwerken verkörpert sind. In mindestens einer Ausführungsform kann ein Deep-Learning-Anwendungsprozessor 2600 eine beliebige Anzahl (einschließlich Null) und Art von ICLs 2620 und ICCs 2630 umfassen.

[0300] In mindestens einer Ausführungsform stellen HBM2s 2640 insgesamt 32 Gigabyte (GB) Speicher zur Verfügung. Ein HBM2 2640(i) ist sowohl der Speichersteuerung 2642(i) als auch der HBM PHY 2644(i) zugeordnet. In mindestens einer Ausführungsform kann eine beliebige Anzahl von HBM2s 2640 eine beliebige Art und Gesamtmenge an Speicher mit hoher Bandbreite bereitstellen und kann einer beliebigen Anzahl (einschließlich Null) und Art von Speicher-Controllern 2642 und HBM PHYs 2644 zugeordnet sein. In mindestens einer Ausführungsform können ein SPI, I²C, GPIO 2660, PCIe-Controller und DMA 2670 und/oder ein PCIe 2680 durch eine beliebige Anzahl und Art von Blöcken ersetzt werden, die eine beliebige Anzahl und Art von Kommunikationsstandards in beliebiger technisch machbaren Weise ermöglichen.

[0301] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform wird der Deep-Learning-Anwendungsprozessor verwendet, um ein Modell maschinellen Lernens, wie beispielsweise ein neuronales Netzwerk, zu trainieren, Informationen vorherzusagen oder zu inferenzieren, die dem Deep-Learning-Anwendungsprozessor 2600 bereitgestellt werden. In mindestens einer Ausführungsform wird der Deep-Learning-Anwendungsprozessor 2600 verwendet, um Informationen basierend auf einem trainierten Modell maschinellen Lernens (z. B. einem neuronalen Netzwerk), das von einem anderen Prozessor oder System oder von dem Deep-Learning-Anwendungsprozessor 2600 trainiert wurde, abzuleiten oder vorherzusagen. In mindestens einer Ausführungsform kann ein Prozessor 2600 verwendet werden, um einen oder mehrere der hier beschriebenen Anwendungsfälle neuronaler Netzwerke durchzuführen.

[0302] **Fig. 27** ist ein Blockdiagramm eines neuromorphen Prozessors 2700 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann ein neuromorpher Prozessor 2700 ein oder mehrere Eingaben von Quellen empfangen, die zu einem neuromorphen Prozessor 2700 extern sind. In mindestens einer Ausführungsform können diese Eingaben an ein oder mehrere Neuronen 2702 innerhalb eines neuromorphen Prozessors 2700 übertragen werden. In mindestens einer Ausführungsform können die Neuronen 2702 und ihre Komponenten unter Verwendung von Schaltungen oder Logik, einschließlich einer oder mehrerer Arithmetik-Logikeinheiten (ALUs), implementiert sein. In mindestens einer Ausführungsform kann ein neuromorpher Prozessor 2700, ohne darauf beschränkt zu sein, Tausende oder Millionen von Instanzen von Neuronen 2702 umfassen, wobei jedoch eine beliebige Anzahl von Neuronen 2702 verwendet werden kann. In mindestens einer Ausführungsform kann jede Instanz eines Neurons 2702 einen Neuroneneingang 2704 und einen Neuronenausgang 2706 umfassen. In mindestens einer Ausführungsform können Neuronen 2702 Ausgaben erzeugen, die an Eingänge anderer Instanzen von Neuronen 2702 übertragen werden können. Beispielsweise können in mindestens einer Ausführungsform die Neuronen-Eingänge 2704 und die Neuronenausgänge 2706 über Synapsen 2708 miteinander verbunden sein.

[0303] In mindestens einer Ausführungsform können Neuronen 2702 und Synapsen 2708 derart miteinander verbunden sein, dass ein neuromorpher Prozessor 2700 zum Verarbeiten oder Analysieren der von einem neuromorphen Prozessor 2700 empfangenen Informationen eingesetzt wird. In mindestens einer Ausführungsform können Neuronen 2702 einen Ausgangsimpuls (oder „Feuern“ oder „Spike“) senden, wenn durch einen Neuroneneingang 2704 empfangene Eingaben einen Schwellenwert überschreiten. In mindestens einer Ausführungsform können Neuronen 2702 Signale, die an den Neuroneneingängen 2704 empfangen wurden, summieren oder integrieren. Beispielsweise können in mindestens einer Ausführungsform Neuronen 2702 als lecke Integrations- und Feuereuronen implementiert sein, wobei dann, wenn eine Summe (als ein „Membranpotential“ bezeichnet) einen Schwellenwert überschreitet, ein Neuron 2702 unter Verwendung einer Übertragungsfunktion, wie beispielsweise einer Sigmoid- oder Schwellenwertfunktion, eine Ausgabe (oder ein „Feuern“) erzeugen kann. In mindestens einer Ausführungsform kann ein leckes Integrations- und Feuereuron Signale, die an den Neuroneneingängen 2704 empfangen wurden, zu einem Membranpotential summieren und kann darüber hinaus einen Zerfallsfaktor (oder ein Leck) anwenden, um ein Membranpotential zu verringern. In mindestens einer Ausführungsform kann ein leckes Integrations- und Feuereuron feuern, wenn mehrere Eingangssignale an Neuroneneingängen 2704 schnell genug empfangen werden, um

einen Schwellenwert zu überschreiten (z. B. bevor ein Membranpotential zu niedrig zum Feuern wird). In mindestens einer Ausführungsform können Neuronen 2702 unter Verwendung von Schaltungen oder Logik implementiert sein, die Eingaben empfangen, Eingaben zu einem Membranpotential integrieren und ein Membranpotential abbauen. In mindestens einer Ausführungsform können Eingaben gemittelt werden oder eine beliebige andere geeignete Übertragungsfunktion kann verwendet werden. Des Weiteren können in mindestens einer Ausführungsform, ohne darauf beschränkt zu sein, Neuronen 2702 Komparatorschaltungen oder Logik umfassen, die einen Ausgangsspike an dem Neuronenausgang 2706 erzeugen, wenn ein Ergebnis der Anwendung einer Übertragungsfunktion auf den Neuroneneingang 2704 einen Schwellenwert überschreitet. In mindestens einer Ausführungsform kann, nachdem ein Neuron 2702 feuert, es zuvor empfangene Eingangsinformationen ignorieren, indem es beispielsweise ein Membranpotenzial auf 0 oder einen anderen geeigneten Standardwert zurücksetzt. In mindestens einer Ausführungsform kann, nachdem das Membranpotential auf 0 zurückgesetzt ist, ein Neuron 2702 nach einer geeigneten Zeitspanne (oder einer Refraktärperiode) den Normalbetrieb wieder aufnehmen.

[0304] In mindestens einer Ausführungsform können Neuronen 2702 durch Synapsen 2708 miteinander verbunden sein. In mindestens einer Ausführungsform können die Synapsen 2708 betrieben werden, um Signale von einem Ausgang eines ersten Neurons 2702 zu einem Eingang eines zweiten Neurons 2702 zu übertragen. In mindestens einer Ausführungsform können Neuronen 2702 Informationen über mehr als eine Instanz einer Synapse 2708 übertragen. In mindestens einer Ausführungsform können ein oder mehrere Instanzen eines Neuronenausgangs 2706 über eine Instanz der Synapse 2708 mit einer Instanz des Neuroneneingangs 2704 im gleichen Neuron 2702 verbunden sein. In mindestens einer Ausführungsform kann eine Instanz des Neurons 2702, die eine Ausgabe erzeugt, die über eine Instanz einer Synapse 2708 übertragen werden soll, als ein „präsynaptisches Neuron“ in Bezug auf diese Instanz einer Synapse 2708 bezeichnet werden. In mindestens einer Ausführungsform kann eine Instanz eines Neurons 2702, die eine über eine Instanz einer Synapse 2708 übertragene Eingabe empfängt, in Bezug auf diese Instanz einer Synapse 2708 als „postsynaptisches Neuron“ bezeichnet sein. Weil eine Instanz eines Neurons 2702 Eingaben von einer oder mehreren Instanzen der Synapse 2708 empfangen kann und ebenfalls Ausgaben über eine oder mehrere Instanzen einer Synapse 2708 übertragen kann, kann eine einzelne Instanz eines Neurons 2702 daher sowohl ein „präsynaptisches Neuron“ als auch ein „postsynaptisches Neuron“ in Bezug auf verschiedene Instanzen von Synapsen 2708 in mindestens einer Ausführungsform sein.

[0305] In mindestens einer Ausführungsform können die Neuronen 2702 in einer oder mehreren Schichten organisiert sein. Jede Instanz eines Neurons 2702 kann einen Neuronenausgang 2706 aufweisen, der sich durch eine oder mehrere Synapsen 2708 zu einem oder mehreren Neuroneneingängen 2704 ausbreiten kann. In mindestens einer Ausführungsform können Neuronenausgänge 2706 von Neuronen 2702 in einer ersten Schicht 2710 mit Neuroneneingängen 2704 von Neuronen 2702 in einer zweiten Schicht 2712 verbunden sein. In mindestens einer Ausführungsform kann eine Schicht 2710 als eine „Feed-Forward-Schicht“ bezeichnet sein. In mindestens einer Ausführungsform kann sich jede Instanz eines Neurons 2702 in einer Instanz einer ersten Schicht 2710 zu jeder Instanz eines Neurons 2702 in einer zweiten Schicht 2712 ausbreiten. In mindestens einer Ausführungsform kann eine erste Schicht 2710 als „vollständig verbundene Feed-Forward-Schicht“ bezeichnet sein. In mindestens einer Ausführungsform kann sich jede Instanz eines Neurons 2702 in einer Instanz einer zweiten Schicht 2712 auf weniger als alle Instanzen eines Neurons 2702 in einer dritten Schicht 2714 verteilen. In mindestens einer Ausführungsform kann eine zweite Schicht 2712 als eine „spärlich verbundene Feed-Forward-Schicht“ bezeichnet sein. In mindestens einer Ausführungsform können sich Neuronen 2702 in einer zweiten Schicht 2712 zu Neuronen 2702 in mehreren anderen Schichten ausbreiten, einschließlich zu Neuronen 2702 in einer (gleichen) zweiten Schicht 2712. In mindestens einer Ausführungsform kann eine zweite Schicht 2712 als „rekurrente Schicht“ bezeichnet sein. Ein neuromorpher Prozessor 2700 kann, ohne darauf beschränkt zu sein, jede geeignete Kombination von rekurrenten Schichten und Feed-Forward-Schichten umfassen, einschließlich, ohne darauf beschränkt zu sein, sowohl spärlich verbundene Feed-Forward-Schichten als auch vollständig verbundene Feed-Forward-Schichten.

[0306] In mindestens einer Ausführungsform kann ein neuromorpher Prozessor 2700, ohne darauf beschränkt zu sein, eine rekonfigurierbare Zwischenverbindungsarchitektur oder dedizierte fest verdrahtete Zwischenverbindungen umfassen, um eine Synapse 2708 mit Neuronen 2702 zu verbinden. In mindestens einer Ausführungsform kann ein neuromorpher Prozessor 2700, ohne darauf beschränkt zu sein, Schaltungen oder Logik umfassen, die ermöglicht, Synapsen bei Bedarf verschiedenen Neuronen 2702 basierend auf neuronaler Netzwerktopologie und Neuronen-Fan-In/Out zuzuordnen. Beispielsweise können in mindestens einer Ausführungsform Synapsen 2708 mit Neuronen 2702 unter Verwendung einer Zwischenverbindungsstruktur, wie beispielsweise einem Netzwerk auf dem Chip, oder mit speziellen Verbindungen verbunden

sein. In mindestens einer Ausführungsform können Synapsen-Zwischenverbindungen und Komponenten derselben unter Verwendung von Schaltungen oder Logik implementiert sein.

[0307] Fig. 28 ist ein Blockdiagramm eines Graphikprozessors 2800, der eine diskrete Graphikverarbeitungseinheit sein kann oder ein Graphikprozessor sein kann, der mit mehreren Verarbeitungskernen integriert sein kann. In mindestens einer Ausführungsform kommuniziert ein Graphikprozessor 2800 über eine speicherabgebildete E/A-Schnittstelle mit Registern auf einem Graphikprozessor 2800 und mit Befehlen, die im Speicher platziert sind. In mindestens einer Ausführungsform umfasst ein Graphikprozessor 2800 eine Speicherschnittstelle 2814, um auf den Speicher zuzugreifen. In mindestens einer Ausführungsform ist die Speicherschnittstelle 2814 eine Schnittstelle zu einem lokalen Speicher, einem oder mehreren internen Caches, einem oder mehreren gemeinsam genutzten externen Caches und/oder zu dem Systemspeicher.

[0308] In mindestens einer Ausführungsform umfasst ein Graphikprozessor 2800 ebenfalls einen Anzeigecontroller 2802 zum Treiben von Anzeigebildern zu einer Anzeigevorrichtung 2820. In mindestens einer Ausführungsform umfasst ein Anzeigecontroller 2802 Hardware für eine oder mehrere Überlagerungsebenen für die Anzeige 2820 und eine Zusammenstellung mehrerer Schichten von Video oder Anwenderschnittstellenelementen. In mindestens einer Ausführungsform kann eine Anzeigevorrichtung 2820 eine interne oder eine externe Anzeigevorrichtung sein. In mindestens einer Ausführungsform ist eine Anzeigevorrichtung 2820 eine am Kopf angebrachte Anzeigevorrichtung, wie beispielsweise eine Anzeigevorrichtung für virtuelle Realität (VR) oder eine Anzeigevorrichtung für erweiterte Realität (AR). In mindestens einer Ausführungsform umfasst ein Graphikprozessor 2800 eine Video-Codec-Engine 2806 zum Codieren, Decodieren oder Transcodieren von Medien in, aus oder zwischen einem oder mehreren Mediencodierungsformaten, die umfassen, ohne jedoch darauf beschränkt zu sein, sowohl Formate der „Moving Picture Experts Group“ (MPEG) wie beispielsweise MPEG-2, „Advanced Video Coding“- (AVC-) Formate wie beispielsweise H.264/MPEG-4 AVC, sowie auch Formate der „Society of Motion Picture & Television Engineers“ (SMPTE) 421M/VC-1 und „Joint Photographic Experts Group“ (JPEG) wie beispielsweise JPEG und Motion JPEG (MJPEG) umfassen.

[0309] In mindestens einer Ausführungsform umfasst ein Graphikprozessor 2800 eine Blockbildübertragungs- (BLIT-) Engine 2804 zum Durchführen zweidimensionaler (2D) Rasterisierungs-Operationen, die beispielsweise Bitgrenzenblockübertragungen umfassen. In mindestens einer Ausführungsform werden 2D-Graphikoperationen jedoch unter Verwendung einer oder mehrerer Komponenten der Graphikverarbeitungs-Engine (GPE) 2810 ausgeführt. In mindestens einer Ausführungsform ist eine GPE 2810 eine Rechen-Engine zum Durchführen von Graphikoperationen, die dreidimensionale (3D-) Graphikoperationen und Medienoperationen umfassen.

[0310] In mindestens einer Ausführungsform umfasst eine GPE 2810 eine 3D-Pipeline 2812 zum Durchführen von 3D-Operationen wie beispielsweise Rendern dreidimensionaler Bilder und Szenen unter Verwendung von Verarbeitungsfunktionen, die auf 3D-Grundformen (beispielsweise Rechteck, Dreieck usw.) arbeiten. Eine 3D-Pipeline 2812 umfasst programmierbare Elemente und Elemente mit fester Funktion, die verschiedene Aufgaben durchführen und/oder Ausführungs-Threads zu einem 3D/Medien-Teilsystem 2815 erzeugen. Während eine 3D-Pipeline 2812 verwendet werden kann, um Medienoperationen durchzuführen, umfasst in mindestens einer Ausführungsform eine GPE 2810 ebenfalls eine Medien-Pipeline 2816, die verwendet wird, um Medienoperationen durchzuführen, wie beispielsweise Videonachverarbeitung und Bildverbesserung.

[0311] In mindestens einer Ausführungsform umfasst eine Medien-Pipeline 2816 Logikeinheiten mit fester Funktion oder programmierbare Logikeinheiten zum Durchführen einer oder mehrerer spezialisierter Medienoperationen, wie beispielsweise Video-Decodierungsbeschleunigung, Video-Entschachtelung und Video-Codierungsbeschleunigung anstelle der oder im Auftrag einer Video-Codec-Engine 2806. In mindestens einer Ausführungsform umfasst eine Medien-Pipeline 2816 zusätzlich eine Thread-Erzeugungseinheit zum Erzeugen von Threads zur Ausführung auf dem 3D/Medien-Teilsystem 2815. In mindestens einer Ausführungsform führen die erzeugten Threads Berechnungen für die Medienoperationen auf einer oder mehreren Graphikausführungseinheiten aus, die in dem 3D/Medien-Teilsystem 2815 umfasst sind.

[0312] In mindestens einer Ausführungsform umfasst ein 3D/Medien-Teilsystem 2815 Logik zum Ausführen von Threads, die durch eine 3D-Pipeline 2812 und eine Medien-Pipeline 2816 erzeugt werden. In mindestens einer Ausführungsform senden eine 3D-Pipeline und eine Medien-Pipeline 2816 Thread-Ausführungsanforderungen an ein 3D/Medien-Teilsystem 2815, das die Thread-Verteilungslogik zum Arbitrieren und Verteilen verschiedener Anforderungen an verfügbare Thread-Ausführungsressourcen umfasst. In mindestens einer

Ausführungsform umfassen die Ausführungsressourcen eine Anordnung von Graphikausführungseinheiten zum Verarbeiten der 3D- und Medien-Threads. In mindestens einer Ausführungsform umfasst ein 3D-Medien-Teilsystem 2815 einen oder mehrere interne Caches für Thread-Anweisungen und Daten. In mindestens einer Ausführungsform umfasst ein Teilsystem 2815 ebenfalls einen gemeinsam genutzten Speicher, der Register und adressierbaren Speicher umfasst, um Daten durch Threads gemeinsam zu nutzen und Ausgabedaten zu speichern.

[0313] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform können Abschnitte der oder die gesamte Inferenzierungs- und/oder Trainingslogik 915 in dem Graphikprozessor 2800 integriert sein. Beispielsweise können in mindestens einer Ausführungsform hier beschriebene Trainings- und/oder Inferenzierungs-Techniken eine oder mehrere ALUs verwenden, die in der 3D-Pipeline 2812 verkörpert sind. Außerdem können in mindestens einer Ausführungsform hier beschriebene Inferenzierungs- und/oder Trainingsoperationen mit einer anderen Logik als der in **Fig. 9A** oder **Fig. 9B** veranschaulichten Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtungsparmeter in On-Chip- oder Off-Chip-Speicher und/oder Registern (gezeigt oder nicht gezeigt) gespeichert sein, die ALUs eines Graphikprozessors 2800 konfigurieren, um einen oder mehrere Algorithmen maschinellen Lernens, Lernalgorithmen, neuronale Netzwerkarchitekturen, Anwendungsfälle oder hier beschriebenen Trainingstechniken auszuführen.

[0314] **Fig. 29** ist ein Blockdiagramm einer Graphikverarbeitungs-Engine 2910 eines Graphikprozessors gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist die Graphikverarbeitungs-Engine (GPE) 2910 eine Version der in **Fig. 28** gezeigten GPE 2810. In mindestens einer Ausführungsform ist eine Medien-Pipeline 2816 optional und kann in einer GPE 2910 nicht ausdrücklich umfasst sein. In mindestens einer Ausführungsform ist ein separater Medien- und/oder Bildprozessor mit einer GPE 2910 gekoppelt.

[0315] In mindestens einer Ausführungsform ist eine GPE 2910 mit einem Befehls-Streamer 2903 gekoppelt oder umfasst ihn, der einen Befehlsstrom zu einer 3D-Pipeline 2812 und/oder Medien-Pipeline 2816 bereitstellt. In mindestens einer Ausführungsform ist ein Befehls-Streamer 2903 mit einem Speicher gekoppelt, der ein Systemspeicher oder einer oder mehrere aus internem Cache-Speicher und gemeinsam genutzter Cache-Speicher sein kann. In mindestens einer Ausführungsform empfängt ein Befehls-Streamer 2903 Befehle von dem Speicher und sendet die Befehle an eine 3D-Pipeline 2812 und/oder Medien-Pipeline 2816. In mindestens einer Ausführungsform sind die Befehle Anweisungen, Primitive oder Mikro-Operationen, die aus einem Ringpuffer geholt werden, der Befehle für eine 3D-Pipeline 2812 und eine Medien-Pipeline 2816 speichert. In mindestens einer Ausführungsform kann ein Ringpuffer zusätzlich Stapelbefehlspuffer umfassen, die Stapel aus mehreren Befehlen speichern. In mindestens einer Ausführungsform können Befehle für eine 3D-Pipeline 2812 außerdem Referenzen auf im Speicher gespeicherte Daten umfassen, wie beispielsweise, ohne jedoch darauf beschränkt zu sein, Vertex- und Geometriedaten für eine 3D-Pipeline 2812 und/oder Bilddaten und Speicherobjekte für eine Medien-Pipeline 2816. In mindestens einer Ausführungsform verarbeiten eine 3D-Pipeline 2812 und eine Medien-Pipeline 2816 Befehle und Daten durch Durchführen von Operationen oder durch Verteilen eines oder mehrerer Ausführungs-Threads an eine Graphikkernanordnung 2914. In mindestens einer Ausführungsform umfasst eine Graphikkernanordnung 2914 einen oder mehrere Blöcke von Graphikkernen (beispielsweise Graphikkern(e) 2915A, Graphikkern(e) 2915B), wobei jeder Block einen oder mehrere Graphikkern(e) umfasst. In mindestens einer Ausführungsform umfasst jeder Graphikkern einen Satz von Graphikausführungsressourcen, die Allzweck- und graphikspezifische Ausführungslogik zum Durchführen von Graphik- und Rechenoperationen sowie auch Beschleunigungslogik für Texturverarbeitung mit fester Funktion und/oder maschinelles Lernen und künstliche Intelligenz umfasst, die Inferenzierungs- und/oder Trainingslogik 915 in **Fig. 9A** und **Fig. 9B** umfassen.

[0316] In mindestens einer Ausführungsform umfasst eine 3D-Pipeline 2812 Logik mit fester Funktion und programmierbare Logik, um ein oder mehrere Shader-Programme zu verarbeiten, wie beispielsweise Vertex-Shader, Geometrie-Shader, Pixel-Shader, Fragment-Shader, Rechen-Shader oder andere Shader-Programme, durch Verarbeiten der Anweisungen und Verteilen von Ausführungs-Threads an eine Graphikkernanordnung 2914. In mindestens einer Ausführungsform stellt eine Graphikkernanordnung 2914 einen einheitlichen Block von Ausführungsressourcen zum Gebrauch zum Verarbeiten dieser Shader-Programme bereit. In mindestens einer Ausführungsform umfasst Mehrzweck-Ausführungslogik (beispielsweise eine Ausführungseinheit(en)) innerhalb eines Graphikkerns(en) 2915A-2915B einer Graphikkernanordnung 2914 Unter-

stützung für verschiedene 3D-API-Shader-Sprachen und kann mehrere gleichzeitige Ausführungs-Threads ausführen, die mehreren Shadern zugeordnet sind.

[0317] In mindestens einer Ausführungsform umfasst eine Graphikkernanordnung 2914 ebenfalls Ausführungslogik zum Durchführen von Medienfunktionen, wie beispielsweise Video- und/oder Bildverarbeitung. In mindestens einer Ausführungsform umfassen die Ausführungseinheiten Allzwecklogik, die programmierbar ist, um parallele Allzweckberechnungsoperationen zusätzlich zu Graphikverarbeitungsoperationen durchzuführen.

[0318] In mindestens einer Ausführungsform können Ausgabedaten, die durch Threads erzeugt werden, die auf einer Graphikkernanordnung 2924 ausgeführt werden, Daten an einen Speicher in einem vereinheitlichten Rückgabepuffer (Unified Return Buffer; URB) 2918 ausgeben. Ein URB 2918 kann Daten für mehrere Threads speichern. In mindestens einer Ausführungsform kann ein URB 2918 verwendet werden, um Daten zwischen unterschiedlichen Threads, die auf einer Graphikkernanordnung 2914 ablaufen, zu senden. In mindestens einer Ausführungsform kann ein URB 2918 zusätzlich zur Synchronisation zwischen Threads auf einer Graphikkernanordnung 2914 und einer Logik mit fester Funktion innerhalb einer Logik 2920 mit gemeinsam genutzter Funktion verwendet werden.

[0319] In mindestens einer Ausführungsform ist eine Graphikkernanordnung 2914 skalierbar, so dass eine Graphikkernanordnung 2914 eine variable Anzahl von Graphikkernen umfasst, von denen jeder eine variable Anzahl von Ausführungseinheiten basierend auf einer Zielleistung und dem Leistungsfähigkeitsniveau einer GPE 2910 aufweist. In mindestens einer Ausführungsform sind Ausführungsressourcen dynamisch skalierbar, so dass Ausführungsressourcen je nach Bedarf aktiviert oder deaktiviert werden können.

[0320] In mindestens einer Ausführungsform ist eine Graphikkernanordnung 2914 mit einer Logik 2920 mit gemeinsam genutzter Funktion gekoppelt, die mehrere Ressourcen umfasst, die zwischen Graphikkernen in einer Graphikkernanordnung 2914 gemeinsam genutzt werden. In mindestens einer Ausführungsform sind gemeinsam genutzte Funktionen, die von gemeinsam genutzter Funktionslogik Logik 2920 durchgeführt werden, in Hardware-Logikeinheiten verkörpert, die spezialisierte ergänzende Funktionalität für eine Graphikkernanordnung 2914 bereitstellen. In mindestens einer Ausführungsform umfasst die Logik 2920 mit gemeinsam genutzter Funktion, ohne jedoch darauf beschränkt zu sein, Logik für Sampler 2921, Math 2922 und Thread-übergreifende Kommunikation (ITC) 2923. In mindestens einer Ausführungsform sind ein oder mehrere Cache(s) 2925 in der Logik 2920 mit gemeinsam genutzter Funktion umfasst oder mit dieser gekoppelt.

[0321] In mindestens einer Ausführungsform wird eine gemeinsam genutzte Funktion verwendet, wenn der Bedarf für eine spezialisierte Funktion zum Aufnehmen innerhalb einer Graphikkernanordnung 2914 nicht ausreichend ist. In mindestens einer Ausführungsform wird eine einzige Instanziierung dieser spezialisierten Funktion als eine eigenständige Instanziierung in der Logik 2920 für gemeinsam genutzte Funktion implementiert und wird von den Ausführungsressourcen innerhalb einer Graphikkernanordnung 2914 gemeinsam genutzt. In mindestens einer Ausführungsform können spezifische gemeinsam genutzte Funktionen innerhalb der Logik 2920 mit gemeinsam genutzter Funktion, die von einer Graphikkernanordnung 2914 extensiv verwendet werden, in der Logik 2916 mit gemeinsam genutzter Funktion innerhalb einer Graphikkernanordnung 2914 umfasst sein. In mindestens einer Ausführungsform kann die Logik 2916 mit gemeinsam genutzter Funktion in einer Graphikkernanordnung 2914 einen Teil der oder die gesamte Logik innerhalb der Logik 2920 mit gemeinsam genutzter Funktion umfassen. In mindestens einer Ausführungsform können alle Logikelemente innerhalb der Logik 2920 mit gemeinsam genutzter Funktion innerhalb der Logik 2916 mit gemeinsam genutzter Funktion einer Graphikkernanordnung 2914 dupliziert sein. In mindestens einer Ausführungsform ist die Logik 2920 mit gemeinsam genutzter Funktion zugunsten der Logik 2916 mit gemeinsam genutzter Funktion innerhalb einer Graphikkernanordnung 2914 ausgeschlossen.

[0322] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform können Abschnitte der oder die gesamte Inferenzierungs- und/oder Trainingslogik 915 in einer Graphikverarbeitungs-Engine 2910 integriert sein. Beispielsweise können in mindestens einer Ausführungsform hier beschriebene Trainings- und/oder Inferenzierungs-Techniken eine oder mehrere der ALUs verwenden, die in der 3D-Pipeline 2812, Graphikkern(en) 2915, gemeinsam genutzter Funktionslogik 2926, gemeinsam genutzter Funktionslogik 2920 oder anderer Logik in **Fig. 29** verkörpert sind. Außerdem können in mindestens einer Ausführungsform die hier beschriebenen Inferenzierungs- und/oder Trainingsoperationen mit einer anderen Logik als der in **Fig. 9A**

oder **Fig. 9B** veranschaulichten Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtungparameter in On-Chip- oder Off-Chip-Speicher und/oder Registern (gezeigt oder nicht gezeigt) gespeichert werden, die ALUs einer Graphikverarbeitungs-Engine 2910 konfigurieren, um einen oder mehrere Algorithmen maschinellen Lernens, neuronale Netzwerkarchitekturen, Anwendungsfälle oder hier beschriebenen Trainingstechniken auszuführen.

[0323] **Fig. 30** ist ein Blockdiagramm von Hardwarelogik eines Graphikprozessorkerns 3000 gemäß mindestens einer hier beschriebenen Ausführungsform. In mindestens einer Ausführungsform ist ein Graphikprozessorkern 3000 innerhalb einer Graphikkernanordnung enthalten. In mindestens einer Ausführungsform kann ein Graphikprozessorkern 3000, manchmal auch als ein Kern-Slice bezeichnet, ein oder mehrere Graphikkernkerne innerhalb eines modularen Graphikprozessors sein. In mindestens einer Ausführungsform ist ein Graphikprozessorkern 3000 beispielhaft für ein Graphikkern-Slice, und ein Graphikprozessor, wie hier beschrieben, kann mehrere Graphikkern-Slices basierend auf einer Sollleistung und Leistungshüllen umfassen. In mindestens einer Ausführungsform kann jeder Graphikkern 3000 einen Festfunktionsblock 3030 umfassen, der mit mehreren, auch als Sub-Slices bezeichneten Teilkernen 3701A-3701F gekoppelt ist, die modulare Blöcke von Allzweck- und Festfunktionslogik umfassen.

[0324] In mindestens einer Ausführungsform umfasst ein Festfunktionsblock 3030 eine Geometrie-/Festfunktions-Pipeline 3036, die von allen Teilkernen in einem Graphikprozessor 3000 beispielsweise bei Implementierungen von Graphikprozessoren mit geringerer Leistung und/oder geringerer Stromaufnahme gemeinsam genutzt werden kann. In mindestens einer Ausführungsform umfasst die Geometrie-/Festfunktions-Pipeline 3036 eine 3D-Festfunktions-Pipeline, eine Video-Frontend-Einheit, einen Thread-Spawner und Thread-Abfertiger sowie einen Unified Return Buffer Manager, der vereinheitlichte Rückgabepuffer verwaltet.

[0325] In mindestens einer Ausführungsform umfasst ein Festfunktionsblock 3030 ebenfalls eine Graphik--SoC-Schnittstelle 3031, einen Graphik-Mikrocontroller 3038 und eine Medien-Pipeline 3039. Eine Graphik--SoC-Schnittstelle 3037 stellt eine Schnittstelle zwischen einem Graphikkern 3000 und anderen Prozessorkernen innerhalb einer integrierten SoC-Schaltung bereit. In mindestens einer Ausführungsform ist ein Graphik--Mikrocontroller 3038 ein programmierbarer Teilprozessor, der konfiguriert sein kann, um verschiedene Funktionen des Graphikprozessors 3000 zu verwalten, einschließlich der Thread-Versendung, des Scheduling und der Präemption. In mindestens einer Ausführungsform umfasst eine Medien-Pipeline 3039 Logik zur Erleichterung der Decodierung, Codierung, Vorverarbeitung und/oder Nachbearbeitung von Multimedia-Daten, einschließlich Bild- und Videodaten. In mindestens einer Ausführungsform implementiert eine Medien-Pipeline 3039 Medienoperationen über Anforderungen an Berechnungs- oder Abtast-Logik innerhalb der Teilkern 3001-3001F.

[0326] In mindestens einer Ausführungsform ermöglicht eine SoC-Schnittstelle 3037 einem Graphikkern 3000 mit Allzweck-Anwendungsprozessorkernen (z. B. CPUs) und/oder anderen Komponenten innerhalb eines SoC, einschließlich Speicherhierarchieelementen, wie beispielsweise gemeinsam genutztem Last-Level-Cache-Speicher, System-RAM und/oder eingebettetem On-Chip- oder On-Package-DRAM, zu kommunizieren. In mindestens einer Ausführungsform kann eine Graphik-SoC-Schnittstelle 3037 ebenfalls Kommunikation mit Vorrichtungen mit fester Funktion innerhalb eines SoC ermöglichen, wie beispielsweise Kameraabbildungspipelines, und ermöglicht die Verwendung und/oder Implementierung globaler Speicheratomare, die zwischen einem Graphikkern 3000 und CPUs innerhalb eines SoC gemeinsam genutzt werden können. In mindestens einer Ausführungsform kann eine SoC-Schnittstelle 3037 ebenfalls Energieverwaltungssteuerungen für einen Graphikkern 3000 implementieren und eine Schnittstelle zwischen einer Taktdomäne eines Graphikkerns 3000 und anderen Taktdomänen innerhalb eines SoC ermöglichen. In mindestens einer Ausführungsform ermöglicht die SoC-Schnittstelle 3037 den Empfang von Befehlspuffern von einem Befehlsstreamer und einem globalen Thread-Versender, die konfiguriert sind, um Befehle und Anweisungen an jeden eines einzelnen oder mehrerer Graphikkernkerne innerhalb eines Graphikprozessors bereitzustellen. In mindestens einer Ausführungsform können Befehle und Anweisungen an eine Medien-Pipeline 3039, wenn Medienoperationen durchzuführen sind, oder an eine Geometrie- und Festfunktions-Pipeline (z. B. eine Geometrie- und Festfunktions-Pipeline 3036, eine Geometrie- und Festfunktions-Pipeline 3014), wenn Graphikverarbeitungsoperationen durchzuführen sind, gesendet werden.

[0327] In mindestens einer Ausführungsform kann ein Graphik-Mikrocontroller 3038 konfiguriert sein, um verschiedene Planungs- und Verwaltungsaufgaben für einen Graphikkern 3000 auszuführen. In mindestens einer Ausführungsform kann der Graphik-Mikrocontroller 3038 eine Graphik- und/oder Rechen-Arbeitslast-Planung auf verschiedenen parallelen Graphik-Engines innerhalb der Ausführungseinheit (EU)-Anordnungen 3002A-3002F, 3004A-3004F innerhalb der Teilkern 3001A-3001F durchführen. In mindestens einer Ausführungsform

rungsform kann eine Host-Software, die auf einem CPU-Kern eines SoC mit einem Graphikkern 3000 ausgeführt wird, Arbeitslasten einer von mehreren Graphikprozessor-Doorbells übergeben, welche einen Planungsvorgang auf einer geeigneten Graphik-Engine auslöst. In mindestens einer Ausführungsform umfassen Planungsoperationen ein Bestimmen, welche Arbeitslast als nächstes auszuführen ist, ein Übermitteln einer Arbeitslast an einen Befehlsstreamer, ein Vorbelegen existierender Arbeitslasten, die auf einer Engine ausgeführt werden, ein Überwachen des Fortschreitens einer Arbeitslast und ein Benachrichtigen der Host-Software, wenn eine Arbeitslast abgeschlossen ist. In mindestens einer Ausführungsform kann ein Graphik-Mikrocontroller 3038 ebenfalls Zustände mit geringer Stromaufnahme oder Leerlaufzustände für einen Graphikkern 3000 ermöglichen, die einem Graphikkern 3000 die Möglichkeit geben, Register innerhalb eines Graphikkerns 3000 über Zustandsübergänge mit geringem Stromverbrauch unabhängig von einem Betriebssystem und/oder einer Graphiktreibersoftware auf einem System zu speichern und wiederherzustellen.

[0328] In mindestens einer Ausführungsform kann ein Graphikkern 3000 mehr als oder weniger als die veranschaulichten Teilkern 3001A-3001F und bis hin zu N modularen Teilkernen aufweisen. Für jeden Satz von N-Teilkernen kann ein Graphikkern 3000 in mindestens einer Ausführungsform auch eine gemeinsam genutzte Funktionslogik 3010, einen gemeinsam genutzten und/oder Cache-Speicher 3012, eine Geometrie-/Festfunktions-Pipeline 3014 sowie zusätzliche Festfunktionslogik 3016 zur Beschleunigung verschiedener Graphik- und Rechenverarbeitungsoperationen umfassen. In mindestens einer Ausführungsform kann die gemeinsam genutzte Funktionslogik 3010 Logikeinheiten (z. B. Abtaster-, Mathematik- und/oder Inter-Thread-Kommunikationslogik) umfassen, die von allen N Teilkernen innerhalb eines Graphikkerns 3000 gemeinsam genutzt werden können. Ein gemeinsam genutzter und/oder Cache-Speicher 3012 kann ein Last-Level-Cache für die N Teilkern 3001A-3001F innerhalb des Graphikkerns 3000 sein und kann ebenfalls als ein gemeinsam genutzter Speicher dienen, auf den mehrere Teilkern zugreifen können. In mindestens einer Ausführungsform kann die Geometrie/Festfunktions-Pipeline 3014 anstelle einer Geometrie/Festfunktions-Pipeline 3036 innerhalb des Festfunktionsblocks 3030 aufgenommen sein und kann gleiche oder ähnliche Logikeinheiten umfassen.

[0329] In mindestens einer Ausführungsform umfasst ein Graphikkern 3000 zusätzliche Festfunktionslogik 3016, die verschiedene Festfunktionsbeschleunigungslogik zur Verwendung durch einen Graphikkern 3000 umfassen kann. In mindestens einer Ausführungsform umfasst die zusätzliche Festfunktionslogik in Nur-Positions-Shading. Bei dem Nur-Positions-Shading existieren mindestens zwei Geometrie-Pipelines, wohingegen in einer Vollgeometrie-Pipeline innerhalb einer Geometrie/Festfunktions-Pipeline 3016, 3036, und eine Cull-Pipeline, welche eine zusätzliche Geometrie-Pipeline ist, die innerhalb der zusätzlichen Festfunktionslogik 3016 umfasst sein kann. In mindestens einer Ausführungsform ist die Cull-Pipeline eine abgespeckte Version einer Vollgeometrie-Pipeline. In mindestens einer Ausführungsform können eine Vollpipeline und eine Cull-Pipeline verschiedene Instanzen einer Anwendung ausführen, wobei jede Instanz einen separaten Kontext hat. In mindestens einer Ausführungsform kann das Nur-Positions-Shading lange Cull-Läufe verworfener Dreiecke verbergen, so dass Shading in einigen Fällen früher abgeschlossen werden kann. Beispielsweise kann in mindestens einer Ausführungsform Cull-Pipeline-Logik innerhalb der zusätzlichen Festfunktionslogik 3016 Positions-Shader parallel zu einer Hauptanwendung ausführen und erzeugt im Allgemeinen kritische Ergebnisse schneller als eine Vollpipeline, da eine Cull-Pipeline ein Positionsattribut von Vertices holt und schattiert, ohne eine Rasterisierung und ein Rendering von Pixeln in einem Einzelbild-Puffer durchzuführen. In mindestens einer Ausführungsform kann eine Cull-Pipeline erzeugte kritische Ergebnisse verwenden, um Sichtbarkeitsinformationen für alle Dreiecke unabhängig davon zu berechnen, ob diese Dreiecke gecullt werden. In mindestens einer Ausführungsform kann eine Vollpipeline (welche in diesem Fall als eine Wiedergabepipeline bezeichnet werden kann) Sichtbarkeitsinformationen verbrauchen, um gecullte Dreiecke zu überspringen, um nur sichtbare Dreiecke zu schattieren, die schließlich an eine Rasterisierungsphase übergeben werden.

[0330] In mindestens einer Ausführungsform kann die zusätzliche Festfunktionslogik 3016 auch eine maschinell lernende Beschleunigungslogik, wie beispielsweise eine Festfunktion-Matrix-Multiplikationslogik, für Implementierungen einschließlich Optimierungen für das Training oder die Inferenzierung des maschinellen Lernens umfassen.

[0331] In mindestens einer Ausführungsform umfasst jeder Graphik-Teilkern 3001A-3001F einen Satz von Ausführungsressourcen, die verwendet werden können, um Graphik-, Medien- und Rechenoperationen im Ansprechen auf Anfragen von der Graphikpipeline, der Medien-Pipeline oder von Shader-Programmen durchzuführen. In mindestens einer Ausführungsform umfassen die Graphik-Teilkern 3001A-3001F mehrere EU-Arrays 3002A-3002F, 3004A-3004F, Thread-Abfertigungs- und Inter-Thread-Kommunikationslogik 3003A-3003F, einen 3D-Abtaster 3005A-3005F, einen Medien-Abtaster 3006A-3006F, einen Shaderprozess-

sor 3007A-3007F und einen gemeinsam genutzten lokalen Speicher (SLM) 3008A-3008F. EU-Arrays 3002A-3002F, 3004A-3004F umfassen jeweils mehrere Ausführungseinheiten, welche Allzweck-Graphikverarbeitungseinheiten sind, die in der Lage sind, Gleitkomma- und Ganzzahl/Festpunkt-Logikoperationen im Dienst einer Graphik-, Medien- oder Rechenoperation durchzuführen, einschließlich von Graphik-, Medien- oder Rechen-Shader-Programmen. In mindestens einer Ausführungsform führt eine TD/IC-Logik 3003A-3003F lokale Thread-Abfertigungs- und Thread-Steuer-Operationen für Ausführungseinheiten innerhalb eines Teilkerns durch und erleichtert die Kommunikation zwischen Threads, die auf Ausführungseinheiten eines Teilkerns ausgeführt werden. In mindestens einer Ausführungsform kann der 3D-Abtaster 3005A-3005F Texturen oder andere 3D-Graphikdaten in den Speicher einlesen. In mindestens einer Ausführungsform kann ein 3D-Abtaster Texturdaten unterschiedlich lesen, basierend auf einem konfigurierten Abtastzustand und einem Texturformat, das einer bestimmten Textur zugeordnet ist. In mindestens einer Ausführungsform kann ein Medien-Abtaster 3006A-3006F ähnliche Leseoperationen durchführen, die auf einem Typ und einem Format basieren, der/das Mediendaten zugeordnet ist. In mindestens einer Ausführungsform kann jeder Graphik-Teilkern 3001A-3001F abwechselnd einen einheitlichen 3D- und Medien-Abtaster umfassen. In mindestens einer Ausführungsform können Threads, die auf Ausführungseinheiten innerhalb jedes der Teilkerns 3001A-3001F ausgeführt werden, den gemeinsam genutzten lokalen Speicher 3008A-3008F innerhalb jedes Teilkerns verwenden, um Threads, die innerhalb einer Thread-Gruppe ausgeführt werden, die Ausführung unter Verwendung eines gemeinsamen Pools von On-Chip-Speicher zu ermöglichen.

[0332] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform können Abschnitte der oder die gesamte Inferenzierungs- und/oder Trainingslogik 915 in den Graphikprozessor 3010 integriert sein. Beispielsweise können in mindestens einer Ausführungsform hier beschriebene Trainings- und/oder Inferenzierungs-Techniken eine oder mehrere der ALUs verwenden, die in einer 3D-Pipeline 3010, einem Graphik-Mikrocontroller 3038, einer Geometrie & Festfunktions-Pipeline 3014 und 3036 oder anderer Logik in **Fig. 29** verkörpert sind. Außerdem können in mindestens einer Ausführungsform die hier beschriebenen Inferenzierungs- und/oder Trainingsoperationen mit einer anderen Logik als der in **Fig. 9A** oder **Fig. 9B** veranschaulichten Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtungparameter in On-Chip- oder Off-Chip-Speicher und/oder Registern (gezeigt oder nicht gezeigt) gespeichert werden, die ALUs eines Graphikprozessors 3000 konfigurieren, um einen oder mehrere Algorithmen maschinellen Lernens, neuronale Netzwerkarchitekturen, Anwendungsfälle oder die hier beschriebenen Trainingstechniken auszuführen.

[0333] **Fig. 31A-31B** veranschaulichen eine Threadausführungslogik 3100 einschließlich einer Anordnung von Verarbeitungselementen eines Graphikprozessorkerns gemäß mindestens einer Ausführungsform. **Fig. 31A** veranschaulicht mindestens eine Ausführungsform, bei welcher die Threadausführungslogik 3100 verwendet wird. **Fig. 31B** veranschaulicht beispielhafte interne Einzelheiten einer Ausführungseinheit gemäß mindestens einer Ausführungsform.

[0334] Wie in **Fig. 31A** veranschaulicht, umfasst die Thread-Ausführungslogik 3100 in mindestens einer Ausführungsform einen Shader-Prozessor 3102, einen Thread-Abfertiger 3104, einen Anweisungs-Cache 3106, eine Anordnung skalierbarer Ausführungseinheiten mit einer Mehrzahl von Ausführungseinheiten 3107A-3107N, einen Abtaster bzw. Sampler 3110, einen Daten-Cache 3112 und einen Datenport 3114. In mindestens einer Ausführungsform kann eine Anordnung skalierbarer Ausführungseinheiten dynamisch skaliert werden, indem eine oder mehrere Ausführungseinheiten (z. B. eine der Ausführungseinheiten 3108A, 3108B, 3108C, 3108D bis 3108N-1 und 3108N) basierend auf Berechnungsanforderungen einer Arbeitslast aktiviert oder deaktiviert werden. In mindestens einer Ausführungsform sind skalierbare Ausführungseinheiten über eine Zwischenverbindungsstruktur, die mit jeder der Ausführungseinheiten verbunden ist, miteinander verbunden. In mindestens einer Ausführungsform umfasst eine Thread-Ausführungslogik 3100 eine oder mehrere Verbindungen zu Speicher, wie beispielsweise Systemspeicher oder Cache-Speicher, über eine oder mehrere eines Anweisungs-Caches 3106, eines Datenports 3114, eines Abtasters 3110 und Ausführungseinheiten 3108A-3108N. In mindestens einer Ausführungsform ist jede Ausführungseinheit (z. B. 3108A) eine eigenständige, programmierbare, Allzweck-Recheneinheit, die in der Lage ist, mehrere gleichzeitige Hardware-Threads auszuführen und gleichzeitig mehrere Datenelemente für jeden Thread parallel zu verarbeiten. In mindestens einer Ausführungsform ist eine Anordnung von Ausführungseinheiten 3108A-3108N skalierbar, um eine beliebige Anzahl einzelner Ausführungseinheiten zu umfassen.

[0335] In mindestens einer Ausführungsform werden Ausführungseinheiten 3108A-3108N primär zum Ausführen von Shader-Programmen verwendet. In mindestens einer Ausführungsform kann ein Shader-Prozessor 3102 verschiedene Shader-Programme verarbeiten und Ausführungsthreads, die mit Shader-Programmen verknüpft sind, über einen Thread-Abfertiger 3104 versenden. In mindestens einer Ausführungsform umfasst ein Thread-Abfertiger 3104 Logik, um Thread-Initiationsanforderungen von Graphik- und Medien-Pipelines zu arbitrieren und angeforderte Threads auf einer oder mehreren Ausführungseinheiten in den Ausführungseinheiten 3108A-3108N zu instanziiieren. Beispielsweise kann in mindestens einer Ausführungsform eine Geometrie-Pipeline Vertex-, Tesselierungs- oder Geometrie-Shader an die Thread-Ausführungslogik zur Verarbeitung senden. In mindestens einer Ausführungsform kann ein Thread-Abfertiger 3104 auch Laufzeit-Thread-Übergabe-Anforderungen von ausführenden Shader-Programmen verarbeiten.

[0336] In mindestens einer Ausführungsform unterstützen die Ausführungseinheiten 3108A-3108N einen Befehlssatz, der native Unterstützung vieler Standard-3D-Graphik-Shader-Anweisungen umfasst, so dass Shader-Programme aus Graphikbibliotheken (z. B. Direct 3D und OpenGL) mit einer minimalen Übersetzung ausgeführt werden. In mindestens einer Ausführungsform unterstützen Ausführungseinheiten Vertex- und Geometrie-Verarbeitung (z. B. Vertex-Programme, Geometrie-Programme, Vertex-Shader), Pixelverarbeitung (z. B. Pixel-Shader, Fragment-Shader) und Universalverarbeitung (z. B. Rechen- und Medien-Shader). In mindestens einer Ausführungsform ist jede der Ausführungseinheiten 3108A-3108N, die eine oder mehrere Arithmetik-Logikeinheiten (ALUs) umfassen, zu mehreren ausgegebenen Single Instruction Multiple Data (SIMD)-Ausführungen in der Lage, und ermöglicht der Multithread-Betrieb trotz Speicherzugriffen mit höherer Latenz eine effiziente Ausführungsumgebung. In mindestens einer Ausführungsform weist jeder Hardware-Thread innerhalb jeder Ausführungseinheit eine dedizierte Registerdatei mit hoher Bandbreite und einen zugehörigen unabhängigen Thread-Zustand auf. In mindestens einer Ausführungsform ist die Ausführung eine Mehrfachausgabe pro Takt an Pipelines, die zu Ganzzahlen, einfach und doppelt genauen Gleitkommaoperationen, SIMD-Zweigfähigkeit, logischen Operationen, transzendenten Operationen und anderen verschiedenen Operationen in der Lage sind. In mindestens einer Ausführungsform veranlasst, während auf Daten aus dem Speicher oder von einer der gemeinsam genutzten Funktionen gewartet wird, die Abhängigkeitslogik innerhalb der Ausführungseinheiten 3108A-3108N, dass ein wartender Thread schläft, bis angeforderte Daten zurückgegeben wurden. In mindestens einer Ausführungsform können Hardware-Ressourcen für die Verarbeitung anderer Threads bereitgestellt werden, während ein wartender Thread schläft. In mindestens einer Ausführungsform kann beispielsweise während einer Verzögerung in Zusammenhang mit einer Vertex-Shader-Operation, eine Ausführungseinheit Operationen für einen Pixel-Shader, einen Fragment-Shader oder eine andere Art von Shader-Programm durchführen, einschließlich eines unterschiedlichen Vertex-Shaders.

[0337] In mindestens einer Ausführungsform arbeitet jede Ausführungseinheit in den Ausführungseinheiten 3108A-3108N an Anordnungen von Datenelementen. In mindestens einer Ausführungsform bezieht sich eine Anzahl von Datenelementen auf „Ausführungsgröße“ oder Anzahl von Kanälen für eine Anweisung. In mindestens einer Ausführungsform ist ein Ausführungskanal eine logische Ausführungseinheit für den Zugriff auf Datenelemente, ein Maskieren und eine Ablaufsteuerung innerhalb von Anweisungen. In mindestens einer Ausführungsform kann eine Anzahl von Kanälen unabhängig von einer Anzahl physikalischer Arithmetik-Logikeinheiten (Arithmetic Logic Units; ALUs) oder Gleitkommaeinheiten (FPUs) für einen bestimmten Graphikprozessor sein. In mindestens einer Ausführungsform unterstützen die Ausführungseinheiten 3108A-3108N Ganzzahl- und Gleitkomma-Datentypen.

[0338] In mindestens einer Ausführungsform umfasst ein Befehlssatz für Ausführungseinheiten SIMD-Befehle. In mindestens einer Ausführungsform können verschiedene Datenelemente als ein gepackter Datentyp in einem Register gespeichert sein und eine Ausführungseinheit wird verschiedene Elemente basierend auf einer Datengröße der Elemente verarbeiten. Beispielsweise werden in mindestens einer Ausführungsform dann, wenn an einem 256-Bit breiten Vektor gearbeitet wird, 256 Bit eines Vektors in einem Register gespeichert und eine Ausführungseinheit arbeitet an einem Vektor als vier separate 64-Bit gepackte Datenelemente (Quad-Word (QW)-Größe-Datenelemente), acht separate 32-Bit gepackte Datenelemente (Double Word (DW)-Größe-Datenelemente), sechzehn separate 16-Bit gepackte Datenelemente (Word (W)-Größe-Datenelemente) oder zweiunddreißig separate 8-Bit-Datenelemente (Byte (B)-Größe-Datenelemente). In mindestens einer Ausführungsform sind jedoch unterschiedliche Vektorbreiten und Registergrößen möglich.

[0339] In mindestens einer Ausführungsform können eine oder mehrere Ausführungseinheiten zu einer fusionierten Ausführungseinheit 3109A-3109N mit Thread-Steuerlogik (3107A-3107N) kombiniert sein, die für fusionierte Ausführungseinheiten (Execution Units; EU) gemeinsam ist. In mindestens einer Ausführungsform können mehrere EUs zu einer EU-Gruppe fusioniert sein. In mindestens einer Ausführungsform kann

jede EU in der fusionierten EU-Gruppe konfiguriert sein, um einen separaten SIMD-Hardware-Thread auszuführen. Die Anzahl von EUs in einer fusionierten EU-Gruppe kann gemäß verschiedener Ausführungsformen variieren. In mindestens einer Ausführungsform können pro EU verschiedene SIMD-Breiten durchgeführt werden, einschließlich, jedoch nicht beschränkt auf, SIMD8, SIMD16 und SIMD32. In mindestens einer Ausführungsform umfasst jede fusionierte Graphikausführungseinheit 3109A-3109N mindestens zwei Ausführungseinheiten. Beispielsweise umfasst in mindestens einer Ausführungsform eine fusionierte Ausführungseinheit 3109A eine erste EU 3107A, eine zweite EU 3108A und eine Thread-Steuerlogik 3107A, die der ersten EU 3107A und der zweiten EU 3108A gemeinsam ist. In mindestens einer Ausführungsform steuert die Thread-Steuerlogik 3107A Threads, die auf einer fusionierten Graphikausführungseinheit 3109A ausgeführt werden, welches jeder EU innerhalb der fusionierten Ausführungseinheiten 3109A-3109N erlaubt, unter Verwendung eines gemeinsamen Befehlszeigerregisters ausgeführt zu werden.

[0340] In mindestens einer Ausführungsform sind ein oder mehrere interne Anweisungs-Caches (z. B. 3106) in der Thread-Ausführungslogik 3100 umfasst, um Thread-Anweisungen für Ausführungseinheiten zwischenzuspeichern. In mindestens einer Ausführungsform sind ein oder mehrere Daten-Caches (z. B. 3112) umfasst, um Thread-Daten während der Thread-Ausführung zwischenzuspeichern. In mindestens einer Ausführungsform ist ein Abtaster 3110 umfasst, um eine Texturabtastung für 3D-Operationen und eine Medienabtastung für Medienoperationen bereitzustellen. In mindestens einer Ausführungsform umfasst der Abtaster 3110 eine spezialisierte Textur- oder Medien-Abtastfunktionalität, um Textur- oder Mediendaten während des Abtastprozesses zu verarbeiten, bevor abgetastete Daten an eine Ausführungseinheit bereitgestellt werden.

[0341] Während der Ausführung senden in mindestens einer Ausführungsform Graphik- und Medien-Pipelines Thread-Initiationsanforderungen über Thread-Übergabe- und Sende-Logik an die Thread-Ausführungslogik 3100. In mindestens einer Ausführungsform wird, sobald eine Gruppe von geometrischen Objekten verarbeitet und in Pixeldaten gerastert wurde, eine Pixelprozessorlogik (z. B. Pixel-Shader-Logik, Fragment-Shader-Logik usw.) innerhalb des Shader-Prozessors 3102 aufgerufen, um Ausgabeinformationen weiter zu berechnen und zu veranlassen, dass Ergebnisse auf Ausgabeoberflächen (z. B. Farbpuffer, Tiefenpuffer, Schablonenpuffer usw.) geschrieben werden. In mindestens einer Ausführungsform berechnet ein Pixel-Shader oder Fragment-Shader Werte verschiedener Vertex-Attribute, die über ein gerastertes Objekt hinweg zu interpolieren sind. In mindestens einer Ausführungsform führt die Pixelprozessorlogik innerhalb des Shader-Prozessors 3102 dann ein von der Anwendungsprogrammierschnittstelle (API) bereitgestelltes Pixel- oder Fragment-Shader-Programm aus. In mindestens einer Ausführungsform sendet, um ein Shader-Programm auszuführen, ein Shader-Prozessor 3102 Threads über den Thread-Abfertiger 3104 an eine Ausführungseinheit (z. B. 3108A). In mindestens einer Ausführungsform verwendet ein Shader-Prozessor 3102 Texturabtastlogik in einem Abtaster 3110, um auf Texturdaten in im Speicher gespeicherten Texturkarten zuzugreifen. In mindestens einer Ausführungsform berechnen arithmetische Operationen auf Texturdaten und zugeführten Geometriedaten Pixelfarbdaten für jedes geometrische Fragment oder werfen ein oder mehrere Pixel aus der weiteren Verarbeitung.

[0342] In mindestens einer Ausführungsform stellt ein Datenport 3114 einen Speicherzugriffsmechanismus für die Thread-Ausführungslogik 3100 bereit, um verarbeitete Daten zur weiteren Verarbeitung auf einer Graphikprozessor-Ausgabepipeline an den Speicher auszugeben. In mindestens einer Ausführungsform umfasst oder koppelt ein Datenport 3114 einen oder mehrere Cache-Speicher (z. B. Daten-Cache 3112), um Daten für einen Speicherzugriff über einen Datenport zwischenzuspeichern.

[0343] Wie in **Fig. 31B** veranschaulicht, kann in mindestens einer Ausführungsform eine Graphikausführungseinheit 3108 eine Befehlsabrufeinheit 3137, eine allgemeine Registerdateianordnung (GRF) 3124, eine architektonische Registerdateianordnung (ARF) 3126, einen Thread-Arbiter 3122, eine Sendeeinheit 3130, eine Verzweigungseinheit 3132, einen Satz von SIMD-Gleitkommaeinheiten (FPUs) 3134 und in mindestens einer Ausführungsform einen Satz von dedizierten ganzzahligen SIMD ALUs 3135 umfassen. In mindestens einer Ausführungsform umfassen die GRF 3124 und die ARF 3126 einen Satz von allgemeinen Registerdateien und Architekturregisterdateien, die jedem gleichzeitigen Hardware-Thread zugeordnet sind, der in einer Graphikausführungseinheit 3108 aktiv sein kann. In mindestens einer Ausführungsform wird ein Architekturzustand pro Thread in der ARF 3126 beibehalten, während Daten, die während einer Thread-Ausführung verwendet werden, in der GRF 3124 gespeichert werden. In mindestens einer Ausführungsform kann ein Ausführungszustand jedes Threads, einschließlich Befehlszeigern für jeden Thread, in threadspezifischen Registern in der ARF 3126 gehalten werden.

[0344] In mindestens einer Ausführungsform weist eine Graphikausführungseinheit 3108 eine Architektur auf, die eine Kombination aus simultanem Multi-Threading (SMT) und feinkörnigem Interleaved Multi-Threa-

ding (IMT) ist. In mindestens einer Ausführungsform weist die Architektur eine modulare Konfiguration auf, die zur Designzeit auf der Grundlage einer Sollanzahl gleichzeitiger Threads und der Anzahl von Registern pro Ausführungseinheit verfeinert werden kann, wobei Ausführungseinheitsressourcen über Logik hinweg verteilt sind, die zum Ausführen mehrerer gleichzeitiger Threads verwendet wird.

[0345] In mindestens einer Ausführungsform kann eine Graphikausführungseinheit 3108 mehrere Anweisungen gemeinsam ausgeben, welche jeweils unterschiedliche Anweisungen sein können. In mindestens einer Ausführungsform kann ein Thread-Arbiter 3122 einer Graphikausführungseinheit-Thread 3108 Anweisungen an eine Sendeeinheit 3130, Verzweigungseinheit 3142 oder SIMD FPU(s) 3134 zur Ausführung senden. In mindestens einer Ausführungsform kann jeder Ausführungsthread auf 128 Universalregister innerhalb der GRF 3124 zugreifen, wobei jedes Register 32 Bytes speichern kann, die als ein SIMD 8-Elementvektor von 32-Bit-Datenelementen zugänglich sind. In mindestens einer Ausführungsform weist jeder Thread der Ausführungseinheit Zugriff auf 4 KByte innerhalb einer GRF 3124 auf, obwohl Ausführungsformen nicht so beschränkt sind und mehr oder weniger Registerressourcen in anderen Ausführungsformen bereitgestellt sein können. In mindestens einer Ausführungsform können bis zu sieben Threads gleichzeitig ausgeführt werden, obwohl auch die Anzahl von Threads pro Ausführungseinheit je nach Ausführungsform variieren kann. In mindestens einer Ausführungsform, in welcher sieben Threads auf 4 KByte zugreifen können, kann die GRF 3124 insgesamt 28 KByte speichern. In mindestens einer Ausführungsform können flexible Adressierungsmodi Registern ermöglichen, gemeinsam adressiert zu werden, um breitere Register wirksam aufzubauen oder schrittförmige rechteckige Blockdatenstrukturen darzustellen.

[0346] In mindestens einer Ausführungsform werden Speicheroperationen, Abtasteroperationen und andere Systemkommunikationen mit längerer Latenz über „Sende“-Anweisungen gesendet, die durch eine Nachrichtenübergabe-Sendeeinheit 3130 ausgeführt werden. In mindestens einer Ausführungsform werden Verzweigungsanweisungen an eine dedizierte Verzweigungseinheit 3132 gesendet, um SIMD-Divergenz und eventuelle Konvergenz zu erleichtern.

[0347] In mindestens einer Ausführungsform umfasst eine Ausführungseinheit 3108 eine oder mehrere SIMD-Gleitkommaeinheiten (FPU(s)) 3134 zum Durchführen von Gleitkommaoperationen. In mindestens einer Ausführungsform unterstützen die FPU(s) 3134 auch Ganzzahlberechnungen. In mindestens einer Ausführungsform können die FPU(s) 3134 bis zu einer Anzahl M von 32-Bit Gleitkomma- (oder Ganzzahl-) Operationen SIMD-ausführen oder bis zu 2M 16-Bit Ganzzahl oder 16-Bit Gleitkomma-Operationen SIMD-ausführen. In mindestens einer Ausführungsform stellt mindestens eine oder mehrere FPUs erweiterte mathematische Fähigkeiten zur Unterstützung von transzendenten mathematischen Funktionen und ein doppeltgenaues 64-Bit-Gleitkomma mit hohem Durchsatz bereit. In mindestens einer Ausführungsform ist auch ein Satz von 8-Bit Ganzzahl SIMD-ALUs 3135 vorhanden, und kann speziell optimiert sein, um Operationen im Zusammenhang mit Berechnungen zum maschinellen Lernen durchzuführen.

[0348] In mindestens einer Ausführungsform können Anordnungen von mehreren Instanzen einer Graphikausführungseinheit 3108 in einer Graphik-Teilkerngruppierung (z. B. einer Sub-Slice) instanziiert sein. In mindestens einer Ausführungsform kann eine Ausführungseinheit 3108 Anweisungen über mehrere Ausführungskanälen hinweg ausführen. In mindestens einer Ausführungsform wird jeder auf einer Graphikausführungseinheit 3108 ausgeführte Thread auf einem unterschiedlichen Kanal ausgeführt.

[0349] Die Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform können Abschnitt der oder die gesamte Inferenzierungs- und/oder Trainingslogik 915 in die Ausführungslogik 3100 integriert sein. Darüber hinaus können in mindestens einer Ausführungsform hier beschriebene Inferenzierungs- und/oder Trainingsoperationen mit einer anderen Logik als der in **Fig. 9A** oder **Fig. 9B** veranschaulichten Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtungsparmeter in On-Chip- oder Off-Chip-Speicher und/oder Registern (gezeigt oder nicht gezeigt) gespeichert werden, die ALUs der Ausführungslogik 3100 konfigurieren, um einen oder mehrere Algorithmen maschinellen Lernens, Lernalgorithmen, neuronale Netzwerkarchitekturen, Anwendungsfälle oder die hier beschriebenen Trainingstechniken durchzuführen.

[0350] **Fig. 32** veranschaulicht eine Parallelverarbeitungs-Einheit („PPU“) 3200 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist die PPU 3200 mit einem maschinenlesbarem Code konfiguriert, der, falls von der PPU 3200 ausgeführt, die PPU 3200 veranlasst, einige oder sämtliche der Prozesse und Techniken durchzuführen, die in dieser Offenbarung durchweg beschrieben sind. In min-

destens einer Ausführungsform ist die PPU 3200 ein Multi-Threaded-Prozessor, der auf einer oder mehreren integrierten Schaltungsvorrichtungen implementiert ist, und der Multi-Threading als eine Latenz-verbergende Architektur benutzt, die ausgestaltet ist, um computerlesbare Anweisungen (auch als maschinenlesbare Anweisungen oder einfach als Anweisungen bezeichnet) an mehreren Threads in einer parallelen Art und Weise zu verarbeiten. In mindestens einer Ausführungsform bezieht sich ein Thread auf einen Ausführungsthread und ist eine Instanziierung eines Satzes von Anweisungen, die konfiguriert sind, um von der PPU 3200 ausgeführt zu werden. In mindestens einer Ausführungsform ist die PPU 3200 eine Graphikverarbeitungseinheit („GPU“), die konfiguriert ist, um eine Graphik-Rendering-Pipeline zur Verarbeitung von dreidimensionalen („3D“) Graphikdaten zu implementieren, um zweidimensionale („2D“) Bilddaten zur Anzeige auf einer Anzeigevorrichtung, wie beispielsweise einer Flüssigkristallanzeige („LCD“)-Vorrichtung, zu erzeugen. In mindestens einer Ausführungsform wird die PPU 3200 benutzt, um Rechnungen, wie beispielsweise Operationen der linearen Algebra und Operationen des Maschinenlernens durchzuführen. **Fig. 32** veranschaulicht ein Beispiel eines parallelen Prozessors lediglich für veranschaulichende Zwecke und sollte als ein nicht einschränkendes Beispiel von Prozessorarchitekturen ausgelegt werden, die im Umfang dieser Offenbarung in Betracht gezogen werden und die jeder geeignete Prozessor einsetzen kann, um dasselbe zu ergänzen und/oder zu ersetzen.

[0351] In mindestens einer Ausführungsform sind eine oder mehrere PPU 3200 konfiguriert, um HPC(High Performance Computing), Rechenzentrum und Maschinenlern-Anwendungen zu beschleunigen. In mindestens einer Ausführungsform ist die PPU 3200 konfiguriert, um alle Systeme und Anwendungen für tiefes Lernen zu beschleunigen, welche die folgenden nicht einschränken Beispiele umfassen: autonome Fahrzeugplattformen, tiefes Lernen, hochgenaue Sprache, Bild, Texterkennungssysteme, intelligente Videoanalyse, molekulare Simulationen, Wirkstoffentdeckung, Krankheitsdiagnose, Wettervorhersage, Analyse großer Datenmengen, Astronomie, Molekulardynamiksimulation, Finanzmodellierung, Robotertechnik, Fabrikautomation, Sprachübersetzung in Echtzeit, Online-Suchoptimierungen personalisierte Benutzerempfehlungen und mehr.

[0352] In mindestens einer Ausführungsform umfasst die PPU 3200, ohne darauf beschränkt zu sein, eine Eingabe/Ausgabe („E/A“)-Einheit 3206, eine Frontend-Einheit 3210, eine Planer-Einheit 3212, eine Arbeitsverteilungs-Einheit 3214, einen Hub 3216, eine Kreuzschiene („Xbar“) 3220, einen oder mehrere allgemeine Verarbeitungscluster („GPCs“) 3218 und eine oder mehrere Partitions-Einheiten („Speicherpartitions-Einheiten“) 3222. In mindestens einer Ausführungsform ist die PPU 3200 mit einem Host-Prozessor oder anderen PPU 3200 über einen oder mehrere Hochgeschwindigkeits-GPU-Zwischenverbindungen („GPU-Zwischenverbindungen“) 3208 verbunden. In mindestens einer Ausführungsform ist die PPU 3200 mit einem Host-Prozessor oder anderen peripheren Vorrichtungen über eine Zwischenverbindung 3202 verbunden. In mindestens einer Ausführungsform ist die PPU 3200 mit einem lokalen Speicher verbunden, der eine oder mehrere Speichervorrichtungen („Speicher“) 3204 umfasst. In mindestens einer Ausführungsform umfassen die Speichervorrichtungen 3204, ohne darauf beschränkt zu sein, eine oder mehrere Direktzugriffsspeicher („DRAM“)-Vorrichtungen. In mindestens einer Ausführungsform sind eine oder mehrere DRAM-Vorrichtungen als HBM(Speicher mit hoher Bandbreite)-Teilsysteme konfiguriert und/oder konfigurierbar, wobei mehrere DRAM-Dies innerhalb jeder Vorrichtung gestapelt sind.

[0353] In mindestens einer Ausführungsform kann sich die Hochgeschwindigkeits-GPU-Zwischenverbindung 3208 auf eine drahtgebundene mehrspurige Kommunikationsverbindung beziehen, die von Systemen verwendet wird, um eine oder mehrere PPU 3200 zu skalieren und zu umfassen, die mit einer oder mehreren zentralen Verarbeitungseinheiten („CPUs“) kombiniert sind, unterstützt Cache-Kohärenz zwischen den PPU 3200 und CPUs sowie CPU-Mastering. In mindestens einer Ausführungsform werden Daten und/oder Befehle mittels der Hochgeschwindigkeits-GPU-Zwischenverbindung 3208 durch den Hub 3216 an/von anderen Einheiten der PPU 3200 übertragen, wie beispielsweise eine oder mehrere Kopier-Engines, Videocodierer, Videodecodierer, Leistungsverwaltungseinheiten und andere Komponenten, die in **Fig. 32** nicht explizit veranschaulicht werden können.

[0354] In mindestens einer Ausführungsform ist die E/A-Einheit 3206 konfiguriert, um Kommunikationen (d.h. Befehle, Daten usw.) von einem Host-Prozessor (in **Fig. 32** nicht gezeigt) über den Systembus 3202 zu übertragen und zu empfangen. In mindestens einer Ausführungsform kommuniziert die E/A-Einheit 3206 direkt mit dem Host-Prozessor über den Systembus 3202 oder durch eine oder mehrere Zwischenvorrichtungen, wie beispielsweise eine Speicherbrücke. In mindestens einer Ausführungsform kann die E/A-Einheit 3206 mit einem oder mehreren anderen Prozessoren, wie beispielsweise eine oder mehrere PPU 3299, über den Systembus 3202 kommunizieren. In mindestens einer Ausführungsform implementiert die E/A-Einheit 3206 eine Peripheral Component Interconnect Express („PCIe“)-Schnittstelle für Kommunikationen über

einen PCIe-Bus und der Systembus 3202 ist ein PCIe-Bus. In mindestens einer Ausführungsform implementiert die E/A-Einheit 3206 Schnittstellen zum Kommunizieren mit externen Vorrichtungen.

[0355] In mindestens einer Ausführungsform decodiert die E/A-Einheit 3206 Pakete, die über den Systembus 3202 empfangen wurden. In mindestens einer Ausführungsform stellen mindestens einige Pakete Befehle dar, die konfiguriert sind, um die PPU 3200 zu veranlassen, verschiedene Operationen durchzuführen. In mindestens einer Ausführungsform überträgt die E/A-Einheit 3206 decodierte Befehle an verschiedene andere Einheiten der PPU 3200, wie durch Befehle spezifiziert. In mindestens einer Ausführungsform werden Befehle an die Frontend-Einheit 3210 und/oder an den Hub 3216 oder andere Einheiten der PPU 3200 übertragen, wie beispielsweise eine oder mehrere Kopier-Engines, einen Video-Codierer, einen Video-Decodierer, eine Leistungsverwaltungseinheit usw. (nicht explizit gezeigt). In mindestens einer Ausführungsform ist die E/A-Einheit 3206 konfiguriert, um Kommunikationen zwischen und unter verschiedenen logischen Einheiten der PPU 3200 weiterzuleiten.

[0356] In mindestens einer Ausführungsform codiert ein von dem Host-Prozessor ausgeführtes Programm einen Befehlsstrom in einem Puffer, welcher der PPU 3200 Arbeitslasten zur Verarbeitung bereitstellt. In mindestens einer Ausführungsform umfasst eine Arbeitslast Anweisungen und Daten, die durch diese Anweisungen zu verarbeiten sind. In mindestens einer Ausführungsform ist der Puffer eine Region in einem Speicher, der von sowohl dem Host-Prozessor als auch der PPU 3200 zugänglich ist (d.h. Lesen/Schreiben) - eine HostSchnittstelleneinheit kann konfiguriert sein, um auf den Puffer in einem Systemspeicher, der mit dem Systembus 3202 verbunden ist, über Speicheranforderungen zuzugreifen, die über den Systembus 3202 durch die E/A-Einheit 3206 übertragen werden. In mindestens einer Ausführungsform schreibt der Host-Prozessor den Befehlsstrom in den Puffer und überträgt dann einen Zeiger zu einem Start des Befehlsstroms an die PPU 3200, so dass die Frontend-Einheit 3210 Zeiger zu einem oder mehreren Befehlsströme empfängt und einen oder mehrere Befehlsströme verwaltet, Befehle aus den Befehlsströmen liest und Befehle an verschiedene Einheiten der PPU 3200 weiterleitet.

[0357] In mindestens einer Ausführungsform ist die Frontend-Einheit 3210 mit einer Planer-Einheit 3212 gekoppelt, die verschiedene GPCs 3218 konfiguriert, um Aufgaben zu verarbeiten, die durch einen oder mehrere Befehlsströme definiert sind. In mindestens einer Ausführungsform ist die Planer-Einheit 3212 konfiguriert, um Zustandsinformation zu verfolgen, die verschiedene Aufgaben betrifft, die von der Planer-Einheit 3212 verwaltet werden, wobei die Zustandsinformation angeben kann, welchem der GPCs 3218 eine Aufgabe zugewiesen ist, ob die Aufgabe aktiv oder inaktiv ist, ob der Aufgabe ein Prioritätsniveau zugeordnet ist, und so weiter. In mindestens einer Ausführungsform verwaltet die Planer-Einheit 3212 die Ausführung mehrerer Aufgaben auf einem oder mehreren der GPCs 3218.

[0358] In mindestens einer Ausführungsform ist die Planer-Einheit 3212 mit einer Arbeitsverteilungs-Einheit 3214 gekoppelt, die konfiguriert ist, um Aufgaben zur Ausführung auf GPCs 3218 zu versenden. In mindestens einer Ausführungsform verfolgt die Arbeitsverteilungs-Einheit 3214 eine Anzahl von eingeplanten Aufgaben, die von der Planer-Einheit 3212 empfangen werden, und die Arbeitsverteilungs-Einheit 3214 verwaltet einen Pool für anstehende Aufgaben und einen Pool für aktive Aufgaben für jeden der GPCs 3218. In mindestens einer Ausführungsform umfasst der Pool für anstehende Aufgaben eine Anzahl von Schlitzen (z.B. 32 Schlitze), die Aufgaben enthalten, die zugewiesen sind, um von einem bestimmten GPC 3218 verarbeitet zu werden; ein Pool für aktive Aufgaben kann eine Anzahl von Schlitzen (z.B. 4 Schlitze) für Aufgaben umfassen, die von den GPCs 3218 aktiv verarbeitet werden, so dass, wenn einer der GPCs 3218 die Ausführung einer Aufgabe abschließt, diese Aufgabe aus dem Pool für aktive Aufgaben für den GPC 3218 geräumt wird und eine der anderen Aufgaben aus dem Pool für anstehende Aufgaben ausgewählt und zur Ausführung auf dem GPC 3218 eingeplant wird. In mindestens einer Ausführungsform, wenn eine aktive Aufgabe auf dem GPC 3218 inaktiv ist, wie beispielsweise während darauf gewartet wird, dass eine Datenabhängigkeit behoben wird, dann wird die aktive Aufgabe aus dem GPC 3218 geräumt und zu dem Pool für anstehende Aufgaben zurückgeführt, während eine andere Aufgabe in dem Pool für anstehende Aufgaben ausgewählt und zur Ausführung auf dem GPC 3218 eingeplant wird.

[0359] In mindestens einer Ausführungsform kommuniziert die Arbeitsverteilungs-Einheit 3214 mit einem oder mehreren GPCs 3218 über die Kreuzschiene bzw. XBar 3220. In mindestens einer Ausführungsform ist die XBar 3220 ein Zwischenverbindung-Netzwerk, das viele der Einheiten der PPU 3200 mit anderen Einheiten der PPU 3200 koppelt und konfiguriert sein kann, um die Arbeitsverteilungs-Einheit 3214 mit einem bestimmten GPC 3218 zu koppeln. In mindestens einer Ausführungsform können eine oder mehrere andere Einheiten der PPU 3200 ebenfalls mit der XBar 3220 über den Hub 3216 verbunden sein.

[0360] In mindestens einer Ausführungsform werden Aufgaben von der Planer-Einheit 3212 verwaltet und an einen der GPCs 3218 durch die Arbeitsverteilungs-Einheit 3214 abgefordert. Der GPC 3218 ist konfiguriert, um eine Aufgabe zu verarbeiten und Ergebnisse zu erzeugen. In mindestens einer Ausführungsform können Ergebnisse von anderen Aufgaben innerhalb des GPC 3218 konsumiert werden, an einen unterschiedlichen GPC 3218 über die XBar 3220 weitergeleitet oder im Speicher 3204 gespeichert werden. In mindestens einer Ausführungsform können Ergebnisse in den Speicher 3204 über die Partitions-Einheiten 3222 geschrieben werden, die eine Speicherschnittstelle zum Lesen und Schreiben von Daten in/aus dem Speicher 3204 implementieren. In mindestens einer Ausführungsform können Ergebnisse an eine andere PPU 3200 oder CPU über die Hochgeschwindigkeit-GPU-Zwischenverbindung 3208 übertragen werden. In mindestens einer Ausführungsform umfasst die PPU 3200, ohne darauf beschränkt zu sein, eine Anzahl U von Speicherpartitions-Einheiten 3222, die gleich einer Anzahl von getrennten und unterschiedlichen Speichervorrichtungen 3204 ist, die mit der PPU 3200 gekoppelt sind. In mindestens einer Ausführungsform wird die Partitionseinheit 3222 ausführlicher nachstehend in Verbindung mit **Fig. 34** beschrieben.

[0361] In mindestens einer Ausführungsform führt ein Host-Prozessor einen Treiber-Kernel aus, der eine Anwendungsprogrammierschnittstelle („API“) implementiert, die einer oder mehreren Anwendungen ermöglicht, die auf dem Host-Prozessor ausgeführt werden, Operationen zur Ausführung auf der PPU 3200 einzuplanen. In mindestens einer Ausführungsform werden mehrere Rechenanwendungen gleichzeitig von der PPU 3200 ausgeführt und die PPU 3200 stellt Isolierung, Dienstqualität (QoS) und unabhängige Adressräume für die mehreren Rechenanwendungen bereit. In mindestens einer Ausführungsform erzeugt eine Anwendung Anweisungen (z.B. in Form von API-Aufrufen), die einen Treiberkernel veranlassen, eine oder mehrere Aufgaben zur Ausführung durch die PPU 3200 zu erzeugen, und der Treiberkernel gibt Aufgaben an einen oder mehrere Streams aus, die von der PPU 3200 verarbeitet werden. In mindestens einer Ausführungsform umfasst jede Aufgabe eine oder mehrere Gruppen von in Beziehung stehender Threads, die hier als ein Warp bezeichnet werden. In mindestens einer Ausführungsform umfasst ein Warp mehrere in Beziehung stehende Threads (z.B. 32 Threads), die parallel ausgeführt werden können. In mindestens einer Ausführungsform können sich kooperierende Threads auf mehrere Threads beziehen, die Anweisungen umfassen, um eine Aufgabe durchzuführen, und die Daten durch einen gemeinsam genutzten Speicher austauschen können. In mindestens einer Ausführungsform werden Threads und kooperierende Threads ausführlicher gemäß mindestens einer Ausführungsform in Verbindung mit **Fig. 34** beschrieben.

[0362] Eine Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform wird ein tief lernender Anwendungsprozessor verwendet, um ein maschinelles Lernmodell zu trainieren, wie beispielsweise ein neuronales Netzwerk, um Informationen vorherzusagen oder zu inferenzieren, die der PPU 3200 bereitgestellt werden. In mindestens einer Ausführungsform wird ein tief lernender Anwendungsprozessor 3200 verwendet, um Informationen basierend auf einem trainierten maschinellen Lernmodell (z.B., neuronales Netzwerk) zu inferenzieren und vorherzusagen, das durch einen anderen Prozessor oder System oder durch die PPU 3200 trainiert wurde. In mindestens einer Ausführungsform kann die PPU 3200 verwendet werden, um ein oder mehrere hier beschriebene Anwendungsfälle eines neuronalen Netzwerks durchzuführen.

[0363] **Fig. 33** veranschaulicht einen allgemeinen Verarbeitungscluster („GPC“) 3300 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist der GPC 3300 der GPC 3218 von **Fig. 32**. In mindestens einer Ausführungsform umfasst jeder GPC 3300, ohne darauf beschränkt zu sein, eine Anzahl von Hardwareeinheiten zur Verarbeitung von Aufgaben, und jeder GPC 3300 umfasst, ohne darauf beschränkt zu sein, einen Pipeline-Manager 3302, eine Vor-Raster-Operationen-Einheit („PROP“) 3304, eine Raster-Engine 3308, eine Arbeitsverteilungs-Kreuzschiene („WDX“) 3316, eine Speicherverwaltungseinheit („MMU“) 3318 und einen oder mehrere Datenverarbeitungscluster („DPCs“) 3306 und jede geeignete Kombination von Teilen.

[0364] In mindestens einer Ausführungsform wird der Betrieb des GPC 3300 durch den Pipeline-Manager 3302 gesteuert. In mindestens einer Ausführungsform verwaltet der Pipeline-Manager 3302 die Konfiguration eines oder mehrerer DPCs 3306 zur Verarbeitung von Aufgaben, die dem GPC 3300 zugeteilt sind. In mindestens einer Ausführungsform konfiguriert der Pipeline-Manager 3302 mindestens einen von einem oder mehreren DPCs 3306, um mindestens einen Abschnitt einer Graphik-Rendering-Pipeline zu implementieren. In mindestens einer Ausführungsform ist ein DPC 3306 konfiguriert, um ein Vertex-Shader-Programm auf einem programmierbaren Streaming-Multiprozessor („SM“) 3314 auszuführen. In mindestens einer Ausführungsform ist der Pipeline-Manager 3302 konfiguriert, um Pakete, die von einer Arbeitsverteilungs-Einheit

empfangen werden, an geeignete logische Einheiten innerhalb des GPC 3300 weiterzuleiten. In mindestens einer Ausführungsform können einige Pakete an Festfunktions-Hardwareeinheiten in dem PROP 3304 und/oder der Raster-Engine 3308 weitergeleitet werden, während andere Pakete an DPCs 3306 zur Verarbeitung durch eine Primitiven-Engine 3312 oder den SM 3314 weitergeleitet werden können. In mindestens einer Ausführungsform konfiguriert der Pipeline-Manager 3302 mindestens einen der DPCs, um ein neuronales Netzwerkmodell und/oder eine Rechen-Pipeline zu implementieren.

[0365] In mindestens einer Ausführungsform ist die PROP-Einheit 3304 konfiguriert, um Daten, die von der Raster-Engine 3308 und den DPCs 3306 erzeugt wurden, an eine Raster-Operationen(„ROP“)-Einheit in einer Partitions-Einheit 3222 weiterzuleiten, die oben ausführlicher in Verbindung mit **Fig. 32** beschrieben ist. In mindestens einer Ausführungsform ist die PROP-Einheit 3304 konfiguriert, um Optimierungen zur Farbmischung durchzuführen, Pixeldaten zu organisieren, Adressenübersetzungen und mehr durchzuführen. In mindestens einer Ausführungsform umfasst die Raster-Engine 3308, ohne darauf beschränkt zu sein, eine Anzahl von Festfunktions-Hardwareeinheiten, die konfiguriert sind, um verschiedene Raster-Operationen in mindestens einer Ausführungsform durchzuführen, und die Raster-Engine 3308 umfasst, ohne darauf beschränkt zu sein, eine Setup-Engine, eine Grobraster-Engine, eine Aussonderungs-Engine, eine Abschneide-Engine, eine Feinraster-Engine und eine Kachel-verschmelzende Engine und jede geeignete Kombination davon. In mindestens einer Ausführungsform empfängt die Setup-Engine transformierte Vertices und erzeugt Ebenengleichungen, die geometrischen Primitiven zugeordnet sind, die durch Vertices definiert werden; Ebenengleichungen werden an die Grobraster-Engine übertragen, um Abdeckungsinformation (z.B. eine (x,y)-Abdeckungsmaske für eine Kachel) für eine Primitive zu erzeugen; die Ausgabe der Grobraster-Engine wird an die Aussonderungs-Engine übertragen, wo Fragmente, die der Primitiven zugeordnet sind, die einen z-Test nicht bestehen, ausgesondert und an eine Abschneide-Engine übertragen werden, wo alle Fragmente, die außerhalb eines Betrachtungsstumpfes liegen, abgeschnitten werden. In mindestens einer Ausführungsform werden diejenigen Fragmente, welche die Abschneidung und Aussonderung überleben, an eine Feinraster-Engine weitergeben, um Attribute für Pixelfragmente basierend auf den Ebenengleichungen zu erzeugen, die durch eine Setup-Engine erzeugt werden. In mindestens einer Ausführungsform umfasst die Ausgabe der Raster-Engine 3308 Fragmente, die durch eine geeignete Entität zu verarbeiten sind, wie beispielsweise durch einem Fragment-Shader, der innerhalb eines DPC 3306 implementiert ist.

[0366] In mindestens einer Ausführungsform umfasst jeder in dem GPC 3300 enthaltene DPC 3306, ohne darauf beschränkt zu sein, einen M-Pipe-Controller („MPC“) 3310; eine Primitiven-Engine 3312; einen oder mehrere SMs 3314; und jede geeignete Kombination davon. In mindestens einer Ausführungsform steuert der MPC 3310 den Betrieb des DPC 3306, wobei von dem Pipeline-Manager 3302 empfangene Pakete an geeignete Einheiten im DPC 3306 weitergeleitet werden. In mindestens einer Ausführungsform werden Pakete, die einer Vertex zugeordnet sind, an die Primitiven-Engine 3312 weitergeleitet, die konfiguriert ist, um der Vertex zugeordnete Vertexattribute aus dem Speicher abzurufen; im Gegensatz dazu können einem Shader-Programm zugeordnete Pakete an den SM 3314 übertragen werden.

[0367] In mindestens einer Ausführungsform umfasst der SM 3314, ohne darauf beschränkt zu sein, einen programmierbaren Streaming-Prozessor, der konfiguriert ist, um Aufgaben zu verarbeiten, die durch eine Anzahl von Threads dargestellt werden. In mindestens einer Ausführungsform ist der SM 3314 multithreaded (umfasst mehrere Threads) und ist konfiguriert, um eine Mehrzahl von Threads (z.B. 32 Threads) von einer bestimmten Gruppe von Threads nebenläufig auszuführen und implementiert eine SIMD(Einzeln-Anweisung, Mehrere-Daten)-Architektur, wobei jeder Thread in einer Gruppe von Threads (d.h. einem Warp) konfiguriert ist, um einen unterschiedlichen Satz von Daten basierend auf dem gleichen Satz von Anweisungen zu verarbeiten. In mindestens einer Ausführungsform führen alle Threads in einer Gruppe von Threads gleiche Anweisungen aus. In mindestens einer Ausführungsform implementiert der SM 3314 eine SIMT(Einzeln-Anweisung, Mehrere-Threads)-Architektur, wobei jeder Thread in einer Gruppe von Threads konfiguriert ist, um einen unterschiedlichen Satz von Daten basierend auf einem gleichen Satz von Anweisungen zu verarbeiten, wobei jedoch einzelnen Threads in der Gruppe von Threads ermöglicht wird, während der Ausführung zu divergieren. In mindestens einer Ausführungsform wird ein Programmzähler, ein Aufrufstapel und ein Ausführungszustand für jeden Warp beibehalten, was eine Nebenläufigkeit zwischen Warps und eine serielle Ausführung innerhalb Warps ermöglicht, wenn Threads innerhalb des Warp divergieren. In einer weiteren Ausführungsform wird ein Programmzähler, ein Aufrufstapel und ein Ausführungszustand für jeden einzelnen Thread beibehalten, was eine gleiche Nebenläufigkeit zwischen allen Threads, innerhalb und zwischen Warps ermöglicht. In mindestens einer Ausführungsform wird der Ausführungszustand für jeden einzelnen Thread beibehalten und Threads, welche die gleichen Anweisungen ausführen, können konvergiert und zur besseren Effizienz parallel ausgeführt werden. Mindestens eine Ausführungsform des SM 3314 wird nachstehend ausführlicher beschrieben.

[0368] In mindestens einer Ausführungsform stellt die MMU 3318 eine Schnittstelle zwischen dem GPC 3300 und der Partitions-Einheit (z.B. Partitions-Einheit 3222 von **Fig. 32**) bereit und die MMU 3318 stellt eine Übersetzung von virtuellen Adressen in physische Adressen, einen Speicherschutz und eine Arbitrierung von Speicheranforderungen bereit. In mindestens einer Ausführungsform stellt die MMU 3318 einen oder mehrere Adressenübersetzungspuffer (Translation Lookaside Buffer; „TLBs“) zum Durchführen einer Übersetzung von virtuellen Adressen in physische Adressen im Speicher bereit.

[0369] Eine Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform wird ein tief lernenden Anwendungsprozessor verwendet, um ein maschinelles Lernmodell zu trainieren, wie beispielsweise ein neuronales Netzwerk, um Informationen vorherzusagen oder zu inferenzieren, die dem GPC 3300 bereitgestellt werden. In mindestens einer Ausführungsform wird der GPC 3300 verwendet, um Informationen basierend auf einem trainierten maschinellen Lernmodell (z.B., neuronales Netzwerk) zu inferenzieren und vorauszusagen, das durch einen anderen Prozessor oder System oder durch den GPC 3300 trainiert wurde. In mindestens einer Ausführungsform kann die PPU 3300 verwendet werden, um ein oder mehrere hier beschriebene Anwendungsfälle eines neuronalen Netzwerks durchzuführen.

[0370] **Fig. 34** veranschaulicht eine Speicherpartitions-Einheit 3400 eine Parallelverarbeitungseinheit („PPU“) gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform umfasst die Partitions-Einheit 3400 eine Raster-Operationen („ROP“)-Einheit 3402, einen Level-2 („L2“)-Cache 3404, eine Speicherschnittstelle 3406 und jede geeignete Kombination davon. In einer Ausführungsform ist die Speicherschnittstelle 3406 mit dem Speicher gekoppelt. In einer Ausführungsform kann die Speicherschnittstelle 3406 32-, 64-, 128-, 1024-Bit-Datenbusse oder ähnliche Implementierungen für einen Hochgeschwindigkeits-Datentransfer implementieren. In mindestens einer Ausführungsform beinhaltet die PPU U Speicherschnittstellen 3406, eine Speicherschnittstelle 3406 pro Paar von Speicherpartitions-Einheiten 3400 ist, wobei jedes Paar von Speicherpartitions-Einheiten 3400 mit einer entsprechenden Speichervorrichtung verbunden ist. Beispielsweise kann in mindestens einer Ausführungsform die PPU mit bis zu Y Speichervorrichtungen, wie beispielsweise Speicherstapel mit hoher Bandbreite oder Graphikdoppeldatenraten, Version 5, synchronen dynamischen Direktzugriffsspeicher („GDDR5 SDRAM“) verbunden sein.

[0371] In mindestens einer Ausführungsform implementiert die Speicherschnittstelle 3406 eine Speicherschnittstelle eines Speichers mit hoher Bandbreite der zweiten Generation („HBM2“) und Y ist gleich einem halben U. In mindestens einer Ausführungsform sind HBM2-Speicherstapel auf der gleicher physischen Packung wie die PPU lokalisiert, die wesentliche Leistungs- und Flächeneinsparungen verglichen mit herkömmlichen GDDR5 SDRAM Systemen bereitstellt. In mindestens einer Ausführungsform umfasst jeder HBM2-Stapel, ohne darauf beschränkt zu sein, vier Speicher-Dies mit $Y = 4$, wobei jeder HBM2-Stapel, ohne darauf beschränkt zu sein, zwei 128-Bit Kanäle pro Die für eine Gesamtzahl von 8 Kanälen und eine Datenbusbreite von 1024 Bit umfasst. In mindestens einer Ausführungsform unterstützt der Speicher einen Fehlerkorrekturcode („ECC“) mit Einzelfehlerkorrektur und Doppelfehlerdetektion („SEDED“), um Daten zu schützen. ECC stellt höhere Zuverlässigkeit für Rechenanwendungen bereit, die gegen Datenverfälschung empfindlich sind.

[0372] In mindestens einer Ausführungsform implementiert eine PPU eine Mehrebenen-Speicherhierarchie. In mindestens einer Ausführungsform unterstützt die Speicherpartitions-Einheit 3400 einen vereinheitlichten Speicher, um einen einzigen vereinheitlichten virtuellen Adressraum für die zentrale Verarbeitungseinheit („CPU“) und den Speicher der PPU bereitzustellen, wobei eine Datenteilung zwischen virtuellen Speichersystemen ermöglicht wird. In mindestens einer Ausführungsform wird die Häufigkeit von Zugriffen durch eine PPU auf den Speicher verfolgt, der auf anderen Prozessoren lokalisiert ist, um sicherzustellen, dass Speicherseiten in den physischen Speicher der PPU bewegt werden, die häufiger auf die Seiten zugreift. In mindestens einer Ausführungsform unterstützt die Hochgeschwindigkeits-GPU-Zwischenverbindung 3208 Adressenübersetzungsdienste, die der PPU ermöglichen, auf Seitentabellen einer CPU direkt zuzugreifen und die einen vollen Zugriff auf den CPU-Speicher durch die PPU bereitstellen.

[0373] In einer Ausführungsform übertragen Kopier-Engines Daten zwischen mehreren PPUs oder zwischen PPUs und CPUs. In einer Ausführungsform können Kopier-Engines Seitenfehler für Adressen erzeugen, die nicht in Seitentabellen abgebildet sind, und die Speicherpartitionseinheit 3400 bedient dann die Seitenfehler, indem sie die Adressen in der Seitentabelle abbildet, woraufhin die Kopier-Engines die Übertragung durchführt. In mindestens einer Ausführungsform wird der Speicher für mehrere Kopier-Engine-Operationen zwi-

schen mehreren Prozessoren fixiert (z.B. nicht auslagerbar), wodurch der verfügbare Speicher erheblich reduziert wird. In mindestens einer Ausführungsform mit Hardware-Seitenfehlern können Adressen an die Kopier-Engines übergeben werden ohne Rücksicht darauf, ob die Speicherseiten im Speicher vorliegen, und der Kopiervorgang ist transparent.

[0374] Daten aus dem Speicher 3204 von **Fig. 32** oder einem anderen Systemspeicher werden von der Speicherpartitions-Einheit 3400 abgerufen und in dem L2-Cache-Speicher 3404 gespeichert, der On-Chip lokalisiert ist und zwischen verschiedenen GPCs gemäß mindestens einer Ausführungsform gemeinsam benutzt wird. Jede Speicherpartitions-Einheit 3400 umfasst in mindestens einer Ausführungsform einen Bereich des L2-Cache-Speichers, der einer entsprechenden Speichervorrichtung zugeordnet ist. In mindestens einer Ausführungsform können Cache-Speicher niedrigerer Ebene dann in verschiedenen Einheiten innerhalb der GPCs implementiert sein. In mindestens einer Ausführungsform kann jeder der SMs 3314 einen Level-1 („L1“)-Cache-Speicher implementieren, wobei der L1-Cache-Speicher ein privater Speicher ist, der einem bestimmten SM 3314 fest zugeordnet ist, und Daten von dem L2-Cache-Speicher 3404 werden abgerufen und in jedem von L1-Cache-Speichern zur Verarbeitung in Funktionseinheiten der SMs 3314 gespeichert. In mindestens einer Ausführungsform ist der L2-Cache-Speicher 3404 mit der Speicherschnittstelle 3406 und dem XBar 3220 gekoppelt.

[0375] Die ROP-Einheit 3402 führt Graphik-Raster-Operationen, welche die Pixelfarbe betreffen, wie beispielsweise Farbenkomprimierung, Pixelmischung und mehr, in mindestens einer Ausführungsform durch. Die ROP-Einheit 3402 implementiert in mindestens einer Ausführungsform ebenfalls Tiefentesten in Verbindung mit der Raster-Engine 3308, wobei eine Tiefe für einen Abtastort, der einem Pixelfragment zugeordnet ist, von einer Aussonderungs-Engine der Raster-Engine 3308 empfangen wird. In mindestens einer Ausführungsform wird die Tiefe gegen eine entsprechende Tiefe in einem Tiefenpuffer für einen Abtastort geprüft, der einem Fragment zugeordnet ist. In mindestens einer Ausführungsform aktualisiert, wenn ein Fragment den Tiefentest für den Abtastort besteht, dann aktualisiert die ROP-Einheit 3402 den Tiefenpuffer und überträgt ein Ergebnis des Tiefentests an die Raster-Engine 3308. Es wird anerkannt, dass sich die Anzahl von Speicherpartitions-Einheiten 3400 von der Anzahl von GPCs unterscheiden kann, und daher kann in mindestens einer Ausführungsform jede ROP-Einheit 3402 mit jedem von GPCs gekoppelt werden. In mindestens einer Ausführungsform verfolgt die ROP-Einheit 3402 Pakete, die von unterschiedlichen GPCs empfangen werden, und bestimmt, welches die ROP-Einheit 3402 erzeugtes Ergebnis zu der Xbar 3220 durchgeleitet wird.

[0376] **Fig. 35** veranschaulicht einen Streaming-Multiprozessor („SM“) 3500 gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist der SM 3500 der SM von **Fig. 33**. In mindestens einer Ausführungsform umfasst der SM 3500, ohne darauf beschränkt zu sein, einen Befehls-Cache-Speicher 3502; eine oder mehrere Planer-Einheiten 3504; eine Registerdatei 3508; einen oder mehrere Verarbeitungskerne („cores“) 3510; eine oder mehrere Spezialfunktionseinheiten („SFUs“) 3512; eine oder mehrere Lade/Speicher-Einheiten („LSUs“) 3514; ein Zwischenverbindung-Netzwerk 3516; einen gemeinsam genutzten Speicher/Level-1 („L1“)-Cache-Speicher 3518; und jede geeignete Kombination davon.

[0377] In mindestens einer Ausführungsform versendet eine Arbeitsverteilungs-Einheit Aufgaben zur Ausführung auf allgemeinen Verarbeitungs-Clustern („GPCs“) von Parallelverarbeitungseinheiten („PPUs“) und jede Aufgabe wird einem bestimmten Datenverarbeitungscluster (Data Processing Cluster; „DPC“) innerhalb eines GPC zugeteilt, und wenn die Aufgabe einem Shader-Programm zugeordnet ist, wird die Aufgabe einem der SMs 3500 zugeteilt. In mindestens einer Ausführungsform empfängt die Planer-Einheit 3504 Aufgaben von der Arbeitsverteilungs-Einheit und verwaltet die Befehlsplanung (instruction scheduling) für einen oder mehrere Thread-Blöcke, die dem SM 3500 zugewiesen sind. In mindestens einer Ausführungsform plant die Planer-Einheit 3504 Thread-Blöcke zur Ausführung als Warps von parallelen Threads, wobei jeder Thread-Block mindestens einem Warp zugeteilt ist. In mindestens einer Ausführungsform führt jeder Warp Threads aus. In mindestens einer Ausführungsform verwaltet die Planer-Einheit 3504 mehrere unterschiedliche Thread-Blöcke, teilt Warps unterschiedlichen Thread-Blöcken zu und versendet dann Anweisungen von der Mehrzahl von unterschiedlichen kooperativen Gruppen an verschiedene Funktionseinheiten (z.B. Verarbeitungskernen 3510, SFUs 3512 und LSUs 3514) während jedes Taktzyklus.

[0378] In mindestens einer Ausführungsform können sich die Cooperative Groups auf ein Programmiermodell zum Organisieren von Gruppen von kommunizierenden Threads beziehen, die es Entwicklern ermöglichen, die Granularität auszudrücken, bei der Threads kommunizieren, wobei der Ausdruck von reicheren, effizienteren Parallelzerlegungen ermöglicht wird. In mindestens einer Ausführungsform unterstützen Cooperative-Start-APIs die Synchronisierung unter Thread-Blöcken zur Ausführung von parallelen Algorithmen. In

mindestens einer Ausführungsform stellen herkömmliche Programmiermodelle einen einzigen, einfachen Aufbau zum Synchronisieren von kooperierenden Threads bereit: eine Barriere über alle Threads eines Thread-Blocks (d.h. die Funktion `syncthreads()`). In mindestens einer Ausführungsform können Programmierer Gruppen von Threads jedoch bei kleineren als Thread-Block-Granularitäten definieren und innerhalb der definierten Gruppen synchronisieren, um größere Leistung, Gestaltungsflexibilität und Software-Wiederverwendung in der Form von kollektiven gruppenweiten Funktionsschnittstellen zu ermöglichen. In mindestens einer Ausführungsform ermöglichen die Cooperative Groups Programmierern, Gruppen von Threads explizit bei Sub-Block- (z. B. so klein wie ein einziger Thread) und Multi-Block-Granularitäten zu definieren und kollektive Operationen, wie beispielsweise Synchronisierung, an den Threads in einer kooperativen Gruppe durchzuführen. Ein Programmiermodell unterstützt eine saubere Zusammensetzung über Softwaregrenzen, so dass Bibliotheken und Dienstprogrammfunktionen innerhalb ihres lokalen Kontexts sicher synchronisieren können, ohne Annahmen über Konvergenz machen zu müssen. In mindestens einer Ausführungsform ermöglichen die Grundelemente kooperativer Gruppen neue Muster von kooperativer Parallelität, die, ohne darauf beschränkt zu sein, Erzeuger-Verbraucher Parallelität, opportunistische Parallelität und globale Synchronisierung über ein gesamtes Gitter von Threadblöcken umfassen.

[0379] In mindestens einer Ausführungsform ist eine Abfertigungseinheit 3506 konfiguriert, um Anweisungen an eine oder mehrere von Funktionseinheiten zu übertragen, und Planer-Einheiten 3504 umfassen, ohne darauf beschränkt zu sein, zwei Abfertigungseinheiten 3506, die ermöglichen, dass zwei unterschiedliche Anweisungen von dem gleichen Warp während jedes Taktzyklus abgefertigt werden. In mindestens einer Ausführungsform umfasst jede Planer-Einheit 3504 eine einzige Abfertigungseinheit 3506 oder zusätzliche Abfertigungseinheiten 3506.

[0380] In mindestens einer Ausführungsform umfasst jeder SM 3500, ohne darauf beschränkt zu sein, eine Registerdatei 3508, die einen Satz von Registern für die Funktionseinheiten des SM 3500 bereitstellt. In mindestens einer Ausführungsform ist die Registerdatei 3508 zwischen jeder von Funktionseinheiten aufgeteilt, so dass jede Funktionseinheit einem zugehörigen Abschnitt der Registerdatei 3508 zugeteilt ist. In einer anderen Ausführungsform ist die Registerdatei 3508 zwischen den unterschiedlichen Warps aufgeteilt, die von dem SM 3500 ausgeführt werden, und die Registerdatei 3508 stellt eine temporäre Speicherung für Operanden bereit, die mit Datenpfaden der Funktionseinheiten verbunden sind. In mindestens einer Ausführungsform umfasst jeder SM 3500, ohne darauf beschränkt zu sein, L Verarbeitungskerne 3510. In mindestens einer Ausführungsform umfasst der SM 3500, ohne darauf beschränkt zu sein, eine große Anzahl (z.B., 128 oder mehr) von distinkten Verarbeitungskernen 3510. In mindestens einer Ausführungsform umfasst jeder Verarbeitungskern 3510, ohne darauf beschränkt zu sein, eine vollständig in einer Pipeline angeordnete (full-y-pipelined) Verarbeitungseinheit mit einfacher, doppelter oder gemischter Präzision, die, ohne darauf beschränkt zu sein, eine Gleitkommaarithmetik-Logikeinheit und eine Ganzzahlarithmetik-Logikeinheit umfasst. In mindestens einer Ausführungsform implementieren die Gleitkommaarithmetik-Logikeinheiten den IEEE 754-2008 Standard für Gleitkommaarithmetik. In mindestens einer Ausführungsform umfassen die Verarbeitungskerne 3510, ohne darauf beschränkt zu sein, 64 Einfach-Präzisions-(32-Bit)-Gleitkommakerne, 64 Ganzzahlkern, 32 Doppel-Präzisions-(64-Bit)-Gleitkommakerne und 8 Tensorkerne.

[0381] Tensorkerne sind konfiguriert, um Matrixoperationen gemäß mindestens einer Ausführungsform durchzuführen. In mindestens einer Ausführungsform sind ein oder mehrere Tensorkerne in Verarbeitungskernen 3510 enthalten. In mindestens einer Ausführungsform sind Tensorkerne konfiguriert, um Matrixarithmetik tiefen Lernens, wie beispielsweise Faltungsoperationen für neuronales Netzwerktraining und Inferenzieren, durchzuführen. In mindestens einer Ausführungsform arbeitet jeder Tensorkern an einer 4x4 Matrix und führt eine Matrix-Multiplikation- und Akkumulation-Operation $D=A \times B + C$ durch, wobei A, B, C und D 4x4 Matrizen sind.

[0382] In mindestens einer Ausführungsform sind Matrix-Multiplikations-Eingaben A und B 16-Bit-Gleitkomma-Matrizen und die Akkumulationsmatrizen C und D sind 16-Bit-Gleitkomma- oder 32-Bit-Gleitkomma-Matrizen. In mindestens einer Ausführungsform arbeiten Tensorkerne an 16-Bit-Gleitkomma-Eingangsdaten mit einer 32-Bit-Gleitkomma-Akkumulation. In mindestens einer Ausführungsform verwendet 16-Bit-Gleitkomma-Multiplikation 64 Operationen und ergibt ein Produkt voller Präzision, das dann unter Verwendung einer 32-Bit-Gleitkomma-Addition mit den anderen Zwischenprodukten für eine 4x4x4-Matrix-Multiplikation akkumuliert wird. Tensorkerne werden verwendet, um in mindestens einer Ausführungsform viel größere zweidimensionale oder höherdimensionale Matrixoperationen durchzuführen, die von diesen kleineren Elementen aufgebaut werden. In mindestens einer Ausführungsform exponiert eine API, wie beispielsweise die CUDA 9 C++ API, spezialisierte Matrix-Lade-, Matrix-Multiplikations- und Matrix-Akkumulations- und Matrix-Speicher-Operationen, um Tensorkerne von einem CUDA-C++ Programm effizient zu verwenden. In mindestens einer

Ausführungsform nimmt auf CUDA-Ebene das Warp-Schnittstellenniveau 16x16 große Matrizen an, die alle 32 Threads eines Warp überspannen.

[0383] In mindestens einer Ausführungsform umfasst jeder SM 3500, ohne darauf beschränkt zu sein, M SFUs 3512, die Sonderfunktionen durchführen (z.B. Attributauswertung, reziproke Quadratwurzel und ähnliches). In mindestens einer Ausführungsform können die SFUs 3512, ohne darauf beschränkt zu sein, eine Baumtraversierungseinheit umfassen, die konfiguriert ist, um eine hierarchische Baumdatenstruktur zu durchlaufen. In mindestens einer Ausführungsform umfassen die SFUs 3512 eine Textureinheit, die konfiguriert ist, um Texturkarten-Filteroperationen durchzuführen. In mindestens einer Ausführungsform sind die Textureinheiten konfiguriert, um Texturkarten (z.B. eine 2D-Anordnung von Texeln) aus dem Speicher zu laden und Texturkarten abzutasten, um abgetastete Texturwerte zum Gebrauch in Shader-Programmen zu erzeugen, die durch den SM 3500 ausgeführt werden. In mindestens einer Ausführungsform werden die Texturkarten in dem gemeinsam genutzten Speicher/L1-Cache-Speicher 3518 gespeichert. In einer Ausführungsform implementieren die Textureinheiten Texturoperationen, wie beispielsweise Filteroperationen, unter Verwendung von Mip-Maps (z.B. Texturkarten von veränderlichem Detaillierungsgrad). In mindestens einer Ausführungsform umfasst jeder SM 3500, ohne darauf beschränkt zu sein, zwei Textureinheiten.

[0384] Jeder SM 3500 umfasst, ohne darauf beschränkt zu sein, N LSUs 3514, die Lade- und Speicheroperationen zwischen dem gemeinsam genutzten Speicher/L1-Cache-Speicher 3518 und der Registerdatei 3508 in mindestens einer Ausführungsform implementieren. Jeder SM 3500 umfasst, ohne darauf beschränkt zu sein, ein Zwischenverbindung-Netzwerk 3516, das jede der Funktionseinheiten mit der Registerdatei 3508 und die LSU 3514 mit der Registerdatei 3508 und dem gemeinsam genutzten Speicher/ L1-Cache-Speicher 3518 in mindestens einer Ausführungsform verbindet. In mindestens einer Ausführungsform ist das Zwischenverbindung-Netzwerk 3516 eine Kreuzschiene, die konfiguriert sein kann, um eine beliebige der Funktionseinheiten mit irgendeinem der Register in der Registerdatei 3508 zu verbinden und die LSUs 3514 mit der Registerdatei 3508 und Speicherorten in dem gemeinsam genutzten Speicher/L1-Cache-Speicher 3518 zu verbinden.

[0385] In mindestens einer Ausführungsform ist der gemeinsam benutzte Speicher/L1-Cache-Speicher 3518 eine On-Chip-Speicheranordnung, die in mindestens einer Ausführungsform Datenspeicherung und Kommunikation zwischen dem SM 3500 und der Primitiven-Engine und zwischen Threads in dem SM 3500 ermöglicht. In mindestens einer Ausführungsform umfasst der gemeinsam benutzte Speicher/L1-Cache-Speicher 3518, ohne darauf beschränkt zu sein, 128KB von Speicherkapazität und ist in einem Pfad von dem SM 3500 zu der Partitions-Einheit. In mindestens einer Ausführungsform wird der gemeinsam benutzte Speicher/L1-Cache-Speicher 3518 verwendet, um in mindestens einer Ausführungsform Lese- und Schreibvorgänge zwischenspeichern. In mindestens einer Ausführungsform sind ein oder mehrere von dem gemeinsam genutzten Speicher/L1-Cache-Speicher 3518, L2-Cache-Speicher und dem Speicher Hintergrundspeicher.

[0386] Ein Kombinieren von Daten-Cache und gemeinsam genutzter Speicherfunktionalität in einen einzigen Speicherblock stellt in mindestens einer Ausführungsform eine verbesserte Leistung für beide Arten von Speicherzugriffen bereit. In mindestens einer Ausführungsform wird die Kapazität als ein Cache für Programme benutzt oder ist nutzbar, die keinen gemeinsam genutzten Speicher verwenden, wie beispielsweise, wenn ein gemeinsam genutzter Speicher konfiguriert ist, dass er die Hälfte der Kapazität verwendet, die Textur- und Lade-/Speicher-Operationen die verbleibende Kapazität verwenden können. Die Integration innerhalb des gemeinsam genutzten Speichers/L1-Caches 3518 ermöglicht dem gemeinsam genutzten Speicher/L1-Cache 3518 als eine Hochdurchsatzleitung zum Streamen von Daten zu arbeiten, während gleichzeitig ein Zugriff auf häufig wiederverwendete Daten mit hoher Bandbreite und geringer Latenzzeit gemäß einer Ausführungsform bereitgestellt wird. In mindestens einer Ausführungsform kann, wenn für Allzweck-Parallelberechnung konfiguriert, im Vergleich mit Graphikverarbeitung eine einfachere Konfiguration verwendet werden. In mindestens einer Ausführungsform werden Festfunktions-Graphikverarbeitungseinheiten umgangen, wobei ein viel einfacheres Programmiermodell erzeugt wird. In der Allzweck-Parallelberechnungs-Konfiguration werden Blöcke von Threads der Arbeitsverteilungs-Einheit direkt DPCs in mindestens einer Ausführungsform zugewiesen und verteilt. In mindestens einer Ausführungsform führen Threads in einem Block ein gleiches Programm aus, unter Verwendung einer eindeutigen Thread-ID in der Berechnung, um sicherzustellen, dass jeder Thread unter Verwendung des SM 3500 eindeutige Ergebnisse erzeugt, um das Programm auszuführen und Berechnungen durchzuführen, eines gemeinsam genutzten Speicher/L1-Cache-Speichers 3518, um zwischen Threads zu kommunizieren, und der LSU 3514, um einen globalen Speicher durch den gemeinsam genutzten Speicher/L1-Cache-Speicher 3518 und die Speicherpartitions-Einheit zu lesen und zu beschreiben. In mindestens einer Ausführungsform schreibt, wenn für Allzweck-Parallelberechnung konfigu-

riert, schreibt der SM 3500 Befehle, welche die Planer-Einheit 3504 verwenden kann, um neue Arbeit auf DPCs zu starten.

[0387] In mindestens einer Ausführungsform ist die PPU enthalten in oder gekoppelt mit einem Tischcomputer, einem Laptop-Computer, einem Tablet-Computer, einem Smartphone (z.B. einer drahtlosen handgehaltenen Vorrichtung), einem persönlichen digitalen Assistenten („PDA“), einer Digitalkamera, einem Fahrzeug, einer Head-Mounted-Display, einer handgehaltenen elektronischen Vorrichtung und mehr. In mindestens einer Ausführungsform ist die PPU auf einem einzelnen Halbleitersubstrat verkörpert. In mindestens einer Ausführungsform ist die PPU in einem System-on-Chip („SoC“) zusammen mit einer oder mehreren anderen Vorrichtungen, wie beispielsweise zusätzlichen PPUs, Speicher, einem Rechner-mit reduziertem-Befehlssatz („RISC“)-CPU, einer Speicherverwaltungseinheit („MMU“), einem Digital/AnalogWandler („DAC“) und ähnlichem enthalten.

[0388] In mindestens einer Ausführungsform kann eine PPU auf einer Graphikkarte enthalten sein, die eine oder mehrere Speichervorrichtungen umfasst. Eine Graphikkarte kann konfiguriert sein, um sich mit einem PCIe-Schlitz auf einer Hauptplatine eines Desktop-Computers schnittstellenmäßig zu verbinden. In mindestens einer Ausführungsform kann eine PPU eine integrierte Graphikverarbeitungseinheit („iGPU“) sein, die in einem Chipsatz einer Hauptplatine enthalten ist.

[0389] Eine Inferenzierungs- und/oder Trainingslogik 915 wird verwendet, um Inferenzierungs- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenzierungs- und/oder Trainingslogik 915 werden nachstehend in Verbindung mit **Fig. 9A** und/oder **Fig. 9B** bereitgestellt. In mindestens einer Ausführungsform wird ein tief lernenden Anwendungsprozessor verwendet, um ein maschinelles Lernmodell zu trainieren, wie beispielsweise ein neuronales Netzwerk, um Informationen vorherzusagen oder zu inferenzieren, die dem SM 3500 bereitgestellt werden. In mindestens einer Ausführungsform wird der SM 3500 verwendet, um Informationen basierend auf einem trainierten maschinellen Lernmodell (z.B., neuronalen Netzwerk) zu inferenzieren und vorauszusagen, das durch einen anderen Prozessor oder System oder durch den SM 3500 trainiert wurde. In mindestens einer Ausführungsform kann der SM 3500 verwendet werden, um ein oder mehrere hier beschriebene Anwendungsfälle eines neuronalen Netzwerks durchzuführen.

[0390] In mindestens einer Ausführungsform kann sich eine einzelne Halbleiterplattform auf eine einzige unitäre Halbleiterbasierte integrierte Schaltung oder Chip beziehen. In mindestens einer Ausführungsform können Mehrfachchipmodule mit erhöhter Konnektivität verwendet werden, welche einen On-Chip-Betrieb simulieren und wesentliche Verbesserungen gegenüber der Benutzung einer herkömmlichen zentralen Verarbeitungseinheit („CPU“) und Busimplementierung machen. In mindestens einer Ausführungsform können verschiedene Module ebenfalls separat oder in verschiedene Kombinationen der Halbleiterplattform gemäß den Wünschen von Benutzern gelegen sein.

[0391] In mindestens einer Ausführungsform werden Computerprogramme in Form von maschinenlesbarem ausführbarem Code oder Computersteuerungslogik-Algorithmen im Hauptspeicher 1504 und/oder einer sekundären Speicherung gespeichert. Computerprogramme, wenn durch einen oder mehrere Prozessoren ausgeführt, ermöglichen dem System 1500, verschiedene Funktionen gemäß mindestens einer Ausführungsform durchzuführen. Der Speicher 1504, eine Speicherung und/oder irgendeine andere Speicherung sind mögliche Beispiele von computerlesbaren Medien. In mindestens einer Ausführungsform kann sich die sekundäre Speicherung auf jede geeignete Speichervorrichtung oder System beziehen, wie beispielsweise ein Festplattenlaufwerk und/oder einen Wechselspeicherlaufwerk, das ein Diskettenlaufwerk, ein Magnetbandlaufwerk, ein Compact-Disk-Laufwerk, ein DVD-Laufwerk (Digital Versatile Disk), eine Aufnahmeeinrichtung oder einen USB-Flash-Speicher (Universal Serial Bus) Flash-Speicher usw. darstellt. In mindestens einer Ausführungsform werden die Architektur und/oder Funktionalität von verschiedenen vorherigen Figuren im Kontext der CPU 1502; des Parallelverarbeitungssystems 1512; einer integrierten Schaltung, die imstande ist, mindestens einen Teil der Fähigkeiten von beiden CPU 1502; eines Parallelverarbeitungssystem 1512; einen Chipsatz (eine Gruppe von integrierten Schaltungen, die zum Arbeiten konzipiert sind und als eine Einheit zum Durchführen verwandter Funktionen usw. verkauft werden); und eine beliebige geeignete Kombination von einer oder mehreren integrierten Schaltungen implementiert.

[0392] In mindestens einer Ausführungsform werden die Architektur und/oder Funktionalität von verschiedenen vorhergehenden Figuren im Kontext eines allgemeinen Computersystems, eines Leiterplattensystems, eines für Unterhaltungszwecke bestimmten Spielkonsolensystems, eines anwendungsspezifischen Systems und mehr implementiert. In mindestens einer Ausführungsform kann das Computersystem 1500 in Form

eines Desktop-Computers, eines Laptop-Computers, eines Tablet-Computers, von Servern, Supercomputern, eines Smartphones (z. B. einer drahtlosen, handgehaltenen Einrichtung), eines persönlichen digitalen Assistenten („PDA“), einer Digitalkamera, eines Fahrzeugs, eines Head-Mounted-Displays, einer handgehaltenen elektronischen Einrichtung, einer Mobiltelefoneinrichtung, eines Fernsehers, einer Workstation, von Spielkonsolen, eines eingebetteten Systems und/oder jeder anderen Art von Logik vorhanden sein.

[0393] In mindestens einer Ausführungsform umfasst ein Parallelverarbeitungssystem 1512, ohne darauf beschränkt zu sein, mehrere Parallelverarbeitungseinheiten (Parallelverarbeitungseinheiten; „PPUs“) 1514 und zugeordneten Speicher 1516. In mindestens einer Ausführungsform sind PPUs 1514 mit einem Host Prozessor oder anderen peripheren Vorrichtungen über eine Zwischenverbindung 1518 und einen Schalter 1520 oder einen Multiplexer verbunden. In mindestens einer Ausführungsform verteilt das Parallelverarbeitungssystem 1512 Rechenaufgaben über PPUs 1514, die parallelisierbar sein können - beispielsweise als Teil der Verteilung von Rechenaufgaben über mehrere Thread-Blöcke einer Graphikverarbeitungseinheit („GPU“). In mindestens einer Ausführungsform wird ein Speicher gemeinsam genutzt und ist (z.B. für Lese- und/oder Schreibzugriff) über einige oder alle PPUs 1514 zugänglich, obwohl bei einem derartigen gemeinsam genutzten Speicher Leistungsnachteile relativ zu der Verwendung von lokalem Speicher und Registern anfallen können, die sich in einer PPU 1514 befinden. In mindestens einer Ausführungsform wird der Betrieb von PPUs 1514 durch Verwendung eines Befehls synchronisiert, wie beispielsweise `syncThreads()`, wobei alle Threads in einen Block (z.B. über mehrere PPUs 1514 ausgeführt) einen bestimmten Punkt der Ausführung des Codes erreichen, bevor fortgesetzt wird.

[0394] Andere Variationen liegen im Sinne der vorliegenden Offenbarung. Somit sind, während die offenbarten Techniken für verschiedene Modifikationen und alternative Konstruktionen anfällig sind, bestimmte dargestellte Ausführungsformen davon in den Zeichnungen gezeigt und wurden oben ausführlich beschrieben. Es versteht sich jedoch, dass es keine Absicht gibt, die Erfindung auf die spezifische Form oder die offenbarten Formen zu beschränken, sondern im Gegenteil die Absicht ist, alle Modifikationen, alternativen Konstruktionen und Äquivalente abzudecken, die in den Geist und den Umfang der Erfindung fallen, wie in den beigefügten Ansprüchen definiert.

[0395] Die Verwendung der Ausdrücke „ein“ und „eine“ und „der/die/das“ und ähnlicher Referenten im Zusammenhang der Beschreibung der offenbaren Ausführungsformen (besonders im Zusammenhang der folgenden Patentansprüche) soll so ausgelegt werden, dass sie sowohl den Singular als auch den Plural abdeckt, sofern in dieser Schrift nicht anderweitig angegeben oder im eindeutigen Widerspruch zum Kontext, und nicht als eine Definition eines Ausdrucks. Die Begriffe „umfassend“, „aufweisend“, „beinhaltend“ und „enthaltend“ sind als offene Begriffe auszulegen (d. h. „einschließlich, aber nicht begrenzt auf“), sofern nicht anderweitig angegeben. Der Begriff „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 Begriffs „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 es nicht anders angegeben ist oder der Kontext dem anderweitig widerspricht, bezeichnet ferner der Begriff „Teilmenge“ nicht notwendigerweise eine richtige Teilmenge des entsprechenden Satzes, sondern können die Teilmenge und der entsprechende Satz auch gleich sein.

[0396] Verbindende Sprache, wie etwa Ausdrücke der Form „wenigstens eines von A, B und C“ oder „mindestens eines von A, B und C“, sind andernfalls in dem Kontext zu verstehen, in dem sie allgemein verwendet werden, um darzustellen, dass ein Objekt, ein Begriff usw. entweder A oder B oder C oder eine beliebige nicht leere Teilmenge des Satzes aus A und B und C sein kann, es sei denn, es ist etwas anderes angegeben oder etwas anderes geht eindeutig aus dem Kontext hervor. Zum Beispiel beziehen sich die verbindenden Ausdrücke „wenigstens eines von A, B und C“ und „mindestens eines von A, B und C“ in dem veranschaulichenden Beispiel eines Satzes, der drei Elemente aufweist, auf einen 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 Begriff „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“.

[0397] Hier beschriebene Vorgänge von Prozessen können in einer beliebigen geeigneten Reihenfolge durchgeführt werden, sofern es durch den Kontext nicht anders angegeben ist 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 Vorgänge 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 ausführbare Anweisungen so ausgeführt, dass verschiedene Anweisungen von verschiedenen Prozessoren ausgeführt werden - beispielsweise speichert ein nicht transitorisches computerlesbares Speichermedium Anweisungen und führt eine zentrale Verarbeitungseinheit („CPU“) einige der Anweisungen aus, während eine Grafikverarbeitungseinheit („GPU“) andere Anweisungen ausführt. In mindestens einer Ausführungsform weisen unterschiedliche Komponenten eines Computersystems separate Prozessoren auf und unterschiedliche Prozessoren führen unterschiedliche Teilmengen von Anweisungen aus.

[0398] Dementsprechend sind in mindestens einer Ausführungsform Computersysteme dazu konfiguriert, einen oder mehrere Dienste umzusetzen, die einzeln oder gemeinsam Vorgänge der in dieser Schrift beschriebenen Prozesse ausführen, und derartige Computersysteme sind mit anwendbarer Hardware und/oder Software konfiguriert, welche die Durchführung von 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 Vorgänge durchführt und sodass eine einzelne Vorrichtung nicht alle Vorgänge durchführt.

[0399] Die Verwendung von beliebigen und allen Beispielen oder beispielhafter Sprache (z. B. „wie etwa“), die hier bereitgestellt wird, ist lediglich bestimmt, Ausführungsformen besser zu veranschaulichen und stellt keine Einschränkung hinsichtlich des Umfangs der Offenbarung dar, sofern nicht anders beansprucht. Keine Sprache in der Spezifikation sollte so ausgelegt werden, dass sie ein beliebiges nicht beanspruchtes Element als für Praxis der Offenbarung wesentlich angibt.

[0400] Mindestens eine Ausführungsform kann im Hinblick auf die folgenden Klauseln beschrieben werden.

1. Ein Verfahren, das einen Prozessor verwendet, der eine oder mehrerer Arithmetik-Logikeinheiten (ALUs) zur Verarbeitung eines Bildes umfasst, umfassend:

Durchführen einer Bildregistrierung, welche die Registrierung von gemeinsamen Merkmalen unter mindestens zwei Bildern durch Trainieren eines neuronalen Netzwerks mit einem Trainingsbild und einem Korrespondenzbild angibt, das durch Simulieren eines Prozesses einer Bildregistrierung erzeugt wurde, das eine Registrierung von gemeinsamen Merkmalen zwischen dem Trainingsbild und dem Korrespondenzbild angibt.

2. Verfahren gemäß Klausel 1, wobei das Trainieren des neuronalen Netzwerks umfasst:

Auswählen eines Satzes von abgetasteten Transformationsparametern, die unter einem Bereich von möglichen Transformationsparametern ausgewählt wurden;

Erzeugen, aus dem Satz von abgetasteten Transformationsparametern, ein erstes Verschiebungsfeld, das Verschiebungen von Pixeln in dem Trainingsbild angibt, die gemäß dem Satz von abgetasteten Transformationsparametern verschoben sind;

Transformieren des Trainingsbildes in das Korrespondenzbild gemäß dem ersten Verschiebungsfeld;

Erzeugen, unter Verwendung des neuronalen Netzwerks, eines zweiten Verschiebungsfelds aus dem Trainingsbild und dem Korrespondenzbild;

Transformieren des Trainingsbildes in ein transformiertes Bild gemäß dem zweiten Verschiebungsfeld;

Erzeugen einer Verlustfunktion aus einer ersten Differenz zwischen dem ersten Verschiebungsfeld und dem zweiten Verschiebungsfeld; und

Verwenden der Verlustfunktion, um das neuronale Netzwerk zu trainieren.

3. Verfahren gemäß Klausel 2, wobei das Erzeugen der Verlustfunktion das Erzeugen der Verlustfunktion von der ersten Differenz und von einer zweiten Differenz zwischen dem Korrespondenzbild und dem transformierten Bild umfasst, wobei die Verlustfunktion eine gewichtete Summe der ersten Differenz und der zweiten Differenz ist.

4. Verfahren gemäß Klausel 3, wobei die erste Differenz ein Mittelwert über einen Bildkoordinatenraum einer Norm von Differenzen von Elementen zwischen dem ersten Verschiebungsfeld und dem zweiten Verschiebungsfeld ist und wobei die zweite Differenz eine negative normierte Kreuzkorrelation zwischen dem Korrespondenzbild und dem transformierten Bild ist.

5. Verfahren gemäß einem der Klauseln 2-4, wobei der Bereich von möglichen Transformationsparametern zum Transformieren des Trainingsbildes in das Korrespondenzbild einer oder mehrerer eines Bereich von Drehwinkeln, einem Bereich von Translationen, einem Bereich von Skalierungsfaktoren und einem Bereich von elastischen Verzerrungen umfasst und wobei Auswählen die Satz von abgetasteten Transformationsparameter umfasst Auswählen ein Winkel innerhalb der Bereich von Drehwinkeln, Auswählen einer Translation innerhalb des Bereichs von Translationen, Auswählen eines Skalierungsfaktors innerhalb des Bereichs von Skalierungsfaktoren und/oder Auswählen einer elastischen Verzerrung innerhalb des Bereichs von elastischen Verzerrungen.

6. Verfahren gemäß einem der Klauseln 1-5, wobei das Trainingsbild ein zweidimensionales oder dreidimensionales Bild einer biologischen Umgebung ist.

7. Medizinisches Bildregistrierungssystem, umfassend:

einen Prozessor, der eine oder mehrere Arithmetik-Logikeinheiten (ALUs) umfasst;

einen oder mehrere Speicher, um Parameter eines neuronalen Netzwerks zu speichern, wobei die Parameter des neuronalen Netzwerks durch Trainieren des neuronalen Netzwerks unter Verwendung eines Registrierungssimulators erzeugt werden, der ein erstes Verschiebungsfeld und ein Korrespondenzbild ausgibt, wobei das erste Verschiebungsfeld Verschiebungen von Pixeln in einem medizinischen Trainingsbild angibt, das gemäß einem Satz von abgetasteten Transformationsparametern verschoben wurde, die unter einem Bereich von möglichen Transformationsparametern ausgewählt wurden, und wobei das Korrespondenzbild das medizinische Trainingsbild ist, das gemäß dem ersten Verschiebungsfeld transformiert wurde, wobei das Trainieren des neuronalen Netzwerks ein Optimieren einer Verlustfunktion von einer ersten Differenz zwischen dem ersten Verschiebungsfeld und einem zweiten Verschiebungsfeld umfasst, das von dem neuronalen Netzwerk ausgegeben wird, das dem neuronalen Netzwerk entspricht, das die Bildregistrierung auf dem medizinischen Trainingsbild und dem Korrespondenzbild durchführt;

eine Eingabe zum Empfangen von zwei oder mehreren Bildern einer biologischen Umgebung, wobei mindestens eine Überlappung der biologischen Umgebung in einem Paar von Bildern der beiden oder mehreren Bildern vorhanden ist; und

eine Ausgabe zum Ausgeben eines Verschiebungsfeldes des Paares von Bildern, die durch das neuronale Netzwerk erzeugt wurden, das unter Verwendung des Registrierungssimulators trainiert wurde.

8. Medizinisches Bildregistrierungssystem gemäß Klausel 7, wobei die Verlustfunktion eine gewichtete Summe der ersten Differenz und einer zweiten Differenz zwischen dem medizinischen Trainingsbild und dem Korrespondenzbild ist.

9. Medizinisches Bildregistrierungssystem gemäß Klausel 8, wobei die erste Differenz ein Mittelwert über einen Bildkoordinatenraum einer Norm von Differenzen von Elementen zwischen dem ersten Verschiebungsfeld und dem zweiten Verschiebungsfeld ist, und wobei die zweite Differenz eine negative normierte Kreuzkorrelation zwischen dem entsprechenden Bild und einem transformierten Bild ist, welches das medizinische Trainingsbild umfasst, das gemäß dem zweiten Verschiebungsfeld transformiert wurden.

10. Medizinisches Bildregistrierungssystem gemäß einem der Klauseln 7-9, wobei der Bereich von möglichen Transformationsparametern einen oder mehrere eines Bereichs von Drehwinkeln, eines Bereichs von Translationen, eines Bereichs von Skalierungsfaktoren und eines Bereich von elastischen Verzerrungen umfasst, und wobei das Auswählen des Satzes von abgetasteten Transformationsparameter das Auswählen eines Winkels innerhalb des Bereichs von Drehwinkeln, das Auswählen einer Translation innerhalb des Bereichs von Translationen, das Auswählen eines Skalierungsfaktors innerhalb des Bereichs von Skalierungsfaktoren und/oder das Auswählen einer elastischen Verzerrung innerhalb des Bereichs von elastischen Verzerrungen umfasst, wobei der Bereich von möglichen Transformationsparameter möglichen Abweichungen zwischen mehreren Bildern von Bildern der biologischen Umgebung entspricht.

11. Medizinisches Bildregistrierungssystem gemäß einem der Klauseln 7-10, ferner umfassend:

einen Bildsegmentierer, der Segmentierungsmasken ausgibt, die Grenzen von Objekten in Bildern darstellen; und

Speicherung in dem einen oder mehreren Speichern für ein Testbild und eine erste Segmentierungsmaske des Testbildes,

wobei der Bildsegmentierer konfiguriert ist, um ein eingegebenes medizinisches Bild zu segmentieren, um eine zweite Segmentierungsmaske für das eingegebene medizinische Bild gemäß einem dritten Verschiebungsfeld zu bilden, das durch das neuronale Netzwerk ausgegeben wird, wobei das dritte Verschiebungsfeld einer Registrierung des Testbildes und des eingegebenen medizinischen Bildes entspricht.

12. Medizinisches Bildregistrierungssystem gemäß einem der Klauseln 7-11, ferner umfassend:

einen Bildsegmentierer, der Segmentierungsmasken ausgibt, die Grenzen von Objekten in Bildern darstellen; und

Speicherung in dem einen oder mehreren Speichern für ein erstes Trainingsbild, ein zweites Trainingsbild, dem ersten Verschiebungsfeld, das der Registrierung des ersten Trainingsbildes entspricht, und einem ersten simulierten Registrierungsbildes, das durch den Registrierungssimulator basierend auf dem ersten Trainingsbild ausgegeben wird, dem zweiten Verschiebungsfeld, das durch das neuronale Netzwerk zur Registrierung des ersten Trainingsbildes und des ersten simulierten Registrierungsbildes ausgegeben wird, und eines dritten Verschiebungsfeldes, das der Registrierung des ersten Trainingsbildes und des zweiten Trainingsbildes entspricht,

wobei das neuronale Netzwerk auf einer dualen Registrierungs-Verlustfunktion trainiert wird, die eine erste Funktion von Differenzen zwischen dem ersten Trainingsbild, das gemäß dem zweiten Verschiebungsfeld transformiert wird, und dem ersten simulierten Registrierungsbildes ist, Differenzen zwischen dem ersten Trainingsbild, das gemäß dem dritten Verschiebungsfeld transformiert wird, und dem zweiten Trainingsbild, und/oder Differenzen zwischen dem zweiten Verschiebungsfeld und dem dritten Verschiebungsfeld.

13. Medizinisches Bildregistrierungssystem gemäß Klausel 12, ferner umfassend eine Speicherung für eine erste Segmentierungsmaske des ersten Trainingsbildes und einer zweiten Segmentierungsmaske des zweiten Trainingsbildes,

wobei das neuronale Netzwerk auf einer Segmentierungs-Verlustfunktion trainiert wird, die eine zweiten Funktion von Differenzen zwischen der ersten Segmentierungsmaske, die gemäß dem zweiten Verschiebungsfeld transformiert wurde, und einer dritten Segmentierungsmaske und/oder Differenzen zwischen der ersten Segmentierungsmaske, die gemäß dem dritten Verschiebungsfeld transformiert wurde, und der zweiten Segmentierungsmaske, wobei die dritte Segmentierungsmaske die erste Segmentierungsmaske ist, die gemäß dem ersten Verschiebungsfeld transformiert wurde.

14. Medizinisches Bildregistrierungssystem gemäß einem der Klauseln 7-13, ferner umfassend:

einen Bildsegmentierer, der Segmentierungsmasken ausgibt, die Grenzen von Objekten in Bildern darstellen; und

Speicherung in dem einen oder mehreren Speichern für ein erstes Trainingsbild, eine erste Segmentierungsmaske des ersten Trainingsbildes, ein zweites Trainingsbild, einer zweiten Segmentierungsmaske des zweiten Trainingsbildes, des ersten Verschiebungsfeldes, das der Registrierung des ersten Trainingsbildes entspricht, und eines ersten simulierten Registrierungsbildes, das durch den Registrierungssimulator basierend auf dem ersten Trainingsbild ausgegeben wird, des zweiten Verschiebungsfeldes, das durch das neuronale Netzwerk zur Registrierung des ersten Trainingsbildes ausgegeben wird, und des ersten simulierten Registrierungsbildes, und eines dritten Verschiebungsfeldes, das der Registrierung des ersten Trainingsbildes und des zweiten Trainingsbildes entspricht,

wobei das neuronale Netzwerk auf einer Segmentierungs-Verlustfunktion trainiert wird, die eine Funktion von Differenzen zwischen der ersten Segmentierungsmaske transformierten gemäß der zweiten Verschiebungsfeld und einer dritten Segmentierungsmaske ist, und/oder Differenzen zwischen der ersten Segmentierungsmaske, die gemäß dem dritten Verschiebungsfeld transformiert wurde, und der zweiten Segmentierungsmaske, wobei die dritte Segmentierungsmaske die erste Segmentierungsmaske ist, die gemäß dem ersten Verschiebungsfeld transformiert wurde.

15. System, umfassend:

einen oder mehrere Prozessoren, um Parameter, die einem oder mehreren neuronalen Netzwerken entsprechen, zum Durchführen einer Bildregistrierung zu berechnen, welche die Registrierung von gemeinsamen Merkmalen unter mindestens zwei Bildern mindestens teilweise angibt, durch Simulieren eines Prozesses der Bildregistrierung, der die Registrierung von gemeinsamen Merkmalen zwischen einem Trainingsbild und einem Korrespondenzbild angibt, Anwenden des Trainingsbildes und des Korrespondenzbildes auf das neuronale Netzwerk, und Bereitstellen einer Verlustfunktion für das neuronale Netzwerk, um zu optimieren, wobei die Verlustfunktion eine erste Funktion von Differenzen zwischen einem simulierten Verschiebungsfeld und einem ausgegeben Verschiebungsfeld ist, welches das eine oder mehrere neuronale Netzwerke basierend auf versuchter Registrierung des Trainingsbildes und des Korrespondenzbildes ausgibt; und

einen oder mehrere Speicher, um die Parameter zu speichern, wobei die Parameter eine zweite Funktion sind, die für die Verlustfunktion optimiert.

16. System gemäß Klausel 15, ferner umfassend:

Speicherung für einen Satz von abgetasteten Transformationsparametern, die unter einem Bereich von möglichen Transformationsparametern ausgewählt wurden, Speicherung für ein erstes Verschiebungsfeld, das Verschiebungen von Pixeln in dem Trainingsbild angibt, die gemäß dem Satz von abgetasteten Transformationsparametern verschoben wurden,

wobei die Verlustfunktion, die verwendet wurde, um das neuronale Netzwerk zu trainieren, eine dritte Funktion einer ersten Differenzfunktion von Differenzen zwischen dem ersten Verschiebungsfeld und einem zweiten Verschiebungsfeld ist, wobei das zweite Verschiebungsfeld durch das von dem ausgegeben Verschiebungsfeld neuronale Netzwerk erzeugt wird.

17. System gemäß Klausel 16, wobei die Verlustfunktion eine gewichtete Summe der ersten Differenzfunktion und einer zweiten Differenzfunktion von Differenzen zwischen dem Korrespondenzbild und einem transformierten Bild des Trainingsbildes ist, das gemäß dem zweiten Verschiebungsfeld transformiert wurde.

18. System gemäß Klausel 17, wobei die erste Differenzfunktion ein Mittelwert über einen Bildkoordinatenraum einer Norm von Differenzen von Elementen zwischen dem ersten Verschiebungsfeld und dem zweiten Verschiebungsfeld ist und wobei die zweite Differenzfunktion eine negative normierte Kreuzkorrelation zwischen dem Korrespondenzbild und dem transformierten Bild ist.

19. System gemäß einem der Klauseln 16-18, wobei der Bereich von möglichen Transformationsparametern zum Transformieren des Trainingsbildes in das Korrespondenzbild einen oder mehrere eines Bereichs von Drehwinkeln, eines Bereichs von Translationen, eines Bereichs von Skalierungsfaktoren und eines Bereichs von elastischen Verzerrungen umfasst, und wobei das Auswählen des Satzes von abgetasteten Transformationsparametern das Auswählen eines Winkels innerhalb des Bereichs von Drehwinkeln, das Auswählen einer Translation innerhalb des Bereichs von Translationen, das Auswählen eines Skalierungsfaktors innerhalb des Bereichs von Skalierungsfaktoren und/oder das Auswählen einer elastischen Verzerrung innerhalb des Bereichs von elastischen Verzerrungen umfasst.

20. System gemäß einem der Klauseln 15-19, wobei das Trainingsbild ein zweidimensionales oder dreidimensionales Bild einer biologischen Umgebung ist.

21. System gemäß einem der Klauseln 15-20, ferner umfassend:

eine Faltungsschicht zum Modifizieren einer letzten neuronalen Netzwerksschicht, das eine Faltung einer letzten Schicht des einen oder mehrerer neuronaler Netzwerke und eine vorhergesagte Segmentierung bereitstellt; und

eine Softmax-Aktivierungsschicht zum Modifizieren der vorhergesagten Segmentierung gemäß einer Softmax-Operation auf die Faltung.

22. Verfahren, umfassend das Trainieren eines oder mehrerer neuronaler Netzwerke durch:

Erhalten eines ersten Trainingsbildes, einer ersten Segmentierungsmaske des ersten Trainingsbildes, eines zweiten Trainingsbildes und einer zweiten Segmentierungsmaske des zweiten Trainingsbildes;

Auswählen eines Satzes von abgetasteten Transformationsparametern, die aus einem Bereich von möglichen Transformationsparametern ausgewählt wurden;

Erzeugen, von dem Satz von abgetasteten Transformationsparametern, ein erstes Verschiebungsfeld, das Verschiebungen von Pixeln in dem ersten Trainingsbild angibt, die gemäß dem Satz von abgetasteten Transformationsparametern verschoben werden;

Erzeugen einer Korrespondenzbildes durch Transformieren des ersten Trainingsbildes gemäß dem ersten Verschiebungsfeld;

Erzeugen, unter Verwendung des einen oder mehrerer neuronaler Netzwerke, eines zweiten Verschiebungsfeldes, das eine erste versuchte Registrierung des ersten Trainingsbildes und des Korrespondenzbildes darstellt;

Erzeugen eines transformierten Bildes durch Transformieren des ersten Trainingsbildes gemäß dem zweiten Verschiebungsfeld;

Erzeugen einer Korrespondenz-Segmentierungsmaske durch Transformieren der ersten Segmentierungsmaske gemäß dem ersten Verschiebungsfeld;

Erzeugen einer transformierten Segmentierungsmaske durch Transformieren der ersten Segmentierungsmaske gemäß dem zweiten Verschiebungsfeld;

Erzeugen einer Verlustfunktion aus einer Funktion einer ersten Differenz zwischen der Korrespondenz-Segmentierungsmaske und der transformierten Segmentierungsmaske; und

Verwenden der Verlustfunktion, um das neuronale Netzwerk zu trainieren.

23. Verfahren gemäß Klausel 22, wobei die Verlustfunktion eine gewichtete Summe der ersten Differenz und einer zweiten Differenz zwischen dem entsprechenden Bild und dem transformierten Bild ist.

24. Verfahren gemäß Klausel 23, wobei die erste Differenz ein Mittelwert über einen Bildkoordinatenraum einer Norm von Differenzen von Elementen zwischen dem ersten Verschiebungsfeld und dem zweiten Verschiebungsfeld ist, und wobei die zweite Differenz eine negative normierte Kreuzkorrelation zwischen dem entsprechenden Bild und dem transformierten Bild ist.

25. Verfahren gemäß einem der Klauseln 22-24, wobei der Bereich von möglichen Transformationsparametern einen oder mehrere eines Bereichs von Drehwinkeln, eines Bereichs von Translationen, eines Bereichs von Skalierungsfaktoren und eines Bereichs von elastischen Verzerrungen umfasst, und wobei das Auswählen des Satzes von abgetasteten Transformationsparametern das Auswählen eines Winkels innerhalb des Bereichs von Drehwinkeln, das Auswählen einer Translation innerhalb des Bereichs von Translationen, das Auswählen eines Skalierungsfaktors innerhalb des Bereichs von Skalierungsfaktoren und/oder das Auswählen einer elastischen Verzerrung innerhalb des Bereichs von elastischen Verzerrungen, wobei der Bereich von möglichen Transformationsparametern möglichen Abweichungen zwischen mehreren Bildern einer Quelle von Bildern entspricht.

26. Verfahren gemäß Klausel 25, wobei die Abweichungen zwischen mehreren Bildern von Bildern der Quelle von Bildern Abweichungen zwischen Bildern einer biologischen Umgebung entsprechen.

27. Verfahren gemäß einem der Klauseln 22-26, ferner umfassend:

Erzeugen, unter Verwendung des einen oder mehrerer neuronaler Netzwerke, eines dritten Verschiebungsfeldes, das eine zweite versuchte Registrierung des ersten Trainingsbildes und des zweiten Trainingsbildes darstellt;

Erzeugen einer dritten Segmentierungsmaske durch Transformieren der ersten Segmentierungsmaske gemäß dem ersten Verschiebungsfeld;

Erzeugen einer vierten Segmentierungsmaske durch Transformieren der ersten Segmentierungsmaske gemäß dem zweiten Verschiebungsfeld;

Erzeugen einer fünften Segmentierungsmaske durch Transformieren der ersten Segmentierungsmaske gemäß dem dritten Verschiebungsfeld;

Bestimmen einer Segmentierungs-Verlustfunktion, die eine erste Funktion einer ersten Differenzfunktion von Differenzen zwischen der dritten Segmentierungsmaske und der vierten Segmentierungsmaske und/oder einer zweiten Differenzfunktion von Differenzen zwischen der fünften Segmentierungsmaske und der zweiten Segmentierungsmaske ist;

Trainieren des einen oder mehrerer neuronaler Netzwerke durch Optimieren der Segmentierungs-Verlustfunktion.

28. Verfahren gemäß Klausel 27, ferner umfassend:

Erzeugen eines ersten transformierten Bildes durch Transformieren des ersten Trainingsbildes gemäß dem zweiten Verschiebungsfeld;

Erzeugen eines zweiten transformierten Bildes durch Transformieren des ersten Trainingsbildes gemäß dem dritten Verschiebungsfeld; und

Trainieren des einen oder mehrerer neuronaler Netzwerke durch Optimieren der Verlustfunktion, wobei die Verlustfunktion die Segmentierungs-Verlustfunktion, eine dritte Differenzfunktion von Differenzen zwischen der dritten Segmentierungsmaske und der vierten Segmentierungsmaske, einer vierten Differenzfunktion von Differenzen zwischen der fünften Segmentierungsmaske und der zweiten Segmentierungsmaske und eine fünfte Differenzfunktion von Differenzen zwischen dem ersten Verschiebungsfeld und dem zweiten Verschiebungsfeld umfasst.

29. Verfahren gemäß Klausel 28, wobei die Verlustfunktion eine gewichtete Summe der ersten Differenzfunktion, der zweiten Differenzfunktion, der dritten Differenzfunktion, der vierten Differenzfunktion und der fünften Differenzfunktion ist.

[0401] Sämtliche Bezugnahmen, einschließlich Veröffentlichungen, Patenanmeldungen und Patente, die hier zitiert sind, werden hiermit durch Bezugnahme in dem gleichen Ausmaß aufgenommen, als ob jede Bezugnahme einzeln und speziell angegeben würde, um durch Bezugnahme aufgenommen zu werden, und in ihrer Gesamtheit hier dargelegt wären.

[0402] In der Beschreibung und den Ansprüchen können die Begriffe „gekoppelt“ und „verbunden“ zusammen mit ihren Ableitungen verwendet werden. Es sei zu verstehen, dass diese Begriffe nicht als Synonyme füreinander bestimmt sind. Stattdessen kann in bestimmten Beispielen „verbunden“ oder „gekoppelt“ verwendet werden, um anzugeben, dass zwei oder mehr Elemente in direktem oder indirektem physischen oder elektrischen Kontakt miteinander sind. „Gekoppelt“ kann ebenfalls bedeuten, dass zwei oder mehr Elemente nicht in direktem Kontakt miteinander sind, jedoch dennoch weiterhin kooperieren oder miteinander wechselwirken.

[0403] Sofern nicht speziell anders angegeben, wird anerkannt, dass die in dieser Spezifikation enthaltenen Begriffe wie „Verarbeiten“, „Rechnen“, „Berechnen“, „Bestimmen“ oder dergleichen sich auf eine Aktion und/oder Verfahren eines Computers oder eines Rechensystems beziehen, wie einem oder mehreren Computern oder einer ähnlichen elektronischen Vorrichtung, die Daten manipulieren oder umwandeln, die als physikalische, wie beispielsweise elektronische, Größen in den Registern des Rechensystems und/oder Speichern in andere Daten, die auf ähnliche Weise als physikalische Größen innerhalb der Speicher des Rechensystems, Registern oder anderer derartiger Informationsspeicherungs-, Übertragungs- oder Anzeigevorrichtungen dargestellt werden.

[0404] Auf eine ähnliche Art und Weise kann sich der Begriff „Prozessor“ auf jede Vorrichtung oder Abschnitt einer Vorrichtung beziehen, die elektronische Daten aus Registern und/oder Speicher verarbeitet und diese elektronischen Daten in andere elektronische Daten transformiert, die in Registern und/oder Speicher gespeichert werden können. Als nicht einschränkende Beispiele kann „Prozessor“ eine CPU oder eine GPU sein. Eine „Rechenplattform“ kann einen oder mehrere Prozessoren umfassen. Wie hier verwendet, können „Software“-Prozesse beispielsweise Software- und/oder Hardwareentitäten umfassen, die Arbeit im Laufe der Zeit durchführen, wie beispielsweise Aufgaben, Threads und intelligente Agenten. Jeder Prozess

kann sich ebenfalls auf mehrere Prozesse zum Ausführen von Anweisungen der Reihe nach oder parallel, kontinuierlich oder intermittierend beziehen. In mindestens einer Ausführungsform werden die Begriffe „System“ und „Verfahren“ hier austauschbar verwendet, insofern als System ein oder mehrere Verfahren verkörpern und Verfahren als ein System betrachtet werden können.

[0405] In dem vorliegenden Dokument können Verweise auf das Erhalten, Erfassen, Empfangen oder Eingeben von analogen oder digitalen Daten in ein Teilsystem, Computersystem oder computerimplementierte Maschine erfolgen. In mindestens einer Ausführungsform kann der Prozess des Erhaltens, Erfassens, Empfangens oder Eingebens von analogen und digitalen Daten in einer Vielfalt von Möglichkeiten erreicht werden, wie beispielsweise durch Empfangen von Daten als ein Parameter eines Funktionsaufrufs oder eines Aufrufs an eine Anwendungsprogrammierschnittstelle. In einigen Implementierungen kann der Prozess des Erhaltens, Erfassens, Empfangen oder Eingebens von analogen oder digitalen Daten durch Transferieren von Daten über eine serielle oder parallele Schnittstelle erreicht werden. In mindestens einer Ausführungsform können Prozesse des Erhaltens, Erfassens, Empfangens oder Eingebens von analogen oder digitalen Daten durch Transferieren von Daten über ein Computernetzwerk von bereitstellender Entität zu erfassender Entität erreicht werden. In mindestens einer Ausführungsform können Verweise ebenfalls auf das Bereitstellen, Ausgeben, übertragen, Senden oder Repräsentieren analoger oder digitaler Daten erfolgen. In verschiedenen Beispielen kann der Prozess des Bereitstellens, Ausgebens, Übertragens, Sendens oder Repräsentierens analoger oder digitaler Daten durch Transferieren von Daten als ein Eingangs- oder Ausgangsparameter eines Funktionsaufrufes, eines Parameters einer Anwendungsprogrammierschnittstelle oder eines Mechanismus zur Interprozesskommunikation erreicht werden.

[0406] Obwohl Beschreibungen hier beispielhafte Implementierungen von beschriebenen Techniken darlegen, können andere Architekturen verwendet werden, um die beschriebene Funktionalität zu implementieren, und sind bestimmt, innerhalb des Umfangs der Offenbarung zu liegen. Des Weiteren könnten, obwohl spezifische Verteilungen von Verantwortlichkeiten obenstehend zum Zweck der Erläuterung definiert sind, verschiedene Funktionen und Verantwortlichkeiten in unterschiedlichen Weisen abhängig von den Umständen verteilt und aufgeteilt werden.

[0407] Des Weiteren sei zu verstehen, obwohl der Gegenstand in einer für strukturelle Merkmale und/oder methodische Handlungen spezifischen Sprache beschrieben wurde, dass der in der Offenbarung definierte Gegenstand nicht notwendigerweise auf beschriebene spezifische Merkmale oder Handlungen beschränkt ist. Vielmehr werden spezifische Merkmale und Handlungen als beispielhafte Formen des Implementierens der Ansprüche offenbart.

ZITATE ENTHALTEN IN DER BESCHREIBUNG

Zitierte Patentliteratur

- US 16/540717 [0001]

Patentansprüche

1. Verfahren, das einen Prozessor verwendet, der eine oder mehrerer Arithmetik-Logikeinheiten (ALUs) zur Verarbeitung eines Bildes umfasst, umfassend:

Durchführen einer Bildregistrierung, welche die Registrierung von gemeinsamen Merkmalen unter mindestens zwei Bildern durch Trainieren eines neuronalen Netzwerks mit einem Trainingsbild und einem Korrespondenzbild angibt, das durch Simulieren eines Prozesses einer Bildregistrierung erzeugt wurde, das eine Registrierung von gemeinsamen Merkmalen zwischen dem Trainingsbild und dem Korrespondenzbild angibt.

2. Verfahren gemäß Anspruch 1, wobei das Trainieren des neuronalen Netzwerks umfasst:

Auswählen eines Satzes von abgetasteten Transformationsparametern, die unter einem Bereich von möglichen Transformationsparametern ausgewählt wurden;

Erzeugen, aus dem Satz von abgetasteten Transformationsparametern, ein erstes Verschiebungsfeld, das Verschiebungen von Pixeln in dem Trainingsbild angibt, die gemäß dem Satz von abgetasteten Transformationsparametern verschoben sind;

Transformieren des Trainingsbildes in das Korrespondenzbild gemäß dem ersten Verschiebungsfeld;

Erzeugen, unter Verwendung des neuronalen Netzwerks, eines zweiten Verschiebungsfelds aus dem Trainingsbild und dem Korrespondenzbild;

Transformieren des Trainingsbildes in ein transformiertes Bild gemäß dem zweiten Verschiebungsfeld;

Erzeugen einer Verlustfunktion aus einer ersten Differenz zwischen dem ersten Verschiebungsfeld und dem zweiten Verschiebungsfeld; und

Verwenden der Verlustfunktion, um das neuronale Netzwerk zu trainieren.

3. Verfahren gemäß Anspruch 2, wobei das Erzeugen der Verlustfunktion das Erzeugen der Verlustfunktion von der ersten Differenz und von einer zweiten Differenz zwischen dem Korrespondenzbild und dem transformierten Bild umfasst, wobei die Verlustfunktion eine gewichtete Summe der ersten Differenz und der zweiten Differenz ist.

4. Verfahren gemäß Anspruch 3, wobei die erste Differenz ein Mittelwert über einen Bildkoordinatenraum einer Norm von Differenzen von Elementen zwischen dem ersten Verschiebungsfeld und dem zweiten Verschiebungsfeld ist und wobei die zweite Differenz eine negative normierte Kreuzkorrelation zwischen dem Korrespondenzbild und dem transformierten Bild ist.

5. Verfahren gemäß Anspruch 2, wobei der Bereich von möglichen Transformationsparametern zum Transformieren des Trainingsbildes in das Korrespondenzbild einer oder mehrerer eines Bereich von Drehwinkeln, einem Bereich von Translationen, einem Bereich von Skalierungsfaktoren und einem Bereich von elastischen Verzerrungen umfasst und wobei Auswählen die Satz von abgetasteten Transformationsparameter umfasst Auswählen ein Winkel innerhalb die Bereich von Drehwinkels, Auswählen einer Translation innerhalb des Bereichs von Translationen, Auswählen eines Skalierungsfaktors innerhalb des Bereichs von Skalierungsfaktoren und/oder Auswählen einer elastischen Verzerrung innerhalb des Bereichs von elastischen Verzerrungen.

6. Verfahren gemäß Anspruch 1, wobei das Trainingsbild ein zweidimensionales oder dreidimensionales Bild einer biologischen Umgebung ist.

7. Medizinisches Bildregistrierungssystem, umfassend:

einen Prozessor, der eine oder mehrere Arithmetik-Logikeinheiten (ALUs) umfasst;

einen oder mehrere Speicher, um Parameter eines neuronalen Netzwerks zu speichern, wobei die Parameter des neuronalen Netzwerks durch Trainieren des neuronalen Netzwerks unter Verwendung eines Registrierungs-simulators erzeugt werden, der ein erstes Verschiebungsfeld und ein Korrespondenzbild ausgibt, wobei das erste Verschiebungsfeld Verschiebungen von Pixeln in einem medizinischen Trainingsbild angibt, das gemäß einem Satz von abgetasteten Transformationsparameter verschoben wurde, die unter einem Bereich von möglichen Transformationsparameter ausgewählt wurden, und wobei das Korrespondenzbild das medizinischen Trainingsbild ist, das gemäß dem ersten Verschiebungsfeld transformiert wurde, wobei das Trainieren des neuronalen Netzwerks ein Optimieren einer Verlustfunktion von einer ersten Differenz zwischen dem ersten Verschiebungsfeld und einem zweiten Verschiebungsfeld umfasst, das von dem neuronalen Netzwerk ausgegeben wird, das dem neuronalen Netzwerk entspricht, das die Bildregistrierung auf dem medizinischen Trainingsbild und dem Korrespondenzbild durchführt;

eine Eingabe zum Empfangen von zwei oder mehreren Bildern einer biologischen Umgebung, wobei min-

destens eine Überlappung der biologischen Umgebung in einem Paar von Bildern der beiden oder mehreren Bildern vorhanden ist; und
eine Ausgabe zum Ausgeben eines Verschiebungsfeldes des Paares von Bildern, die durch das neuronale Netzwerk erzeugt wurden, das unter Verwendung des Registrierungs-simulators trainiert wurde.

8. Medizinisches Bildregistrierungssystem gemäß Anspruch 7, wobei die Verlustfunktion eine gewichtete Summe der ersten Differenz und einer zweiten Differenz zwischen dem medizinischen Trainingsbild und dem Korrespondenzbild ist.

9. Medizinisches Bildregistrierungssystem gemäß Anspruch 8, wobei die erste Differenz ein Mittelwert über einen Bildkoordinatenraum einer Norm von Differenzen von Elementen zwischen dem ersten Verschiebungsfeld und dem zweiten Verschiebungsfeld ist, und wobei die zweite Differenz eine negative normierte Kreuzkorrelation zwischen dem entsprechenden Bild und einem transformierten Bild ist, welches das medizinische Trainingsbild umfasst, das gemäß dem zweiten Verschiebungsfeld transformiert wurden.

10. Medizinisches Bildregistrierungssystem gemäß Anspruch 7, wobei der Bereich von möglichen Transformationsparametern einen oder mehrere eines Bereichs von Drehwinkeln, eines Bereichs von Translationen, eines Bereichs von Skalierungsfaktoren und eines Bereich von elastischen Verzerrungen umfasst, und wobei das Auswählen des Satzes von abgetasteten Transformationsparameter das Auswählen eines Winkels innerhalb des Bereichs von Drehwinkeln, das Auswählen einer Translation innerhalb des Bereichs von Translationen, das Auswählen eines Skalierungsfaktors innerhalb des Bereichs von Skalierungsfaktoren und/oder das Auswählen einer elastischen Verzerrung innerhalb des Bereichs von elastischen Verzerrungen umfasst, wobei der Bereich von möglichen Transformationsparameter möglichen Abweichungen zwischen mehreren Bildern von Bildern der biologischen Umgebung entspricht.

11. Medizinisches Bildregistrierungssystem gemäß Anspruch 7, ferner umfassend:
einen Bildsegmentierer, der Segmentierungsmasken ausgibt, die Grenzen von Objekten in Bildern darstellen; und
Speicherung in dem einen oder mehreren Speichern für ein Testbild und eine erste Segmentierungsmaske des Testbildes,
wobei der Bildsegmentierer konfiguriert ist, um ein eingegebenes medizinisches Bild zu segmentieren, um eine zweite Segmentierungsmaske für das eingegebene medizinische Bild gemäß einem dritten Verschiebungsfeld zu bilden, das durch das neuronale Netzwerk ausgegeben wird, wobei das dritte Verschiebungsfeld einer Registrierung des Testbildes und des eingegebenen medizinischen Bildes entspricht.

12. Medizinisches Bildregistrierungssystem gemäß Anspruch 7, ferner umfassend:
einen Bildsegmentierer, der Segmentierungsmasken ausgibt, die Grenzen von Objekten in Bildern darstellen; und
Speicherung in dem einen oder mehreren Speichern für ein erstes Trainingsbild, ein zweites Trainingsbild, dem ersten Verschiebungsfeld, das der Registrierung des ersten Trainingsbildes entspricht, und einem ersten simulierten Registrierungs-bildes, das durch den Registrierungs-simulator basierend auf dem ersten Trainingsbild ausgegeben wird, dem zweiten Verschiebungsfeld, das durch das neuronale Netzwerk zur Registrierung des ersten Trainingsbildes und des ersten simulierten Registrierungs-bildes ausgegeben wird, und eines dritten Verschiebungsfeldes, das der Registrierung des ersten Trainingsbildes und des zweiten Trainingsbildes entspricht,
wobei das neuronale Netzwerk auf einer dualen Registrierungs-Verlustfunktion trainiert wird, die eine erste Funktion von Differenzen zwischen dem ersten Trainingsbild, das gemäß dem zweiten Verschiebungsfeld transformiert wird, und dem ersten simulierten Registrierungs-bildes ist, Differenzen zwischen dem ersten Trainingsbild, das gemäß dem dritten Verschiebungsfeld transformiert wird, und dem zweiten Trainingsbild, und/oder Differenzen zwischen dem zweiten Verschiebungsfeld und dem dritten Verschiebungsfeld.

13. Medizinisches Bildregistrierungssystem gemäß Anspruch 12, ferner umfassend eine Speicherung für eine erste Segmentierungsmaske des ersten Trainingsbildes und einer zweiten Segmentierungsmaske des zweiten Trainingsbildes, wobei das neuronale Netzwerk auf einer Segmentierungs-Verlustfunktion trainiert wird, die eine zweite Funktion von Differenzen zwischen der ersten Segmentierungsmaske, die gemäß dem zweiten Verschiebungsfeld transformiert wurde, und einer dritten Segmentierungsmaske und/oder Differenzen zwischen der ersten Segmentierungsmaske, die gemäß dem dritten Verschiebungsfeld transformiert wurde, und der zweiten Segmentierungsmaske, wobei die dritte Segmentierungsmaske die erste Segmentierungsmaske ist, die gemäß dem ersten Verschiebungsfeld transformiert wurde.

14. Medizinisches Bildregistrierungssystem gemäß Anspruch 7, ferner umfassend: einen Bildsegmentierer, der Segmentierungsmasken ausgibt, die Grenzen von Objekten in Bildern darstellen; und Speicherung in dem einen oder mehreren Speichern für ein erstes Trainingsbild, eine erste Segmentierungsmaske des ersten Trainingsbildes, ein zweites Trainingsbild, einer zweiten Segmentierungsmaske des zweiten Trainingsbildes, des ersten Verschiebungsfeldes, das der Registrierung des ersten Trainingsbildes entspricht, und eines ersten simulierten Registrierungsbildes, das durch den Registrierungssimulator basierend auf dem ersten Trainingsbild ausgegeben wird, des zweiten Verschiebungsfeldes, das durch das neuronale Netzwerk zur Registrierung des ersten Trainingsbildes ausgegeben wird, und des ersten simulierten Registrierungsbildes, und eines dritten Verschiebungsfeldes, das der Registrierung des ersten Trainingsbildes und des zweiten Trainingsbildes entspricht, wobei das neuronale Netzwerk auf einer Segmentierungs-Verlustfunktion trainiert wird, die eine Funktion von Differenzen zwischen der ersten Segmentierungsmaske transformierten gemäß der zweiten Verschiebungsfeld und einer dritten Segmentierungsmaske ist, und/oder Differenzen zwischen der ersten Segmentierungsmaske, die gemäß dem dritten Verschiebungsfeld transformiert wurde, und der zweiten Segmentierungsmaske, wobei die dritte Segmentierungsmaske die erste Segmentierungsmaske ist, die gemäß dem ersten Verschiebungsfeld transformiert wurde.

15. System, umfassend: einen oder mehrere Prozessoren, um Parameter, die einem oder mehreren neuronalen Netzwerken entsprechen, zum Durchführen einer Bildregistrierung zu berechnen, welche die Registrierung von gemeinsamen Merkmalen unter mindestens zwei Bildern mindestens teilweise angibt, durch Simulieren eines Prozesses der Bildregistrierung, der die Registrierung von gemeinsamen Merkmalen zwischen einem Trainingsbild und einem Korrespondenzbild angibt, Anwenden des Trainingsbildes und des Korrespondenzbildes auf das neuronale Netzwerk, und Bereitstellen einer Verlustfunktion für das neuronale Netzwerk, um zu optimieren, wobei die Verlustfunktion eine erste Funktion von Differenzen zwischen einem simulierten Verschiebungsfeld und einem ausgegebenen Verschiebungsfeld ist, welches das eine oder mehrere neuronale Netzwerke basierend auf versuchter Registrierung des Trainingsbildes und des Korrespondenzbildes ausgibt; und einen oder mehrere Speicher, um die Parameter zu speichern, wobei die Parameter eine zweite Funktion sind, die für die Verlustfunktion optimiert.

16. System gemäß Anspruch 15, ferner umfassend: Speicherung für einen Satz von abgetasteten Transformationsparametern, die unter einem Bereich von möglichen Transformationsparametern ausgewählt wurden, Speicherung für ein erstes Verschiebungsfeld, das Verschiebungen von Pixeln in dem Trainingsbild angibt, die gemäß dem Satz von abgetasteten Transformationsparametern verschoben wurden, wobei die Verlustfunktion, die verwendet wurde, um das neuronale Netzwerk zu trainieren, eine dritte Funktion einer ersten Differenzfunktion von Differenzen zwischen dem ersten Verschiebungsfeld und einem zweiten Verschiebungsfeld ist, wobei das zweite Verschiebungsfeld durch das von dem ausgegebenen Verschiebungsfeld neuronale Netzwerk erzeugt wird.

17. System gemäß Anspruch 16, wobei die Verlustfunktion eine gewichtete Summe der ersten Differenzfunktion und einer zweiten Differenzfunktion von Differenzen zwischen dem Korrespondenzbild und einem transformierten Bild des Trainingsbildes ist, das gemäß dem zweiten Verschiebungsfeld transformiert wurde.

18. System gemäß Anspruch 17, wobei die erste Differenzfunktion ein Mittelwert über einen Bildkoordinatenraum einer Norm von Differenzen von Elementen zwischen dem ersten Verschiebungsfeld und dem zweiten Verschiebungsfeld ist und wobei die zweite Differenzfunktion eine negative normierte Kreuzkorrelation zwischen dem Korrespondenzbild und dem transformierten Bild ist.

19. System gemäß Anspruch 16, wobei der Bereich von möglichen Transformationsparametern zum Transformieren des Trainingsbildes in das Korrespondenzbild einen oder mehrere eines Bereichs von Drehwinkeln, eines Bereichs von Translationen, eines Bereichs von Skalierungsfaktoren und eines Bereichs von elastischen Verzerrungen umfasst, und wobei das Auswählen des Satzes von abgetasteten Transformationsparametern das Auswählen eines Winkels innerhalb des Bereichs von Drehwinkeln, das Auswählen einer Translation innerhalb des Bereichs von Translationen, das Auswählen eines Skalierungsfaktors innerhalb des Bereichs von Skalierungsfaktoren und/oder das Auswählen einer elastischen Verzerrung innerhalb des Bereichs von elastischen Verzerrungen umfasst.

20. System gemäß Anspruch 15, wobei das Trainingsbild ein zweidimensionales oder dreidimensionales Bild einer biologischen Umgebung ist.

21. System gemäß Anspruch 15, ferner umfassend:
eine Faltungsschicht zum Modifizieren einer letzten neuronalen Netzwerksschicht, das eine Faltung einer letzten Schicht des einen oder mehrerer neuronaler Netzwerke und eine vorhergesagte Segmentierung bereitstellt; und
eine Softmax-Aktivierungsschicht zum Modifizieren der vorhergesagten Segmentierung gemäß einer Softmax-Operation auf die Faltung.

22. Verfahren, umfassend das Trainieren eines oder mehrerer neuronaler Netzwerken durch:
Erhalten eines ersten Trainingsbildes, einer ersten Segmentierungsmaske des ersten Trainingsbildes, eines zweiten Trainingsbildes und einer zweiten Segmentierungsmaske des zweiten Trainingsbildes;
Auswählen eines Satzes von abgetasteten Transformationsparametern, die aus einem Bereich von möglichen Transformationsparametern ausgewählt wurden;
Erzeugen, von dem Satz von abgetasteten Transformationsparametern, ein erstes Verschiebungsfeld, das Verschiebungen von Pixeln in dem ersten Trainingsbild angibt, die gemäß dem Satz von abgetasteten Transformationsparametern verschoben werden;
Erzeugen einer Korrespondenzbildes durch Transformieren des ersten Trainingsbildes gemäß dem ersten Verschiebungsfeld;
Erzeugen, unter Verwendung des einen oder mehrerer neuronaler Netzwerke, eines zweiten Verschiebungsfeldes, das eine erste versuchte Registrierung des ersten Trainingsbildes und des Korrespondenzbildes darstellt;
Erzeugen eines transformierten Bildes durch Transformieren des ersten Trainingsbildes gemäß dem zweiten Verschiebungsfeld;
Erzeugen einer Korrespondenz-Segmentierungsmaske durch Transformieren der ersten Segmentierungsmaske gemäß dem ersten Verschiebungsfeld;
Erzeugen einer transformierten Segmentierungsmaske durch Transformieren der ersten Segmentierungsmaske gemäß dem zweiten Verschiebungsfeld;
Erzeugen einer Verlustfunktion aus einer Funktion einer ersten Differenz zwischen der Korrespondenz-Segmentierungsmaske und der transformierten Segmentierungsmaske; und
Verwenden der Verlustfunktion, um das neuronale Netzwerk zu trainieren.

23. Verfahren gemäß Anspruch 22, wobei die Verlustfunktion eine gewichtete Summe der ersten Differenz und einer zweiten Differenz zwischen dem entsprechenden Bild und dem transformierten Bild ist.

24. Verfahren gemäß Anspruch 23, wobei die erste Differenz ein Mittelwert über einen Bildkoordinatenraum einer Norm von Differenzen von Elementen zwischen dem ersten Verschiebungsfeld und dem zweiten Verschiebungsfeld ist, und wobei die zweite Differenz eine negative normierte Kreuzkorrelation zwischen dem entsprechenden Bild und dem transformierten Bild ist.

25. Verfahren gemäß Anspruch 22, wobei der Bereich von möglichen Transformationsparametern einen oder mehrere eines Bereichs von Drehwinkeln, eines Bereichs von Translationen, eines Bereichs von Skalierungsfaktoren und eines Bereichs von elastischen Verzerrungen umfasst, und wobei das Auswählen des Satzes von abgetasteten Transformationsparametern das Auswählen eines Winkels innerhalb des Bereichs von Drehwinkeln, das Auswählen einer Translation innerhalb des Bereichs von Translationen, das Auswählen eines Skalierungsfaktors innerhalb des Bereichs von Skalierungsfaktoren und/oder das Auswählen einer elastischen Verzerrung innerhalb des Bereichs von elastischen Verzerrungen, wobei der Bereich von möglichen Transformationsparametern möglichen Abweichungen zwischen mehreren Bildern einer Quelle von Bildern entspricht.

26. Verfahren gemäß Anspruch 25, wobei die Abweichungen zwischen mehreren Bildern von Bildern der Quelle von Bildern Abweichungen zwischen Bildern einer biologischen Umgebung entsprechen.

27. Verfahren gemäß Anspruch 22, ferner umfassend:
Erzeugen, unter Verwendung des einen oder mehrerer neuronaler Netzwerke, eines dritten Verschiebungsfeldes, das eine zweite versuchte Registrierung des ersten Trainingsbildes und des zweiten Trainingsbildes darstellt;
Erzeugen einer dritten Segmentierungsmaske durch Transformieren der ersten Segmentierungsmaske gemäß dem ersten Verschiebungsfeld;

Erzeugen einer vierten Segmentierungsmaske durch Transformieren der ersten Segmentierungsmaske gemäß dem zweiten Verschiebungsfeld;
Erzeugen einer fünften Segmentierungsmaske durch Transformieren der ersten Segmentierungsmaske gemäß dem dritten Verschiebungsfeld;
Bestimmen einer Segmentierungs-Verlustfunktion, die eine erste Funktion einer ersten Differenzfunktion von Differenzen zwischen der dritten Segmentierungsmaske und der vierten Segmentierungsmaske und/oder einer zweiten Differenzfunktion von Differenzen zwischen der fünften Segmentierungsmaske und der zweiten Segmentierungsmaske ist;
Trainieren des einen oder mehrerer neuronaler Netzwerke durch Optimieren der Segmentierungs-Verlustfunktion.

28. Verfahren gemäß Anspruch 27, ferner umfassend:

Erzeugen eines ersten transformierten Bildes durch Transformieren des ersten Trainingsbildes gemäß dem zweiten Verschiebungsfeld;
Erzeugen eines zweiten transformierten Bildes durch Transformieren des ersten Trainingsbildes gemäß dem dritten Verschiebungsfeld; und
Trainieren des einen oder mehrerer neuronaler Netzwerke durch Optimieren der Verlustfunktion, wobei die Verlustfunktion die Segmentierungs-Verlustfunktion, eine dritte Differenzfunktion von Differenzen zwischen der dritten Segmentierungsmaske und der vierten Segmentierungsmaske, einer vierten Differenzfunktion von Differenzen zwischen der fünften Segmentierungsmaske und der zweiten Segmentierungsmaske und eine fünfte Differenzfunktion von Differenzen zwischen dem ersten Verschiebungsfeld und dem zweiten Verschiebungsfeld umfasst.

29. Verfahren gemäß Anspruch 28, wobei die Verlustfunktion eine gewichtete Summe der ersten Differenzfunktion, der zweiten Differenzfunktion, der dritten Differenzfunktion, der vierten Differenzfunktion und der fünften Differenzfunktion ist.

Es folgen 49 Seiten Zeichnungen

Anhängende Zeichnungen

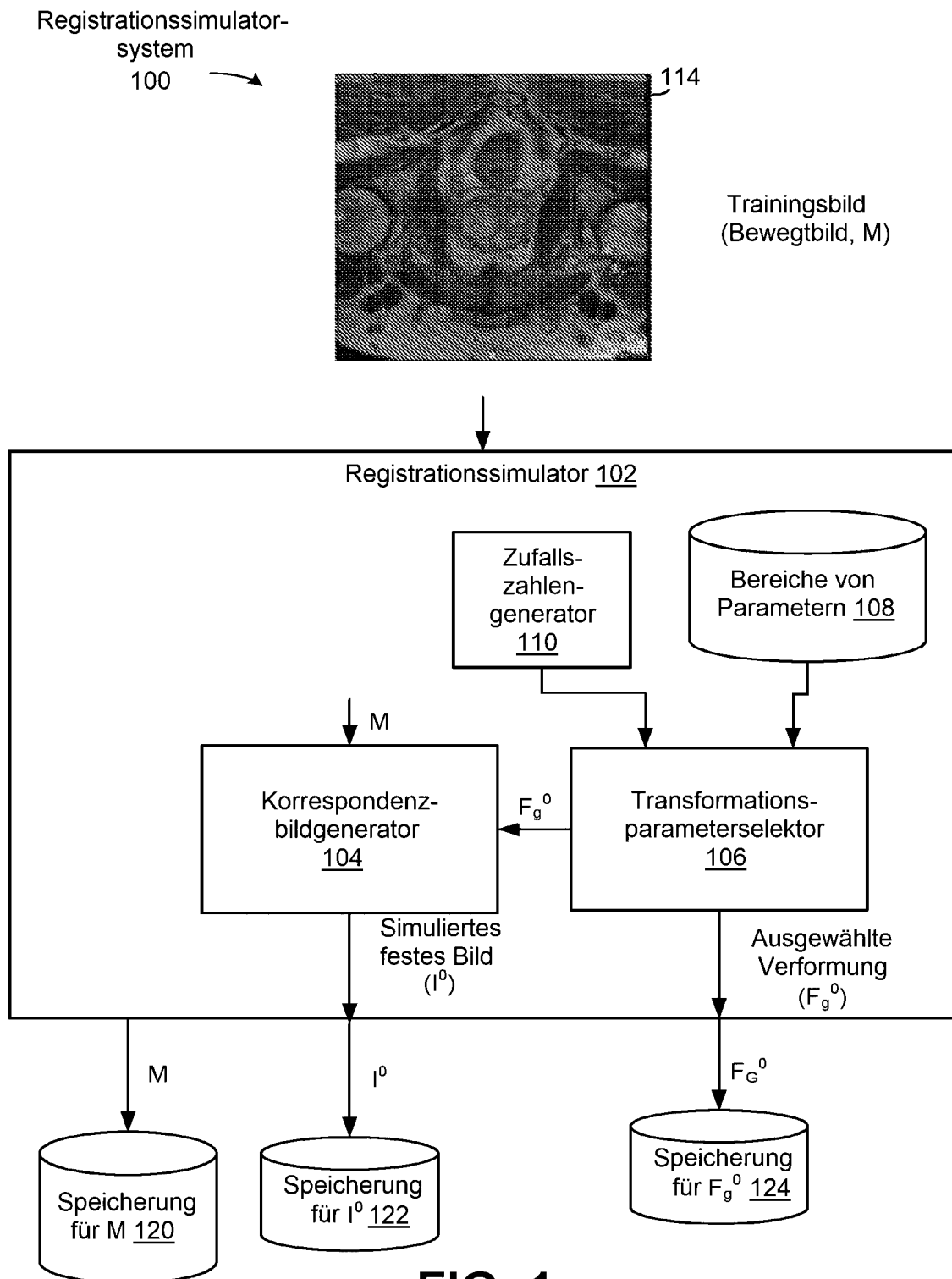


FIG. 1

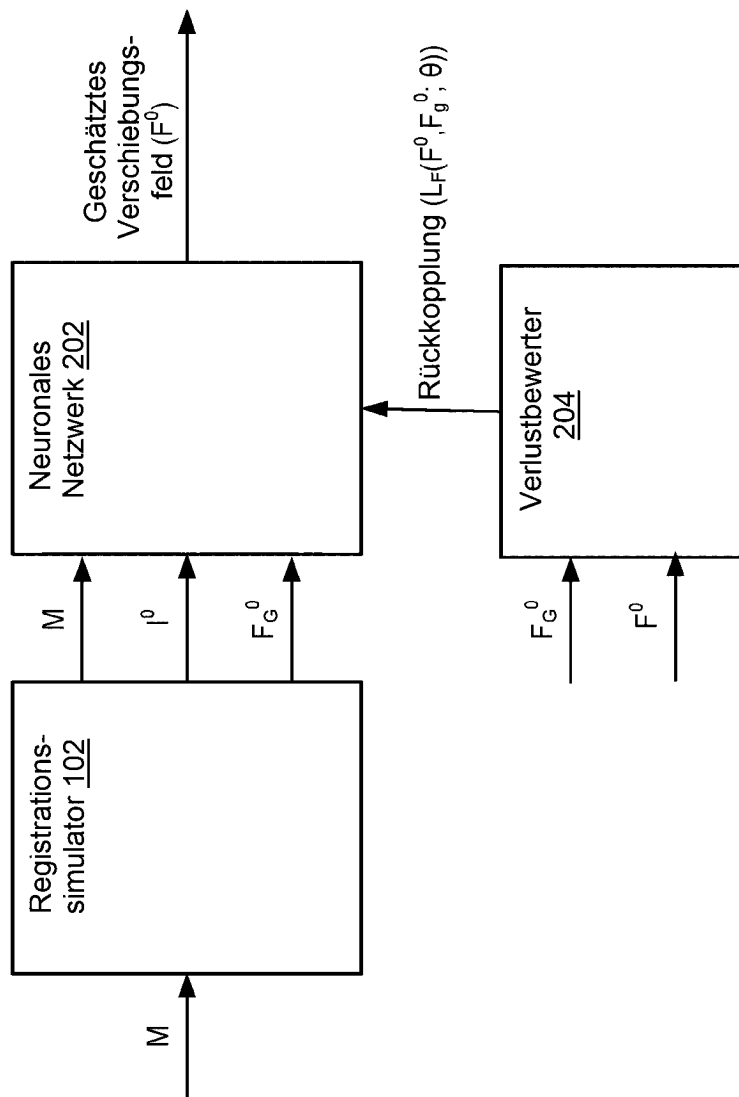


FIG. 2

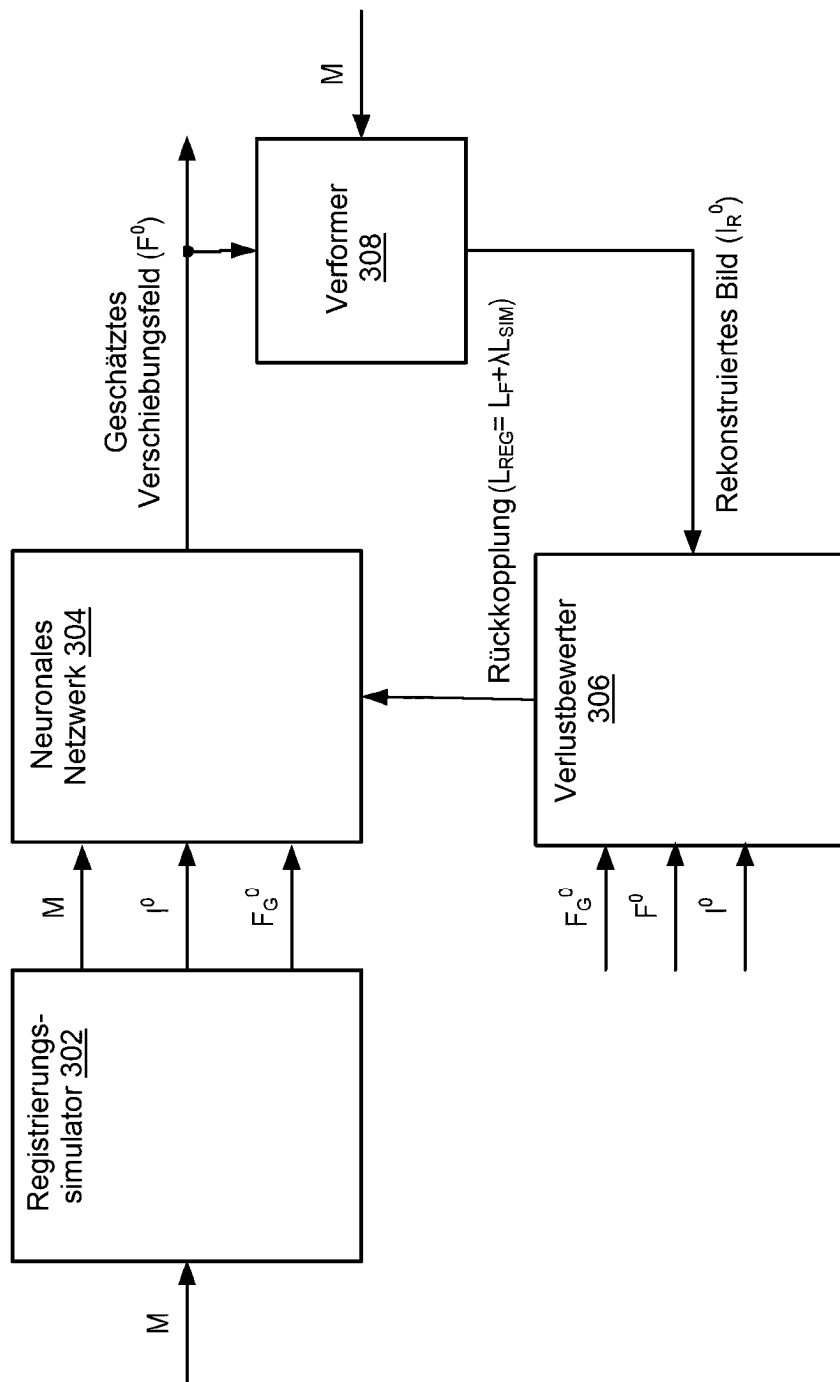


FIG. 3

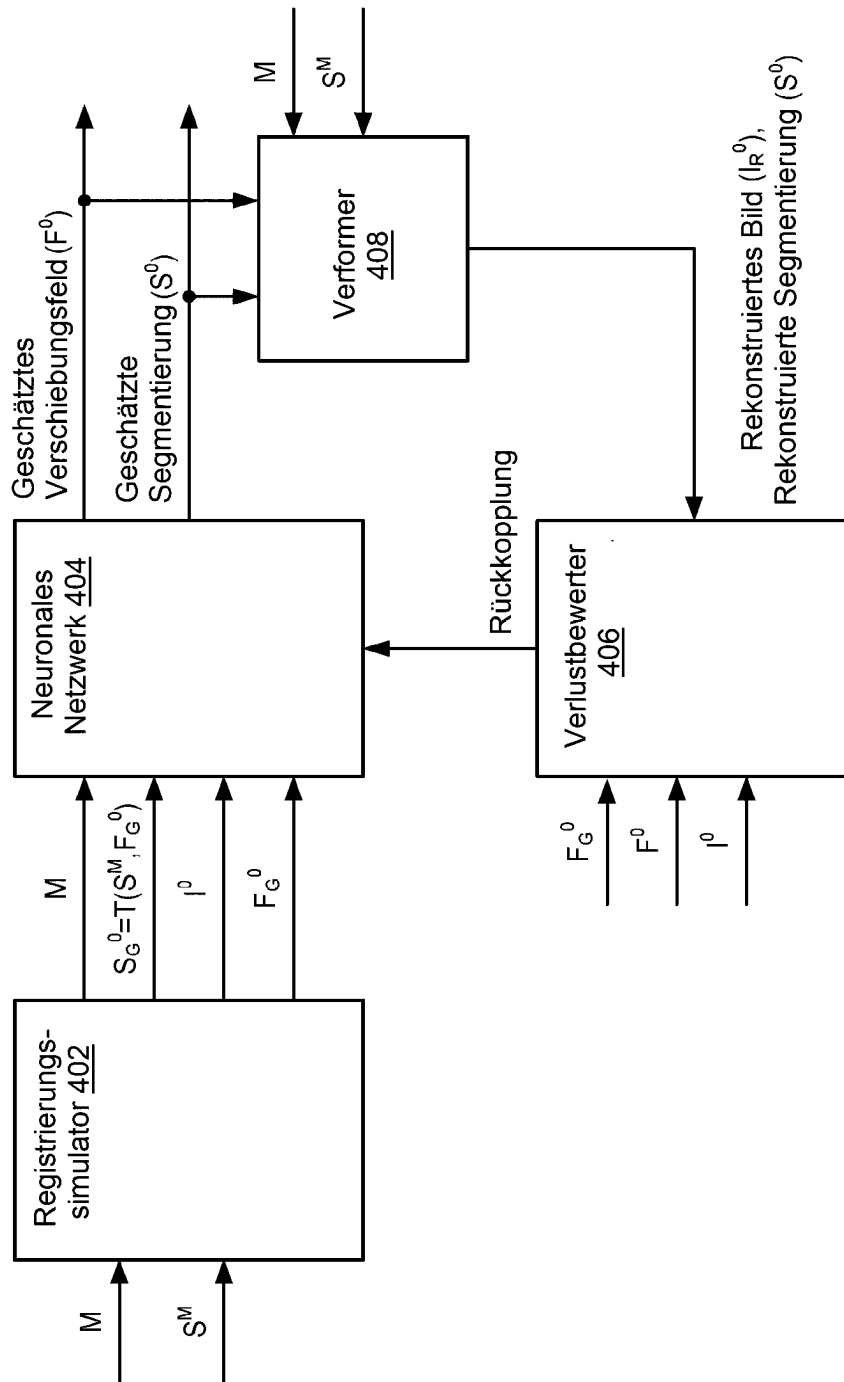


FIG. 4

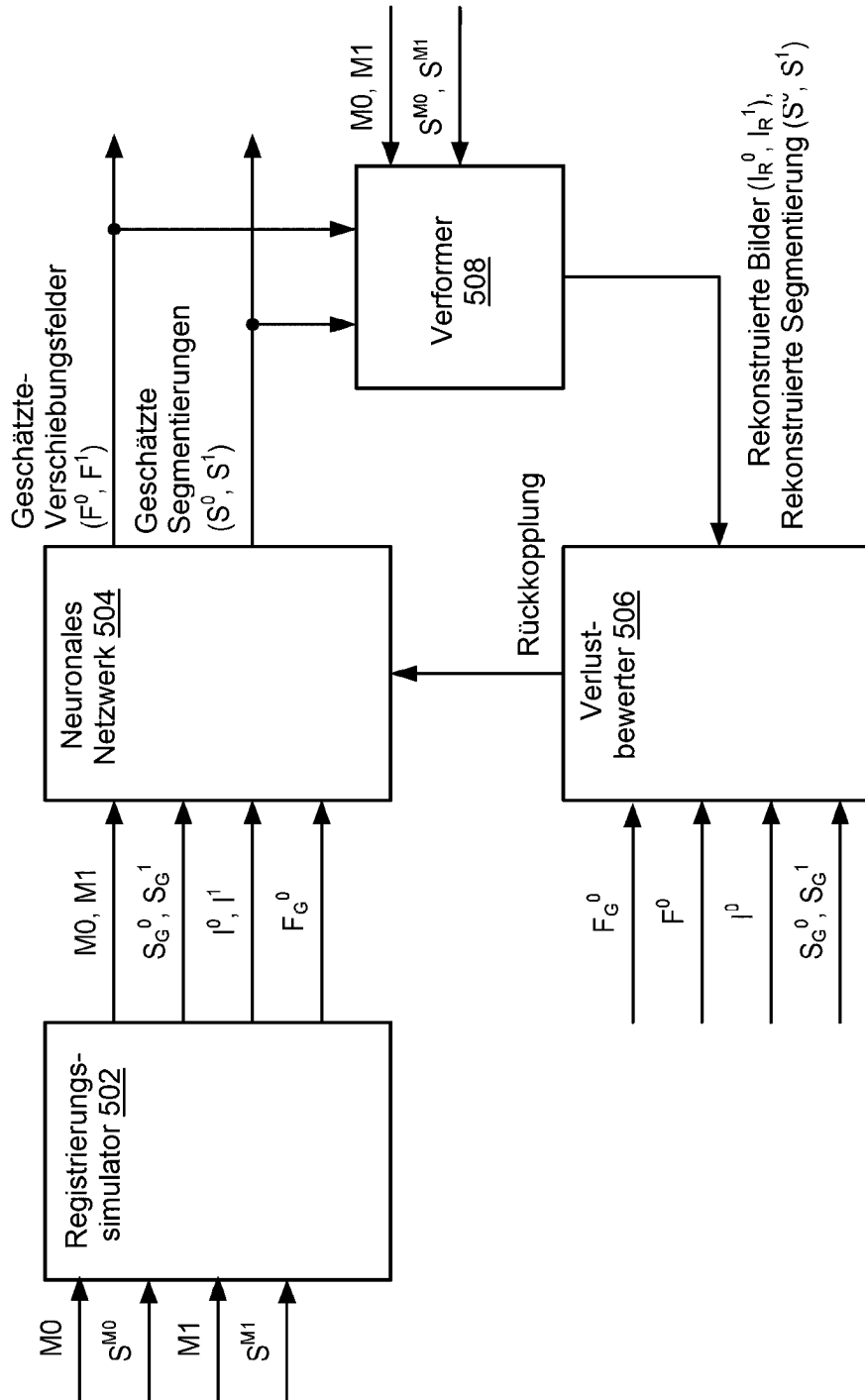


FIG. 5

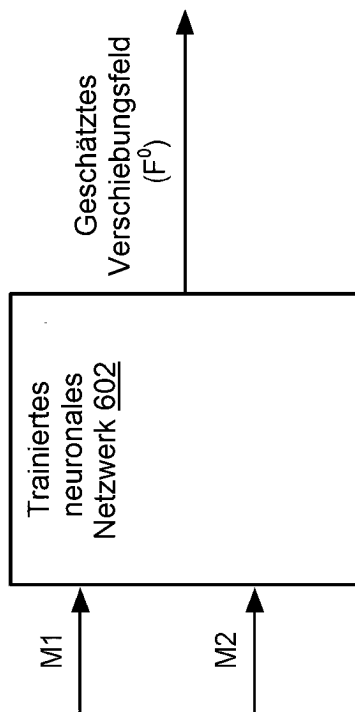


FIG. 6

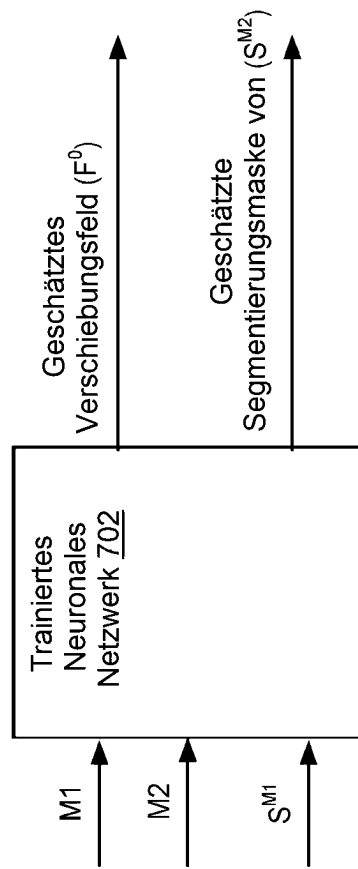
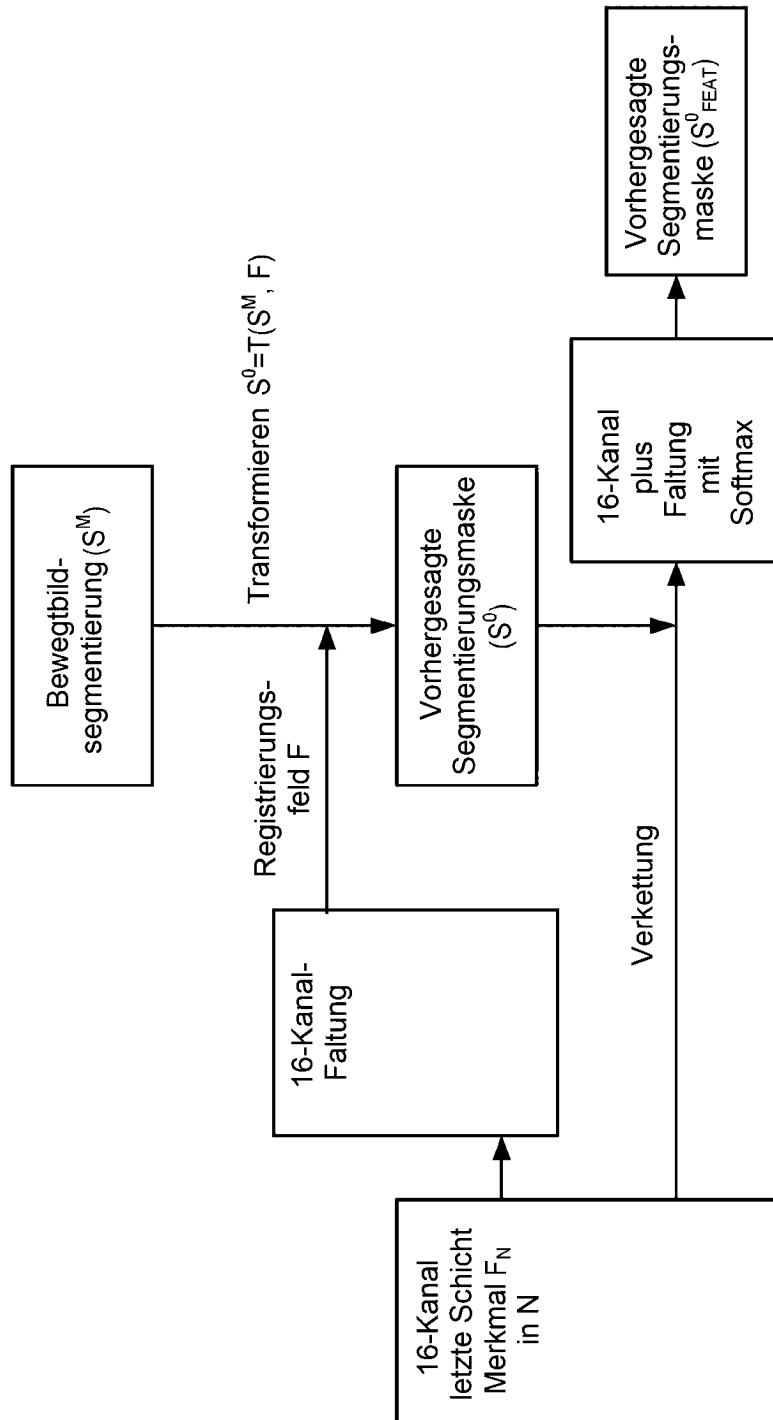


FIG. 7

**FIG. 8**

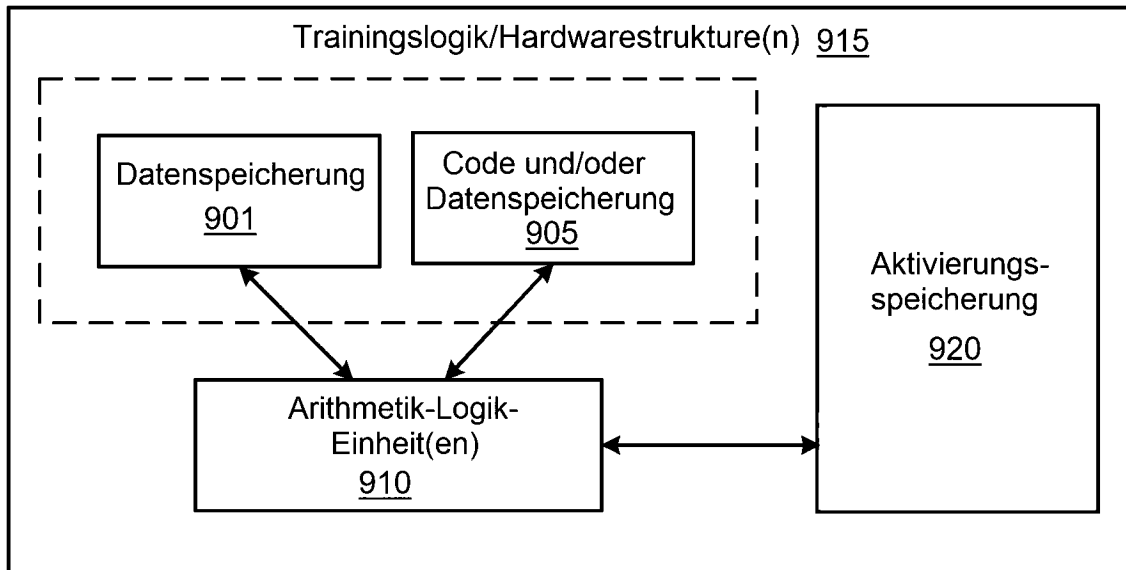


FIG. 9A

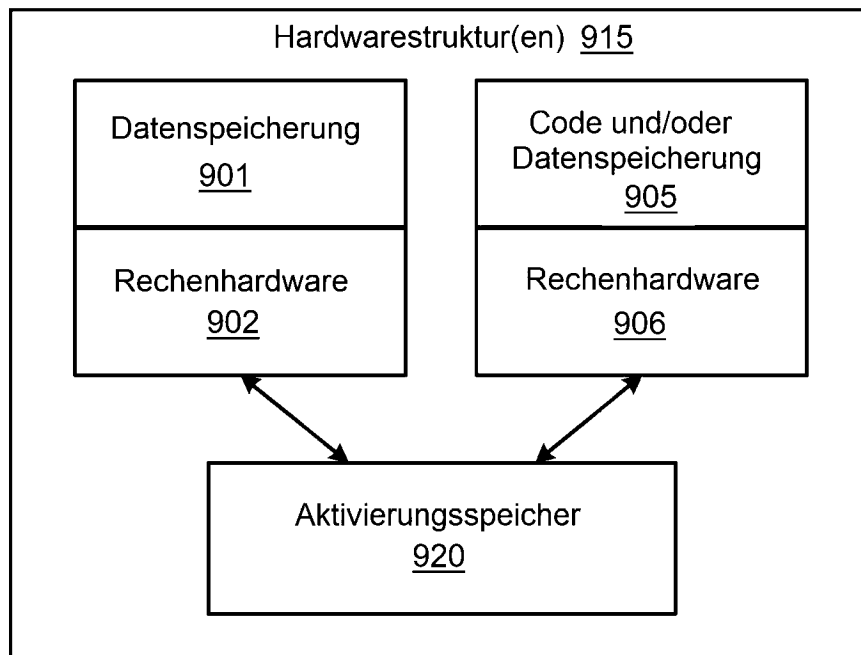


FIG. 9B

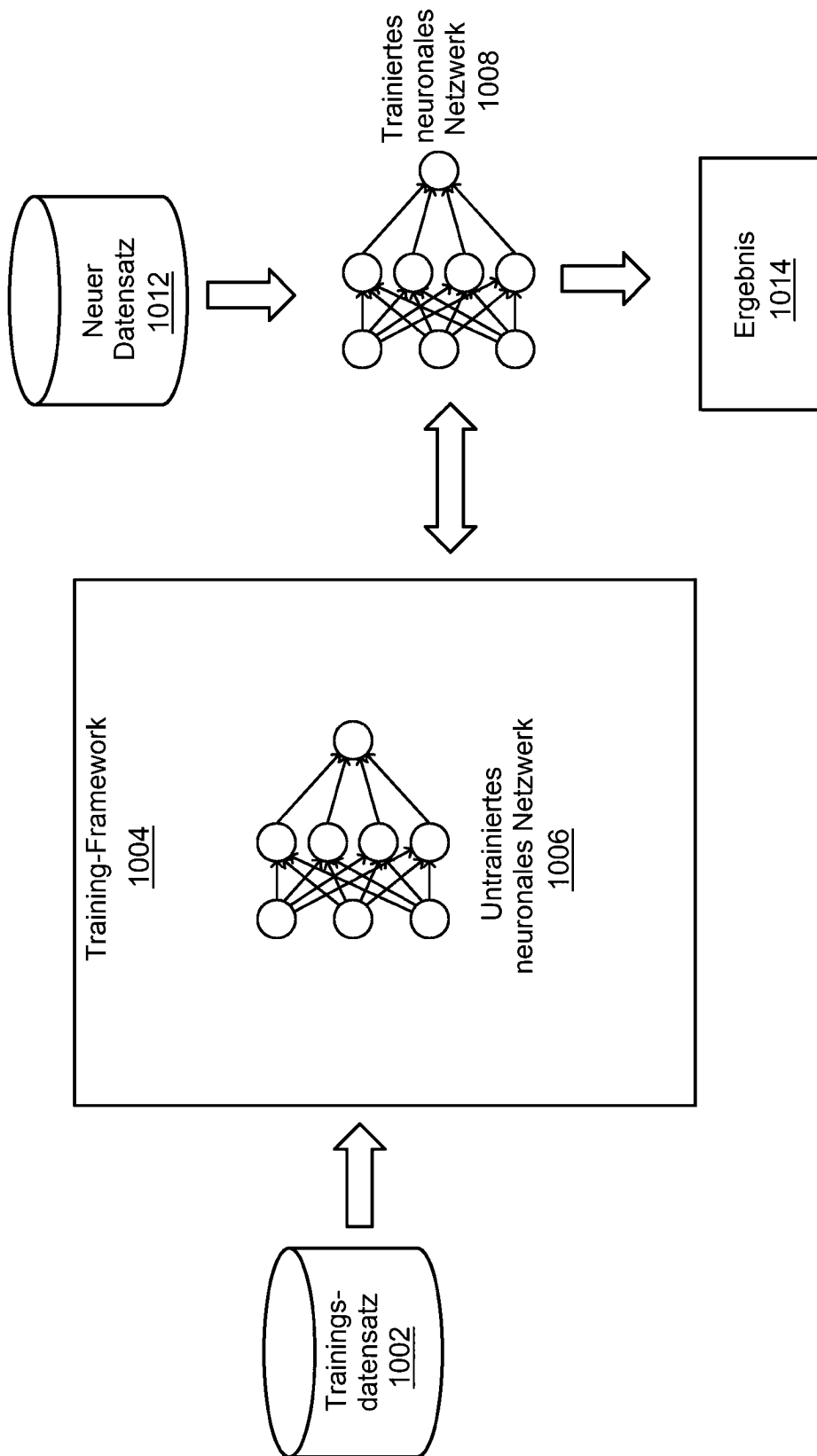


FIG. 10

Rechenzentrum
1100

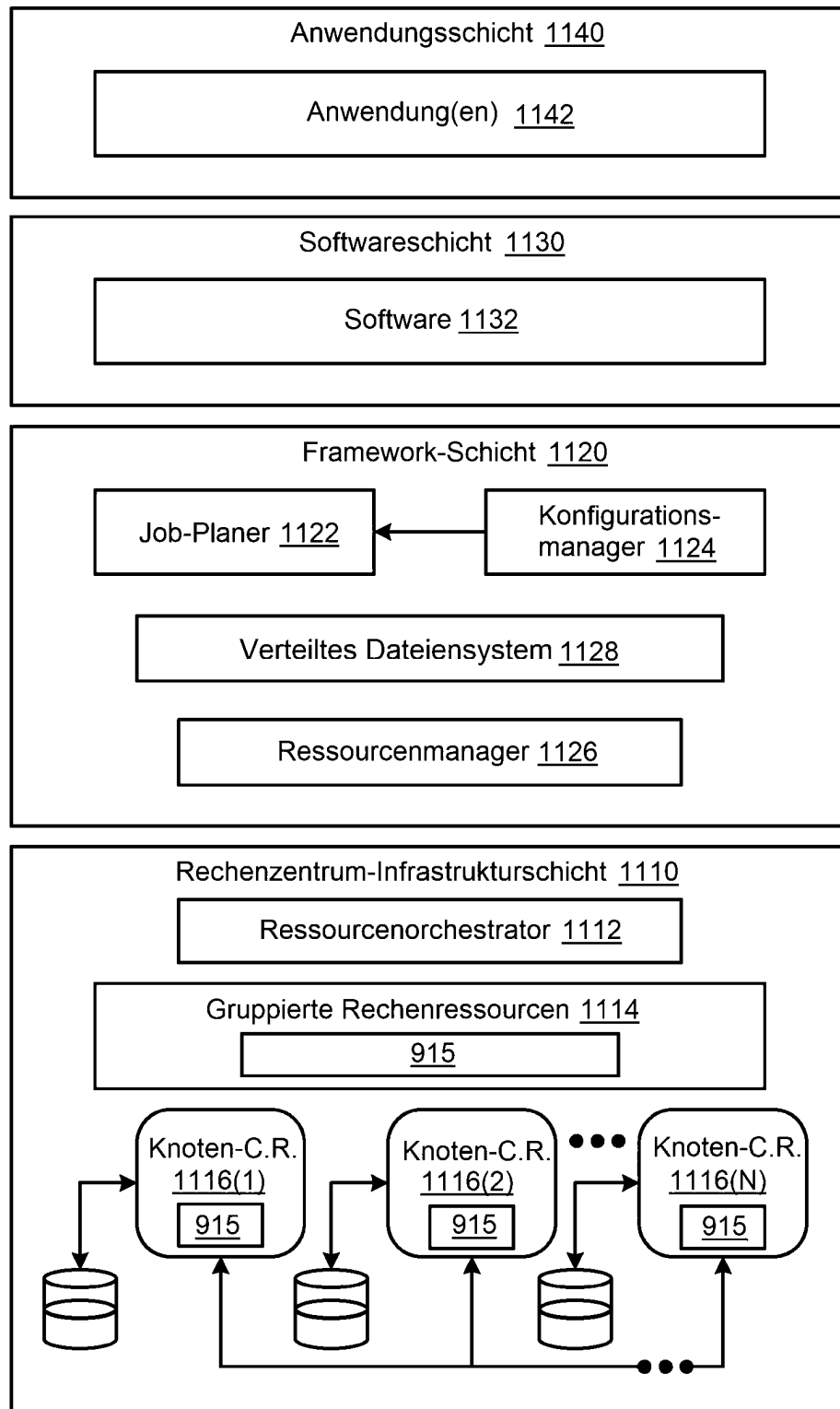


FIG. 11

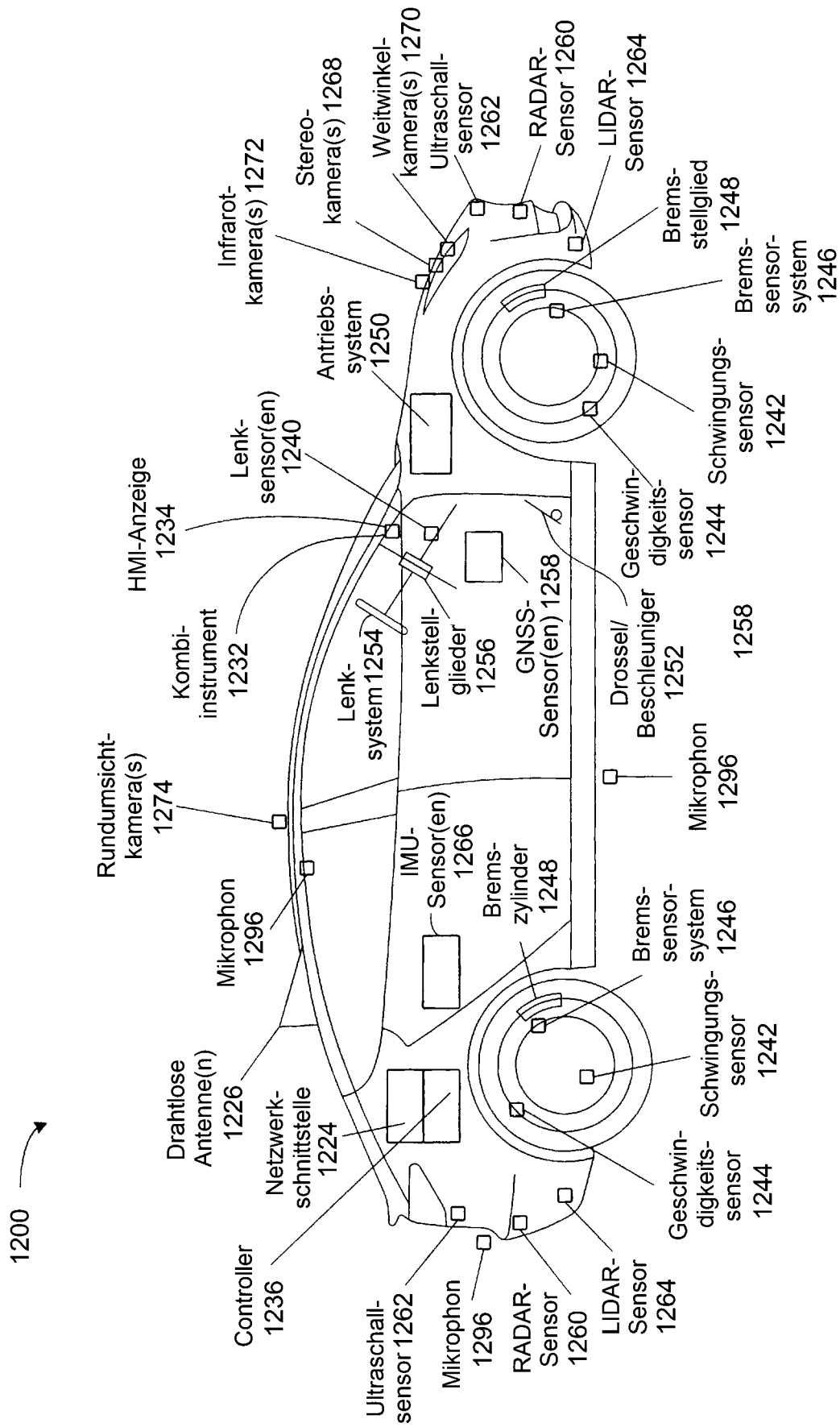


FIG. 12A

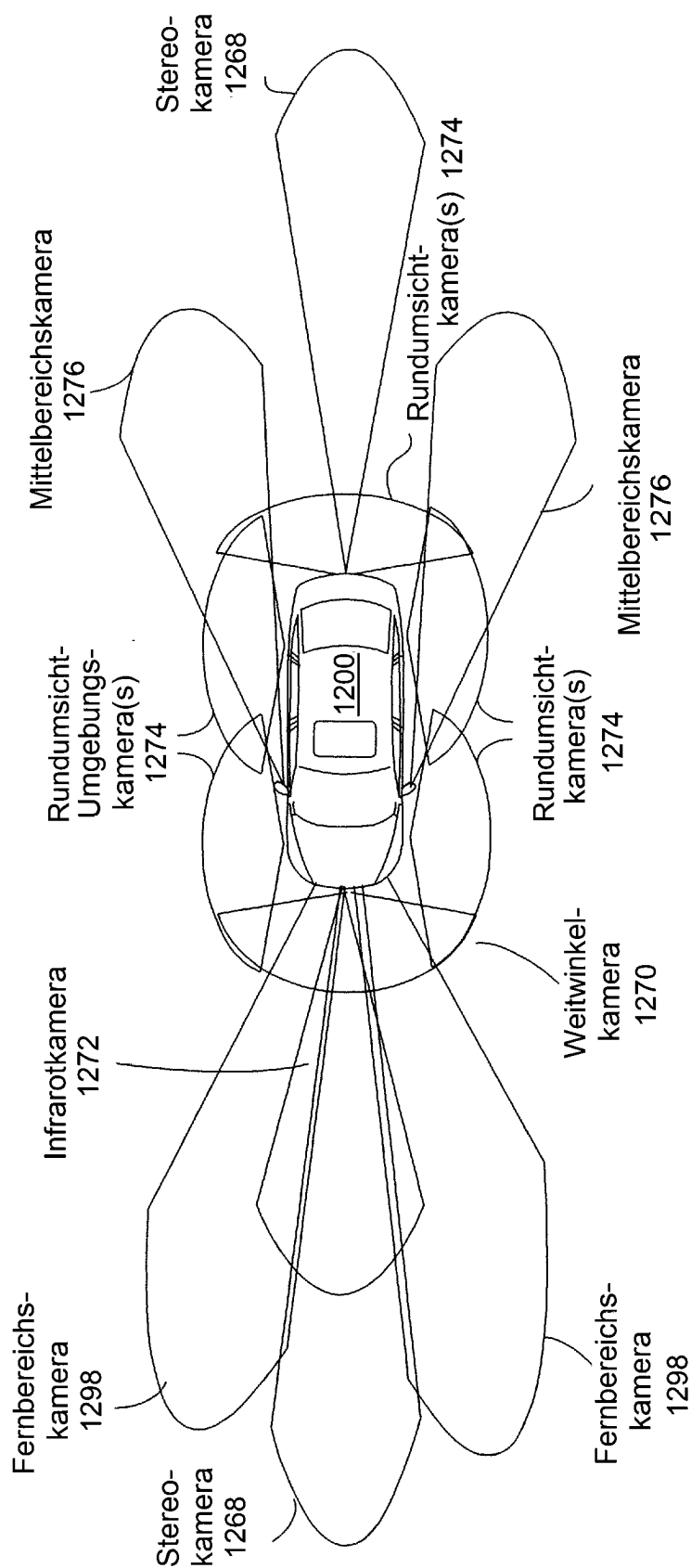


FIG. 12B

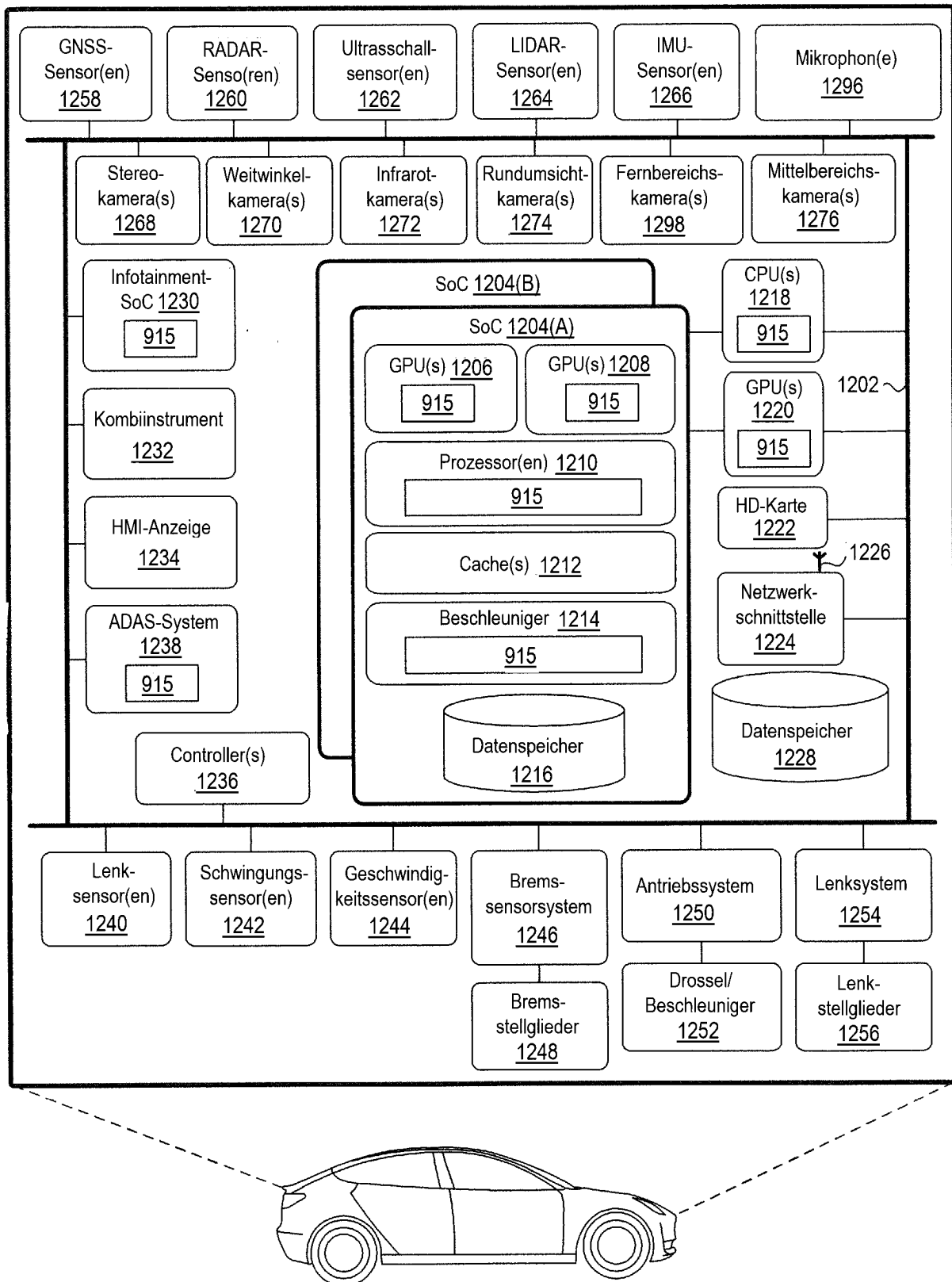


FIG. 12C

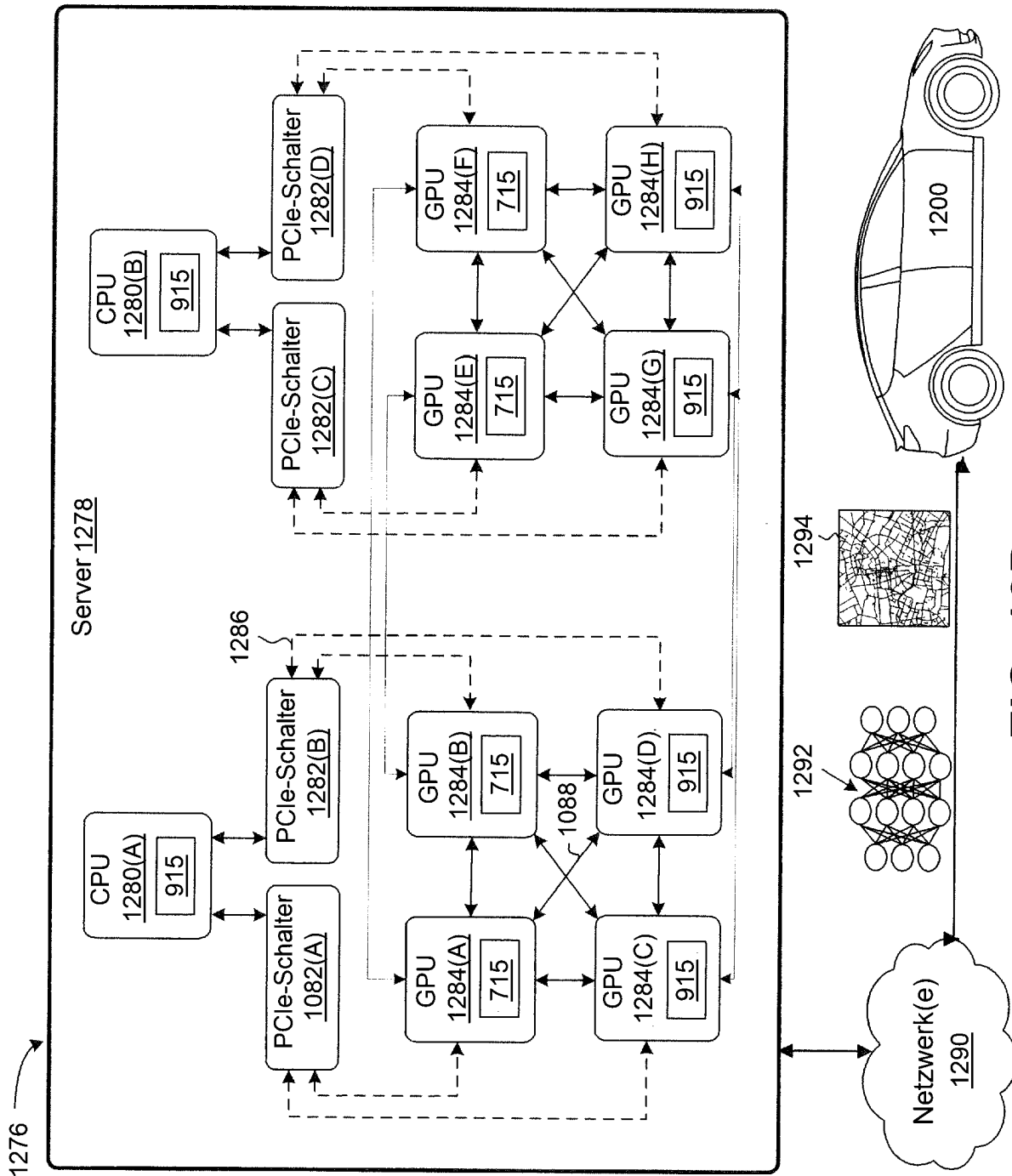


FIG. 12D

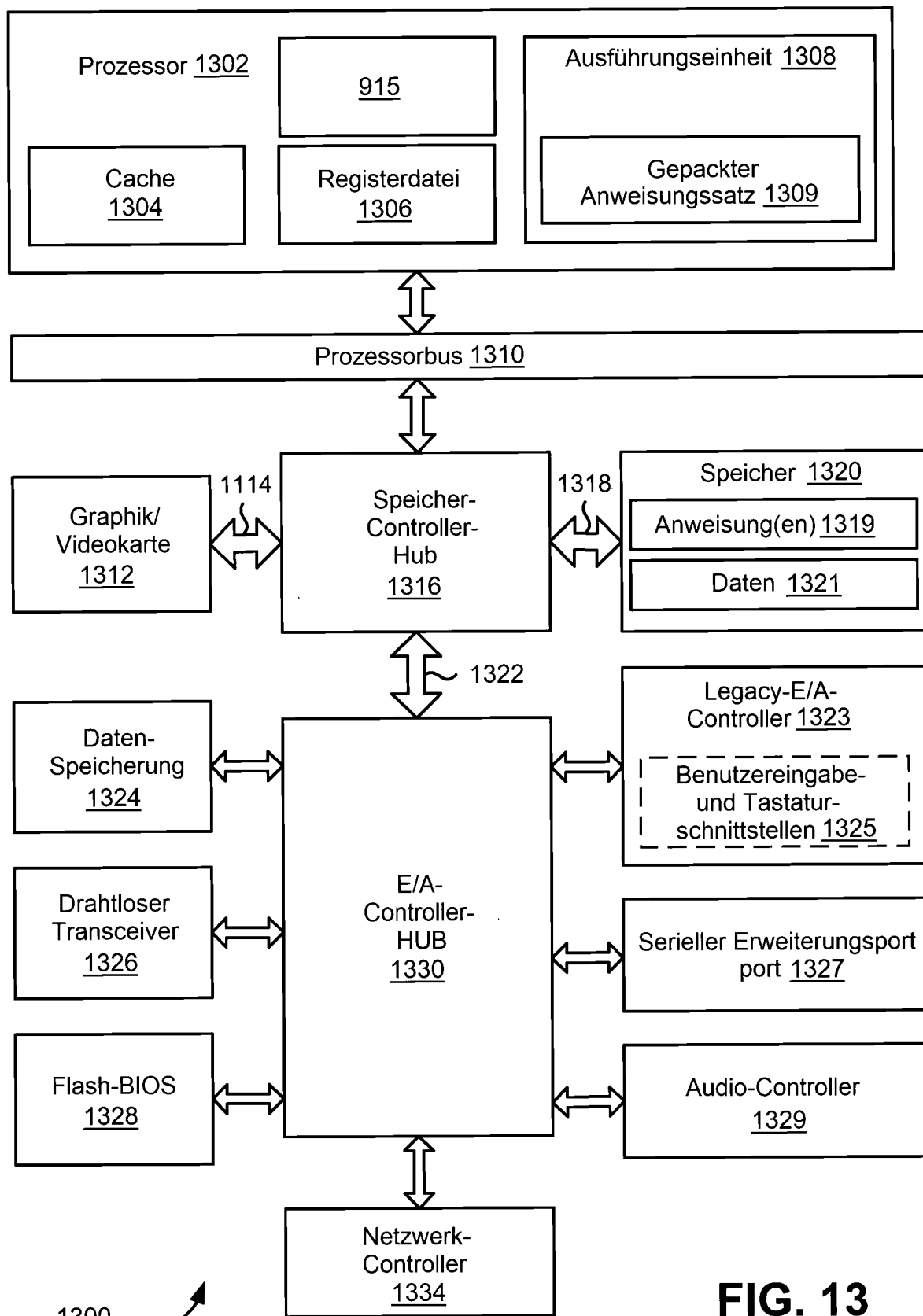
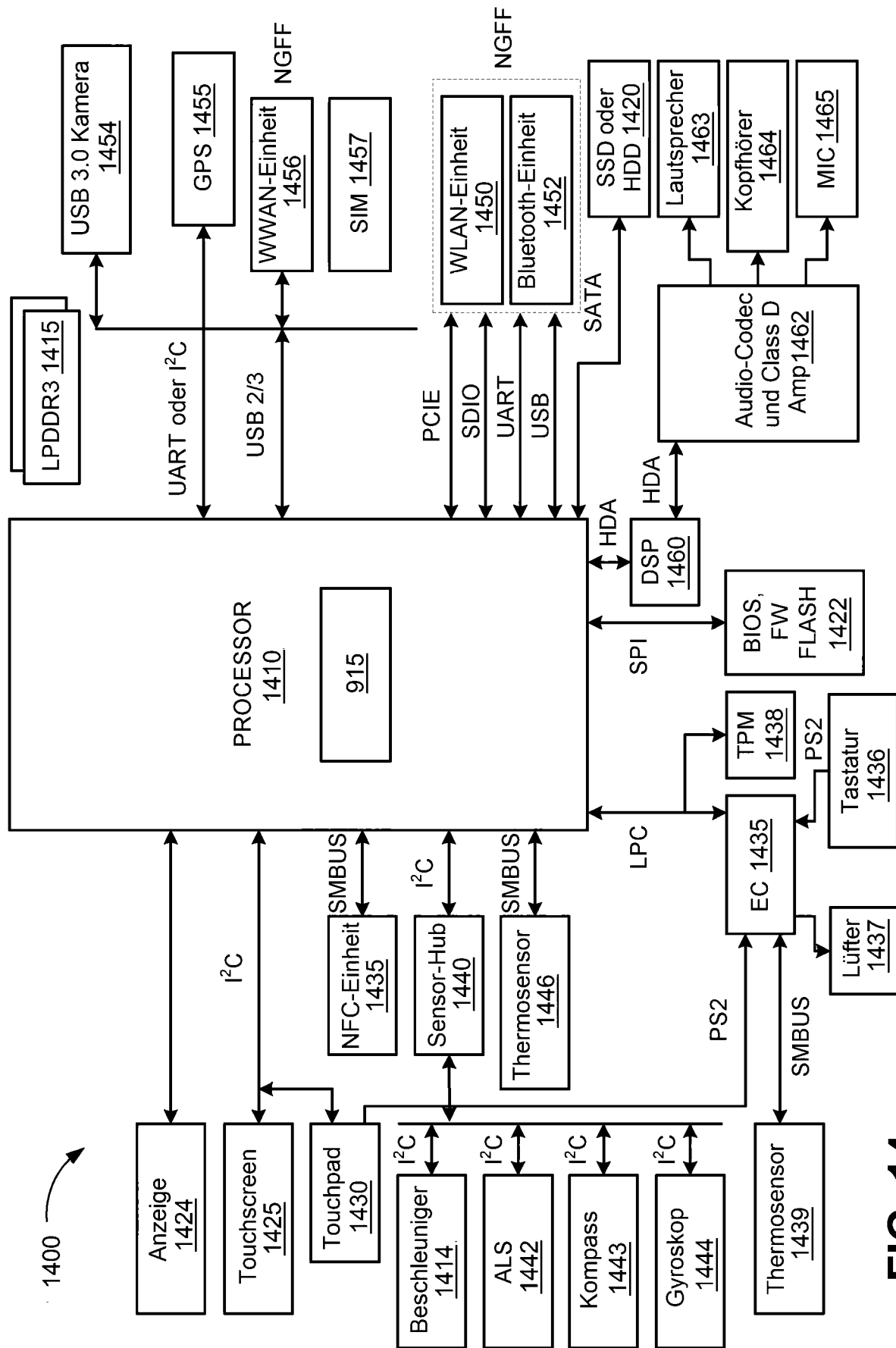
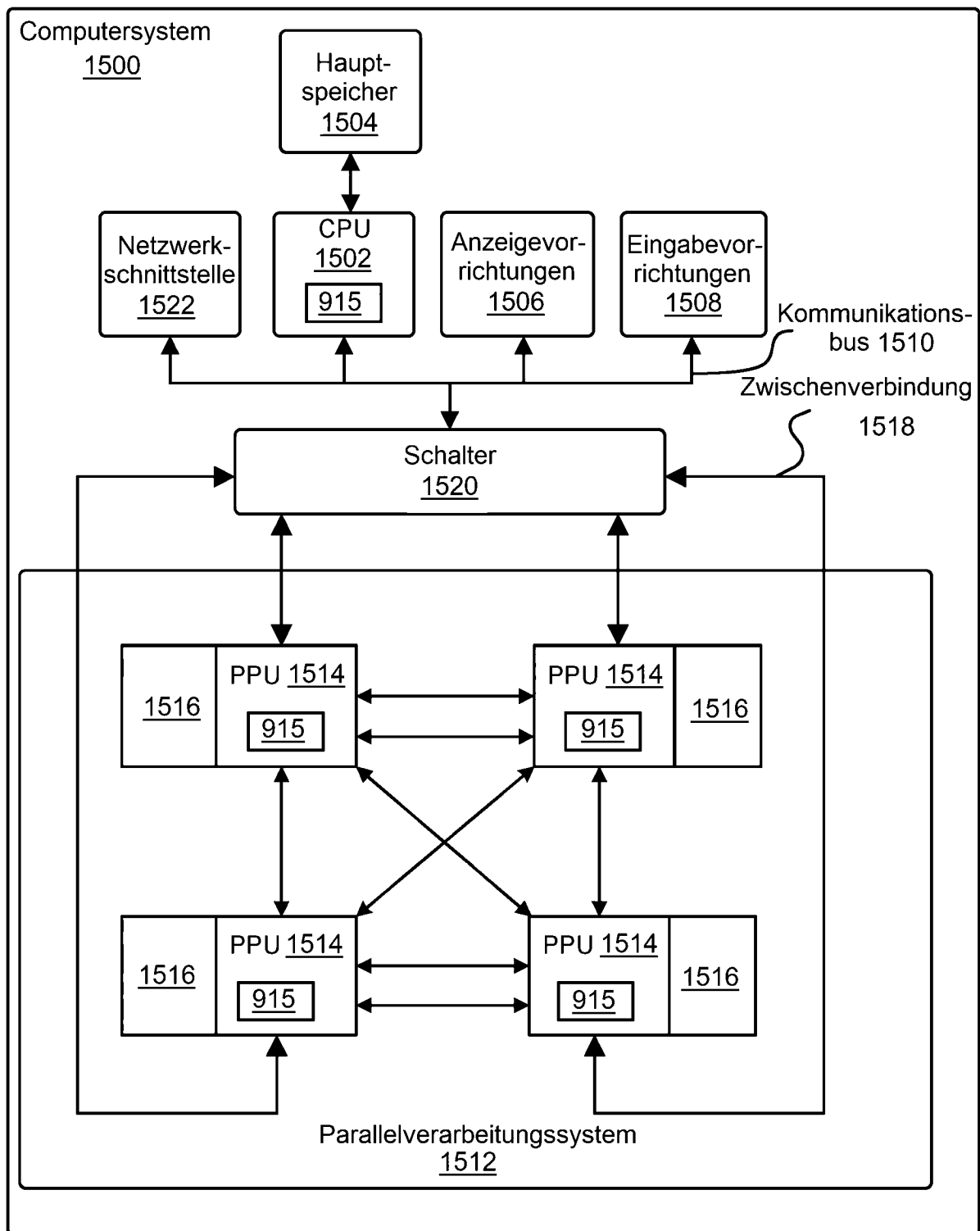


FIG. 13

**FIG. 14**

**FIG. 15**

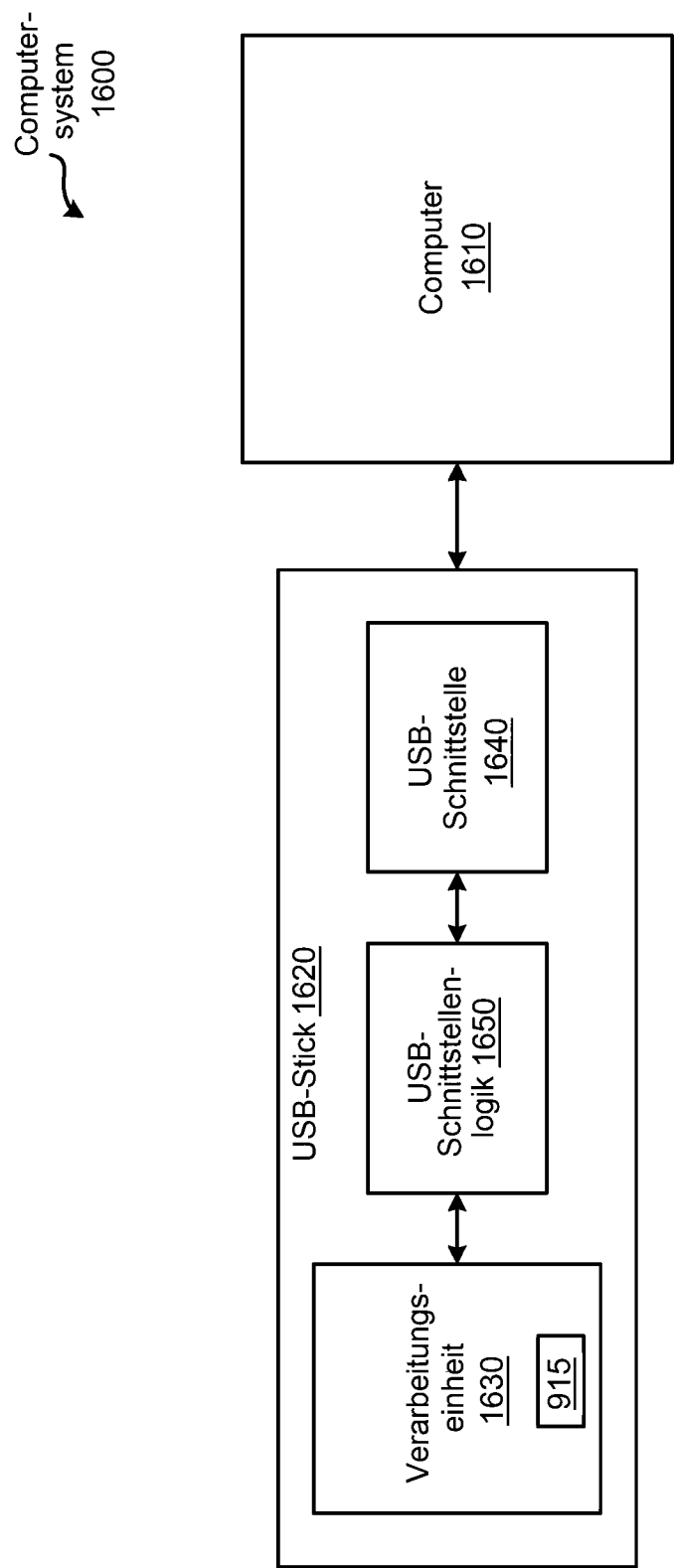


FIG. 16

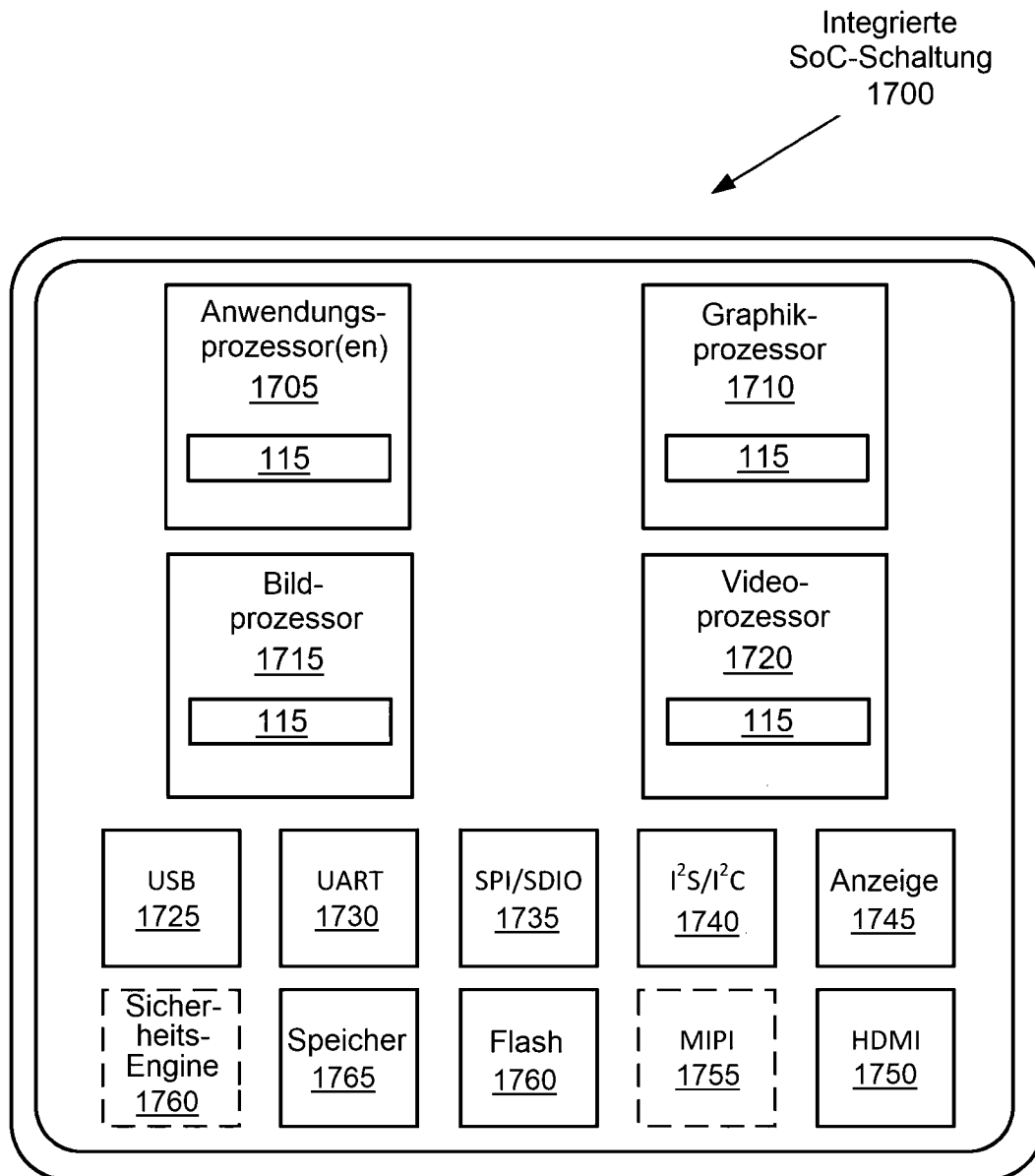


FIG. 17

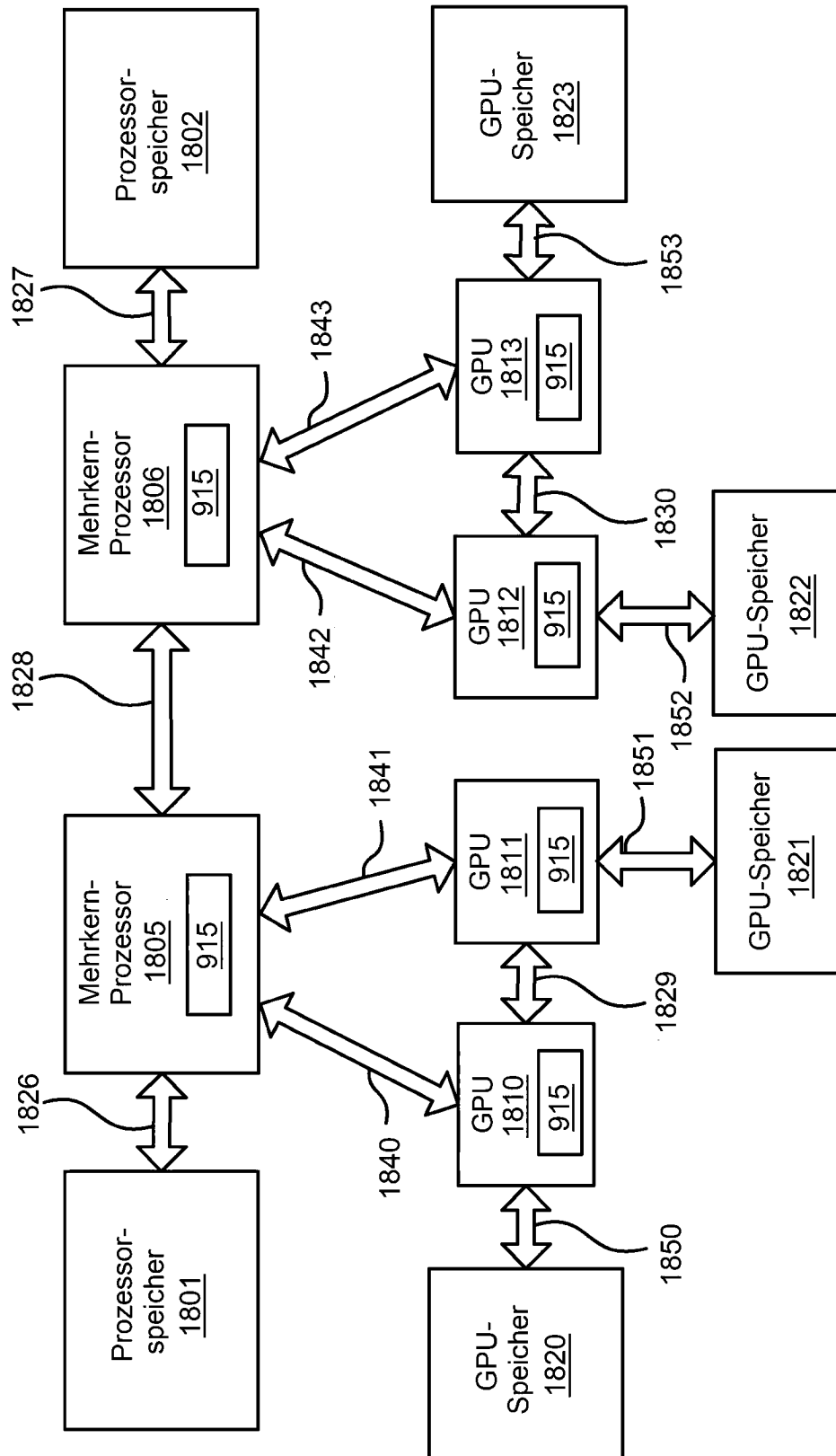
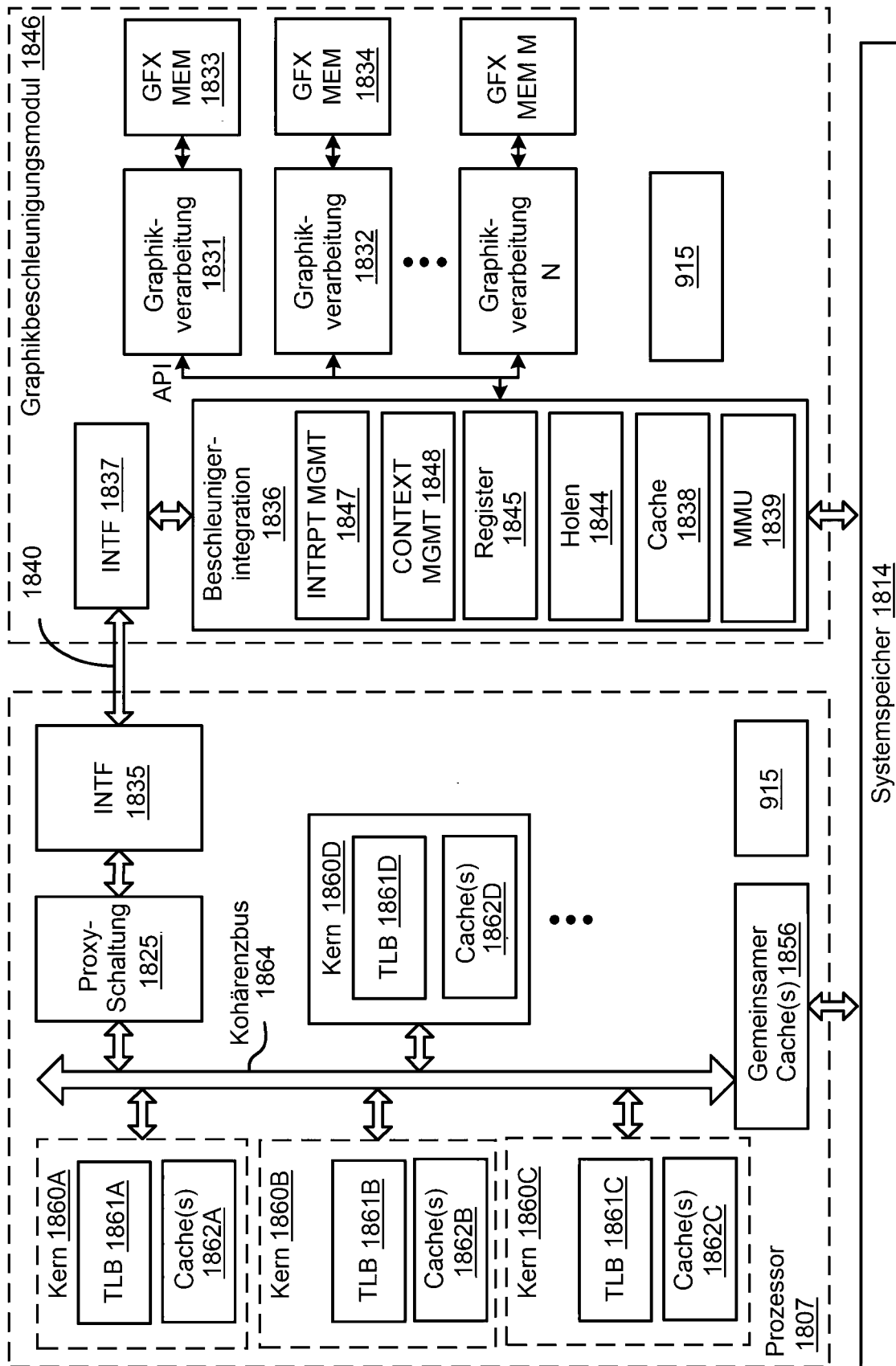
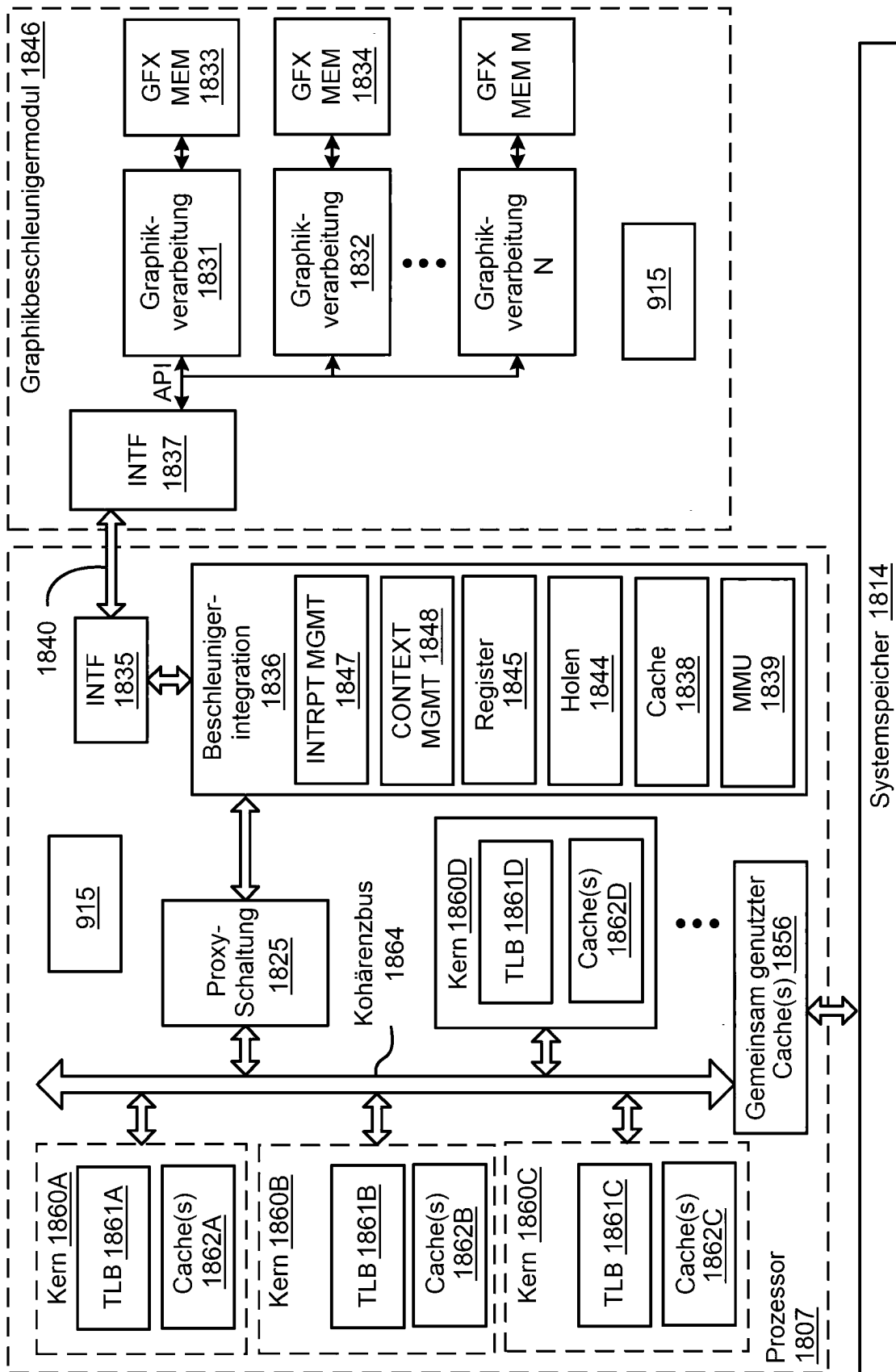
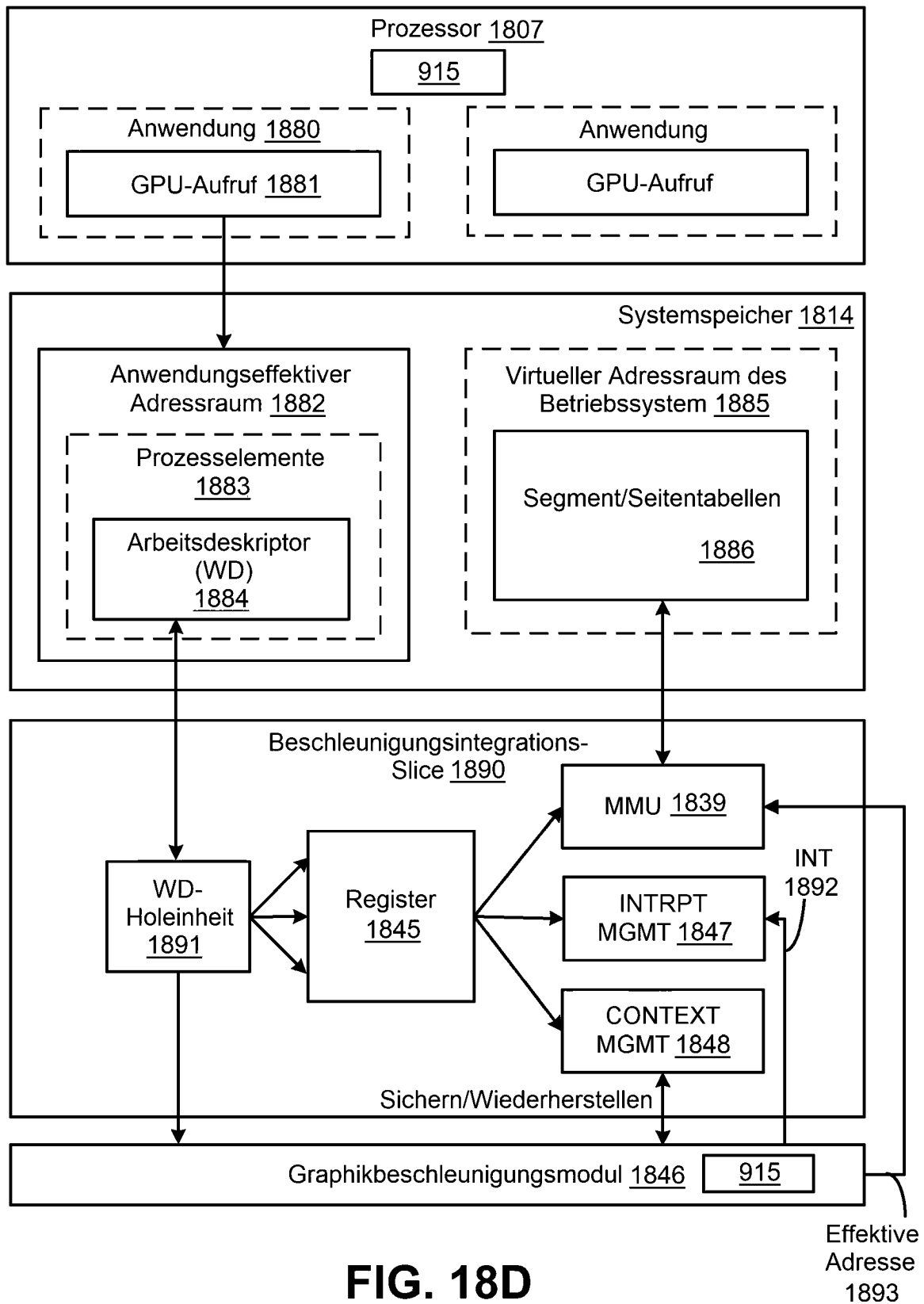
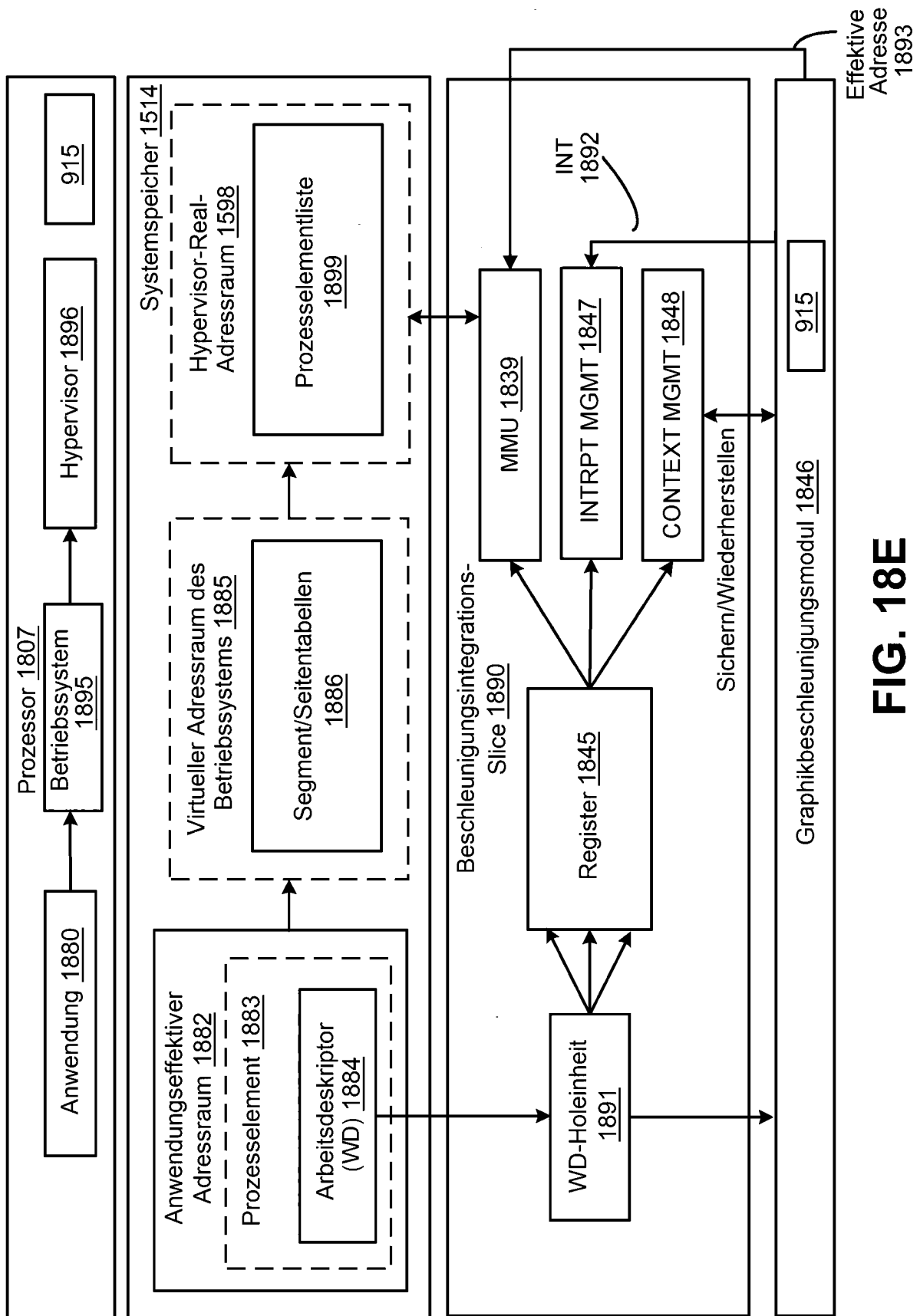


FIG. 18A

**FIG. 18B**

**FIG. 18C**

**FIG. 18D**

**FIG. 18E**

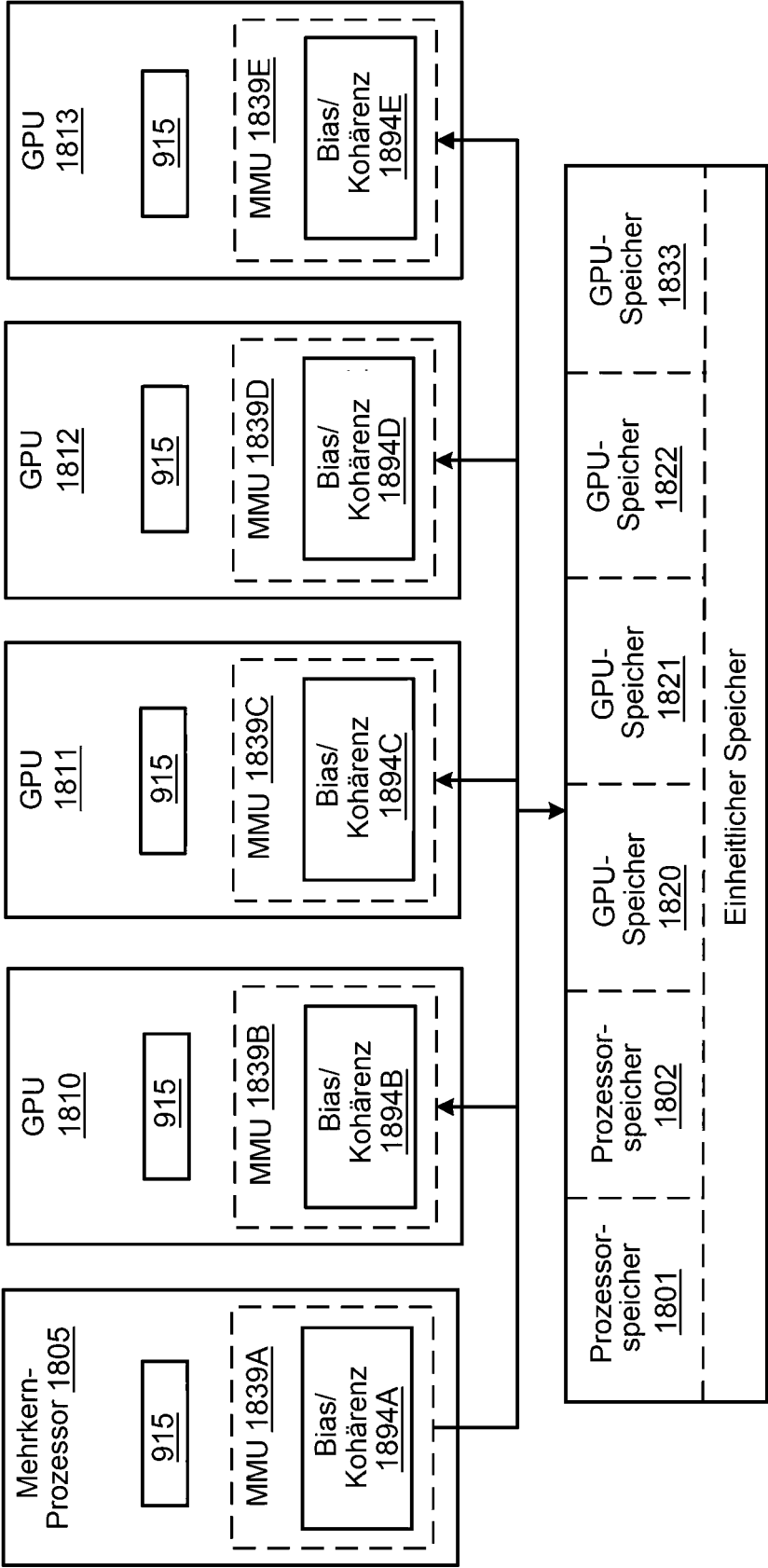


FIG. 18F

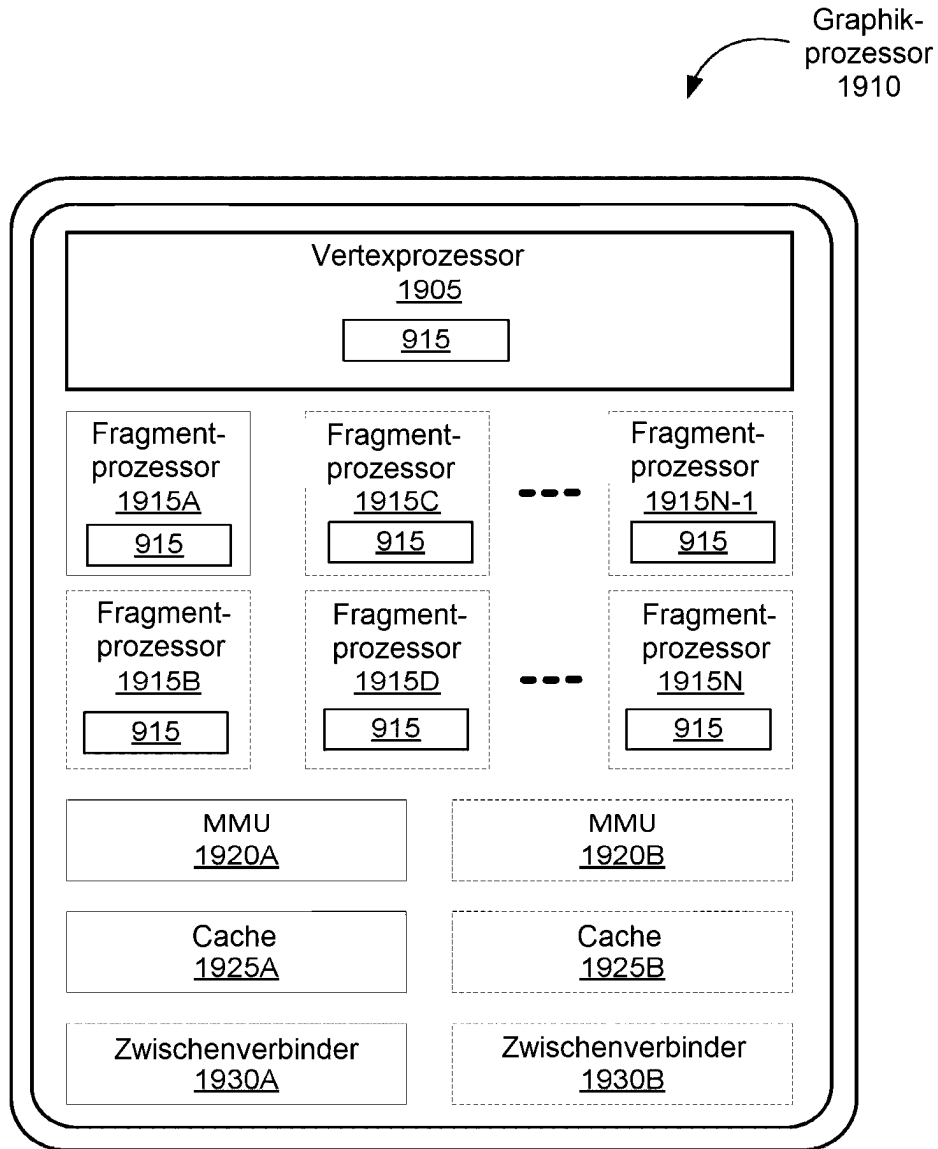


FIG. 19A

Graphik-
Prozessor
1940

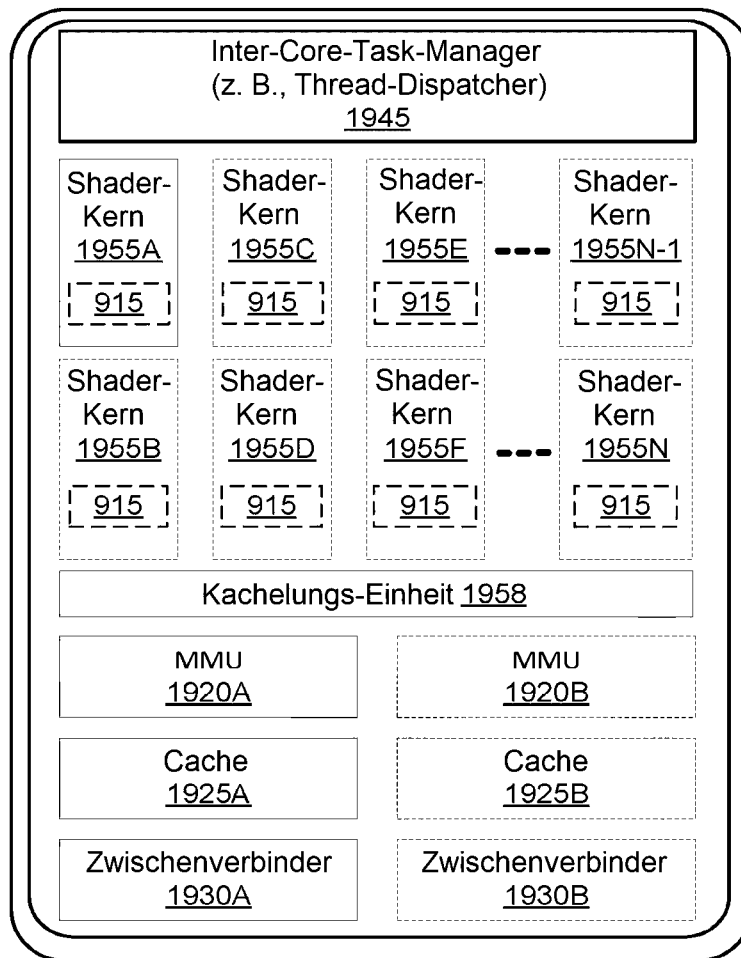
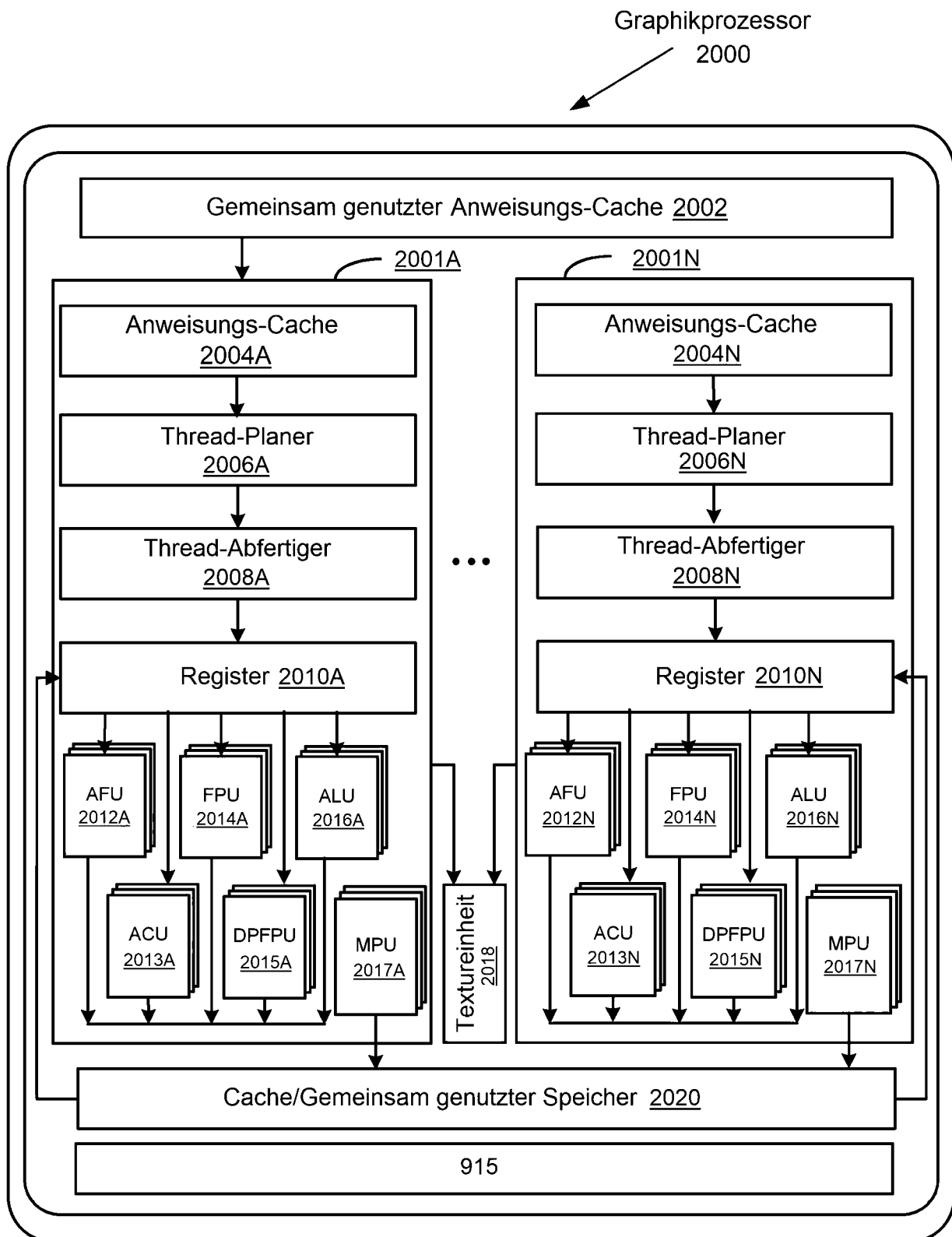


FIG. 19B



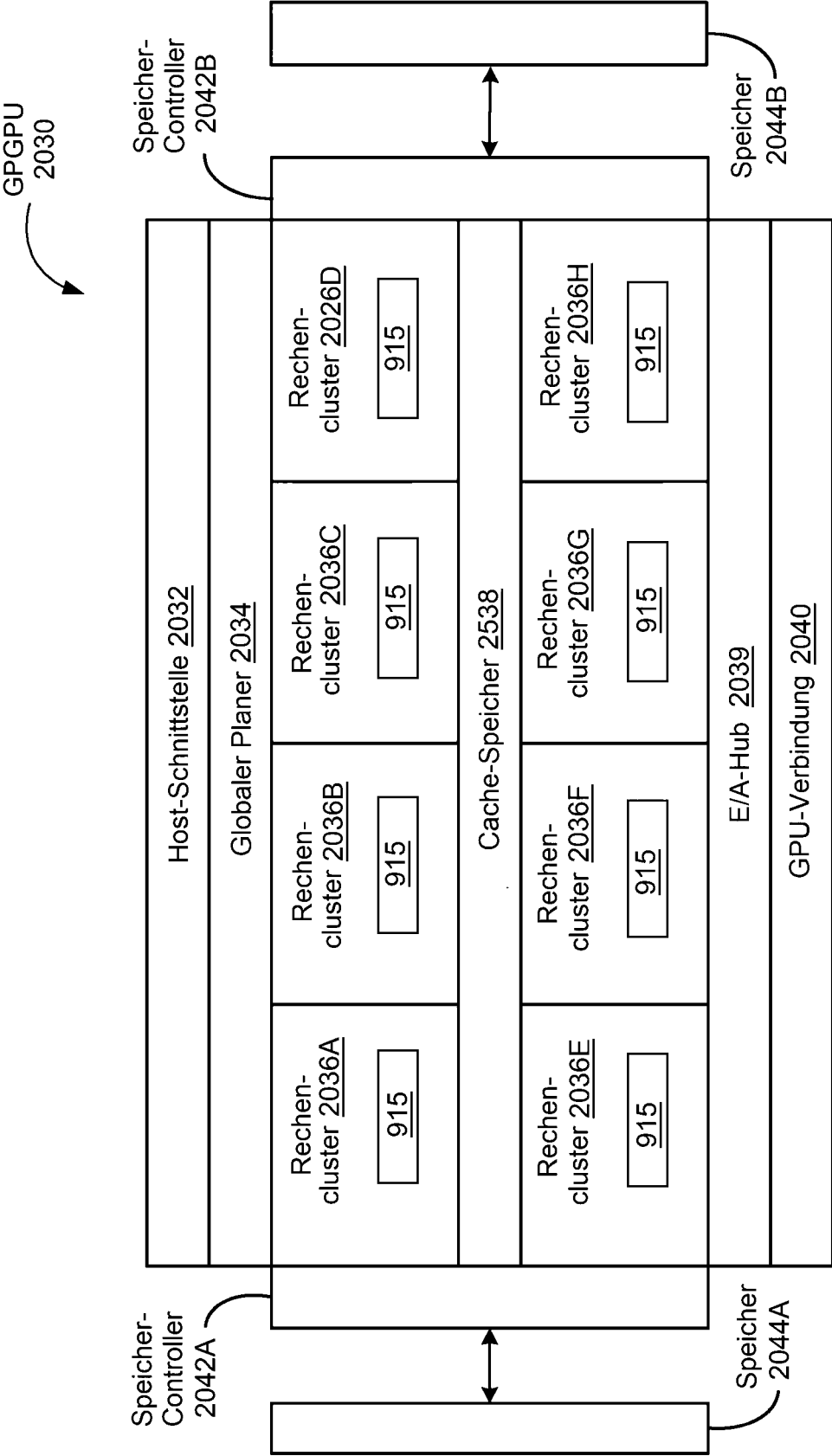
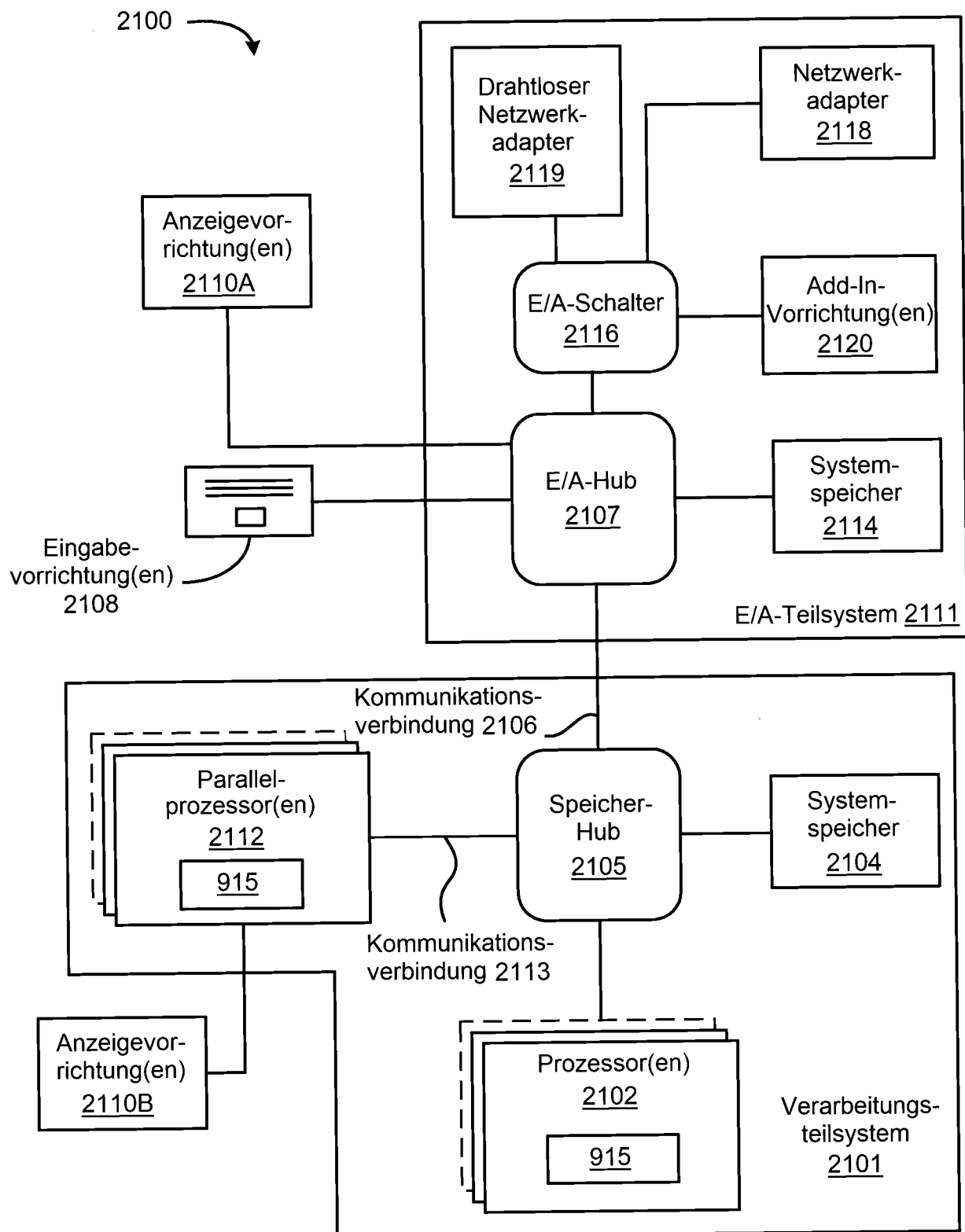


FIG. 20B

**FIG. 21**

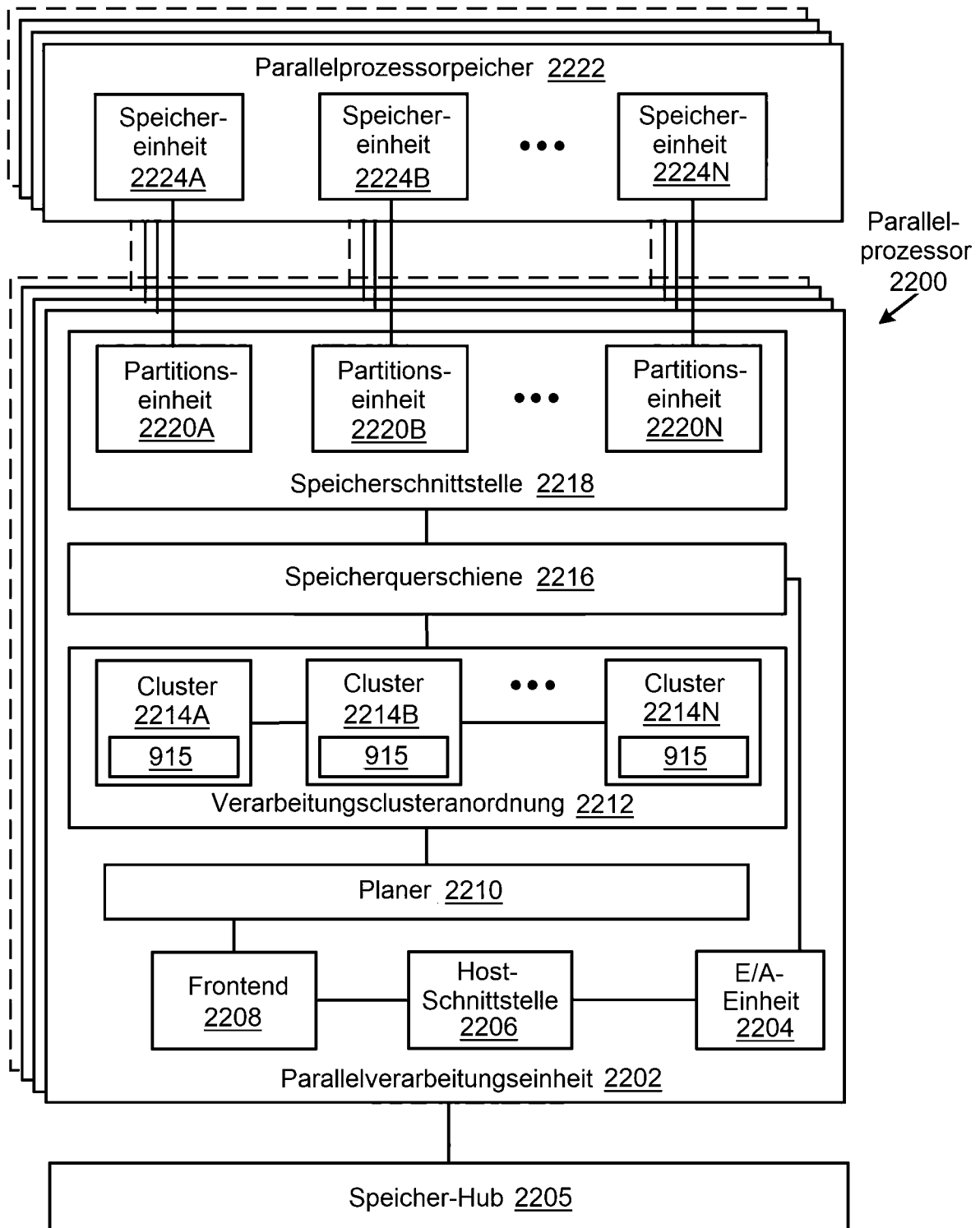


FIG. 22A

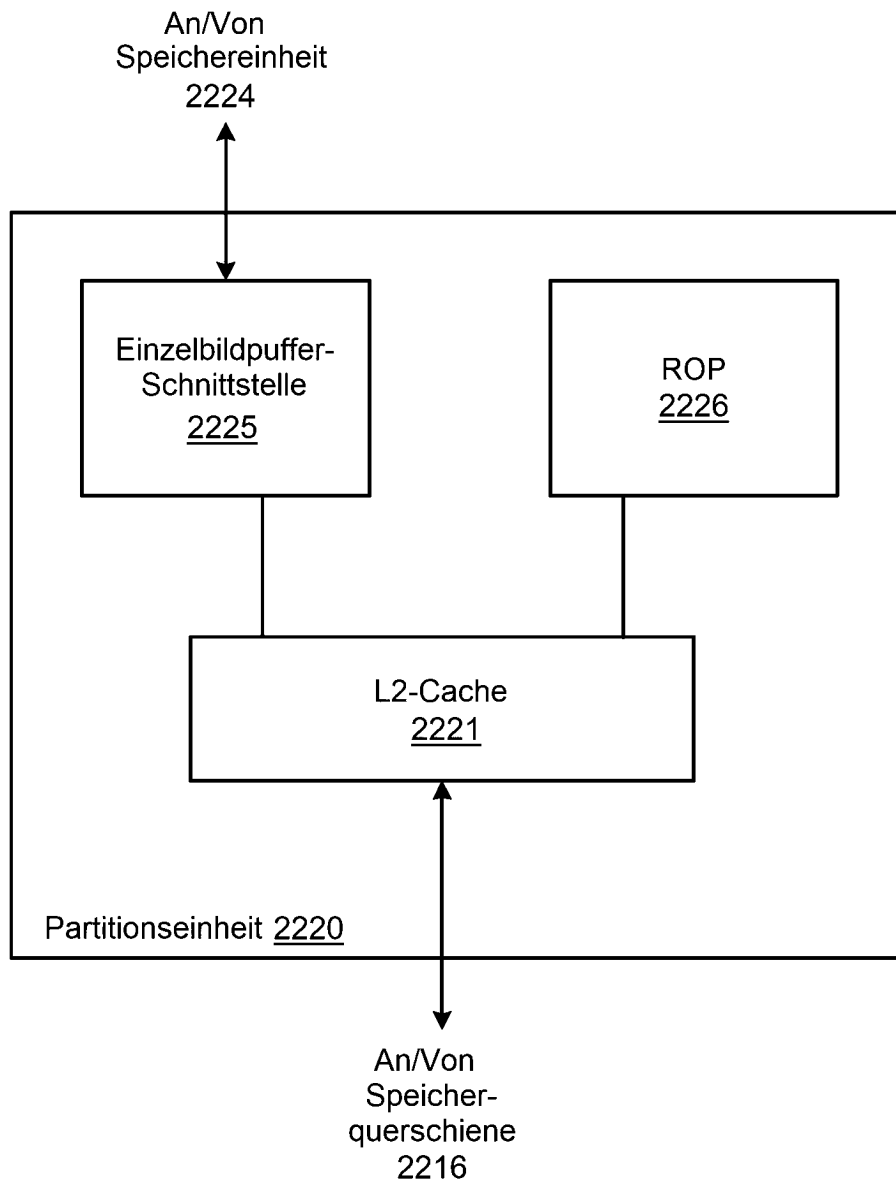


FIG. 22B

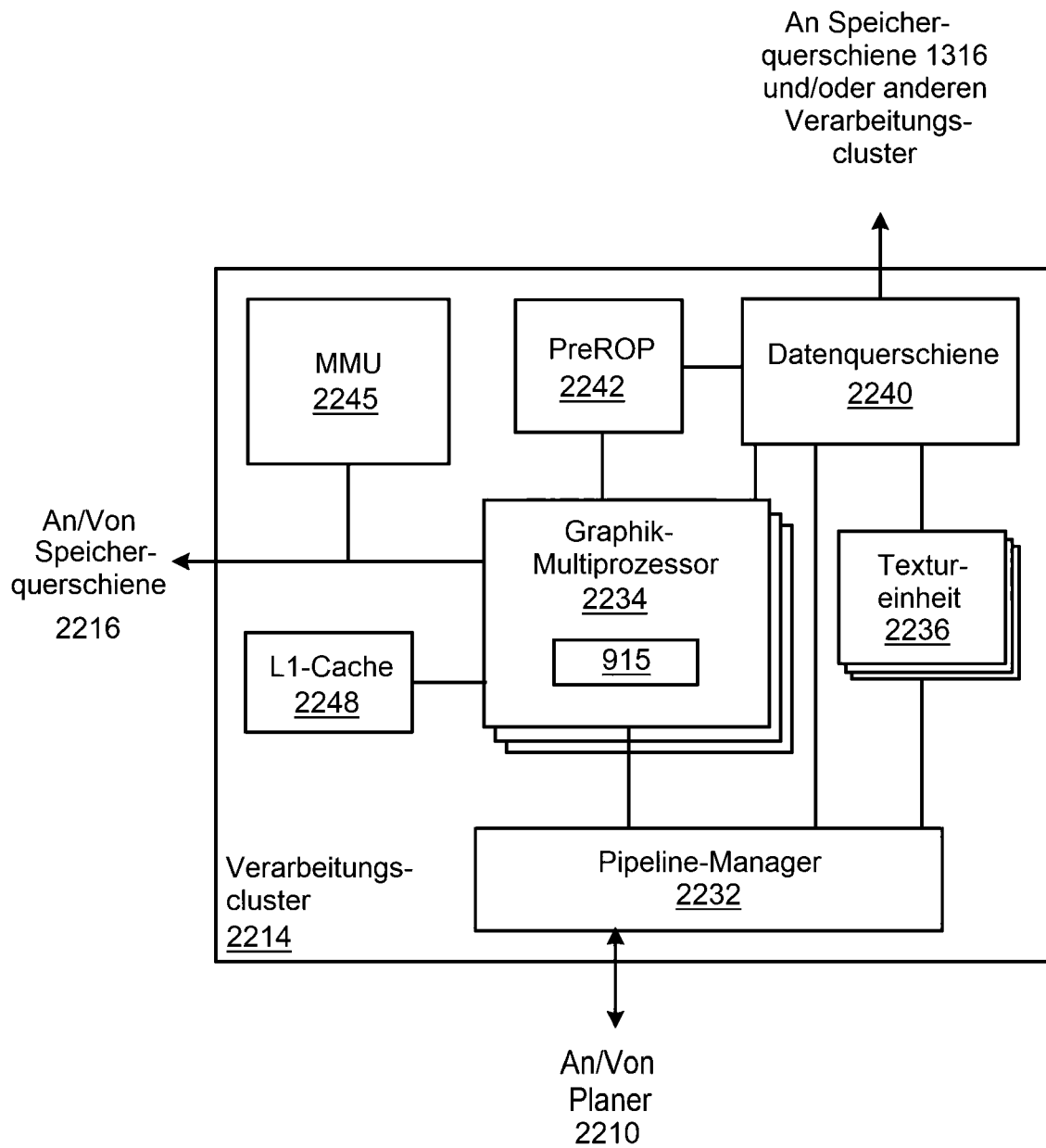
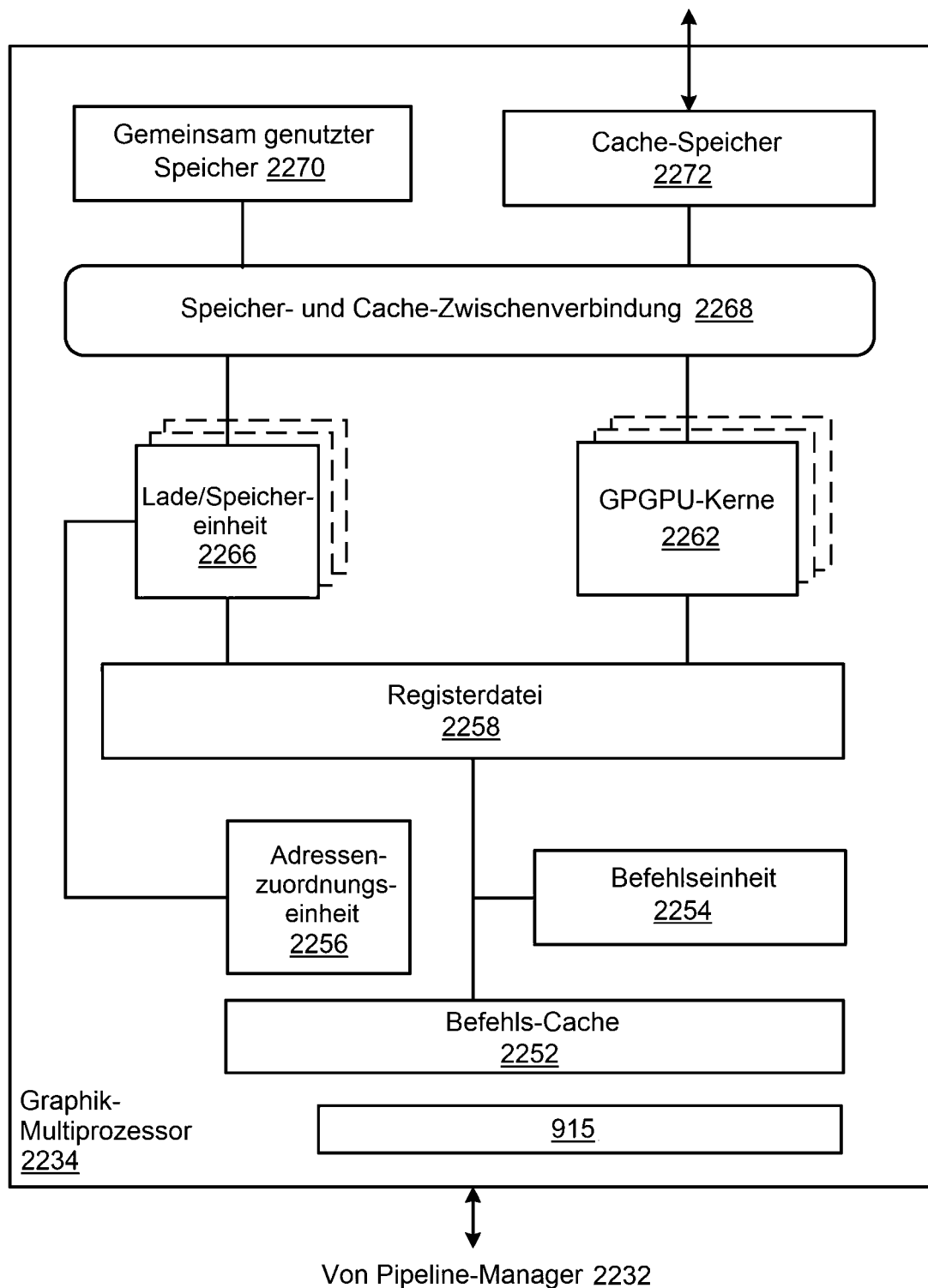


FIG. 22C

**FIG. 22D**

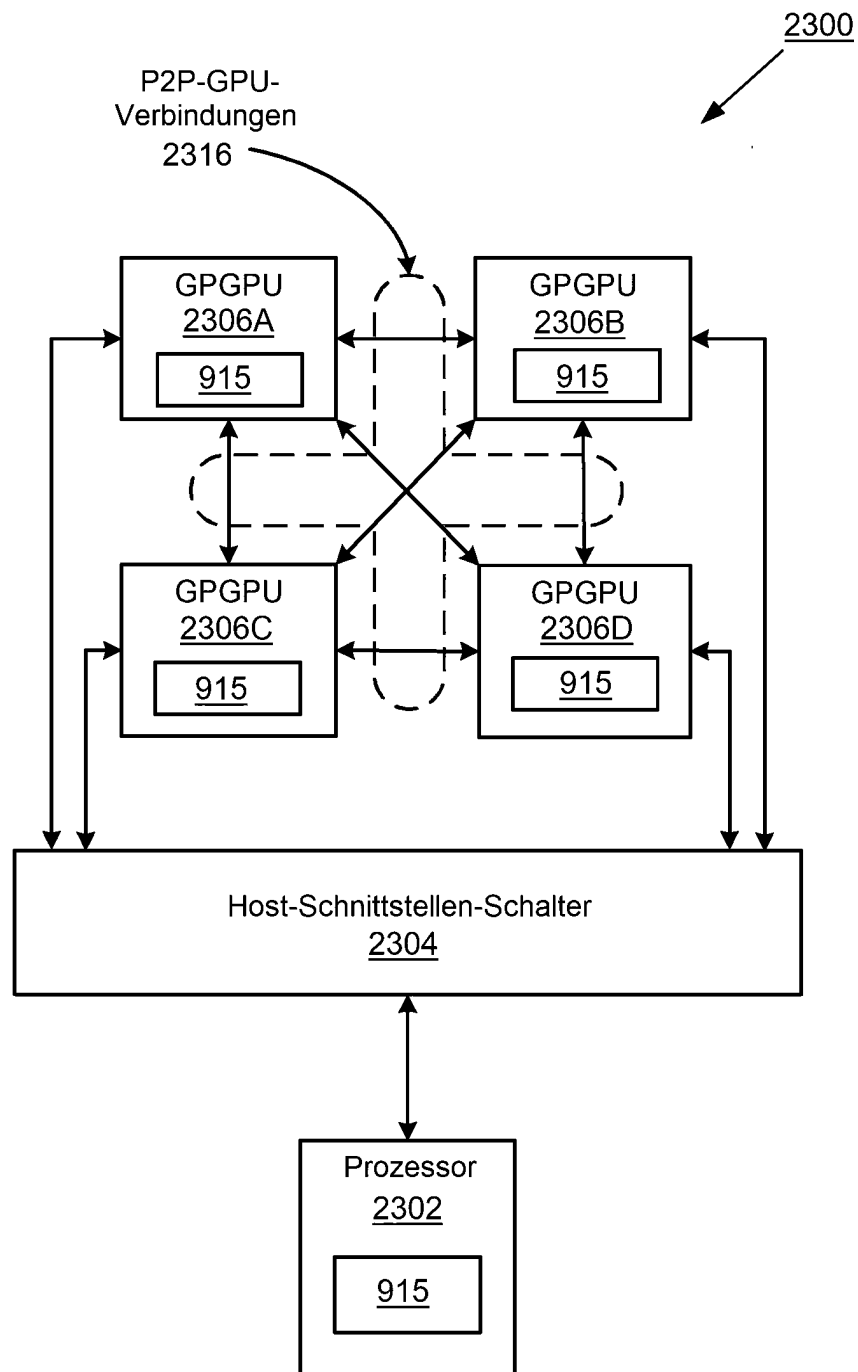


FIG. 23

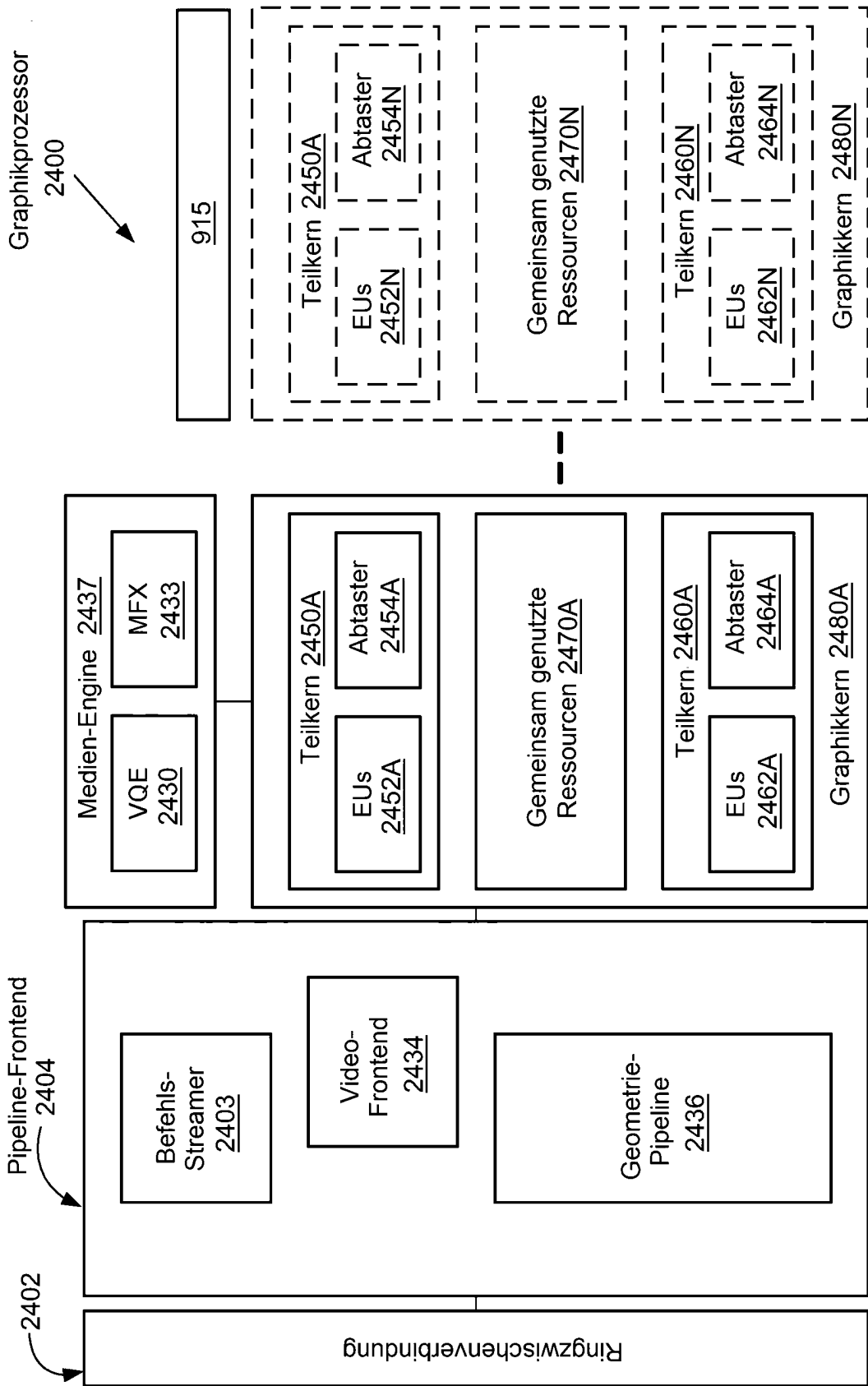


FIG. 24

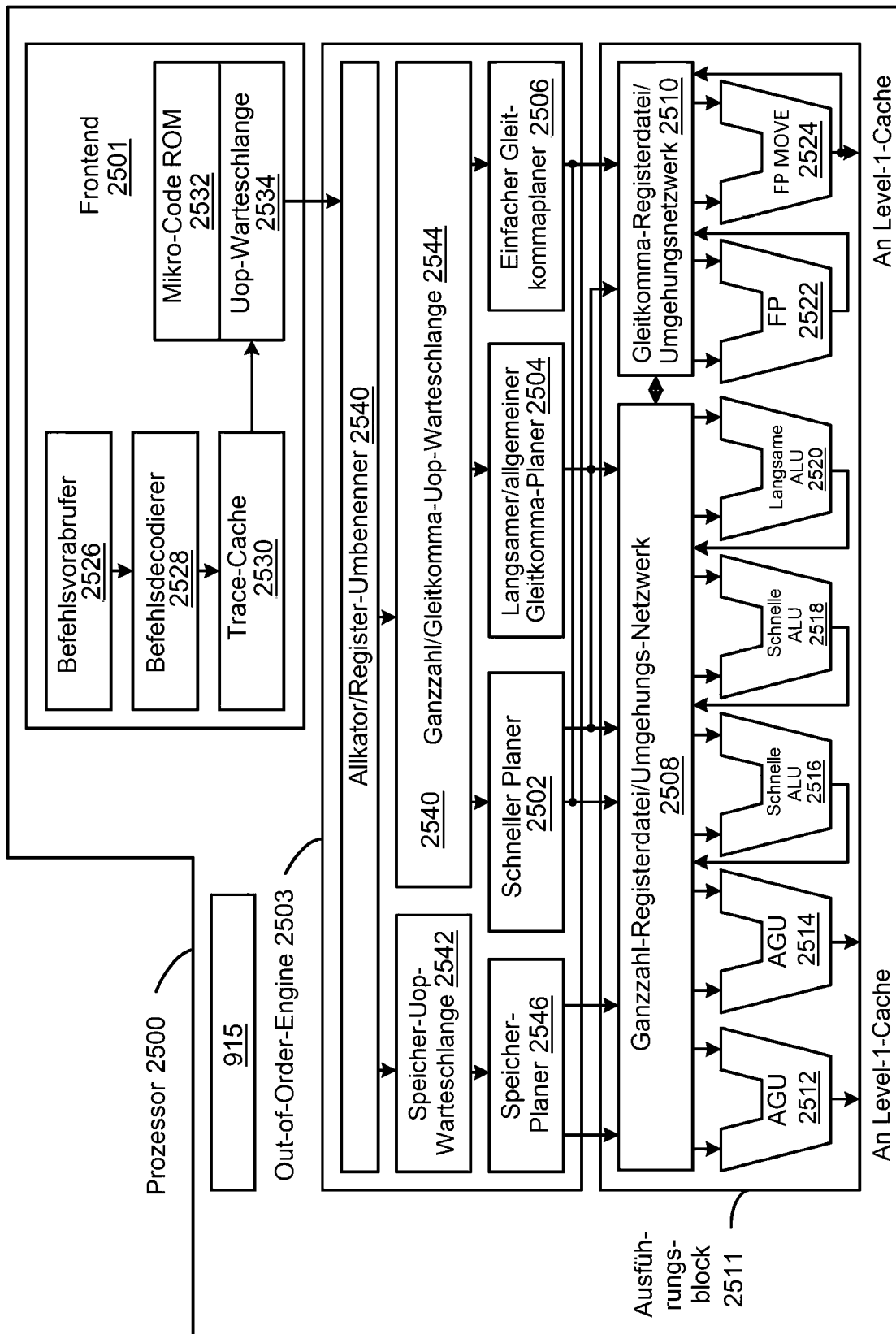


FIG. 25

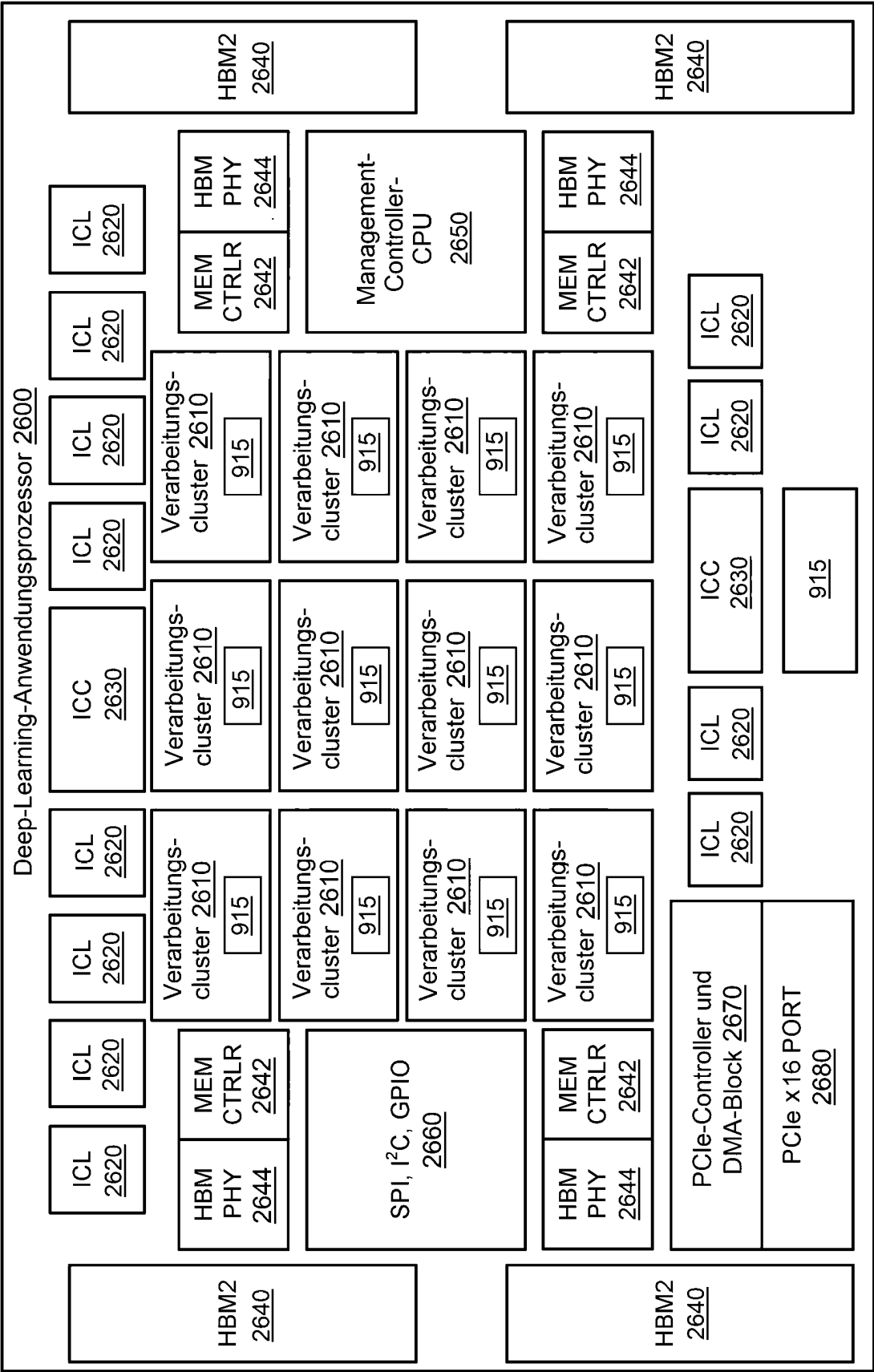
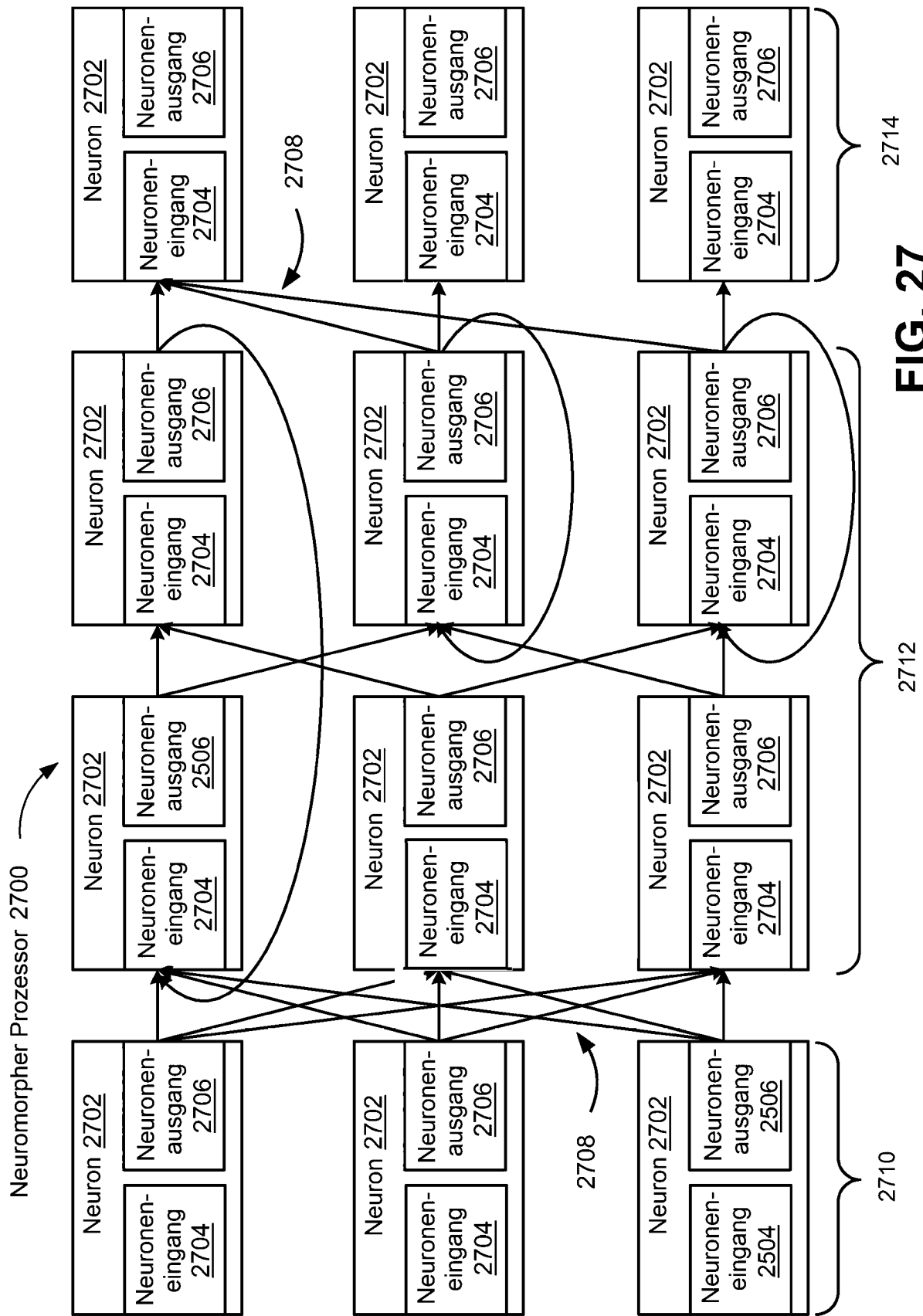


FIG. 26



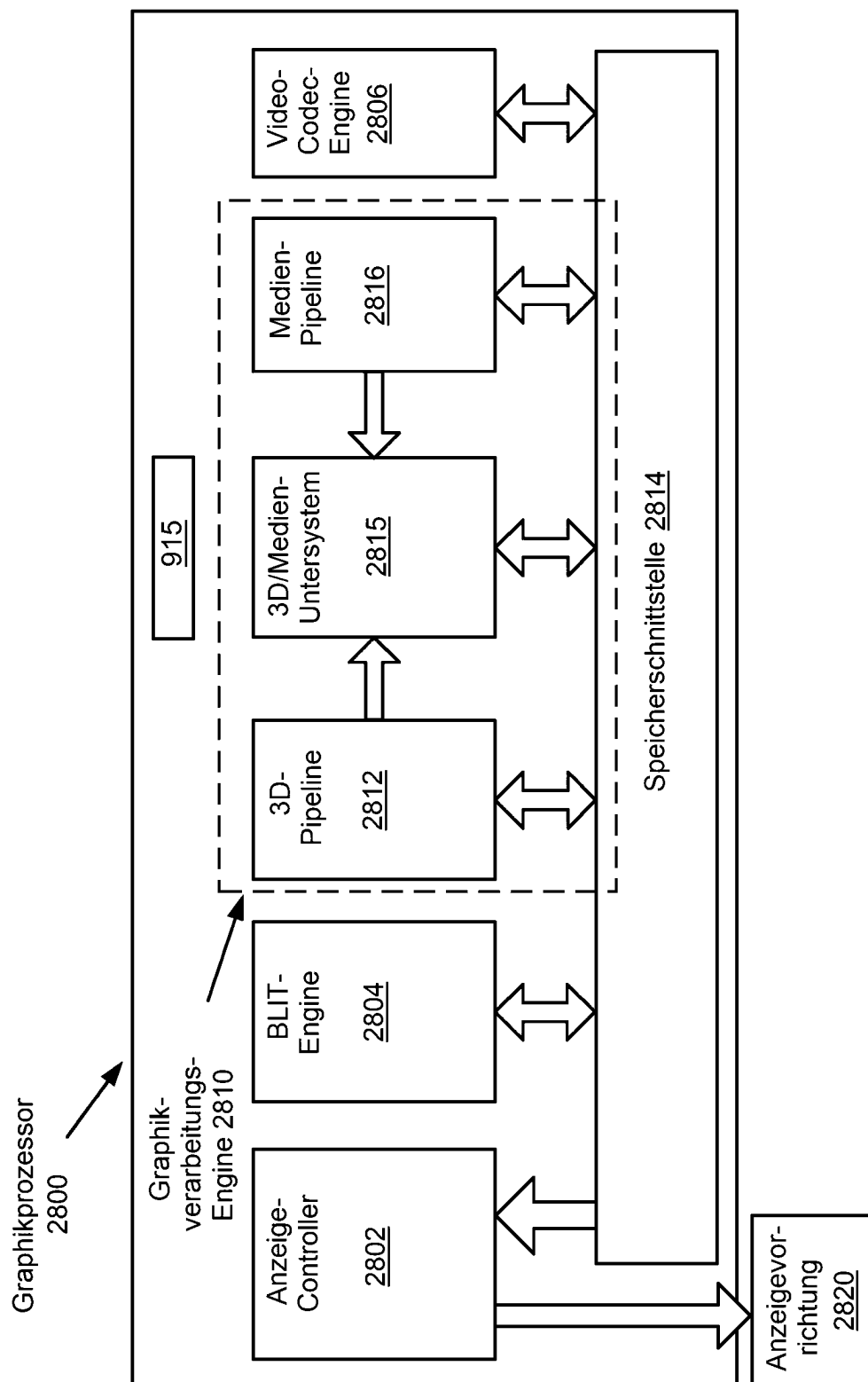


FIG. 28

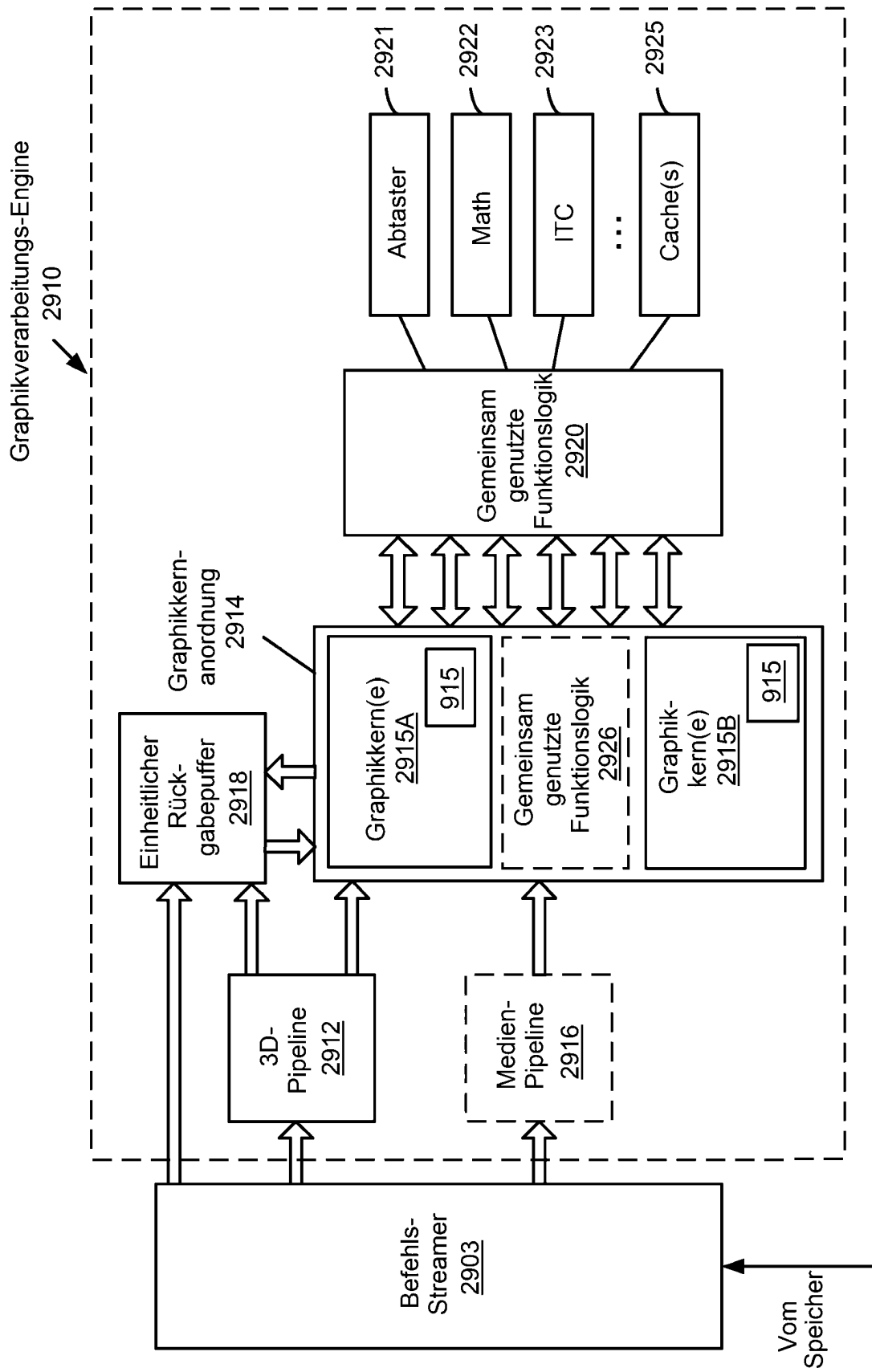


FIG. 29

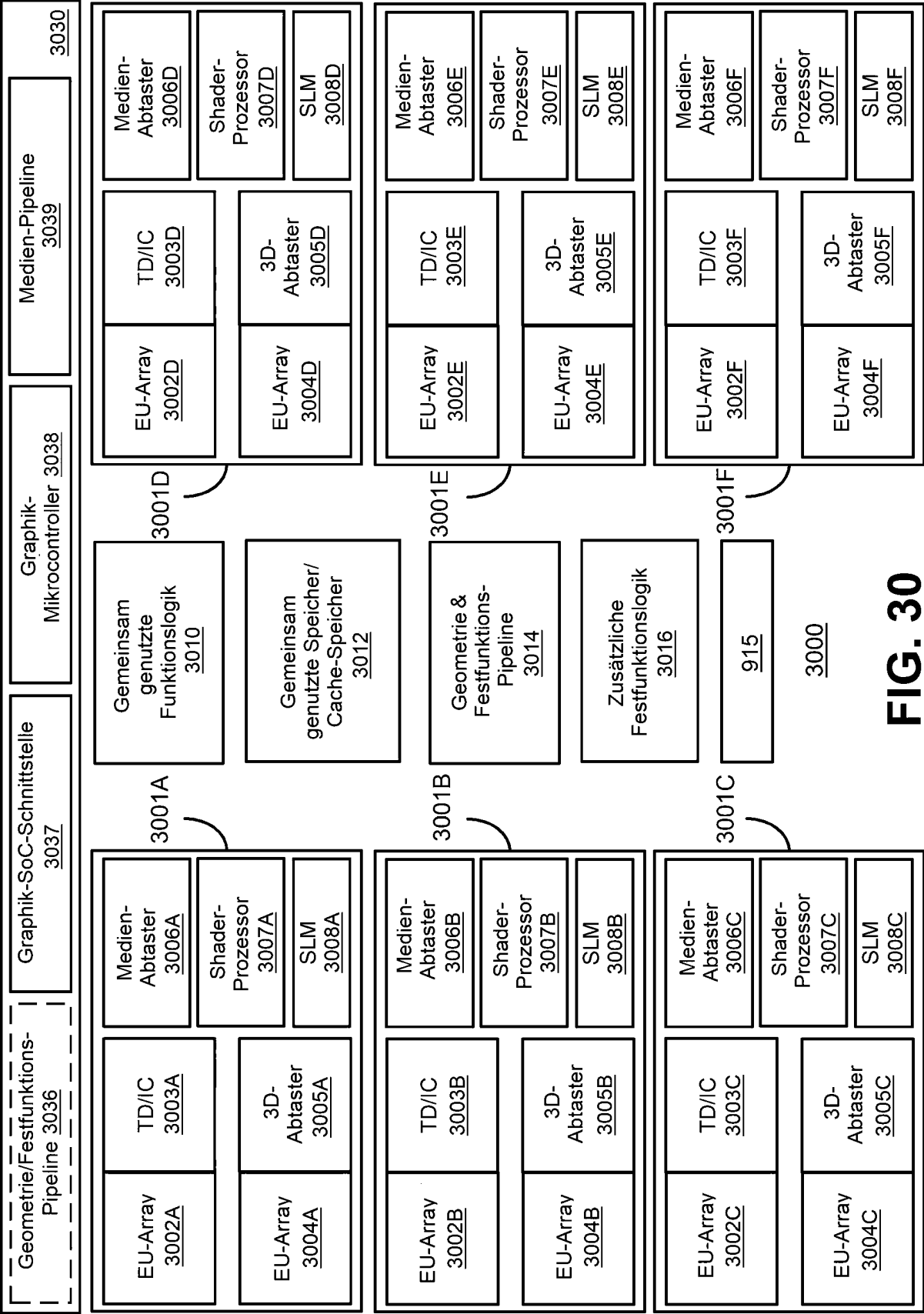


FIG. 30

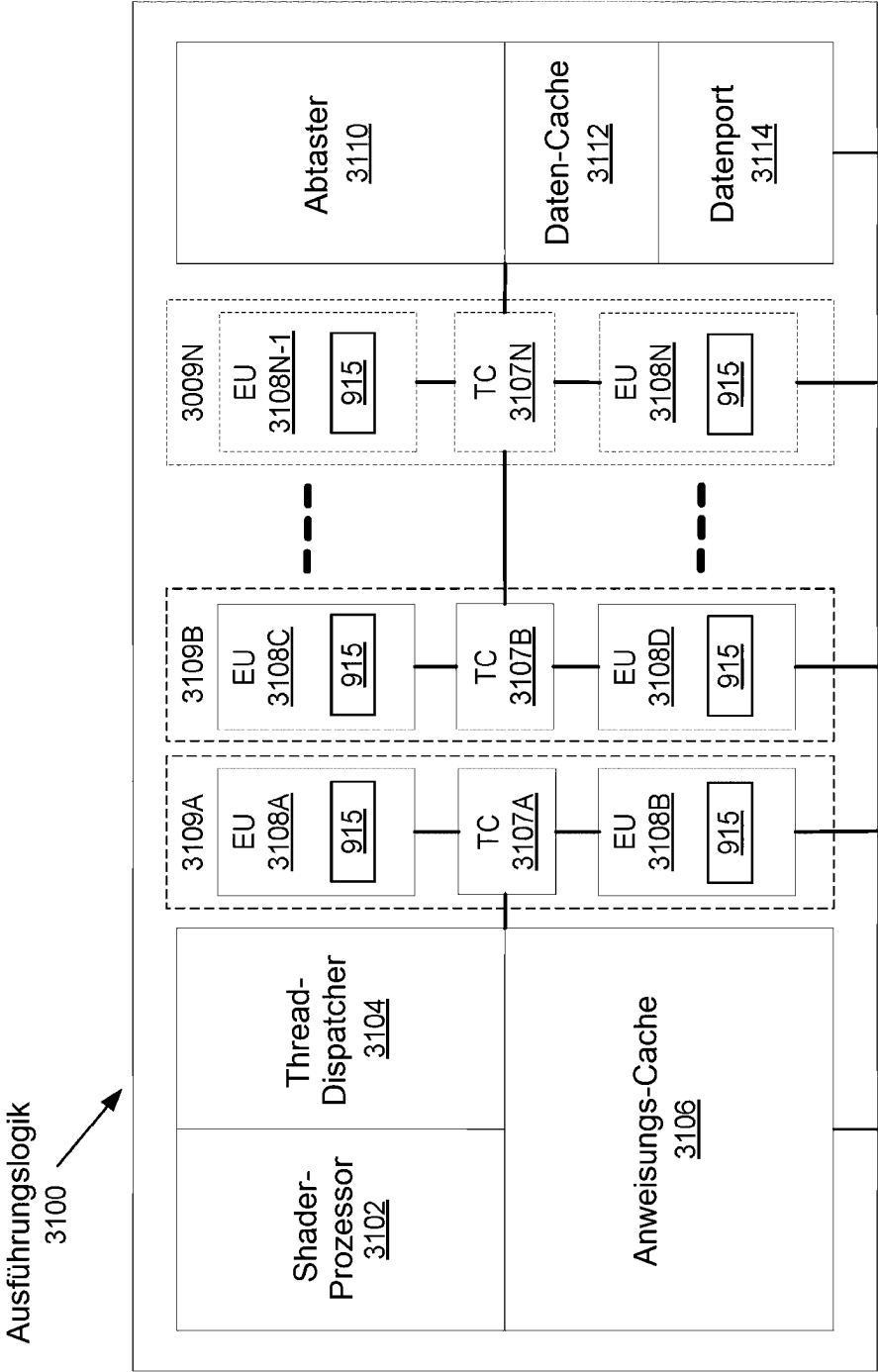


FIG. 31A

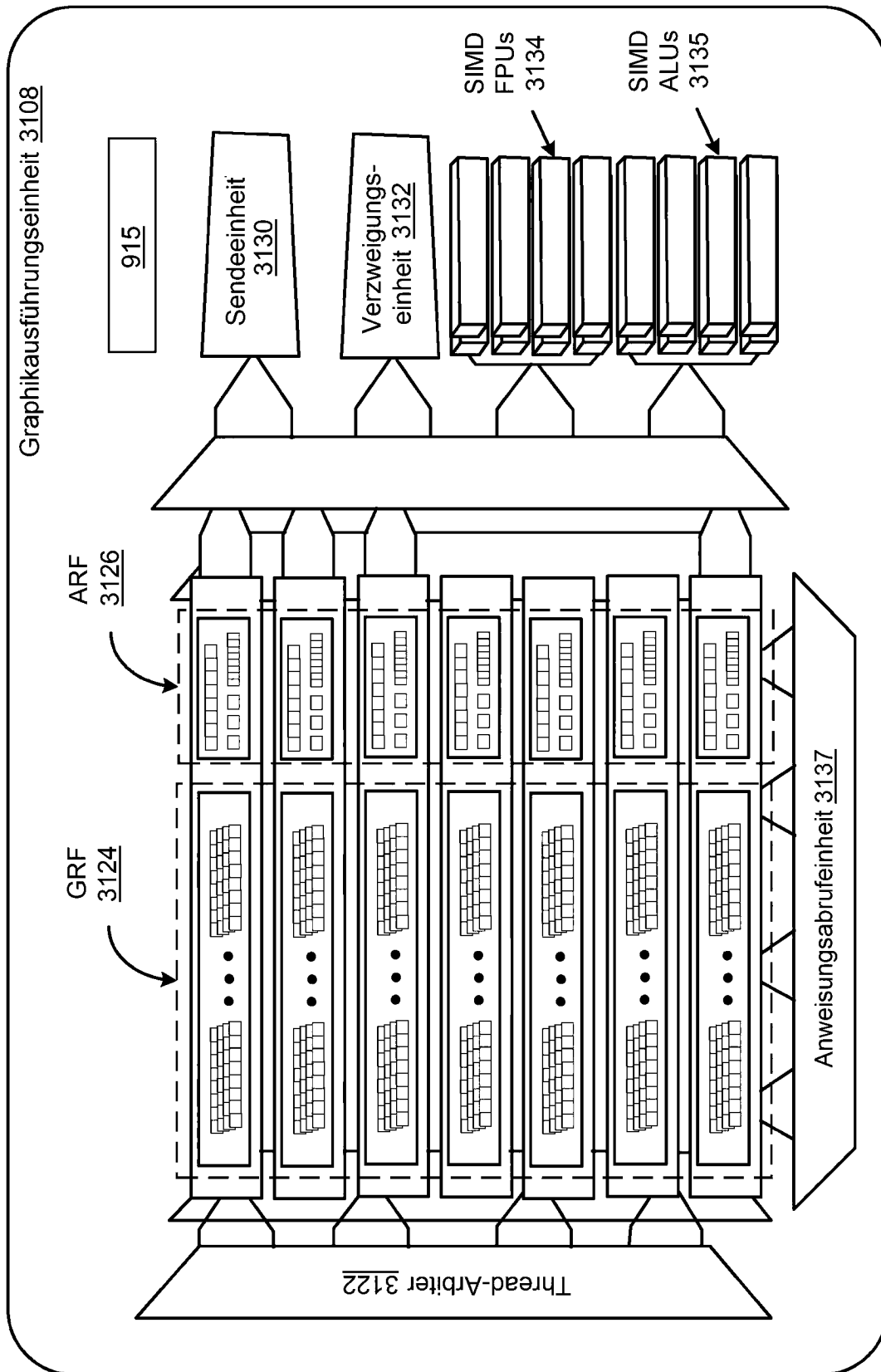
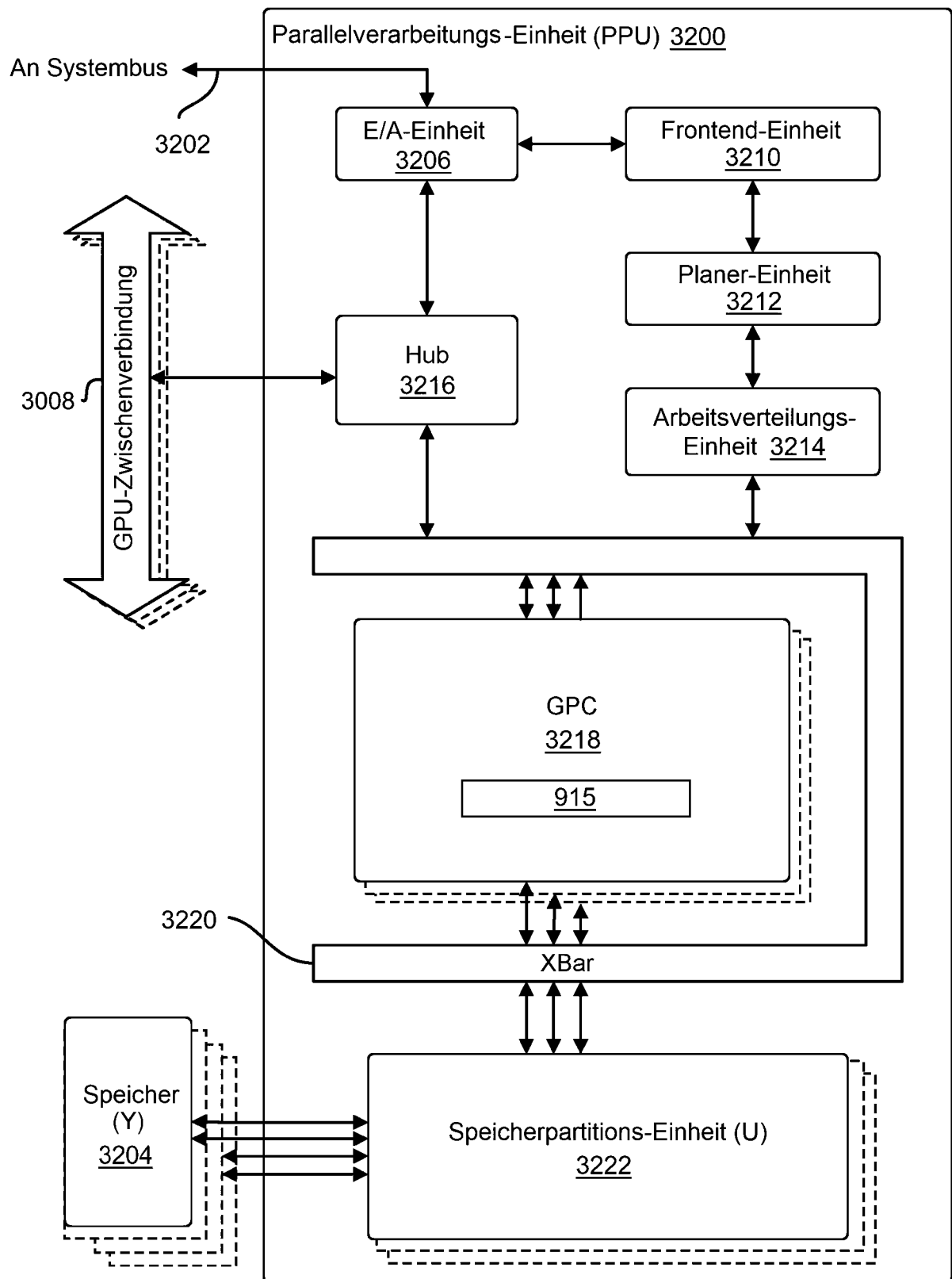


FIG. 31B

**FIG. 32**

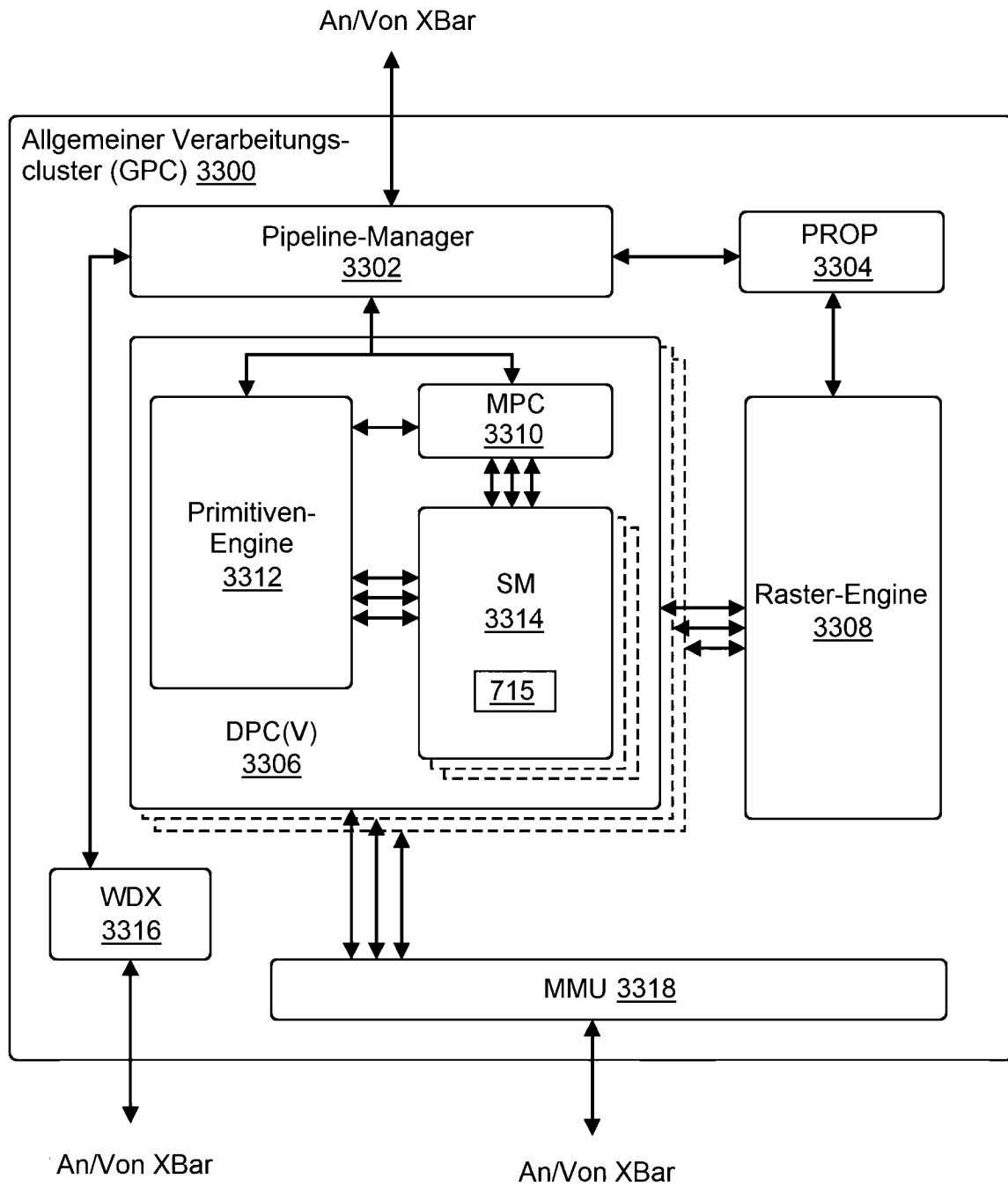


FIG. 33

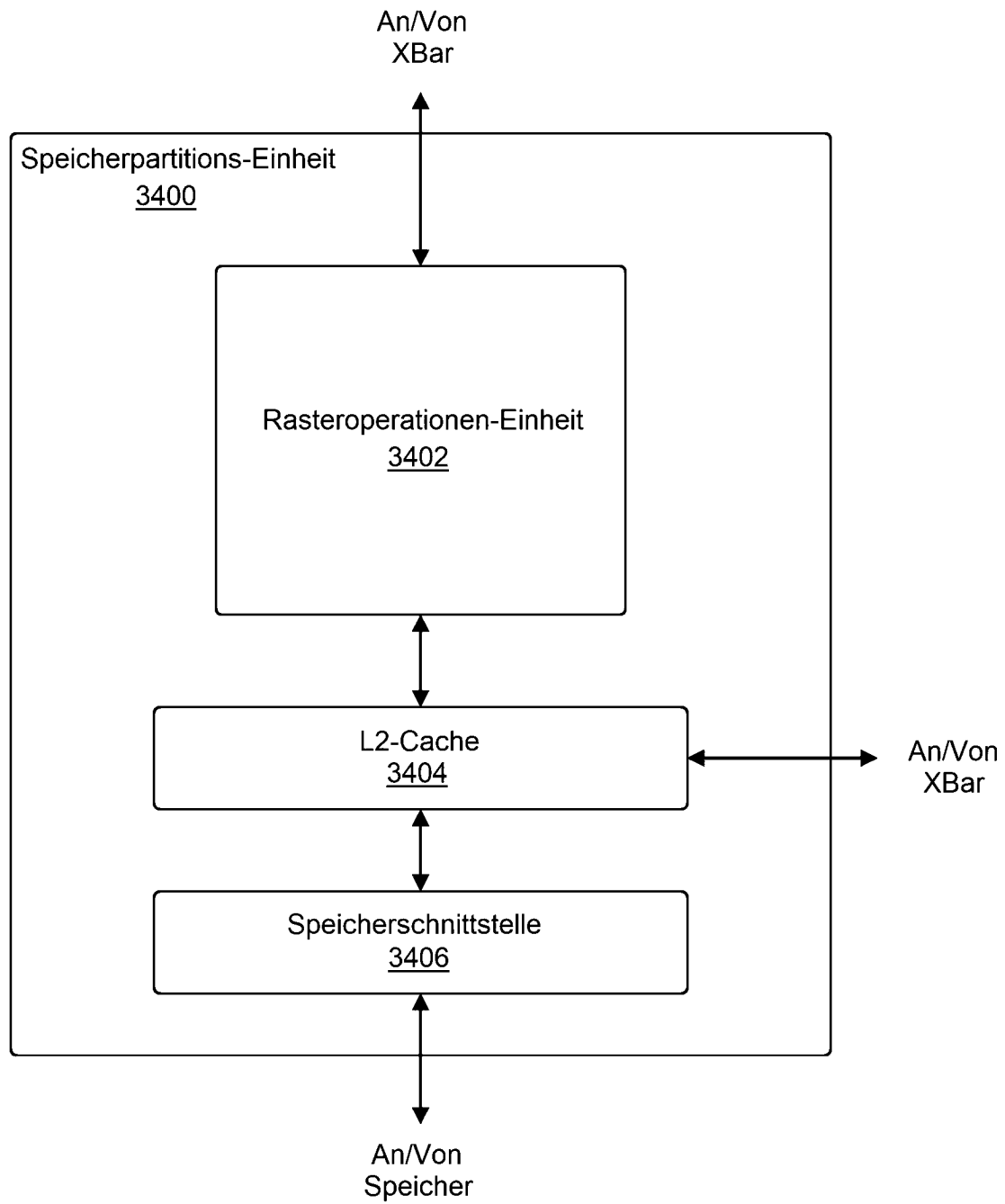
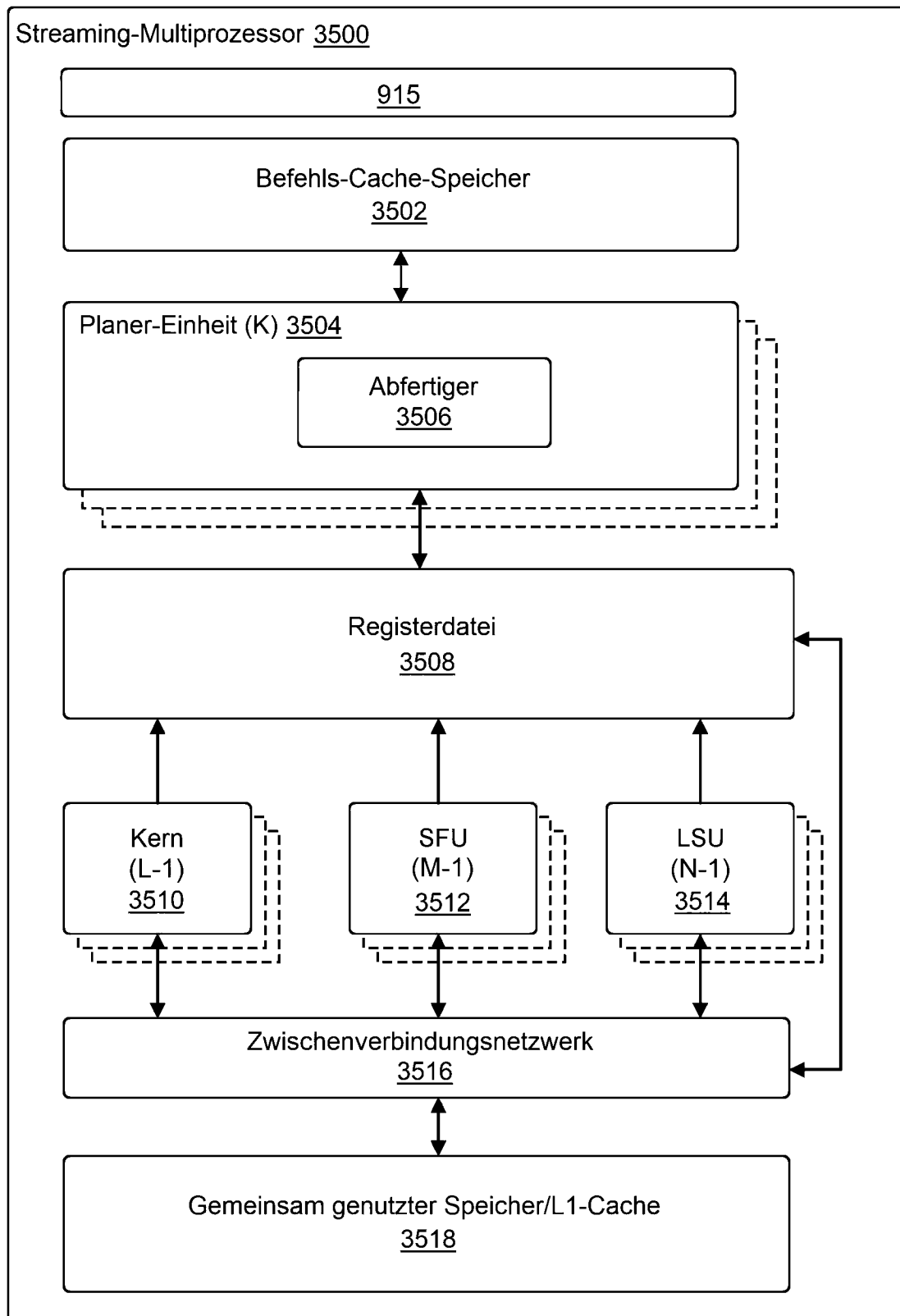


FIG. 34

**FIG. 35**