



(10) **DE 10 2021 114 012 A1** 2021.12.09

(12)

Offenlegungsschrift

(21) Aktenzeichen: **10 2021 114 012.9**

(22) Anmeldetag: **31.05.2021**

(43) Offenlegungstag: **09.12.2021**

(51) Int Cl.: **G06F 1/20 (2006.01)**
G06F 9/50 (2006.01)

(30) Unionspriorität:
16/891,999 03.06.2020 US

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

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

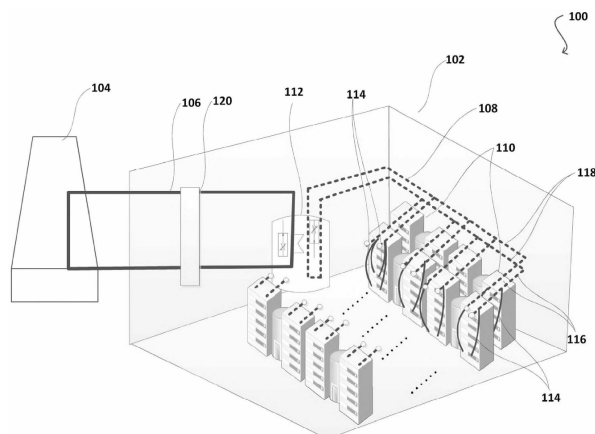
(72) Erfinder:
Heydari, Ali, Santa Clara, CA, US

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **INTELLIGENTE FLÜSSIGKEITSGEKÜHLTE COMPUTER- PODS FÜR EIN MOBILES RECHENZENTRUM**

(57) Zusammenfassung: Ein mobiles Rechenzentrumkühlsystem wird offenbart. Das mobile Rechenzentrumkühlsystem umfasst mindestens einen Behälter, der seinerseits einen Behälterverteiler umfasst, um Kühlmittel, das einem Kühlturm zugeordnet ist, zu einem oder mehreren flüssigkeitsgekühlten Gestellen innerhalb des mindestens einen Behälters zu zirkulieren, und eine Fluidkopplung des Behälterverteilers mit einem zweiten Behälterverteiler eines zweiten Behälters zu ermöglichen.



Beschreibung

GEBIET

[0001] Mindestens eine Ausführungsform betrifft mobile Kühlsysteme für ein mobiles Rechenzentrum. In mindestens einer Ausführungsform umfasst mindestens ein Behälter einen Behälterverteiler, um Kühlmittel, das einem Kühlturm zugeordnet ist, zu einem oder mehreren flüssigkeitsgekühlten Gestellen innerhalb des mindestens einen Behälters zu zirkulieren, und um eine Fluidkopplung des Behälterverters mit einem zweiten Behälterverteiler eines zweiten Behälters zu ermöglichen.

HINTERGRUND

[0002] Rechenzentrumskühlsysteme verwenden typischerweise Gebläse, um Luft durch Serverkomponenten zu zirkulieren. Bestimmte Supercomputer oder andere Computer hoher Kapazität können Wasser oder andere Kühlsysteme als Luftkühlsysteme verwenden, um Wärme von den Serverkomponenten oder Gestellen des Rechenzentrums zu einem Bereich extern zu dem Rechenzentrum abzuführen. Die Kühlsysteme können eine Kältemaschine innerhalb des Rechenzentrumsbereichs umfassen. Der Bereich extern zu dem Rechenzentrum kann ein Kühlturm oder ein anderer externer Wärmetauscher sein, der erwärmtes Kühlmittel (auch als verbrauchtes Kühlmittel bezeichnet) von dem Rechenzentrum empfängt und die Wärme durch Zwangsluft oder andere Mittel an die Umgebung (oder ein externes Kühlmedium) zerstreut, bevor das gekühlte Kühlmittel zurück in das Rechenzentrum rezirkuliert wird. In einem Beispiel bilden die Kältemaschine und der Kühlturm zusammen eine abkühlende Einrichtung mit Pumpen, die auf die Temperatur reagieren, die von an das Rechenzentrum angelegte externe Vorrichtungen gemessen wird. Luftkühlsysteme ziehen Wärme nicht ausreichend ab, um eine wirksame oder effiziente Kühlung in Rechenzentren zu unterstützen, und Flüssigkeitskühlsysteme können für einen schnellen Einsatz hinsichtlich des Potentials für erhebliche Schäden an Serverkomponenten oder Gestellen durch elektrischen Kurzschluss, Überschwemmung oder andere Problemen unhandlich sein.

Figurenliste

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

Fig. 1 ist ein Blockdiagramm eines beispielhaften Rechenzentrums mit einem Kühlsystem, das Verbesserungen unterliegt, die in mindestens einer Ausführungsform beschrieben werden;

Fig. 2 ist ein Blockdiagramm eines mobilen Rechenzentrumsystems oder Kühlsystems, das einen Behälter mit einem Behälterverteiler aufweist, um die Zirkulation eines Kühlmittels zu ermöglichen, das einem mobilen Kühlturm zugeordnet ist, gemäß mindestens einer Ausführungsform;

Fig. 3 ist eine topologische Veranschaulichung von Behältern oder Pods eines mobilen Rechenzentrumssystems oder Kühlsystems, die in einer Anordnung konfiguriert sind, welche die Zirkulation eines Kühlmittels, das einem mobilen Kühlturm für einen oder mehrere Behälter oder Pods ermöglicht, gemäß mindestens einer Ausführungsform;

Fig. 4 ist eine Veranschaulichung eines schnell eingesetzten mobilen Rechenzentrums mit einem mobilen Rechenzentrumskühlsystem gemäß mindestens einer Ausführungsform;

Fig. 5 ist ein Prozessablauf von Schritten, die für ein Verfahren zum Verwenden oder Herstellen des mobilen Kühlsystems oder zum Einsetzen eines mobilen Rechenzentrumssystems mit einem Kühlsystem wie beispielsweise gemäß **Fig. 2-4** und **Fig. 6A-17D** verfügbar sind, gemäß mindestens einer Ausführungsform;

Fig. 6A veranschaulicht ein beispielhaftes Rechenzentrum, bei dem mindestens eine Ausführungsform von **Fig. 2-5** verwendet werden kann;

Fig. 6B, Fig. 6C veranschaulichen eine Inferenz- und/oder Trainingslogik, wie beispielsweise in **Fig. 6A** und in mindestens einer Ausführungsform der vorliegenden Offenbarung verwendet, zum Ermöglichen und/oder Unterstützen eines mobilen Rechenzentrums mit einem mobilen Rechenzentrumskühlsystem, gemäß verschiedener Ausführungsformen;

Fig. 7A ist ein Blockdiagramm, das ein beispielhaftes Computersystem veranschaulicht, das ein System mit miteinander verbundenen Vorrichtungen und Komponenten, ein System-on-Chip (SOC) oder eine Kombination davon sein kann, das mit einem Prozessor gebildet wird, der Ausführungseinheiten umfassen kann, um eine Anweisung auszuführen, um ein mobiles Rechenzentrum mit einem mobilen Rechen-

zentrumkühlsystem, wie hier beschrieben, zu unterstützen und/oder zu ermöglichen, gemäß mindestens einer Ausführungsform;

Fig. 7B ist ein Blockdiagramm, das eine elektronische Vorrichtung zum Benutzen eines Prozessors veranschaulicht, um ein mobiles Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem zu unterstützen oder zu ermöglichen, wie hier beschrieben, gemäß mindestens einer Ausführungsform;

Fig. 7C ist ein Blockdiagramm, das eine elektronische Vorrichtung zum Benutzen eines Prozessors veranschaulicht, um ein mobiles Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem zu unterstützen oder zu ermöglichen, wie hier beschrieben, gemäß mindestens einer Ausführungsform;

Fig. 8 veranschaulicht ein weiteres beispielhaftes Computersystem gemäß mindestens einer Ausführungsform, um verschiedene Prozesse und Verfahren für ein mobiles Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem zu implementieren, wie in dieser gesamten Offenbarung beschrieben;

Fig. 9A veranschaulicht eine beispielhafte Architektur, bei der GPUs mit Mehrkern-Prozessoren über Hochgeschwindigkeit-Verbindungen zum Ermöglichen und/oder Unterstützen eines mobilen Rechenzentrums mit einem mobilen Rechenzentrumkühlsystem kommunikativ gekoppelt sind, gemäß mindestens einer Ausführungsform der Offenbarung hier;

Fig. 9B veranschaulicht zusätzliche Einzelheiten für eine Zwischenverbindung zwischen einem Mehrkern-Prozessor und einem Graphikbeschleunigungsmodul gemäß einer beispielhaften Ausführungsform;

Fig. 9C veranschaulicht eine andere beispielhafte Ausführungsform, bei der eine Beschleunigerintegrations-schaltung innerhalb eines Prozessors integriert ist, um ein mobiles Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem zu ermöglichen und/oder zu unterstützen, gemäß mindestens einer Ausführungsform der Offenbarung hier;

Fig. 9D veranschaulicht eine beispielhafte

Beschleunigerintegrations-Slice **990**, um ein mobiles Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem zu ermöglichen und/oder zu unterstützen, gemäß mindestens einer Ausführungsform der Offenbarung hier;

Fig. 9E veranschaulicht zusätzliche Einzelheiten für eine beispielhafte Ausführungsform eines gemeinsam genutzten Modells, um ein mobiles Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem zu ermöglichen und/oder zu unterstützen, gemäß mindestens einer Ausführungsform der Offenbarung hier;

Fig. 9F veranschaulicht zusätzliche Einzelheiten für eine beispielhafte Ausführungsform eines vereinheitlichten Speichers, der über einen gemeinsamen virtuellen Speicheradressraum adressierbar ist, der verwendet wird, um auf physikalischen Prozessorspeicher und GPU-Speicher zuzugreifen, um ein mobiles Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem zu ermöglichen und/oder zu unterstützen, gemäß mindestens einer Ausführungsform der Offenbarung hier;

Fig. 10A veranschaulicht beispielhafte integrierte Schaltungen und zugeordnete Graphikprozessoren gemäß hier beschriebener Ausführungsformen für ein mobiles Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem;

Fig. 10B-10C veranschaulichen beispielhafte integrierte Schaltungen und zugeordnete Graphikprozessoren gemäß mindestens einer Ausführungsform, um ein mobiles Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem zu unterstützen und/oder zu ermöglichen;

Fig. 10D-10E veranschaulichen eine zusätzliche beispielhafte Graphikprozessorlogik gemäß mindestens einer Ausführungsform, um ein mobiles Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem zu unterstützen und/oder zu ermöglichen;

Fig. 11A ist ein Blockdiagramm, das ein Rechensystem veranschaulicht, um ein mobiles Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem zu unterstützen und/oder zu ermöglichen, gemäß mindestens einer Ausführungsform;

Fig. 11B veranschaulicht einen parallelen Prozessor, um ein mobiles Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem zu unterstützen und/oder zu ermöglichen, gemäß mindestens einer Ausführungsform;

Fig. 11C ist ein Blockdiagramm einer Partitionseinheit gemäß mindestens einer Ausführungsform;

Fig. 11D zeigt einen Graphikmultiprozessor, der für ein mobiles Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem verwendet wird, gemäß mindestens einer Ausführungsform;

Fig. 11E zeigt einen Graphikmultiprozessor gemäß mindestens einer Ausführungsform;

Fig. 12A veranschaulicht ein Multi-GPU-Rechensystem gemäß mindestens einer Ausführungsform;

Fig. 12B ist ein Blockdiagramm eines Graphikprozessors gemäß mindestens einer Ausführungsform;

Fig. 13 ist ein Blockdiagramm, das eine Mikroarchitektur für einen Prozessor veranschaulicht, die Logikschaltungen umfassen kann, um Anweisungen durchzuführen, gemäß mindestens einer Ausführungsform;

Fig. 14 veranschaulicht einen tief lernenden bzw. Deep-Learning-Anwendungsprozessor gemäß mindestens einer Ausführungsform;

Fig. 15 ist ein Blockdiagramm eines neuromorphischen Prozessors gemäß mindestens einer Ausführungsform;

Fig. 16A ist ein Blockdiagramm eines Verarbeitungssystems gemäß mindestens einer Ausführungsform;

Fig. 16B ist ein Blockdiagramm eines Prozessors mit einem oder mehreren Prozessorkernen, eines integrierten Speichercontrollers und eines integrierten Graphikprozessors gemäß mindestens einer Ausführungsform;

Fig. 16C ist ein Blockdiagramm von Hardwarelogik eines Graphikprozessorkerns gemäß mindestens einer Ausführungsform;

Fig. 16D-16E veranschaulichen eine Thread-Ausführungslogik, die eine Anordnung von Verarbeitungselementen eines Graphikprozessorkern umfasst, gemäß mindestens einer Ausführungsform;

Fig. 17A veranschaulicht eine Parallelverarbeitungseinheit gemäß mindestens einer Ausführungsform;

Fig. 17B veranschaulicht einen allgemeinen Verarbeitungscluster gemäß mindestens einer Ausführungsform;

Fig. 17C veranschaulicht eine Speicherpartitionseinheit einer Parallelverarbeitungseinheit gemäß mindestens einer Ausführungsform; und

Fig. 17D veranschaulicht einen Streaming-Multi-Prozessor gemäß mindestens einer Ausführungsform.

AUSFÜHRLICHE BESCHREIBUNG

[0004] Eine Luftkühlung von hochdichten Servern ist ineffizient und unwirksam hinsichtlich der hohen Wärmeanforderungen, die von heutigen Rechenkomponenten verursacht werden. Von daher strebt die vorliegenden Offenbarung Perspektiven bei flüssigen Kühlmitteln und zugeordneten Systemen zur Kühlung von Rechenkomponenten an, wie beispielsweise einer Graphikverarbeitungseinheit (Graphics Processing Unit; GPU), einer Zentralverarbeitungseinheit (Central Processing Unit; CPU) oder Schaltkomponenten. Diese Rechenkomponenten werden in Servern verwendet, die in Servereinschüben auf Gestellen in einem Rechenzentrum zusammengebaut werden. Da die Rechenkomponenten durch technologische Fortschritte miniaturisiert sind, nehmen die Servereinschübe und die Gestelle mehr und mehr Rechenkomponenten auf, um dadurch die Ableitung von mehr Wärme, die pro Komponente erzeugt wird, als in vorherigen Systemen zu erfordern. Ein Thema, das in der vorliegenden Offenbarung angesprochen wird, ist eine Unfähigkeit, eine Kühlung auf Anfrage während nachteiliger saisonaler Wirkungen auf das Rechenzentrum bereitzustellen.

[0005] Ferner wird ein Rechenzentrum-Flüssigkeitskühlsystem durch eine Kühlmaschinenanlage oder -system unterstützt, das aufwändig sein kann, weil es mit einer Überbereitstellung von Anforderungen ausgestaltet ist. Die Überbereitstellungs-Anforderungen können auf aufgrund fehlender geeigneter Steuerung zur Entfernung von Wärme aus dem Server, wie beispielsweise aus einer oder mehreren Komponenten (z.B. GPU, CPU usw.), aus den kollektiven Komponenten in den Serverkästen oder den kollektiven Komponenten innerhalb der Gestelle. Eine effiziente Verwendung von Flüssigkeitskühlsystemen hoher Wärmekapazität kann durch die vielen Zwischenmerkmale in dem Kühlsystem, die flüssige Kältemaschinen, Pumpen, Kühlmittelverteilungseinheiten (Coolant Distribution Units; CDUs) und Wärmetauscher umfassen, überschattet werden.

[0006] Der Einsatz von Rechenzentrumshardware, um schnell wachsenden Anforderungen auf den Gebieten der künstlichen Intelligenz und des Maschinenlernens anzusprechen, erfordern wesentliche Rechenfähigkeiten mit extrem dichten Servergestellen von GPU/CPU-Servern mit einer sehr schnellen Vernetzungsausrüstung. Die Herausforderungen können jedoch in der Form von Zugänglichkeit und Verfügbarkeit von gefragten Rechenzentrum präsentiert werden. Außerdem können die Herausforderungen durch die Kühlanforderungen von derartigen gefragten Rechenzentren stark verstärkt werden, die mit dem Wissen eingesetzt werden, dass

sie stark verwendet werden, um rechenintensive Operationen durchzuführen, die Wärme bei Kapazität erzeugen und die nach Bedarf gekühlt werden müssen.

[0007] In mindestens einer Ausführungsform wird eine portierbare Supercomputing- oder rechenintensive Infrastruktur durch einen Behälter oder Pod ermöglicht, der mit Anpassungen versehen ist, die eine Alles-in-Eins-Infrastruktur umfassen, wie beispielsweise Fähigkeiten zum Skalieren und Unterstützen von extrem hochdichten Supercomputing- oder rechenintensiven Anforderungen. Der portierbare Behälter oder Pod ist schnell überall und zu beliebiger Zeit einsetzbar und wird durch eingebaute mobile Kühlsysteme möglich gemacht, die durch ein Kühlmittel, wie beispielsweise Wasser oder ein Wasser-gemischtes Kühlmittel, gekühlt werden können. Die Ausgestaltung und der Einsatz von Serverhardware für den Zweck von Supercomputing- oder rechenintensiven Operationen umfassen räumliches Planen für Gestelle, die erforderlich sind, um die Berechnung von Trainings- und/oder Inferenzierungsbedürfnissen von Arbeitslasten der künstlichen Intelligenz (Artificial Intelligence; AI) und des Maschinlernens (ML) zu ermöglichen. In mindestens einer Ausführungsform ist das mobile Rechenzentrum der vorliegenden Offenbarung angepasst, um eine andere Infrastruktur als die des mobiles Kühlung zu adressieren, die Leistungsanforderungen des Rechenzentrums umfassen, und ist mindestens in einem Behälter oder Pod mit stabiler Umgebungsunterstützung lokalisiert, um den Behälter oder Pod auf einer Plattform aufzuhängen. In mindestens einer Ausführungsform ist der Behälter oder Pod nach Bedarf lieferbar und einsetzbar.

[0008] In mindestens einer Ausführungsform wird ein mobiles Rechenzentrumkühlsystem beschrieben, das mindestens einen Behälter aufweist, der einen Behälterverteiler umfasst, um Kühlmittel zu zirkulieren, das einem Kühlturm zugeordnet ist, der angebracht ist an oder benachbart mindestens einem Behälter mit einem oder mehreren flüssigkeitsgeköhlten Gestellen lokalisiert ist, um eine Fluidkopplung des Behälterverters mit einem zweiten Behälterverteiler eines zweiten Behälters zu ermöglichen. In mindestens einer Ausführungsform wird mindestens ein Prozessor für ein mobiles Kühlsystem beschrieben. Der mindestens eine Prozessor umfasst mindestens eine Logikeinheit, um ein oder mehrere neuronale Netzwerk mit verborgenen Schichten von Neuronen zu trainieren. Das eine oder mehrere neuronale Netzwerke sind zum Bewerten von Temperaturanforderungen eines oder mehrerer flüssigkeitsgeköhlten Gestelle, die in einem Behälter zu hosten sind, zum Bewerten von Strömungsraten oder Strömungsvolumen eines Kühlmittel basierend teilweise auf ihren Zuordnungen zu den Temperaturanforderungen und zum Bewerten mindestens einer physikalischen Beschränkung des Behälters oder eines zweiten Behälters, der einen Kühlturm hostet, um das eine oder mehrere flüssigkeitsgeköhlte Gestelle zu kühlen. Der mindestens eine Prozessor stellt eine Ausgabe bereit, die Kühlmittelanforderungen für das eine oder mehrere flüssigkeitsgeköhlte Gestelle angibt, die innerhalb mindestens einer physikalischen Beschränkung des Behälters oder des zweiten Behälters sind. Ferner wird in mindestens einer Ausführungsform mindestens ein Prozessor als mit mindestens einer Logikeinheit beschrieben, um einen oder mehrere Strömungscontroller zu steuern, die einem Behälterverteiler zugeordnet sind, um Kühlmittel, das einem Kühlturm zugeordnet ist, zu einem oder mehreren flüssigkeitsgeköhlten Gestellen innerhalb mindestens eines Behälters zu zirkulieren, und dem Kühlmittel zu ermöglichen, von dem Behälterverteiler zu einem zweiten Behälterverteiler eines zweiten Behälters zu strömen.

[0009] Fig. 1 ist ein Blockdiagramm eines beispielhaften Rechenzentrums **100** mit einem Kühlsystem, das Verbindungen unterliegt, die in mindestens einer Ausführungsform beschrieben sind. Das Rechenzentrum **100** kann ein oder mehrere Räume **102** mit Gestellen **110** und Hilfsausrüstung sein, um ein oder mehrere Server auf einem oder mehreren Servereinschüben unterzubringen. Das Rechenzentrum **100** wird durch einen Kühlturm **104** unterstützt, der extern zu dem Rechenzentrum **100** lokalisiert ist. Der Kühlturm **104** leitet Wärme von innerhalb des Rechenzentrums **100** durch Wirken auf eine primäre Kühlschleife **106** ab. Ferner wird eine Kühlverteilungseinheit (Cooling Distribution Unit; CDU) **112** zwischen der primären Kühlschleife **106** und einer zweiten oder sekundären Kühlschleife **108** verwendet, um einen Entzug der Wärme aus der zweiten oder sekundären Kühlschleife **108** in die primäre Kühlschleife **106** zu ermöglichen. Die sekundäre Kühlschleife **108** ist imstande, vollständig auf verschiedene Verrohrungen in dem Servereinschub zuzugreifen, wie in einem Aspekt erforderlich. Die Schleifen **106**, **108** sind als Strichzeichnungen veranschaulicht, wobei jedoch ein Fachmann erkennen würde, dass ein oder mehreren Verrohrungsmerkmale verwendet werden können. In einem Fall können biegsame Rohre Polyvinylchlorid (PVC) zusammen mit zugeordneter Verrohrung verwendet werden, um das Fluid in jedem der Schleifen **106**, **108** zu bewegen. Eine oder mehrere Kühlmittelpumpen können in mindestens einer Ausführungsform verwendet werden, um Druckunterschiede innerhalb der Schleifen **106**, **108** aufrechtzuerhalten, um die Bewegung des Kühlmittels gemäß Temperatursensoren in verschiedenen Orten, einschließlich in dem Raum, in einem oder mehreren Gestellen **110** und/oder in Serverkästen oder Servereinschüben innerhalb der Gestelle **110** zu ermöglichen.

[0010] In mindestens einer Ausführungsform kann das Kühlmittel in der primären Kühlschleife **106** und in der sekundären Kühlschleife **108** mindestens Wasser und ein Zusatzstoff, beispielsweise Glykol oder Propylenglykol sein. Im Betrieb weist jede der primären und der sekundären Kühlschleifen ihr eigenes Kühlmittel auf. In einem Aspekt kann das Kühlmittel in den sekundären Kühlschleifen proprietär für Anforderungen der Komponenten in dem Servereinschub oder Gestellen **110** sein. Die CDU **112** ist imstande für eine ausgefeilte Steuerung des Kühlmittels, unabhängig oder gleichläufig, in den Schleifen **106**, **108** fähig. Beispielsweise kann die CDU angepasst sein, um die Strömungsrate zu steuern, so dass das(die) Kühlmittel entsprechend verteilt wird(werden), um Wärme zu entziehen, die innerhalb der Gestelle **110** erzeugt wird. Ferner wird mehr biegsames Verschlauchung **114** von der sekundären Kühlschleife **108** bereitgestellt, um in jeden Servereinschub einzutreten und um den elektrischen und/oder Rechenkomponenten Kühlmittel bereitzustellen. In der vorliegenden Offenbarung werden die elektrischen und/oder Rechenkomponenten austauschbar verwendet, die sich auf die Wärme-erzeugenden Komponenten beziehen, die von dem vorliegenden Rechenzentrumkühlsystem profitieren. Die Verschlauchung **118**, die Teil der sekundären Kühlschleife **108** bildet, kann als Raumverteiler bezeichnet werden. Separat kann die Verschlauchung **116**, die sich von der Verschlauchung **118** erstreckt, ebenfalls Teil der sekundären Kühlschleife **108** sein, wobei sie jedoch als Reihenverteiler bezeichnet wird. Die Verschlauchung **114** tritt in die Gestelle als Teil der sekundären Kühlschleife **108** ein, wobei sie jedoch als Kühlverteiler des Gestells bezeichnet wird. Ferner erstrecken sich die Reihenverteiler **116** zu allen Gestellen entlang einer Reihe in dem Rechenzentrum **100**. Die Verrohrung der sekundären Kühlschleife **108**, einschließlich der Verteiler **118**, **116** und **114**, kann durch mindestens eine Ausführungsform der vorliegenden Offenbarung verbessert werden. Eine optionale Kältemaschine **120** kann in der primären Kühlschleife innerhalb des Rechenzentrums **102** bereitgestellt werden, um Kühlung die vor dem Kühlturm zu unterstützen. Zu dem Ausmaß, in dem zusätzliche Schleifen in der primären Steuerschleife existieren, würde ein Fachmann beim Lesen der vorliegenden Offenbarung erkennen, dass die zusätzlichen Schleifen eine Kühlung extern zu dem Gestell und extern zu der sekundären Kühlschleife bereitstellen; und zusammen mit der primären Kühlschleife für diese Offenbarung genommen werden können.

[0011] In mindestens einer Ausführungsform kann im Betrieb Wärme, das innerhalb von Servereinschüben der Gestelle **110** erzeugt wird, an ein Kühlmittel übertragen werden, das die Gestelle **110** über eine biegsame Verschlauchung des Reihenverteilers **114** der zweiten Kühlschleife **108** verlässt. Wenn angemessen, bewegt sich das zweite Kühlmittel (in der sekundären Kühlschleife **108**) von der CDU **112** zum Kühlen der Gestelle **110** zu den Gestellen **110**. Das der CDU **112** zugeordnete zweite Kühlmittel läuft von einer Seite des Raumverteilers mit der Verschlauchung **118** zu einer Seite des Gestells **110** über den Reihenverteiler **116** und durch eine Seite des Servereinschubs über die Verschlauchung **114**. Das verbrauchte zweite Kühlmittel (oder austretendes zweites Kühlmittel, das die Wärme von den Rechenkomponenten trägt) tritt aus einer anderen Seite des Servereinschubs aus (z.B., tritt in die linke Seite des Gestells ein und tritt aus der rechten Seite des Gestells für den Servereinschub nach Laufen durch den Servereinschub oder durch Komponenten auf dem Servereinschub aus). Das verbrauchte zweite Kühlmittel, das den Servereinschub oder das Gestell **110** verlässt, kommt aus der anderen Seite (z.B., der Austrittsseite) der Verschlauchung **114** aus und bewegt sich zu einer parallelen, jedoch ebenfalls austretenden Seite des Reihenverteilers **116**. Von dem Reihenverteiler **116** bewegt sich das verbrauchte zweite Kühlmittel in einem parallelen Abschnitt des Raumverteilers **118** entgegengesetzt der Richtung des ankommenden zweiten Kühlmittels (das ebenfalls das erneuerte zweite Kühlmittel sein kann) und zu der CDU **112**.

[0012] In mindestens einer Ausführungsform tauscht das verbrauchte zweite Kühlmittel seine Wärme mit einem primären Kühlmittel in der primären Kühlschleife **106** über die CDU **112**. Das verbrauchte zweite Kühlmittel wird erneuert (z.B. relativ gekühlt im Vergleich zu der Temperatur bei der Stufe des verbrauchten zweiten Kühlmittels) und ist bereit, durch die zweite Kühlschleife **108** zu den Rechenkomponenten zurück zyklisiert zu werden. Verschiedene Strömungs- und Temperatursteuermerkmale in der CDU **112** ermöglichen die Steuerung der Wärme, die von dem verbrauchten zweiten Kühlmittel oder der Strömung des zweiten Kühlmittels in und aus der CDU **112** getauscht wird. Die CDU **112** ist ebenfalls imstande, eine Strömung des primären Kühlmittels in der primären Kühlschleife **106** zu steuern. Von daher ist es möglich, neue Gestelle oder Server hinzuzufügen oder inaktive Gestelle oder Server nach Bedarf zu aktivieren, um zusätzliche Rechenanforderungen gemäß Nachfrage bereitzustellen. In einem derartigen fixierten Rechenzentrum sind die Kühlanforderungen im Wesentlichen verfügbar, um der größten Nachfrage von dem Rechenzentrum zu entsprechen, weil es einen wesentlichen Bereich gibt, um die erforderlichen Kühlmerkmale hinzuzufügen, wenn das Rechenzentrum skaliert wird. Wenn die Nachfrage unbekannt ist oder physikalische Beschränkungen an dem Rechenzentrum geltend gemacht werden, kann das Kühlsystem jedoch nicht den wärmebegrenzenden Anforderungen des Rechenzentrums hinsichtlich der Nachfragen entsprechen, die auf das Rechenzentrum platziert werden.

[0013] Fig. 2 ist ein Blockdiagramm, das ein mobiles Rechenzentrumsystem oder Kühlsystem **200** veranschaulicht, das einen Behälter **202, 204** mit einem Behälterverteiler **210** aufweist, um die Zirkulation des einem mobilen Kühlturm **206** zugeordneten Kühlmittels zu ermöglichen, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform veranschaulicht Fig. 2 einen Querschnitt oder eine Schnittansicht des mobilen Rechenzentrumsystems. Der Behälter **202, 204** kann zusammen mit dem mobilen Kühlturm **206** und der Anhängerladefläche **220** ein mobiles Rechenzentrumsystem oder ein mobiles Rechenzentrumkühlsystem bilden. In mindestens einer Ausführungsform kann das mobile Rechenzentrumkühlsystem jedoch auf einer oder mehreren Anhängerladeflächen lokalisiert sein und kann mehr Behälter als den Behälter **202, 204** umfassen. In mindestens einer Ausführungsform ist der Kühlturm, wie beispielsweise der Kühlturm **234**, auf einem Dach des einen oder mehrerer Behälter auf der Anhängerladefläche lokalisiert. In mindestens einer Ausführungsform ist der Kühlturm luftgekühlt oder ist ein Rückkühler. In mindestens einer Ausführungsform können Abschnitte **202, 204** des Behälters vereinheitlicht werden, so dass die Gestelle **208A-N** und die CDU **230** physikalisch zueinander sichtbar sind. In mindestens einer Ausführungsform umfasst der Behälter **202, 204** einen Behälterverteiler **210**. Der Behälterverteiler **210** zirkuliert dem Kühlturm **206** zugeordnetes Kühlmittel zu einem oder mehreren flüssigkeitsgekühlten Gestellen **208A-N**. Die flüssigkeitsgekühlten Gestelle können hier auch einfach als Gestelle bezeichnet werden.

[0014] In mindestens einer Ausführungsform können die Gestelle **208A-N** innerhalb des Abschnitts **202** des Behälters **202, 204** sein und können Kühlmittel über Rohre empfangen, die dem Behälterverteiler **210** zugeordnet sind. Ein Beispiel des Rohrs ist als Einlassrohr **232** und Auslassrohr **234** für das Gestell **208B** veranschaulicht. Das Kühlmittel kann gepumpt, zirkuliert oder gefördert werden, um durch einen oder mehrere Strömungscontroller **212A-N** und/oder **214A-N** zu strömen. In mindestens einer Ausführungsform kann lediglich das Einlassrohr mit einem Strömungscontroller gekoppelt werden, kann lediglich das Auslassrohr mit einem Strömungscontroller gekoppelt werden oder sowohl das Einlassrohr als auch das Auslassrohr können mit dem Strömungscontroller gekoppelt werden, um das Kühlmittel den Gestellen **208A-N** bereitzustellen. In mindestens einer Ausführungsform ermöglicht der Behälterverteiler **210** die Fluidkopplung des Behälterverteilers **210** mit einem zweiten Behälterverteiler eines zweiten Behälters über einen oder mehrere Fluidkoppler **224A; 224B**. In mindestens einer Ausführungsform sind sämtliche Koppler, einschließlich der Koppler **224A, B**, nicht tropfende Koppler, die im Plug-and-Play-Format verwendet werden können, so dass Kühlsysteme skaliert werden können, um Kapazität ohne Verlust von Kühlmittel hinzuzufügen oder zu entfernen.

[0015] In mindestens einer Ausführungsform kann mindestens ein Merkmal oder eine Anforderung des Kühlturms **206** gemäß einer Kühlanforderung oder einem Kühlmerkmal des einen oder mehrerer Gestelle **208A-N** und einer physikalischen Beschränkung oder einem physikalischen Merkmal der Anhängerladefläche **220** bestimmt werden. In mindestens einer Ausführungsform ist das mindestens eine Merkmal oder eine Anforderung des Kühlturms **206** eines oder mehreres von einer Kapazität des dem Kühlturm **206** zugeordneten Kühlmittels, einer Kapazität der Kühlturmpumpen, die dem Kühlturm **206** zugeordnet oder innerhalb des Kühlturms ist, physikalische Abmessungen, die den Wärmetauscherspulen innerhalb des Kühlturms **206** zugeordnet sind, physikalischen Abmessungen, die dem Kühlturm zugeordnet sind, und eine Strömungsrate oder Strömungsraten und ein Strömungsvolumen oder Strömungsvolumen des durch den Kühlturm zirkulierenden Kühlmittels.

[0016] In mindestens einer Ausführungsform kann die Kühlanforderung oder das Kühlmerkmal eines sein von: (a) einer Temperatur, die zum optimalen Betrieb einer Komponente, eines Server oder eines Gestells der Gestelle **208A-N** erforderlich ist, (b) einem Unterschied oder einer Änderung in Temperaturen, die für die Komponente, den Server oder das Gestell der Gestelle **208A-N** erreicht werden soll, (c) einem Zeitraum, in dem eine Temperatur beibehalten werden kann, (d) einem Zeitraum, in dem die Änderung in Temperaturen erreicht werden kann, und (e) einer Betriebstemperatur für die Komponente, den Server oder das Gestell der Gestelle **208A-N**.

[0017] In mindestens einer Ausführungsform ist die physikalische Beschränkung oder das physikalische Merkmal der Anhängerladefläche **220** eines von: einem Raum, in Abmessungen, der innerhalb der Anhängerladefläche **220** dem Kühlturm verfügbar ist, einer Gewichtsbeschränkung der Anhängerladefläche **220**, einer Gewichtsbeschränkung einer oder mehrerer Federn **226A; 226B** der Anhängerladefläche **220**, einem Abstand zwischen dem Kühlturm an einem Ort auf der Anhängerladefläche **220** (oder einer unabhängigen Anhängerladefläche) und dem Behälter **202, 204** und irgendwelche zusätzlichen Abständen zwischen dem Kühlturm an dem Ort auf der Anhängerladefläche **220** (oder der unabhängigen Anhängerladefläche) und einer Wärmequelle in einer beabsichtigten Umgebung für das mobile Rechenzentrumsystem. In mindestens einer Ausführungsform ist der Kühlturm **206** auf seiner eigenen Anhängerladefläche, wie beispielsweise der oben referenzierten unabhängigen Anhängerladefläche. Die unabhängige Anhängerladefläche kann benachbart der Anhängerladefläche **220** lokalisiert sein. In mindestens einer Ausführungsform kann die unabhängige Anhängerladefläche

so nahe wie möglich an der Anhängerladefläche **220** lokalisiert sein, um die effiziente Kühlung der Gestelle **208A-N** innerhalb des Behälters **202**, **204** zu fördern. In mindestens einer Ausführungsform ermöglicht die Gewichtsbeschränkung eine Bestimmung der Menge eines oder mehrerer Kühlmittel, die in dem mobilen Kühlsystem für die Kühlanforderungen verfügbar sein können. Beispielsweise kann das Gewicht des Kühlmittels einschränken, wie viel Kühlung durch das mobile Kühlsystem ermöglicht werden kann. Ferner sind größere Längen- und Breitereinschränkungen erforderlich, um die Mobilität des mobilen Kühlsystems zu unterstützen. In mindestens einer Ausführungsform kann der Transport des mobilen Kühlsystems auf existierenden Autobahnen durch die Belastungsvariante von Standardtoleranzen (wie beispielsweise LKW-Transporttoleranzen durch das Department of Transportation) begrenzt sein. Die vorliegende Offenbarung rechnet mit diesen physikalischen Beschränkungen, während die Kühlanforderungen des prozessintensiven Rechensystems der Gestelle innerhalb des mobilen Rechenzentrums anerkannt werden.

[0018] In mindestens einer Ausführungsform kann das Kühlmittel in den Behälter **202**, **204** über die externen Kühlmittelrohre **236** zirkuliert werden. Die CDU **230** empfängt und verteilt das Kühlmittel oder ein sekundäres Kühlmittel über den Behälterverteiler **210**. In mindestens einer Ausführungsform ist das Kühlmittel, das dem Kühlturm **206** zugeordnet ist, in den externen Kühlrohren **236** ein primäres Kühlmittel. In mindestens einer Ausführungsform ist der Kühlturm **206** Teil einer primären Kühschleife. In mindestens einer Ausführungsform ist der Behälterverteiler **210** Teil einer sekundären Kühschleife. In mindestens einer Ausführungsform bildet die CDU **230** einen Austauschpunkt für die primäre und die sekundäre Kühschleife. In mindestens einer Ausführungsform kann das Kühlmittel der sekundären Kühschleife, das als das primäre Kühlmittel bezeichnet wird, das gleiche oder ein ähnliches Kühlmittel als ein sekundäres Kühlmittel der sekundären Kühschleife sein. In mindestens einer Ausführungsform verlässt das sekundäre Kühlmittel nicht den Behälter **202**, **204**. Stattdessen wird das sekundäre Kühlmittel durch (oder Austauschen von Wärme mit) das primäre Kühlmittel gekühlt. Das primäre Kühlmittel verlässt die CDU **230** und den Behälter **202**, **204** über die externen Kühlmittelrohre **236**, um durch den Kühlturm **206** gekühlt zu werden. In mindestens einer Ausführungsform ist der Kühlturm **206** ein Luftkühlsystem, um Wärme aus dem primären Kühlmittel zu extrahieren, das durch ein Gebläse-gekühltes, zickzack-förmiges Rohr läuft, das sich durch den Kühlturm **206** fortbewegt.

[0019] In mindestens einer Ausführungsform ermöglicht die CDU **230** eine Zirkulation des sekundären Kühlmittels, in dem es aus der CDU-Austrittsrohrleitung **216**, durch den Behälterverteiler **210** und zurück in die CDU **230** über die Eintrittsrohrleitung **218** geleitet wird. In mindestens einer Ausführungsform kann einer der Strömungscontroller **214A-N** der Gestelleintritts-Rohrleitungen aktiviert werden, um die Zirkulation des sekundären Kühlmittels in ein jeweiliges eines der Gestelle **208A-N** zu modifizieren. In mindestens einer Ausführungsform kann einer der Strömungscontroller **212A-N** auf den Gestellaustritts-Rohrleitungen aktiviert werden, um die Zirkulation des sekundären Kühlmittels aus dem jeweiligen einen der Gestelle **208A-N** zu modifizieren.

[0020] In mindestens einer Ausführungsform kann eine Anforderung für den schnellen Einsatz von Supercomputing- oder rechenintensiven Ressourcen gemacht werden. Die Anforderung kann Rechenmerkmale umfassen, wie beispielsweise die Rechengeschwindigkeit, Schaltgeschwindigkeiten, Datenübertragungsgeschwindigkeiten usw. Von der Anforderung werden Konfigurationsinformationen für einen oder mehrere Server innerhalb eines oder mehrerer Gestelle, wie beispielsweise den Gestellen **208A-N** des mobilen Rechenzentrumsystems **200**, bestimmt. In mindestens einer Ausführungsform können ein oder mehrere Behälter **202**, **204** in der einen oder mehreren Anhängerladeflächen als geeignet bestimmt werden, in das eine oder mehrere Gestelle zu passen, die für die Anforderung geplant werden. In mindestens einer Ausführungsform können optimale Temperaturen einer oder mehrerer Komponenten, die den Gestellen zugeordnet sind, die als Reaktion auf die Anforderung geplant werden, bestimmt werden. In mindestens einer Ausführungsform wird ein Bereich von Temperaturen bestimmt, der die optimalen Temperaturen der einen oder mehreren Komponenten umfassen kann. In mindestens einer Ausführungsform kann ein Mittel oder ein anderer statistischer Temperaturwert aus den Temperaturen oder den Temperaturbereichen bestimmt werden.

[0021] In mindestens einer Ausführungsform wird basierend teilweise auf dem statistischen Temperaturwert, den Temperaturen oder den bestimmten Temperaturbereichen eine Kühlanforderung oder ein Kühlmerkmal bestimmt. In mindestens einer Ausführungsform kann die Kühlanforderung oder das Kühlmerkmal die Änderung in der Temperatur sein, die erforderlich ist, um den optimalen Betrieb des einen oder mehrerer Komponenten in dem einen oder mehreren Gestellen aufrechtzuerhalten, die für die Anforderung bestimmt sind. Alternativ oder mit der Änderung in der Temperatur kann eine Geschwindigkeit der Änderung in der Temperatur eine Kühlanforderung oder ein Kühlmerkmal sein. In mindestens einer Ausführungsform kann eine Kühlanforderung oder Merkmal ein Zeitraum sein, für den eine Temperatur beibehalten werden muss. Diese und andere Kühlanforderungen oder Merkmale können aus dem statistischen Temperaturwert, den Temperaturen oder den Temperaturbereichen bestimmt werden.

[0022] Ferner wird in mindestens einer Ausführungsform eine physikalische Beschränkung oder ein physikalisches Merkmal bestimmt, die/das die Anhängerladefläche zugeordnet ist, die den einen oder mehrere Behälter **202**, **204** hostet. In mindestens einer Ausführungsform kann es basierend teilweise auf der Kühlanforderung oder dem Kühlmerkmal und basierend teilweise auf dem Raum, der von dem einen oder mehreren Gestellen belegt wird, der für die Anforderung geplant wird, einen Restraum auf einer Anhängerladefläche geben, um den Kühlturm zu hosten. Wenn die Kühlanforderungen oder die Merkmale jedoch anspruchsvoll sind (z.B., niedrige Kühltemperatur während einer heißen Umgebungstemperatur und der Anforderung, die niedrige Kühltemperatur für einen verlängerten Zeitraum beizubehalten), dann kann mehr als ein Kühlturm oder ein größerer Kühlturm erforderlich sein. Von daher können in mindestens einer Ausführungsform ein oder mehrere Kühltürme (wie beispielsweise der Kühlturm **206** und zusätzliche Kühltürme auf anderen Anhängerladeflächen) bewertet werden, um der Anforderung zu entsprechen.

[0023] In mindestens einer Ausführungsform ist mindestens ein dritter Behälter, eine Anhängerladefläche oder der Behälter **202**, **204**, der die Gestelle **208A-N** hostet, angepasst, um den Kühlturm (veranschaulicht als Kühlturm **206**) zu hosten. Der Kühlturm **206** ist angepasst, um die Kühlanforderungen oder Merkmale zu erfüllen, für die ein oder mehrere Gestelle **208A-N** bestimmt sind. Der Kühlturm **206** ist ebenfalls für das mindestens eine physikalische Merkmal des mindestens einen dritten Behälters, der Anhängerladefläche oder dem Behälter **206** angepasst, der die Gestelle **208A-N** hostet. In mindestens einer Ausführungsform muss der Kühlturm **206** in der Lage sein, die erforderliche Kühlung bereitzustellen, um dadurch die Kühlanforderungen des einen oder mehrerer Gestelle **208A-N** zu erfüllen, sowie auch in die physikalischen Merkmale zu passen, die in dem mindestens einen dritten Behälter, der Anhängerladefläche oder dem Behälter **202**, **204**, der die Gestelle **208A-N** hostet, verfügbar sind. In mindestens einer Ausführungsform werden, wenn mehrere Kühltürme erforderlich sind und weil die mehreren Kühltürme eine singuläre Kühlfunktion durchführen - zum Kühlen des einen oder mehrerer Gestelle **208A-N**, werden die mehreren Kühltürme als ein Kühlturm bezeichnet.

[0024] In mindestens einer Ausführungsform umfasst das mobile Rechenzentrumkühlsystem, wie beispielsweise das System **200**, den mindestens eine primäre Kühlschleife, der dem Kühlturm **206** zugeordnet ist, und der mindestens einen sekundären Kühlschleife, die dem Behälterverteiler **210** zugeordnet ist. Die mindestens eine Kühlverteilungseinheit (Cooling Distribution Unit; CDU) **230** ist dem mindestens einen Behälter **202**, **204** an dem Abschnitt **204** zugeordnet oder befindet sich in diesem, der am nächsten oder benachbart zu den Gestellen **208A-N** ist, um Wärme zwischen der primären Kühlschleife und der sekundären Kühlschleife zu tauschen.

[0025] In mindestens einer Ausführungsform erstrecken sich Fluidkoppler **224A**; **224B** von dem Behälterverteiler oder dem Behälter. Die Fluidkoppler **224A**; **224B** sind ein Paar von Kopplern, die als Einlass und Auslass von dem Behälterverteiler **210** zu einem benachbarten Behälterverteiler, wie beispielsweise einem zweiten Behälterverteiler eines benachbarten Behälters, dienen. In mindestens einer Ausführungsform sind die Fluidkoppler **224A**; **224B** angepasst, um eine Fluidkopplung zwischen dem Behälterverteiler und dem zweiten Behälterverteiler bereitzustellen. In mindestens einer Ausführungsform sind der Behälter, der den Behälterverteiler **210** hostet, und ein zweiter Behälter, der den zweiten Behälterverteiler hostet, auf den gleichen oder unterschiedlichen Anhängerladeflächen. In mindestens einer Ausführungsform kann einer der Fluidkoppler **224A**; **224B** verwendet werden, um Kühlmittel, das dem Behälterverteiler **210** zugeordnet ist, zu dem zweiten Behälterverteiler, zu einer zweiten CDU in dem zweiten Behälter und zurück zu der CDU **230** des Behälters **202**, **204** zu leiten. Auf diese Art und Weise kann die zweite CDU, die mit ihrem eigenen Kühlturm gekoppelt ist, eine zusätzliche Kühlung des Kühlmittels bereitstellen, bevor das Kühlmittel zurück zu der CDU **230** des Behälters **202**, **204** kommt.

[0026] In mindestens einer Ausführungsform umfasst das mobile Rechenzentrumkühlsystem **200** mindestens eine Anhängerladefläche **220** mit Ständern **222** und einer Anhängerkupplung **228**. Die Ständer können bereitgestellt werden, um die Anhängerladefläche **220** stabil zu halten, wobei jedoch Räder, die mit Kreide angemerkt sind, um Bewegung zu verhindern, ebenfalls implementiert sein können. In mindestens einer Ausführungsform können die Räder mit angelegten Bremsen und mit hydraulisch verlängerten Ständern **222**, um den Boden zu kontaktieren, gesichert werden. Auf diese Art und Weise kann Stabilität erreicht werden, während das Rechenzentrum in Betrieb ist. In mindestens einer Ausführungsform kann mindestens eine Feder (mit regulären Schraubenfedern **226A** und/oder Blattfedern **226B** veranschaulicht) verwendet werden, um Aufprallschutz für die Gestelle, die CDU und den Kühlturm während des Transports des mobilen Rechenzentrationssystems **200** bereitzustellen. In mindestens einer Ausführungsform können Luftbälge, die Luftfederung bereitstellen, zusammen mit oder anstatt der mechanischen Federn verwendet werden, um den gleichen Aufprallschutz zu erreichen.

[0027] In mindestens einer Ausführungsform umfasst das mobiles Rechenzentrumkühlsystem **200** ein lernendes Subsystem mit mindestens einem Prozessor und einem Speicher mit Anweisungen zur Ausführung auf dem mindestens einen Prozessor, um mindestens einem Prozessor zu ermöglichen, Funktionen durchzuführen. In mindestens einer Ausführungsform kann mindestens ein Prozessor Teil eines Rechenzentrum-Managementsystems (DMS) zur Verwaltung von Rechenzentrumsfunktionen sein. In mindestens einer Ausführungsform ist der mindestens eine Prozessor angepasst bzw. wird veranlasst, Temperaturanforderungen von einem oder mehreren zweiten flüssigkeitsgekühlten Gestellen zu bewerten. Das eine oder mehrere zweite flüssigkeitsgekühlte Gestelle können von vorherigen oder von Testkonfigurationen für Komponenten innerhalb Servereinschüben des einen oder mehreren zweiten flüssigkeitsgekühlten Gestellen sein. Ferner ist der mindestens eine Prozessor angepasst bzw. wird veranlasst, Strömungsraten oder Strömungsvolumen des Kühlmittels zu bewerten, das für einen vorliegenden Vorgang bestimmt ist, basierend teilweise auf ihren Zuordnungen zu den Temperaturanforderungen von der zuvor oder Testkonfigurationen.

[0028] In mindestens einer Ausführungsform ist der mindestens eine Prozessor ebenfalls angepasst bzw. wird veranlasst, um mindestens eine physikalische Beschränkung des Behälters, einer Anhängerladefläche oder eines dritten Behälters zu bewerten, der den Kühlturm hostet, wie in anderen Ausführungsformen erläutert. In mindestens einer Ausführungsform sind die Ergebnisse der Bewertungen eine Ausgabe, die von dem mindestens einen Prozessor bereitgestellt wird. Die Ausgabe ist zum Erleichtern der Zirkulation des Kühlmittels zu dem einen oder mehreren Gestellen des vorliegenden Vorgangs. Tatsächlich wird das Lernen von einem Test oder einem zuvor verwendeten Fall verwendet, um für Wärme, die von einer Kundenanforderung zugeordneten Gestellkonfiguration erzeugt wird, eine entsprechende Kühlturmanforderung zu kennzeichnen. In mindestens einer Ausführungsform ist die Ausgabe mindestens einer Kühlturmanforderung zugeordnet, wobei sie jedoch ebenfalls mindestens einer physikalischen Beschränkung des Behälters, der Anhängerladefläche oder dem dritten Behälter zugeordnet ist. Auf diese Art und Weise ist es möglich, einen schnellen Einsatz eines mobilen Rechenzentrumsystems mit entsprechender Kühlung bereitzustellen, um Forderungen des mobilen Rechenzentrumsystems zu erfüllen.

[0029] In mindestens einer Ausführungsform umfasst das mobile Rechenzentrumkühlsystem **200** den einen oder mehrere Strömungscontroller **212A-N**; **214A-N**, um das Kühlmittel durch den Behälterverteiler **210** und das eine oder mehrere flüssigkeitsgekühlte Gestelle **208A-N** zu zirkulieren. Das lernende Subsystem des mindestens einen Prozessors und des Speichers, das Anweisungen umfasst, kann ein Maschinenlernmodell ausführen, um die oben referenzierten Bewertungen durchzuführen. Als Ergebnis verarbeitet das Maschinenlernmodell Temperaturen, die Temperaturanforderungen zugeordnet sind, unter Verwendung mehrerer Neuronenebenen des Maschinenlernmodells mit den Temperaturen und mit zuvor zugeordneten Strömungsraten oder Strömungsvolumen für das Kühlmittel. Das neuronale Netzwerk und andere Maschinenlernmodellmerkmale für das lernende Subsystem werden hier an anderer Stelle mindestens mit Bezug auf **Fig. 14** und **Fig. 15** erläutert.

[0030] Das Maschinenlernmodell stellt die Ausgabe, in mindestens einer Ausführungsform, die einer Strömungsrate oder einem Strömungsvolumen für das Kühlmittel zugeordnet ist, dem einen oder mehreren Strömungscontrollern zur Verfügung. Die Ausgabe ist von einer Bewertung der zuvor zugeordneten Strömungsraten oder Strömungsvolumen und der mindestens einen physikalischen Beschränkung des Behälters, der Anhängerladefläche oder des dritten Behälters. Das Maschinenlernmodell ist in der Lage, Anforderungen oder Merkmale des Kühlturms, wie beispielsweise eine vorhergesagte Strömungsrate oder ein vorhergesagtes Strömungsvolumen und Größenanforderungen des Kühlturms, vorherzusagen. Mit Bezug auf die Größen- oder Abstandsanforderung kann die Ausgabe in mindestens einer Ausführungsform angeben, imstande zu sein, neben einen Behälter zu passen, der Gestelle hostet, der die Kühlung erfordert, lediglich imstande zu sein, auf seine eigene Anhängerladefläche zu passen, oder imstande zu sein, in einen Raum neben einem unterschiedlichen Behälter mit weniger Gestellen auf einer unterschiedlichen Anhängerladefläche zu passen, jedoch imstande zu sein, die Kühlung weniger Gestelle und der Gestelle zu handhaben, die dazu gedacht sind, gekühlt zu werden. In mindestens einer Ausführungsform gibt die Strömungsrate oder das Strömungsvolumen die Größe des Kühlturms und seine Kapazität zum Halten und Zirkulieren des Kühlmittels an.

[0031] In mindestens einer Ausführungsform sind kalte Platten, die Servereinschüben des einen oder mehreren Gestellen zugeordnet sind, angepasst, um dem Behälterverteiler zugeordnetes Kühlmittel zu empfangen. In mindestens einer Ausführungsform sind die kalten Platten mit einer oder mehreren prozessintensiven Komponenten gekoppelt, wie beispielsweise Speicherkomponenten, Zentralverarbeitungseinheiten (CPUs), Graphikverarbeitungseinheiten (GPUs) und Schaltern. Die kalten Platten umfassen Flüssigkeitskanäle und nicht tropfende Koppler, um ein sekundäres Kühlmittel von mindestens einer dem Behälterverteiler zugeordneten CDU zu empfangen. Von daher sind die prozessintensiven Komponenten innerhalb des einen oder mehreren Behälter flüssigkeitsgekühlt. In mindestens einer Ausführungsform ist die sekundäre Kühlturmschleife den kalten

Platten zugeordnet, um ein System darzustellen, das 0% bis 80% Flüssigkeitskühlung aufweist, wobei die primäre Kühlschleife Gebläse- oder Luft-basierter Kühlung zugeordnet wird, die etwa 40% bis 20% des Systems darstellt. In mindestens einer Ausführungsform wird 100% Eintauchkühlen in der vorliegenden Offenbarung ermöglicht, wobei die prozessintensiven Komponenten vollständig in dem sekundären Kühlmittel eingetaucht sind. Weil es für das sekundäre Kühlmittel erforderlich sein kann, die Komponenten und andere Elektronik zu kontaktieren, ist es ein Dielektrikum-reiches Kühlmittel im Vergleich mit dem primären Kühlmittel, das in mindestens einer Ausführungsform Wasser sein kann.

[0032] Fig. 3 ist eine topologische Veranschaulichung von Behältern oder Pods **302A-D** eines mobilen Rechenzentrumsystems oder Kühlsystems **300**, die in einer Anordnung konfiguriert sind, die Zirkulation von Kühlmittel ermöglicht, die einem mobilen Kühlturm für die Behälter oder Pods zugeordnet ist, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist das mobile Rechenzentrumkühlsystem **300** mit mehreren Behältern oder Pods **302A-D** mit Gestellen **306A-D** konfiguriert. In mindestens einer Ausführungsform werden die Gestelle **306A-D** über ein Kühlmittel gekühlt, das von einem jeweiligen der Behälterverteiler **318A-D** über einen jeweiligen von Reihenverteilern **304A-D** bereitgestellt wird. In mindestens einer Ausführungsform erstreckt sich jeder der Behälterverteiler **318A-D** durch den Umfang seines jeweiligen Behälters **302A-D**. Von daher können die Reihenverteiler **304A-D** in mindestens einer Ausführungsform nicht erforderlich sein. In mindestens einer Ausführungsform hosten die Behälterverteiler **318A-D** jeweilige CDUs oder sind mit einer jeweiligen CDU **310A, B** eines jeweiligen einen oder mehreren Behältern **302A-D** gekoppelt. In mindestens einer Ausführungsform kann sich eine CDU **310A** in einem Kühlmittel befinden und dieses verteilen, das einem oder mehreren Behältern **302A-D** zugeordnet ist, oder jeder des einen oder mehreren Behältern **302A-D** kann seine eigene CDU aufweisen.

[0033] In mindestens einer Ausführungsform, wie beispielsweise in Fig. 3, kann jeder Behälter **302A-D** angepasst sein, um eine primäre Kühlschleife über eine jeweilige externe Rohrleitung **312A; B** von einer jeweiligen einer oder mehreren CDUs **310A, B** zu empfangen, die ihrerseits mit einer sekundären Kühlschleife gekoppelt sind, die einem oder mehreren externen Kühltürmen, wie beispielsweise dem Kühlturm **320**, zugeordnet ist. Während der Kühlturm **320** von Fig. 3 als ein singuläre Turm veranschaulicht ist, kann jede CDU **310A; 310B** ihrem eigenen Kühlturm zugeordnet sein. Die primäre Kühlschleife kann unterschiedliche Konfigurationen aufweisen, wie beispielsweise, von einer CDU **310A** zu jedem Satz von Gestellen **306B, 306A**; oder von einer CDU **310A** zu dem Gestell **306B**, wobei eine separate CDU mit dem Gestell **306A** gekoppelt ist; oder von der CDU **310A** zu Gestellen **306A-D**, wobei die zusätzliche CDU **310B** für Redundanz im Fall eines Ausfalls einer primärer CDU **310A** konfiguriert ist. Die primäre Kühlschleife kann zusätzliche Unterschleifen umfassen, wie beispielsweise Schleifen mit dem Reihenverteiler **304A-D**, wobei sich die primäre Kühlschleife jedoch vollständig zu den Reihenverteilern durch Bereitstellen eines singulären Kühlmittels durch die Behälterverteiler **318A-D** und die Reihenverteiler **304A-D** erstrecken kann. In mindestens einer Ausführungsform ist die CDU nahe zu (zum Beispiel in einem abgeschlossenen Behälterbereich **308A**) und oder innerhalb eines oder mehreren der Behälter **302A-D** lokalisiert. Der Kühlturm **320** kann extern zu dem Behälter oder in abgeschlossenen Bereichen lokalisiert sein. In mindestens einer Ausführungsform ist, ebenfalls wie in dem Beispiel von Fig. 2 veranschaulicht, der Kühlturm **320** oben auf einem oder mehreren Behälter lokalisiert. In mindestens einer Ausführungsform können eine oder mehrere CDUs vollständig in einem Behälter zugewiesen sein, um Gestelle von mehreren Behältern zu unterstützen. Die CDUs können mit mindestens einem bemessenen Kühlturm gekoppelt sein, der oben auf dem Behälter lokalisiert ist. Die Kombination des Kühlturms oben auf dem Behälter und der CDUs innerhalb des Behälters bilden ein mobiles Kühlsystem, das mit benachbarten Trailern zusammen lokalisiert werden kann, die lediglich Behälter mit Gestellen aufweisen.

[0034] In mindestens einer Ausführungsform veranschaulicht Fig. 3 mehrere Anhängerladeflächen, die benachbart zueinander lokalisiert sind, um eine Kopplung und eine gemeinsame Nutzung von Kühlressourcen zu ermöglichen. In mindestens einer Ausführungsform kann eine einzige Anhängerladefläche mit Kapazität, mehrere Behälter oder Pods sowie auch die Kühltürme zu handhaben, in der Konfiguration von Fig. 3 veranschaulicht werden. In beiden Fällen nutzen die Behälter oder Pods **302A-D** mindestens Kühlressourcen von mindestens zwei CDUs **310A; B** gemeinsam. In mindestens einer Ausführungsform terminiert jede primäre Kühlschleife von dem Kühlturm **320** bei einer CDU **310A; B** innerhalb seines eigenen Behälters oder einer anderen Einschließung **308A** oder in einem benachbarten Behälter **302A; D**. Eine sekundäre Kühlschleife erstreckt sich durch die Behälterverteiler **318A-D**, wobei jeder Behälterverteiler **318A, B, C, D** Kühlmittel direkt den Gestellen **306A, B, C, D** oder indirekt durch Reihenverteiler **304A, B, C, D** bereitstellt. Das Kühlmittel, das den Behältern in der sekundären Kühlschleife zugeordnet ist, zirkuliert durch beide der CDUs der jeweiligen Behälter **302B; D**, während die Behälter **302A; C** keine CDUs aufweisen können. Alternativ kann in mindestens einer Ausführungsform die sekundäre Kühlschleife zusätzliche CDUs in Behältern **302A; C** umfassen, wobei das Kühlmittel der CDU von Behältern **302B; D** zugeordnet ist, die ein den CDUs zugeordneten Kühlmittel in

Behältern **302A; C** kühlen. Die Behälterverteiler **318A-D** können zueinander oder benachbarten der Behälterverteiler über Fluidkoppler **316** gekoppelt sein, die zwischen den Behälterverteilern **318A; C** veranschaulicht sind.

[0035] In mindestens einer Ausführungsform terminiert die primäre Kühlschleife in einer jeweiligen CDU eines jeweiligen Behälters **302A-D**. In mindestens einer Ausführungsform erstreckt sich die primäre Kühlschleife von dem Kühlturm **320** zu der CDU **310A** in oder benachbart zu dem Behälter **302B**, wobei ein teilender Koppler dem Kühlmittel der sekundären Kühlschleife ermöglicht, sich zu einer CDU in dem Behälter **302A** zu erstrecken. In mindestens einer Ausführungsform ermöglichen weitere Rohrleitungen des Behälterverteilers **318A-D** der primären Kühlschleife eines Kühlturms, sich durch die CDUs jedes Behälters **302A-D**, wie beispielsweise der CDU **310B**, zu erstrecken. Dann bilden die Reihenverteiler **304A-D** die sekundäre Kühlschleife in diesem Fall. Separat oder gleichläufig kann in mindestens dieser oder einer Ausführungsform die CDU **310B** verwendet werden, um die Kühlanforderungen der Behälter **302A-D** über die Behälterverteiler **318A-D** anzusprechen, die eine sekundäre Kühlschleife mit den Reihenverteilern **304A-D** bilden, während der Kühlturm **320** in einer primären Kühlschleife mit der CDU **310B** ist. Dies ermöglicht Redundanzpläne in dem Fall, dass einer der Kühltürme **310A; B** ausfällt.

[0036] In mindestens einer Ausführungsform wird die Konfiguration von **Fig. 3** durch mindestens einen Prozessor aktiviert, der mindestens eine Logikeinheit mit trainierten neuronalen Netzwerken oder anderen Maschinenlernmodellen umfassen kann, die trainiert ist, ein Kühlsystem zu bestimmen, um den Temperaturanforderungen eines mobilen Rechenzentrums, das als Reaktion auf eine Kundenanforderung eingesetzt wird, zu entsprechen. Das neuronale Netzwerk oder andere Maschinenlernmodelle können anfangs trainiert werden und können ferner auf einer fortlaufenden Grundlage trainiert werden, um ihre Genauigkeit zu verbessern. Der mindestens eine Prozessor kann bereitgestellt werden, um ein oder mehrere neuronale Netzwerke mit verborgenen Schichten von Neuronen zum Bewerten von Temperaturanforderungen eines oder mehrerer flüssigkeitsgekühlten Gestellen **306A-D** zu trainieren, die in einem jeweiligen Behälter zu hosten sind. Der mindestens eine Prozessor wird ferner an anderer Stelle in dieser Offenbarung mit Bezugnahme auf mindestens die Prozessoren und Trainingsschemata für neuronale Netzwerk in **Fig. 14** und **Fig. 15** beschrieben. Das Trainieren kann ebenfalls verwendet werden, um Strömungsraten oder Strömungsvolumen eines Kühlmittels basierend teilweise auf ihren Zuordnungen zu den Temperaturanforderungen zu bewerten.

[0037] In mindestens einer Ausführungsform ist das Kühlmittel das Kühlmittel in der primären Kühlschleife, wobei die Bewertung von Strömungsraten oder Strömungsvolumen jedoch ebenfalls auf das Kühlmittel in der sekundären Kühlschleife anwendbar sein kann. Das Training kann ebenfalls verwendet werden, um mindestens eine physikalische Beschränkung des Behälters, einer Anhängerladefläche oder eines zweiten Behälters zu bewerten, der einen oder mehrere Kühltürme **320** und die eine oder mehrere CDUs **310A; B** hostet, verwendet werden, um das eine oder mehrere flüssigkeitsgekühlte Gestelle **306A-D** zu kühlen. Eine Ausgabe des mindestens einen Prozessors, welches das trainierte oder mehrere neuronale Netzwerke aufweist, ist zum Erleichtern der Zirkulation des Kühlmittels in einem oder mehreren der primären und der sekundären Kühlschleifen. Von daher ist in mindestens einer Ausführungsform die Ausgabe mindestens einer Kühlturmanforderung zugeordnet, wie beispielsweise der Kapazität, der Strömungsrate oder dem Strömungsvolumen des für die Gestelle **306A-D** verfügbaren Kühlmittel. In mindestens einer Ausführungsform ist die Ausgabe ebenfalls der mindestens einen physikalischen Beschränkung des Behälters, der Anhängerladefläche oder des zweiten Behälters zugeordnet, wie beispielsweise Größenbeschränkungen des Behälters, der Anhängerladefläche oder des zweiten Behälters.

[0038] In mindestens einer Ausführungsform kann der mindestens eine Prozessor unter Verwendung vorheriger oder Testwerte für die Temperaturanforderungen, die Strömungsraten oder Strömungsvolumen und der mindestens einen physikalischen Beschränkung trainiert werden. Wenn ein trainiertes neuronales Netzwerk dann auf mindestens eine Temperaturanforderung angewandt wird, die den Gestellen **306A-D** in Behältern **302A-D** zugeordnet ist, kann eine Ausgabe der erforderlichen Strömungsraten oder Strömungsvolumen des Kühlmittels bereitgestellt werden, um der Temperaturanforderung zu entsprechen, und die Ausgabe kann ebenfalls für eine Größe eines derartigen Kühlturms bereitgestellt werden, die der physikalischen Beschränkung zugeordnet ist. In mindestens einer Ausführungsform ist die Ausgabe eine Extrapolation der vorherigen oder bereitgestellten Testwerte, wobei das neuronale Netzwerk einen Fehler beim Erreichen der Kühlung bestimmt, welche die geringste bei der durch die Ausgabe bestimmten Konfiguration ist. Diese Ausgabe kann verwendet werden, um die Konfiguration des mobilen Rechenzentrumsystems **300** zu bestimmen, wie beispielsweise, um lediglich einen Kühlturm **320** bereitzustellen, der bestimmt wird, ausreichend zu sein, um die Kühlung mit mindestens zwei CDUs **310A; B** zu koordinieren. Der Kühlturm **320** und die CDUs **310A; B** sind von spezifischen Größen, weil bestimmt ist, dass der Kühlturm **320** und die beiden CDUs **310A; B** ausreichend sind,

um die Temperaturanforderungen von Gestellen **306A-D** zu erfüllen, und weil der Raum, der neben den jeweiligen Behältern **302B; D** (für die CDUs) und neben oder auf den Behältern (für den Kühlturm) verfügbar ist, die physikalischen Beschränkung sogar erfüllt, wenn kein Raum benachbart den Behältern **302A; C** verfügbar ist.

[0039] In mindestens einer Ausführungsform umfasst das mobile Rechenzentrumsystem oder das mobile Rechenzentrumkühlsystem **300** den mindestens einen Prozessor in seinen Gestellen **306A-D** zum kontinuierlichen Verbessern des neuronalen Netzwerks basierend teilweise auf aktueller Information, die von dem eingesetzten mobilen Rechenzentrumkühlsystem **300** verfügbar sind. Der mindestens eine Prozessor wird zum Bewerten der Temperaturanforderungen des einen oder mehrerer flüssigkeitsgekühlter Gestelle und der Strömungsraten oder Strömungsvolumen des Kühlmittels und zum Bereitstellen der Ausgabe mit einer Zuordnung zu mindestens einer Temperatur verwendet, die von dem einen oder mehreren flüssigkeitsgekühlten Gestellen durch die Zirkulation des Kühlmittels in mindestens der primären Kühlterschleife erreichbar ist, jedoch ebenfalls auf ein Kühlmittel der sekundären Kühlterschleife angewandt werden kann.

[0040] In mindestens einer Ausführungsform stellt der mindestens eine Prozessor des mobilen Rechenzentrumkühlsystems **300** eine Anweisungsausgabe zum Kommunizieren einer Ausgabe der mindestens einen Logikeinheit mit dem einen oder mehreren Strömungscontrollern bereit, um die Zirkulation des Kühlmittels innerhalb eines Behälterverteilers oder von dem Behälterverteiler zu einem zweiten Behälterverteiler der anderen fluidisch verbundenen Behälter zu erleichtern, welche die anderen flüssigkeitsgekühlten Gestelle (z.B., Behälter **302A, C**) hosten. In mindestens einer Ausführungsform des mobilen Rechenzentrumkühlsystems **300** ist der mindestens eine Prozessor angepasst, um einen Temperaturwert eines Temperatursensors innerhalb eines oder mehrerer der Behälter **302A-D** zu empfangen, und ist angepasst, um die Zirkulation des Kühlmittels in den primären oder den sekundären Kühlterschleifen zu erleichtern, um das eine oder mehrere flüssigkeitsgekühlte Gestelle zu kühlen.

[0041] **Fig. 4** ist eine Veranschaulichung eines schnell eingesetzten mobilen Rechenzentrums **400**, das ein mobiles Rechenzentrumsystem oder Kühlsystem gemäß mindestens einer Ausführungsform aufweist. **Fig. 4** veranschaulicht in mindestens einer Ausführungsform Lastwagen **406, 412** und ihre angekuppelten Anhängerladeflächen **414, 416**. Jede Anhängerladefläche kann einen oder mehreren Behälter **404B** für die Gestelle und einen oder mehrere Abschnitte **404A** für die CDUs und die extern angebrachten (oder lokalisierten) Kühltürme **418** aufweisen. In mindestens einer Ausführungsform werden die Fluidrohrleitungen **410** zusammen mit Behälterverteilern über Anhängerladeflächen **414, 416** zugeordnet. In mindestens einer Ausführungsform kann der Lastwagen von der Anhängerladefläche entkuppelt werden, die freistehend bleibt, wie durch die Anhängerladefläche **402** benachbart zu den Anhängerladeflächen **414, 416** veranschaulicht. Die freistehende Anhängerladefläche **402** weist ebenfalls den Behälter **402B** der Gestelle und einen Abschnitt **402A** für seine CDU und den extern angebrachten (oder lokalisierten) Kühlturm **418** auf, der in mindestens einer Ausführungsform bereitgestellt werden kann und der nicht bereitgestellt werden kann, wenn die CDU in mindestens einer anderen Ausführungsform von der Anhängerladefläche **402** Kühlmittel empfängt und verteilt, das einem Kühlturm einer benachbarten Anhängerladefläche zugeordnet ist. Die Konfiguration ermöglicht ein Skalieren des Rechenzentrums mit zusätzlichen Gestellen oder mit zusätzlicher Kühlung, wie durch die zusätzlichen Gestellen gefordert. Leistung und andere Infrastruktur kann einem Generator bereitgestellt werden, der benachbart zu den CDU-Abschnitten in jeder Anhängerladefläche lokalisiert sein kann. Datenleitungen **408** können sich von den Behältern erstrecken, um eine externe Kommunikation mit dem schnell eingesetzten Rechenzentrum zu ermöglichen. Drahtlose Kommunikation kann jedoch ebenfalls verwendet werden, um entweder Operationen oder Arbeitslastanforderungen mit dem Rechenzentrum **400** zu kommunizieren.

[0042] In mindestens einer Ausführungsform kann Kühlmittel der sekundären Kühlterschleife von einem oder mehreren der drei veranschaulichten Kühltürme **418** mit einer oder mehreren der CDUs über die Behälterverteiler und die Fluidrohrleitungen **410** gemeinsam genutzt werden. Von daher kann der Behälterverteiler Teil der sekundären Kühlterschleife sein, und interne Reihenverteiler, die das sekundäre Kühlmittel an die Gestelle kommunizieren, sind Teil der sekundären Kühlterschleife. In mindestens einer Ausführungsform weist jede Anhängerladefläche ihre eigene CDU auf, wobei jedoch lediglich ein Kühlturm (von den drei Kühltürmen **418**) zum Bereitstellen eines primären Kühlmittels an jede der CDUs über die Fluidrohrleitungen **410** aktiviert ist. Von daher können in mindestens einer Ausführungsform die verbleibenden deaktivierten Kühltürme redundante Kühltürme sein, die online zu bringen sind, wenn ein Ausfall eines aktivierten Kühlturms auftritt. Auf ähnliche Weise ist in mindestens einer Ausführungsform eine der CDUs auf einer Anhängerladefläche eine aktivierte CDU, die angepasst ist, um sekundäres Kühlmittel an die Gestelle sämtlicher Anhängerladeflächen zu kommunizieren, wobei die anderen CDUs deaktiviert sind und lediglich in einer redundanten oder Reserve-Konfiguration aktiviert werden. Die Behälterverteiler können darin getrennte Kanalisierungs- oder Rohrleitungen aufweisen, um dem Kühlturm zugeordnetes Kühlmittel anderen Behälterverteilern über Fluidrohrleitungen oder

der Kopplung **410** zu kommunizieren. In mindestens einer Ausführungsform können einige der Kanalisierungs- oder Rohrleitungen für die CDU verwendet werden, um sekundäres Kühlmittel an andere CDUs zu kommunizieren, und ein Teil der Kanalisierungs- oder Rohrleitungen kann für den Kühlturm verwendet werden, um primäres Kühlmittel an die anderen CDUs zu kommunizieren. Wenn die anderen CDUs sekundäres Kühlmittel empfangen, können sie ihr eigenes sekundäres Kühlmittel, das gekühlt ist, über das empfangene sekundäre Kühlmittel verteilen oder zirkulieren. Wenn die anderen CDUs primäres Kühlmittel empfangen, zirkulieren sie ihr eigenes sekundäres Kühlmittel, das über das empfangene primäre Kühlmittel gekühlt wird.

[0043] Fig. 5 ist ein Prozessablauf von Schritten, die für ein Verfahren **500** zum Verwenden oder Herstellen des mobilen Kühlsystems oder zum Einsetzen eines mobilen Rechenzentrums mit einem mobilen Kühlsystem, wie beispielsweise von Fig. 2-4 und Fig. 6A-17D, gemäß mindestens einer Ausführungsform verfügbar sind. In mindestens einer Ausführungsform stellt ein Schritt in dem Verfahren **502** einen Behälterverteiler bereit, um ein einem Kühlturm zugeordnetes Kühlmittel zu einem oder mehreren flüssigkeitsgekühlten Gestellen innerhalb des mindestens einen Behälters zu zirkulieren. Ein weiterer Schritt **504** des Verfahrens ermöglicht einem oder mehreren Kühlmitteln, in einer sekundären Kühlschleife und einer primären Kühlschleife zu zirkulieren, sind zugeordneten mit die Behälterverteiler. In mindestens einer Ausführungsform wird der Schritt **504** durch eine Ausgabe eines lernenden Subsystems aktiviert, das ein Maschinenlernmodell auf mindestens einem Prozessor ausführt, der eine Logikeinheit aufweist.

[0044] In mindestens einer Ausführungsform führt das lernende Subsystem mehrere Bewertungen durch, um die Ausgabe bereitzustellen. Das lernende Subsystem bewertet Temperaturanforderungen eines oder mehrerer zweiter flüssigkeitsgekühlten Gestellen, bewertet Strömungsraten oder Strömungsvolumen des Kühlmittels basierend teilweise auf ihren Zuordnungen zu den Temperaturanforderungen und bewertet mindestens eine physikalische Beschränkung des Behälters, einer Anhängerladefläche oder eines dritten Behälters, der den Kühlturm hostet. Die bereitgestellte Ausgabe erleichtert dann die Zirkulation des Kühlmittels, weil es mindestens einer Kühlturmanforderung zugeordnet ist, wie beispielsweise einer Strömungsrate und/oder einem Strömungsvolumen, und der mindestens einen physikalischen Beschränkung des Behälters, der Anhängerladefläche oder des dritten Behälters zugeordnet ist, wie beispielsweise eine dimensionale Anforderung innerhalb des mobilen Rechenzentrums zu passen.

[0045] In mindestens einer Ausführungsform kann der Schritt **504** teilweise durch ein neuronales Netzwerk oder ein anderes Maschinenlernmodell durchgeführt werden, das auf mindestens einem Prozessor ausgeführt wird. Dann stellt Schritt **504** eine Ausgabe einem oder mehreren Strömungscontroller bereit und ist imstande, den einen oder mehrere Strömungscontroller zu verwenden, um das Kühlmittel durch den Behälterverteiler und das eine oder mehrere flüssigkeitsgekühlte Gestelle zu zirkulieren. Das ausgeführte Maschinenlernmodell verarbeitet Temperaturen, den Temperaturanforderungen zugeordnet sind, unter Verwendung von mehreren Neuronenebenen des Maschinenlernmodells. Die Neuronen werden mit den Temperaturen zugeführt und weisen zuvor zugeordnete Strömungsraten oder Strömungsvolumen für das Kühlmittel auf. Das Maschinenlernmodell führt eine Korrelation und Extrapolation einer Art zwischen den Temperaturen und den zuvor zugeordneten Strömungsraten oder Strömungsvolumen durch. Ferner stellt das Maschinenlernmodell die Ausgabe, die einer Strömungsrate oder einem Strömungsvolumen für das Kühlmittel zugeordnet ist, dem einen oder mehreren Strömungscontrollern zur Verfügung. In mindestens einer Ausführungsform ist die Ausgabe daher eine Bewertung der zuvor zugeordneten Strömungsraten oder Strömungsvolumen und der mindestens einen physikalischen Beschränkung des Behälters, der Anhängerladefläche oder des dritten Behälters.

[0046] In mindestens einer Ausführungsform stellt Schritt **506** einen Fluidkoppler bereit, um eine Fluidkopplung des Behälterverteilers mit einem zweiten Behälterverteiler eines zweiten Behälters zu ermöglichen. Bei Schritt **508** wird Verifikation der Anforderung für die Kühlung in dem zweiten Behälter durchgeführt. Schritt **510** ermöglicht die Fluidkopplung des Behälterverteilers mit dem zweiten Behälterverteiler des zweiten Behälters. Wenn die Verifikation bestimmt, dass keine Kühlung erforderlich ist, bleiben die Fluidkoppler lediglich zwischen dem Behälter und dem zweiten Behälter bereitgestellt, wie bei Schritt **506**.

[0047] In mindestens einer Ausführungsform umfasst das Verfahren **500** ein Bereitstellen mindestens eines dritten Behälters, einer Anhängerladefläche oder des Behälters, um den Kühlturm zu umfassen. Der Kühlturm ist angepasst, um mindestens eine Kühlturmanforderung zu erfüllen, die für das eine oder mehrere flüssigkeitsgekühlten Gestellen bestimmt ist, und ist angepasst, um mindestens ein physikalisches Merkmal des mindestens einen dritten Behälters, der Anhängerladefläche oder des mindestens einen Behälters zu erfüllen. In mindestens einer Ausführungsform umfasst das Verfahren **500** die Aktivierung mindestens einer primären Kühlschleife, der dem Kühlturm zuzuordnen ist, und umfasst die Aktivierung des Behälterverteilers, der mindestens einer sekundären Kühlschleife zuzuordnen ist. Ein weiterer Schritt kann durchgeführt werden, der mindestens

eine Kühlmittelverteilungseinheit (CDU) aktiviert, die dem mindestens einen Behälter zum Tauschen von Wärme zwischen der mindestens einen primären Kühlturschleife und der mindestens einen sekundären Kühlturschleife zugeordnet ist.

[0048] In mindestens einer Ausführungsform umfasst das Verfahren **500** das Bestimmen eines Merkmals des Kühlturms basierend teilweise auf mindestens einem physikalischen Merkmal, das dem mindestens einen Behälter, einer Anhängerladefläche oder einem dritten Behälter zugeordnet ist, der angepasst ist, den Kühlturm zu hosten, und basiert teilweise auf einem zweiten Merkmal, das einem oder mehreren flüssigkeitsgekühlten Gestellen zugeordnet ist. Wie an anderer Stelle in der Offenbarung erläutert, kann das Merkmal des Kühlturms Abmessungen, Strömungsrate oder Strömungsvolumen sein, die fähig sind, von dem Kühlturm zur Verfügung gestellt zu werden. In mindestens einer Ausführungsform kann das zweite Merkmal die Temperaturverringern oder Wartungsanforderungen des einen oder mehrerer flüssigkeitsgekühlten Gestelle sein. Von daher ist es erforderlich, dass der Kühlturm auf das mobile Rechenzentrum passt, wobei es jedoch ebenfalls erforderlich ist, die erforderliche Kühlung den Gestellen bereitzustellen.

[0049] In mindestens einer Ausführungsform ist das Verfahren **500** bei Schritt **510** imstande, unter Verwendung mindestens eines Prozessors einen oder mehrere Strömungscontroller zu steuern, die dem Behälterverteiler zugeordnet sind, um das dem Kühlturm zugeordnete Kühlmittel zu dem einen oder mehreren flüssigkeitsgekühlten Gestellen zu zirkulieren, und dem Kühlmittel zu ermöglichen, von dem Behälterverteiler zu dem zweiten Behälterverteiler des zweiten Behälters über die Fluidkoppler zu strömen. In mindestens einer Ausführungsform erstrecken sich die Fluidkoppler von dem Behälterverteiler oder dem Behälter und sind für die Fluidkopplung zwischen dem Behälterverteiler und dem zweiten Behälterverteiler angepasst.

[0050] In mindestens einer Ausführungsform wird mindestens ein Prozessor für ein mobiles Kühlsystem offenbart. Mindestens ein Prozessor umfasst mindestens eine Logikeinheit, um einen oder mehrere Strömungscontroller zu steuern, die einem Behälterverteiler zugeordnet sind, um Kühlmittel zu zirkulieren, das einem Kühlturm to einer oder mehreren flüssigkeitsgekühlten Gestellen innerhalb mindestens eines Behälters und dem Kühlmittel zu ermöglichen, von dem Behälterverteiler zu einem zweiten Behälterverteiler eines zweiten Behälters zu strömen. In mindestens einer Ausführungsform kann der Strömungscontroller eine Eingabe von der mindestens einen Prozessor empfangen, der sich in einem Vorrichtungscontroller eines DMS befindet, das den einen oder mehrere Strömungscontroller steuert und ebenfalls Sensoren zugeordnet ist, um die Temperatur eines oder mehrerer Orte eines Gestells, des Serverkastens oder Servereinschubs zu überwachen. In mindestens einer Ausführungsform umfasst das DMS mindestens einen Prozessor, welcher der Prozessor sein kann, der den Strömungscontrollern oder einem separaten Prozessor zugeordnet sein kann, der einem lernenden Subsystem angepasst ist. Das lernende Subsystem bewertet Temperaturanforderungen eines oder mehrerer zweiter flüssigkeitsgekühlten Gestellen, bewertet Strömungsraten oder Strömungsvolumen des Kühlmittels basierend teilweise auf ihren Zuordnungen zu den Temperaturanforderungen und bewertet mindestens eine physikalische Beschränkung des Behälters, einer Anhängerladefläche oder eines dritten Behälters, der den Kühlturm hostet. Das lernende Subsystem stellt eine Ausgabe zum Erleichtern der Zirkulation des Kühlmittels bereit. Die Ausgabe kann den Strömungscontrollern bereitgestellt werden und ist mindestens einer Kühlturmanforderung zugeordnet und der mindestens einen physikalischen Beschränkung des Behälters, der Anhängerladefläche oder des dritten Behälters zugeordnet.

[0051] In mindestens einer Ausführungsform weisen die Prozessoren eine kommunikative Kopplung mit einem Rechenzentrum-Managementsystem (DMS) auf. Das DMS ist innerhalb des einen oder mehreren flüssigkeitsgekühlten Gestellen aktiviert oder zugeordnet. Die kommunikative Kopplung ist angepasst, um Temperatureingaben zu empfangen und Steuerausgaben für den einen oder mehrere Strömungscontroller zu kommunizieren, um die Zirkulation des Kühlmittels zu erleichtern. In mindestens einer Ausführungsform besteht die kommunikative Kopplung aus Kugelkontakten oder anderen Stiften, die eine Eingabe-Ausgabe von den Prozessoren bereitstellen.

[0052] In mindestens einer Ausführungsform hostet ein Rechenzentrum-Managementsystem (DMS) mindestens einen Prozessor eines Vorrichtungscontrollers. In mindestens einer Ausführungsform ist das DMS ein verteiltes System, das von einem Vorrichtungscontroller zu Subcontroller kommuniziert, die den Strömungscontrollern einer Kühlturschleife direkt zugeordnet sind. In mindestens einer Ausführungsform ist der Vorrichtungscontroller ein verteiltes Netzwerk von Prozessoren und Speichern mit Anweisungen, die von den Prozessoren durchgeführt werden. Jeder Prozessor in dem verteilten Netzwerk befindet sich benachbart zu und steuert einen jeweiligen Strömungscontroller oder mehrere Strömungscontrollern. In mindestens einer Ausführungsform sind ein oder mehrere der Prozessoren zum Trainieren und Ausführen neuronaler Netzwerke, die an anderer Stelle in dieser Offenbarung erläutert werden. Von daher ist es möglich, dass mindestens ein oder

mehrere neuronale Netzwerke aktiv zu einem beliebigen Zeitpunkt innerhalb eines oder mehrerer Prozessoren ausgeführt werden, während eine Sicherungskopie eines oder mehrerer neuronaler Netzwerke trainiert wird, um Informationen aktiv zu korrelieren und extrapolierte Strömungsrateninformationen aktiv auszugeben, die verwendet werden können, um Strömungscontroller der jeweiligen Kühlschleife zu steuern.

[0053] In mindestens einer Ausführungsform kann ein Strömungscontroller auf einer Austrittsseite der Rohrleitungen in jedem der Rechenzentrumsmerkmale bereitgestellt werden. Des Weiteren kann ein Strömungscontroller auf sowohl den Eintritts- als auch den Austrittsseiten der Rohrleitungen in jedem der Rechenzentrumsmerkmale bereitgestellt werden. Von daher führt, wenn der Strömungscontroller auf der Austrittsseite ist, er eine Saugaktion und keine Drückaktion durch. Wenn zwei Strömungscontroller hintereinander arbeiten, gibt es sowohl Saug- als auch Druckaktionen. Die erreichten Strömungsraten können durch die Tandemströmungscontroller höher sein. Die Konfiguration oder Anpassung der Strömungscontroller kann beispielsweise durch Anforderungen der Komponenten bestimmt werden.

[0054] In mindestens einer Ausführungsform erleichtert die sekundäre Kühlschleife eine Vorgabe- oder Standardbewegung des Kühlmittels in einem Behälterverteiler. In mindestens einer Ausführungsform kann das lernende Subsystem über den tief lernenden Anwendungsprozessor implementiert werden, wie beispielsweise den Prozessor **1400** in **Fig. 14**, und kann die Neuronen **1502** und Komponenten davon verwenden, die unter Verwendung von Schaltungen oder Logik implementiert werden, die eine oder mehreren Arithmetik-Logikeinheiten (ALUs) umfasst, wie in **Fig. 15** beschrieben. Von daher umfasst das lernende Subsystem mindestens einen Prozessor zum Bewerten von Temperaturanforderungen mit Strömungsraten oder Strömungsvolumen, die einem oder mehreren Kühltürmen zugeordnet sind. Die Strömungsraten oder Strömungsvolumen können einer primären Kühlschleife zugeordnet sein, wobei sie jedoch ebenfalls einem zweiten Kühlmittel einer sekundären Kühlschleife zugeordnet sein können. In mindestens einer Ausführungsform erfordert die Bestimmung der Strömungsrate und des Strömungsvolumens der sekundären Kühlschleife wiederum die Bestimmung der entsprechenden Strömungsraten und Strömungsvolumen für die sekundäre Kühlschleife. Auf diese Art und Weise ist es möglich, die Kühlung zu erlangen, die von dem Kühlturm für die Gestelle vorgesehen ist.

[0055] Des Weiteren führt das lernende Subsystem ein Maschinenlernmodell aus, das Temperaturen verarbeitet, die von vorherigen Anwendungen vielleicht in einer Testumgebung gesammelt wurden, um die Temperaturen innerhalb eines Servers, eines Gestells oder eines Kühlmittels der sekundären Kühlschleife zu steuern. Die gesammelten Temperaturen werden Temperaturanforderungen zugeordnet und umfassen (a) erreichte Temperaturen für bestimmte Strömungsraten (und zugeordnete Strömungsvolumen) eines Kühlmittels (und sekundären Kühlmittels); (b) erreichter Unterschied in Temperaturen für bestimmte Strömungsraten oder Strömungsvolumen für spezifische Zeiträume; und (c) Anfangstemperaturen und entsprechende Strömungsraten oder Strömungsvolumen, die angewandt werden, um den Server, das Gestell oder das Kühlmittel in optimalem Betrieb zu halten.

[0056] In mindestens einer Ausführungsform können Aspekte der Verarbeitung für das tief lernende Subsystem die gesammelten Informationen verwenden, die in Einklang mit den Merkmalen verarbeitet wurden, die mit Bezugnahme auf **Fig. 14**, **Fig. 15** erläutert wurden. In einem Beispiel verwendet die Verarbeitung mehrere Neuronenebenen des Maschinenlernmodells, die mit einer oder mehreren der oben angemerkten Temperaturanforderungen und den entsprechenden Strömungsraten oder Strömungsvolumen von Kühlmittel innerhalb einer oder mehreren Kühlschleifen, sowie auch den physikalischen Beschränkungen des(der) Behälters oder Anhängerladeflächen geladen sind. Das lernende Subsystem führt ein Training durch, das als eine Bewertung von Temperaturänderungen dargestellt werden kann, die vorherigen Strömungsraten oder Strömungsvolumen (oder Änderungen darin) gemäß Anpassungen zugeordnet sind, die an dem einen oder mehreren Strömungscontroller vorgenommen wurden, die den jeweiligen Kühlschleifen zugeordnet sind.

[0057] In mindestens einer Ausführungsform können die Neuronenebenen Werte speichern, die dem Bewertungsprozess zugeordnet sind, und können eine Zuordnung oder Korrelation zwischen den Temperaturanforderungen und den Strömungsraten oder Strömungsvolumen sowie auch physikalische Beschränkungen des einem Kühlturm verfügbaren physischen Raums darstellen, um die Temperaturanforderungen zu erreichen. Das lernende Subsystem ist einmal trainiert beispielsweise imstande, bei der Anwendung einer Strömungsrate oder eines Strömungsvolumens des Kühlmittels in einem oder mehreren Kühlschleifen die Kühlung auf eine Temperatur (oder einer Änderung, wie beispielsweise einer Verringerung in Temperatur) gemäß den Temperaturanforderungen zu erreichen. Die Temperaturanforderungen und zuvor zugeordneten Strömungsraten oder Strömungsvolumen für Kühlmittel, die verwendet werden, um die Temperaturanforderungen zu erreichen (z.B., Temperaturänderungen), können von dem lernenden Subsystem verwendet werden, um die Ausgabe bereitzustellen, die einer erforderlichen Strömungsrate oder Strömungsvolumen für ein oder mehrere Kühlmittel

tel einer oder mehrerer Kühlschleifen zugeordnet ist, um eine Kühlung zu erreichen, die durch eine Temperatur (beispielweise eine Verringerung darstellt) als eine vorliegende Temperatur für ein Gestell, einen Server oder ein Kühlmittel innerhalb eines oder mehrerer Abschnitte einer jeweiligen Kühlschleife reflektiert wird.

[0058] In mindestens einer Ausführungsform ist ein Ergebnis des lernenden Subsystems als Reaktion auf eine erfasste Temperatur eines Gestells, eines Servers oder des Kühlmittels eine Ausgabe an den einen oder mehrere Strömungscontroller in einer oder mehreren Kühlschleifen, eine Modifikation einer Strömungsrate eines Kühlmittels einer jeweiligen Kühlschleife. Die Modifikation der Strömungsrate oder des Strömungsvolumen ermöglicht einer bestimmten Strömung des Kühlmittels, den Bereich des Gestells, des Servers, der sekundären Kühlschleife zu erreichen, der Kühlung erfordert. Die modifizierte Strömungsrate oder das modifizierte Strömungsvolumen kann beibehalten werden, bis die Temperatur in dem Bereich eine Temperatur erreicht, die der Strömungsrate oder Strömungsvolumen des Kühlmittels zugeordnet ist, die dem lernenden Subsystem bekannt ist. Alternativ kann die modifizierte Strömungsrate oder das modifizierte Strömungsvolumen beibehalten werden, bis sich die Temperatur in dem Bereich um einen bestimmten Wert ändert. Alternativ kann die modifizierte Strömungsrate oder Strömungsvolumen beibehalten werden, bis die Temperatur in dem Bereich eine Nenntemperatur für das Kühlmittel, den Server oder das Gestell erreicht.

[0059] In mindestens einer Ausführungsform umfasst der Vorrichtungscontroller einen Prozessor mit mindestens einer Logikeinheit, um einen oder mehrere Strömungscontroller zu steuern, die einer oder mehreren Kühlschleifen zugeordnet sind. In mindestens einer Ausführungsform kann der Vorrichtungscontroller ein Prozessor mit dem Rechenzentrum sein, wie beispielsweise ein Prozessor **702** von **Fig. 7A**. Der Strömungscontroller erleichtert die Bewegung eines jeweiligen Kühlmittel einer jeweiligen Kühlschleife, um die Kühlung eines Bereichs in dem mobilen Rechenzentrum als Reaktion auf eine in dem Bereich erfasste Temperatur zu ermöglichen. In mindestens einer Ausführungsform ist der Prozessor ein Prozessorkern eines Mehrkern-Prozessors, wie beispielsweise die Mehrkern-Prozessoren **905, 906** in **Fig. 9A**. In mindestens einer Ausführungsform kann die mindestens eine Logikeinheit angepasst sein, um einen Temperaturwert von einem Temperatursensor innerhalb des Servers, des Gestells oder Kühlmittels zu empfangen, welcher der sekundären Kühlschleife zugeordnet ist und angepasst sein kann, um die Bewegung des Kühlmittels in einer unterschiedlichen Strömungsrate oder einem unterschiedlichen Strömungsvolumen zu erleichtern, um den Server, das Gestell oder das der sekundären Kühlschleife zugeordnete Kühlmittel zusätzlich zu kühlen.

[0060] In mindestens einer Ausführungsform kann ein Prozessor, wie beispielsweise die Prozessorkerne der Mehrkern-Prozessoren **905, 906** in **Fig. 9A**, ein lernendes Subsystem umfassen, um Temperaturen des Servers, des Gestells, des Kühlmittels oder sogar einer Komponente innerhalb des Servers mit Strömungsraten zu bewerten, die einem oder mehreren Strömungscontrollern einer jeweiligen Kühlschleife zugeordnet sind. Das lernende Subsystem stellt eine Ausgabe bereit, die einer Strömungsrate zum Erleichtern der Bewegung eines jeweiligen Kühlmittels zugeordnet ist, durch Steuern des einen oder mehrerer Strömungscontroller, der (die) einer jeweiligen Kühlschleife zugeordnet ist(sind). In mindestens einer Ausführungsform führt das lernende Subsystem ein Maschinenlernmodell aus, um die Temperatur unter Verwendung mehrere Neuronenebenen des Maschinenlernmodells mit den Temperaturanforderungen und mit zuvor zugeordneten Strömungsraten oder Strömungsvolumen für das Kühlmittel zu verarbeiten. Das Maschinenlernmodell kann unter Verwendung der in **Fig. 15** beschriebenen Neuronenstruktur und dem tief lernenden Prozessor implementiert werden, wie in **Fig. 14** beschrieben. Das Maschinenlernmodell stellt die Ausgabe zur Verfügung, die der Strömungsrate oder dem Strömungsvolumen zugeordnet ist, die ebenfalls eine für den Kühlturm gemachte physikalische Beschränkung widerspiegeln kann. Des Weiteren ermöglicht eine Anweisungsausgabe des Prozessors, wie beispielsweise einer Klemme eines Verbinderbusses oder einer Kugel einer Kugelgitteranordnung, die Kommunikation der Ausgabe mit dem einen oder mehreren Strömungscontrollern, um eine Strömungsrate oder ein Strömungsvolumen des Kühlmittels zu modifizieren, das der jeweiligen Kühlschleife zugeordnet ist, und um eine bestimmte Strömungsrate oder ein bestimmtes Strömungsvolumen für das Kühlmittel als Reaktion auf die Ausgabe zu veranlassen.

[0061] In mindestens einer Ausführungsform ist die vorliegende Offenbarung für mindestens einen Prozessor für ein Kühlsystem. Der mindestens eine Prozessor umfasst mindestens eine Logikeinheit, um ein neuronales Netzwerk mit verborgenen Schichten von Neuronen zu trainieren, um einem Bereich des Rechenzentrums (z.B., einem Server, einer Komponente innerhalb des Servers, einem Gestell oder einem Kühlmittel einer sekundären Kühlschleife) zugeordnete Temperaturen mit zuvor zugeordneten Strömungsraten für ein Kühlmittel einer jeweiligen Kühlschleife zu bewerten, das verwendet wird, um den Bereich zusätzlich zu kühlen. Wie an anderer Stelle in dieser Offenbarung beschrieben und mit Bezugnahmen auf **Fig. 2-5, Fig. 14** und **Fig. 15** vorgenommen, kann das Training durch Schichten von Neuronen durchgeführt werden, denen Eingaben der Temperaturen eines oder mehrerer Bereiche in dem Rechenzentrum und zugeordnete Strömungsraten oder

Strömungsvolumen eines Kühlmittels bereitgestellt werden, die einer vorhergehenden Anwendung vielleicht in einer Testumgebung zugeordnet sind. Die Temperaturanforderungen können eine Starttemperatur (und zugeordnete Strömungsrate des für die Starttemperatur angewandten Kühlmittels, um die Temperatur auf eine Nenntemperatur des einen oder mehreren Bereiche zu bringen), eine Endtemperatur (nachdem ein Kühlmittel für eine Strömungsrate oder ein Strömungsvolumen für einen Zeitraum angewandt wird) und einen Unterschied in Temperaturen umfassen, der für die Strömungsrate oder das Strömungsvolumen und den Zeitraum der Anwendung des Kühlmittels erreicht wurde. Eines oder mehrere dieser Merkmale können verwendet werden, um ein neuronales Netzwerk zu trainieren, um zu bestimmen, wenn eine Strömung des Kühlmittels anzuwenden und/oder wenn die Strömung des Kühlmittels zu stoppen ist, z.B., wenn eine für einen Bereich erfasste Temperatur die Starttemperatur erreicht, wenn die für den Bereich erfasste Temperatur die Endtemperatur erreicht, und/oder die für den Bereich erfasste Temperatur den Unterschied in Temperatur widerspiegelt, die für den Bereich geeignet ist (wie beispielsweise eine Verringerung in der Temperatur auf eine Nenntemperatur).

[0062] In mindestens einer Ausführungsform umfasst der mindestens eine Prozessor mindestens eine Logikeinheit, die zum Trainieren eines neuronalen Netzwerks konfiguriert oder angepasst ist, und einen Mehrkern-Prozessor. In mindestens einer Ausführungsform kann die mindestens eine Logikeinheit innerhalb eines Prozessorkerns sein, der verwendet werden kann, um eine Temperatur eines Bereichs in dem Rechenzentrum gegen das neuronale Netzwerk zu bewerten und um eine Anweisung auszugeben, um die Montage eines Kühlturms zur mobilen Kühlung des Bereichs zu erleichtern. Von daher ist es möglich, obwohl der Bereich sogar Kühlmittel bereits empfängt, das einer sekundären Kühlschleife zugeordnet ist, zusätzliche Kühlmittel zum ergänzenden Kühlen und zum Steuern der Temperatur des Bereichs über eine Anforderung oder Anweisung für einen mobilen Kühlturm zu empfangen. Als Reaktion auf die Anforderung oder die Anweisung kann das Kühlmittel bei einer Strömungsrate oder Volumen bereitgestellt werden, die für die vorliegende Temperatur oder einer Zieltemperatur für den Bereich bestimmt werden. In mindestens einer Ausführungsform weist der Prozessor des Vorrichtungscontrollers eine Anweisungsausgabe in der Form einer Klemme oder einer Kugel zum Kommunizieren der Ausgabe mit einem oder mehreren Strömungscontrollern auf, die der jeweiligen Kühlschleife zugeordnet sind. Der eine oder die mehreren Strömungscontroller modifizieren als Ergebnis eine dem Kühlmittel zugeordnete Strömungsrate oder zugeordnetes Strömungsvolumen. Die Modifikation ist daher als Reaktion auf die Ausgabe und erleichtert die Kühlung eines oder mehrerer Bereiche der mobilen Rechenzentrums. In mindestens einer Ausführungsform ist die mindestens eine Logikeinheit angepasst, um einen Temperaturwert eines Temperatursensors zu empfangen, der einem Vorrichtungscontroller zugeordnet ist.

RECHENZENTRUM

[0063] Fig. 6A veranschaulicht ein beispielhaftes Rechenzentrum 600, in welchem mindestens eine Ausführungsform verwendet werden kann. In mindestens einer Ausführungsform umfasst das Rechenzentrum 600 eine Rechenzentrum-Infrastrukturschicht 610, eine Framework-Schicht 620, eine Softwareschicht 630 und eine Anwendungsschicht 640. In mindestens einer Ausführungsform können, wie beispielsweise mit Bezug auf Fig. 2 beschrieben, Merkmale in Komponenten 204-214 innerhalb oder in Zusammenarbeit mit dem beispielhaften Rechenzentrum 600 durchgeführt werden. In mindestens einer Ausführungsform können die Infrastrukturschicht 610, die Frameworkschicht 620, die Softwareschicht 630 und die Anwendungsschicht 640 teilweise oder vollständig über Rechenkomponenten auf Servereinschüben, die in Gestellen 210 des Rechenzentrums 200 lokalisiert sind, bereitgestellt werden. Dies ermöglicht Kühlsystemen der vorliegenden Offenbarung, Kühlung zu bestimmten der Rechenkomponenten auf eine effiziente und wirksame Art und Weise zu leiten. Ferner können Aspekte des Rechenzentrums, einschließlich der Rechenzentrum-Infrastrukturschicht 610, der Frameworkschicht 620, der Softwareschicht 630 und der Anwendungsschicht 640, verwendet werden, um das mobile Rechenzentrum mit einem mobile Rechenzentrumzentrumkühlsystem zu unterstützen, das mindestens mit Bezugnahme auf Fig. 2A-5 obenstehend erläutert wird. Von daher kann die Erläuterung mit Bezugnahme auf Fig. 6A-17D verstanden werden, beispielsweise auf die Hardware- und Softwaremerkmale Anwendung zu finden, die erforderlich sind, um das mobile Rechenzentrum zu ermöglichen oder zu unterstützen, welches das Kühlsystem des mobilen Rechenzentrumzentrums von Fig. 2-5.

[0064] In mindestens einer Ausführungsform kann, wie in Fig. 6A gezeigt, kann die Rechenzentrum-Infrastrukturschicht 610 einen Ressourcenorchestrator 612, gruppierte Rechenressourcen 614 und Knotenrechenressourcen („Knoten-C.R.“) 616(1)-616(N) umfassen, wobei „N“ eine beliebige positive ganze Zahl darstellt. In mindestens einer Ausführungsform können die Knoten-C.R.s 616(1)-616(N), sind jedoch nicht beschränkt auf, eine beliebige Anzahl von Zentralverarbeitungseinheiten („CPUs“) oder andere Prozessoren (einschließlich Beschleunigern, feldprogrammierbaren Gate-Arrays (FPGAs), Graphikprozessoren usw.), Speichervorrichtungen, (z.B. dynamische Nur-Lese-Speicher), Speichervorrichtungen (z.B. Festkörper- oder Festplattenlaufwerke), Netzwerk-Eingabe-/Ausgabe („NW E/A“-)Vorrichtungen, Netzwerkschalter, virtuelle Maschinen („VMs“),

Leistungsmodule und Kühlmodule usw. umfassen. In mindestens einer Ausführungsform können ein oder mehrere Knoten-C.R.s aus den Knoten-C.R.s **616(1)-616(N)** ein Server mit einer oder mehreren der oben erwähnten Rechenressourcen sein.

[0065] In mindestens einer Ausführungsform können gruppierte Rechenressourcen **614** separate Gruppierungen von Knoten-C.R.s. umfassen, die in einem oder mehreren Gestelle (nicht gezeigt) untergebracht sind, oder in vielen Gestelle, die in Rechenzentren an verschiedenen geografischen Standorten untergebracht sind (ebenfalls nicht gezeigt). Getrennte Gruppierungen von Knoten-C.R.s. innerhalb gruppierter Rechenressourcen **614** können 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 Gestelle gruppiert sein, um Rechenressourcen zur Unterstützung einer oder mehrerer Arbeitslasten bereitzustellen. In mindestens einer Ausführungsform können ein oder mehrere Gestelle ebenfalls eine beliebige Anzahl von Leistungsmodulen, Kühlmodulen und Netzwerkschaltern in beliebiger Kombination umfassen.

[0066] In mindestens einer Ausführungsform kann der Ressourcenorchestrator **612** eine oder mehrere Knoten-C.R.s **616(1)-616(N)** und/oder gruppierte Rechenressourcen **614** konfigurieren oder anderweitig steuern. In mindestens einer Ausführungsform kann der Ressourcenorchestrator **612** eine Software-Design-Infrastruktur („SDI“-)Verwaltungseinheit für das Rechenzentrum **600** umfassen. In mindestens einer Ausführungsform kann der Ressourcenorchestrator Hardware, Software oder eine Kombination davon umfassen.

[0067] In mindestens einer Ausführungsform umfasst, wie in **Fig. 6A** gezeigt, die Framework-Schicht **620** einen Job-Scheduler **622**, einen Konfigurationsverwalter **624**, einen Ressourcenverwalter **626** und ein verteiltes Dateisystem **628**. In mindestens einer Ausführungsform kann die Framework-Schicht **620** ein Framework zur Unterstützung der Software **632** der Softwareschicht **630** und/oder eine oder mehrere Anwendungen **642** der Anwendungsschicht **640** umfassen. In mindestens einer Ausführungsform kann die Software **632** oder kann (können) die Anwendung(en) **642** 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 **620** eine Art Framework für eine freie und quelloffene Software-Webanwendung-Framework wie beispielsweise Apache Spark™ (nachfolgend als „Spark“ bezeichnet) sein, welches das verteilte Dateisystem **628** für eine groß angelegte Datenverarbeitung (z.B. „big data“) nutzen kann. In mindestens einer Ausführungsform kann der Job-Scheduler **632** einen Spark-Treiber umfassen, um die Planung von Arbeitslasten zu erleichtern, die von verschiedenen Schichten des Rechenzentrums **600** unterstützt werden. In mindestens einer Ausführungsform kann der Konfigurationsverwalter **624** in der Lage sein, verschiedene Schichten wie beispielsweise die Softwareschicht **630** und die Framework-Schicht **620** einschließlich Spark und des verteilten Dateisystems **628** zur Unterstützung der groß angelegten Datenverarbeitung zu konfigurieren. In mindestens einer Ausführungsform kann der Ressourcenverwalter **626** in der Lage sein, geclusterte oder gruppierte Rechenressourcen zu verwalten, die zur Unterstützung des verteilten Dateisystems **628** und des Job-Planers **622** gemappt oder zugeteilt sind. In mindestens einer Ausführungsform können geclusterte oder gruppierte Rechenressourcen die gruppierte Rechenressource **614** auf der Rechenzentrum-Infrastrukturschicht **610** umfassen. In mindestens einer Ausführungsform kann sich der Ressourcenverwalter **626** mit dem Ressourcenorchestrator **612** koordinieren, um diese gemappten oder zugeteilten Rechenressourcen zu verwalten.

[0068] In mindestens einer Ausführungsform kann die in der Softwareschicht **630** enthaltene Software **632** Software, die von mindestens Teilen der Knoten-C.R.s **616(1)-616(N)** verwendet wird, gruppierte Rechenressourcen **614** und/oder das verteilte Dateisystem **628** der Framework-Schicht **620** umfassen. Eine oder mehrere Arten von Software können, ohne darauf beschränkt zu sein, Internet-Webseiten-Suchsoftware, E-Mail-Virens Scanner-Software, Datenbanksoftware und Streaming- Videoinhalt-Software umfassen.

[0069] In mindestens einer Ausführungsform kann(können) die in der Anwendungsschicht **640** enthaltene(n) Anwendung(en) **642** eine oder mehrere Arten von Anwendungen, die von mindestens Teilen der Knoten-C.R.s **616(1)-616(N)** verwendet werden, gruppierte Rechenressourcen **614** und/oder das verteilte Dateisystem **628** der Framework-Schicht **620** umfassen. Eine oder mehrere Arten von Anwendungen können, ohne darauf beschränkt zu sein, eine beliebige Anzahl von genomischen Anwendungen, ein kognitives Berechnen und Anwendungen maschinellen Lernens, 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.

[0070] In mindestens einer Ausführungsform kann jeder der Konfigurationsmanager **634**, des Ressourcenmanager **636** und des Ressourcenorchestrators **612**, 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 **600** davon befreien, möglicherweise schlechte Konfigurationsentscheidungen zu treffen und möglicherweise ungenutzte und/oder schlecht funktionierende Abschnitte eines Rechenzentrums zu vermeiden.

[0071] In mindestens einer Ausführungsform kann das Rechenzentrum **600** 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. In mindestens einer Ausführungsform kann in mindestens einer Ausführungsform ein Modell maschinellen Lernens trainiert werden, indem Gewichtsparameter gemäß einer neuronalen Netzwerkarchitektur unter Verwendung der vorstehend in Bezug auf das Rechenzentrum **600** 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 das Rechenzentrum **600** beschriebenen Ressourcen durch Verwenden von Gewichtsparametern abzuleiten oder vorherzusagen, die durch eine oder mehrere der hier beschriebenen Trainingstechniken berechnet werden. Wie zuvor erläutert, können Techniken tiefen Lernens verwendet werden, um die intelligente Steuerung der Strömungscontroller in dem mobilen Rechenzentrum zu unterstützen, das ein Kühlsystem für eine mobiles Rechenzentrum aufweist, durch Überwachen Bereichstemperaturen des Rechenzentrums. Tiefes Lernen kann unter Verwendung eines beliebigen entsprechenden Lernnetzwerks und den Rechenfähigkeiten des Rechenzentrums **600** vorangebracht werden. Von daher kann ein tiefes neuronales Netzwerk (DNN), ein rekurrentes neuronales Netzwerk (RNN) oder ein faltendes neuronales Netzwerk (CNN) entweder gleichzeitig oder nebenläufig unter Verwendung der Hardware in dem Rechenzentrum unterstützt werden. Sobald ein Netzwerk trainiert und erfolgreich bewertet ist, um Daten beispielsweise innerhalb einer Teilmenge oder eines Slice zu erkennen, kann das trainierte Netzwerk ähnliche repräsentative Daten zur Verwendung mit den gesammelten Daten bereitstellen.

[0072] In mindestens einer Ausführungsform kann das Rechenzentrum CPUs, anwendungsspezifische integrierte Schaltungen (ASICs), GPUs, FPGAs oder andere Hardware verwenden, um Training und/oder Inferenzieren unter Verwendung der vorstehend beschriebenen Ressourcen durchzuführen. 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 Strömungsraten, Temperatur und Ortsinformation oder andere Dienste künstlicher Intelligenz, zu trainieren oder eine Inferenzieren derselben durchzuführen.

INFERENZ- UND TRAININGSLOGIK

[0073] Die Inferenz- und/oder Trainingslogik kann verwendet werden, um Inferenz- und/oder Trainingsoperationen durchzuführen, die mit einer oder mehreren Ausführungsformen verbunden sind. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik im System **Fig. 6A** zum Inferenzieren oder Vorhersagen von Operationen verwendet werden, die mindestens teilweise auf Gewichtsparametern basieren, die unter Verwendung von Trainingsoperationen eines neuronalen Netzwerk, neuronaler Netzwerkfunktionen und/oder Architekturen oder hier beschriebenen Verwendungsfällen des neuronalen Netzwerk berechnet wurden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik **615** umfassen, ohne darauf beschränkt zu sein, Hardwarelogik, bei der Rechenressourcen fest zugeordnet sind oder anderweitig exklusiv in Verbindung mit Gewichtswerten oder anderen Informationen verwendet werden, die einer oder mehreren Schichten von Neuronen innerhalb eines neuronalen Netzwerks entsprechen. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik **615** in Verbindung mit einer anwendungsspezifischen integrierten Schaltung (ASIC), wie beispielsweise Tensorflow® Processing Unit von Google, einer Inferenzverarbeitungseinheit (Inference Processing Einheit; IPU) von Graphcore™ oder einem Nirvana® (z.B. „Lake Crest“) Prozessor von Intel Corp. verwendet werden.

[0074] In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik **615** in Verbindung mit Hardware der Zentralverarbeitung (Central Processing Unit; CPU), Hardware der Graphikverarbeitungseinheit (GPU) oder anderer Hardware, wie beispielsweise feldprogrammierbaren Gate Arrays (FPGAs), verwendet werden. In mindestens einer Ausführungsform umfasst die Inferenz- und/oder Trainingslogik **615**, ohne darauf beschränkt zu sein, Code- und/oder Datenspeichermodule, die verwendet werden können, um Code (z.B., ei-

nen Graph-Code), Gewichtswerte und/oder anderen Informationen zu speichern, die Bias-Werte, Gradienteninformationen, Momentenwerte und/oder andere Parameter- oder Hyperparameterinformationen umfassen. In mindestens einer Ausführungsform ist jedes der Code- und/oder Datenspeichermodule einer fest zugeordneten Rechenressource zugeordnet. In mindestens einer Ausführungsform umfasst die fest zugeordnete Rechenressource Rechenhardware, die ferner eine oder mehrere ALUs umfasst, die mathematische Funktionen, wie beispielsweise lineare algebraische Funktionen, nur an Informationen durchführen, die in Code- und/oder Datenspeichermodulen gespeichert sind und Ergebnisse davon in einem Aktivierungsspeichermodul der Inferenz- und/oder Trainingslogik **615** gespeichert sind.

[0075] Fig. **6B**, Fig. **6C** veranschaulichen Inferenz- und/oder Trainingslogik, wie beispielsweise in Fig. **6A** und in mindestens einer Ausführungsform der vorliegenden Offenbarung verwendet, gemäß mindestens einer Ausführungsform. Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die mindestens einer Ausführungsform zugeordnet sind. Einzelheiten hinsichtlich der Inferenz- und/oder Trainingslogik **615** werden nachstehend in Verbindung mit Fig. **6B** und/oder **6C** bereitgestellt. Die Inferenz- und/oder Trainingslogik **615** von Fig. **6B** und Fig. **6C** unterscheiden sich durch die Verwendung der Arithmetiklogikeinheiten (ALUs) **610** versus der Rechenhardware **602**, **606**. In mindestens einer Ausführungsform umfasst jede der Rechenhardware **602** und der Rechenhardware **606** eine oder mehrere ALUs, die mathematische Funktionen, wie beispielsweise lineare algebraische Funktionen, lediglich an Informationen durchführen, die in der Code- und/oder Datenspeicherung **601** und Code- und/oder Datenspeicherung **605** jeweils gespeichert sind, deren Ergebnis in der Aktivierungsspeicherung **620** gespeichert wird. Von daher können Fig. **6B** und Fig. **6C** Alternativen sein und können austauschbar verwendet werden, sofern nichts anderes angegeben.

[0076] In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik **615**, ohne darauf beschränkt zu sein, eine Code- und/oder Datenspeicherung **601** umfassen, um Vorwärts- und/oder Ausgangsgewichte und/oder Eingangs-/Ausgangsdaten und/oder andere Parameter zu speichern, um Neuronen oder Schichten eines neuronalen Netzwerks zu konfigurieren, das in Aspekten einer oder mehrerer Ausführungsformen trainiert und/oder zum Inferenzieren verwendet wird. In mindestens einer Ausführungsform kann die Trainingslogik **615** eine Code- und/oder eine Datenspeicherung **601** umfassen oder mit dieser gekoppelt sein, um einen Graphencode oder eine andere Software zum Steuern des Timings und/oder der Reihenfolge zu speichern, in welcher Gewichts- und/oder andere Parameterinformationen zu laden sind, um Logik, einschließlich Ganzzahl- und/oder Gleitkommaeinheiten (zusammen arithmetische Logikeinheiten (ALUs)), zu konfigurieren. In mindestens einer Ausführungsform lädt Code, wie beispielsweise Graphencode, Gewichts- oder andere Parameterinformationen, in Prozessor-ALUs basierend auf einer Architektur eines neuronalen Netzwerks, dem dieser Code entspricht. In mindestens einer Ausführungsform speichern der Code- und/oder die Datenspeicherung **601** Gewichtsparameter und/oder Eingabe-/Ausgabedaten jeder Schicht eines neuronalen Netzwerks, das in Verbindung mit einer oder mehreren Ausführungsformen während der Vorwärtspropagierung von Eingabe-/Ausgabedaten und/oder Gewichtsparametern während des Trainings und/oder des Inferenzierens unter Verwendung von Aspekten einer oder mehrerer Ausführungsformen trainiert oder verwendet wurde. In mindestens einer Ausführungsform kann ein beliebiger Teil der Code- und/oder Datenspeicherung **601** in anderen On-Chip- oder Off-Chip-Datenspeicher, einschließlich des L1-, L2- oder L3-Cache oder Systemspeichers eines Prozessors, enthalten sein.

[0077] In mindestens einer Ausführungsform kann ein beliebiger Teil der Code- und/oder Datenspeicherung **601** intern oder extern zu einem oder mehreren Prozessoren oder anderen Hardware-Logikvorrichtungen oder Schaltungen sein. In mindestens einer Ausführungsform kann der Code und/oder die Code- und/oder Datenspeicherung **601** ein Cache-Speicher, ein dynamisches Direktzugriffsspeicher („DRAM“), ein statischer Direktzugriffsspeicher („SRAM“), ein nichtflüchtiger Speicher (z.B. Flash-Speicher) oder andere Speicherung sein. In mindestens einer Ausführungsform kann die Wahl, ob der Code und/oder die Code- und/oder Datenspeicherung **601** 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 verfügbarer Speicherung, Latenzanforderungen der durchgeführten Trainings- und/oder Inferenzfunktionen, der Losgröße der beim Inferenzieren und/oder dem Training eines neuronalen Netzwerks verwendeten Daten oder einer Kombination dieser Faktoren abhängen.

[0078] In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik **615**, ohne darauf beschränkt zu sein, eine Code- und/oder Datenspeicherung **605** zum Speichern von Rückwärts- und/oder Ausgangsgewichts- und/oder Eingangs-/Ausgangsdaten umfassen, die Neuronen oder Schichten eines neuronalen Netzwerks entsprechen, das in Aspekten einer oder mehrerer Ausführungsformen trainiert und/oder zum Inferenzieren verwendet wird. In mindestens einer Ausführungsform speichert die Code- und /oder Datenspei-

cherung **605** Gewichtsparameter und/oder Eingangs-/Ausgangsdaten jeder Schicht eines neuronalen Netzwerks, das in Verbindung mit einem oder mehreren Ausführungsformen während der Rückwärtspropagierung von Eingangs-/Ausgangsdaten und/oder von Gewichtsparametern während des Trainings und/oder des Inferenzierens unter Verwendung von Aspekten einer oder mehrerer Ausführungsformen trainiert oder verwendet wird. In mindestens einer Ausführungsform kann die Trainingslogik **615** die Code- und/oder Datenspeicherung **605** umfassen oder mit dieser gekoppelt sein, um einen Graphencode oder andere Software zum Steuern des Timings und/oder der Reihenfolge zu speichern, in der Gewichts- und/oder andere Parameterinformationen zum Konfigurieren von Logik einschließlich Ganzzahl- und/oder Gleitkommaeinheiten (zusammen als arithmetische Logikeinheiten (ALUs) bezeichnet) zu laden sind.

[0079] In mindestens einer Ausführungsform lädt der Code, wie beispielsweise der Graphencode, basierend auf einer Architektur eines neuronalen Netzwerks, dem dieser Code entspricht, Gewichts- oder andere Parameterinformationen in Prozessor-ALUs. In mindestens einer Ausführungsform kann ein beliebiger Teil der Code- und/oder Datenspeicherung **605** mit einer anderen On-Chip- oder Off-Chip-Datenspeicherung, einschließlich des L1-, L2- oder L3-Cache oder Systemspeichers eines Prozessors, enthalten sein. In mindestens einer Ausführungsform kann jeder Teil der Code- und/oder Datenspeicherung **605** intern oder extern zu einem oder mehreren Prozessoren oder anderen Hardware-Logikvorrichtungen oder -Schaltungen sein. In mindestens einer Ausführungsform kann die Code- und/oder Datenspeicherung **605** ein Cache-Speicher, DRAM, SRAM, ein 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.

[0080] In mindestens einer Ausführungsform können die Code- und/oder Datenspeicherung **601** und die Code- und/oder Datenspeicherung **605** separate Speicherstrukturen sein. In mindestens einer Ausführungsform können die Code- und/oder Datenspeicherung **601** und die Code- und/oder Datenspeicherung **605** dieselbe Speicherstruktur sein. In mindestens einer Ausführungsform können die Code- und/oder Datenspeicherung **601** und die Code- und/oder Datenspeicherung **605** teilweise dieselbe Speicherstruktur und teilweise separate Speicherstrukturen sein. In mindestens einer Ausführungsform kann ein beliebiger Teil der Code- und/oder Datenspeicherung **601** und der Code- und/oder Datenspeicherung **605** mit anderen On-Chip- oder Off-Chip-Datenspeicherungen, einschließlich des L1-, L2- oder L3-Cache oder Systemspeichers eines Prozessors, enthalten sein.

[0081] In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik **615**, ohne darauf beschränkt zu sein, eine oder mehrere Arithmetiklogikeinheiten („ALU(s)“) **610** umfassen, einschließlich Ganzzahl- und/oder Gleitkommaeinheiten, um logische und/oder mathematische Operationen durchzuführen, die mindestens teilweise auf Trainings- und/oder Inferenzcode (beispielsweise Graphencode) basieren oder durch diesen angegeben werden, deren Ergebnis Aktivierungen (z.B. Ausgangswerte von Schichten oder Neuronen innerhalb eines neuronalen Netzwerks) erzeugen kann, die in einer Aktivierungsspeicherung **620** gespeichert sind, die Funktionen von Eingangs-/Ausgangs- und/oder Gewichtsparameterdaten sind, die in der Code- und/oder Datenspeicherung **601** und/oder der Code- und/oder Datenspeicherung **605** gespeichert sind. In mindestens einer Ausführungsform werden in der Aktivierungsspeicherung **620** gespeicherte Aktivierungen in Übereinstimmung mit linearer algebraischer und/oder matrixbasierter Mathematik erzeugt, die von den ALU(s) **610** im Ansprechen auf das Durchführen von Anweisungen oder anderem Code durchgeführt wird, wobei Gewichtswerte, die in der Code- und/oder Datenspeicherung **605** und/oder der Datenspeicherung **601** gespeichert sind, als Operanden zusammen mit anderen Werten, wie beispielsweise Biaswerten, Gradienteninformationen, Impulswerten oder anderen Parametern oder Hyperparametern, verwendet werden, von denen einige oder alle in der Code- und/oder Datenspeicherung **605** oder dem Code und/oder der Code- und/oder Datenspeicherung **601** oder einer anderen Speicherung auf oder außerhalb des Chips gespeichert sein können.

[0082] In mindestens einer Ausführungsform ist bzw. sind die ALU(s) **610** in einem oder mehreren Prozessoren oder anderen Hardware-Logikvorrichtungen oder -Schaltungen enthalten, während in einer anderen Ausführungsform die ALU(s) **610** extern zu einem Prozessor oder einer anderen Hardware-Logikvorrichtung oder Schaltung sein kann bzw. können, die sie verwendet (z.B. ein Co-Prozessor). In mindestens einer Ausführungsform können die ALUs **610** in den Ausführungseinheiten eines Prozessors oder anderweitig in einer Bank von ALUs enthalten sein, auf welche die Ausführungseinheiten eines Prozessors entweder innerhalb desselben Prozessors oder verteilt auf verschiedene Prozessoren unterschiedlichen Typs (z.B. Zentraleinheiten, Graphikverarbeitungseinheiten, feste Funktionseinheiten usw.) zugreifen können. In mindestens einer

Ausführungsform können die Code- und/oder Datenspeicherung **601**, die Code- und/oder Datenspeicherung **605** und die Aktivierungsspeicherung **620** einen Prozessor oder eine andere Hardware-Logikvorrichtung oder -schaltung gemeinsam nutzen, wohingegen sie in einer anderen Ausführungsform in unterschiedlichen Prozessoren oder anderen Hardware-Logikvorrichtungen oder -Schaltungen oder einer Kombination aus gleichen und unterschiedlichen Prozessoren oder anderen Hardware-Logikvorrichtungen oder -Schaltungen sein können. In mindestens einer Ausführungsform kann ein beliebiger Teil der Aktivierungsspeicherung **620** mit einer anderen On-Chip- oder Off-Chip-Datenspeicherung, einschließlich des L1-, L2- oder L3-Cache oder Systemspeichers eines Prozessors, enthalten sein. Ferner kann Inferenz- und/oder Trainingscode mit anderem Code gespeichert sein, auf den ein Prozessor oder eine andere Hardware-Logik oder -Schaltung zugreifen kann und der unter Verwendung der Abruf-, Decodier-, Planungs-, Ausführungs-, Ausscheidungs- und/oder anderer logischer Schaltungen eines Prozessors abgerufen und/oder verarbeitet wird.

[0083] In mindestens einer Ausführungsform kann die Aktivierungsspeicherung **620** ein Cache-Speicher, DRAM, SRAM, nichtflüchtiger Speicher (z.B. Flash-Speicher) oder eine andere Speicherung sein. In mindestens einer Ausführungsform kann die Aktivierungsspeicherung **620** ganz oder teilweise innerhalb oder außerhalb eines oder mehrerer Prozessoren oder anderer logischer Schaltungen sein. In mindestens einer Ausführungsform kann die Wahl, ob der Aktivierungsspeicherung **620** beispielsweise intern oder extern zu einem Prozessor ist oder aus DRAM, SRAM, Flash-Speicher oder einer anderen Speichertyp besteht, von der On-Chip gegenüber Off-Chip verfügbarer Speicherung, Latenzanforderungen der durchgeführten Trainings- und/oder Inferenzfunktionen, der Losgröße der Daten, die beim Inferenzieren und/oder dem Training eines neuronalen Netzwerks verwendet werden, oder einer Kombination dieser Faktoren abhängen. In mindestens einer Ausführungsform kann die in **Fig. 6C** dargestellte Inferenz- und/oder Trainingslogik **615** in Verbindung mit der Hardware einer anwendungsspezifischen integrierten Schaltung („ASIC“), wie beispielsweise der Tensorflow®-Verarbeitungseinheit 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. 6B** dargestellte Inferenz- und/oder Trainingslogik **615** in Verbindung mit Hardware der Zentralverarbeitungseinheit (CPU), mit Hardware der Graphikverarbeitungseinheit (GPU) oder anderer Hardware, wie beispielsweise feldprogrammierbaren Gate Arrays (FPGAs), verwendet werden.

[0084] In mindestens einer Ausführungsform kann, wie in **Fig. 6C** veranschaulicht, die Inferenz- und/oder Trainingslogik **615**, ohne darauf beschränkt zu sein, eine Hardwarelogik umfassen, bei der Rechenressourcen fest zugeordnet sind oder anderweitig exklusiv in Verbindung mit Gewichtswerten oder anderen Informationen verwendet werden, die einer oder mehreren Schichten von Neuronen innerhalb eines neuronalen Netzwerks entsprechen. In mindestens einer Ausführungsform kann die in **Fig. 6C** veranschaulichte Inferenz- und/oder Trainingslogik **615** in Verbindung mit einer anwendungsspezifischen integrierten Schaltung (ASIC), wie beispielsweise der Tensorflow® Processing 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. 6B** veranschaulichte Inferenz- und/oder Trainingslogik **615** in Verbindung mit Hardware der Zentralverarbeitungseinheit (CPU), Hardware der Graphikverarbeitungseinheit (GPU) oder anderer Hardware, wie beispielsweise feldprogrammierbaren Gate Arrays (FPGAs), verwendet werden. In mindestens einer Ausführungsform umfasst die Inferenz- und/oder Trainingslogik **615**, ohne darauf beschränkt zu sein, die Code- und/oder Datenspeicherung **601** und die Code- und/oder Datenspeicherung **605**, die verwendet werden können, um Code (z.B., Graphencode), Gewichtswerte und/oder andere Informationen zu speichern, die Bias-Werte, Gradienteninformationen, Momentenwerten und/oder andere Parameter- oder Hyperparameterinformationen umfassen. In mindestens einer in **Fig. 6C** veranschaulichten Ausführungsform ist jede der Code- und/oder Datenspeicherung **601** und der Code- und/oder Datenspeicherung **605** einer fest zugeordneten Rechenressource zugeordnet, wie beispielsweise der Rechenhardware **602** bzw. Rechenhardware **606**.

[0085] In mindestens einer Ausführungsform entsprechen jeweils die Code- und/oder Datenspeicherung **601** und **605** und die entsprechende Rechenhardware **602** und **606** jeweils verschiedenen Schichten eines neuronalen Netzwerks, so dass eine aus einem „Speicher-/Berechnungspaar 601/602“ der Code- und/oder Datenspeicherung **601** und der Rechenhardware **602** resultierende Aktivierung als eine Eingabe in das nächste „Speicher-/Berechnungspaar 605/606“ der Code- und/oder Datenspeicherung **605** und der Rechenhardware **606** bereitgestellt wird, um die konzeptionelle Organisation eines neuronalen Netzwerks zu spiegeln. In mindestens einer Ausführungsform kann jedes der Speicher-Berechnungspaare **601/602** und **605/606** mehr als einer neuronalen Netzwerkschicht entsprechen. In mindestens einer Ausführungsform können zusätzliche Speicher-/Berechnungspaare (nicht gezeigt) im Anschluss an oder parallel zu den Speicher-/Berechnungspaaren **601/602** und **605/606** in die Inferenz- und/oder Trainingslogik **615** aufgenommen sein.

COMPUTERSYSTEME

[0086] Fig. 7A 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, um die intelligente Steuerung des mobilen Rechenzentrums mit einem mobilen Rechenzentrumkühlsystem zu unterstützen und/oder zu ermöglichen, gemäß mindestens einer Ausführungsform umfassen kann. In mindestens einer Ausführungsform kann ein Computersystem 700A, ohne darauf beschränkt zu sein, eine Komponente umfassen, wie beispielsweise einen Prozessor 702, 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 das Computersystem 700A Prozessoren, wie beispielsweise die PENTIUM®-Prozessorfamilie, Xeon™, 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 das Computersystem 700A 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.

[0087] In mindestens einer Ausführungsform kann das beispielhafte Computersystem 700A eine oder mehrere der Komponenten 110-116 (von Fig. 1) aufnehmen, um Verarbeitungsaspekte für die intelligente Steuerung für das mobile Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem zu unterstützen. Aus mindestens diesem Grund veranschaulicht in einer Ausführungsform Fig. 7A ein System, das miteinander verbundene Hardwarevorrichtungen oder „Chips“ umfasst, wohingegen in anderen Ausführungsformen Fig. 7A may eine beispielhaftes System-on-chip („SoC“) veranschaulichen kann. In mindestens einer Ausführungsform können Vorrichtungen mit proprietären Zwischenverbindungen, standardisierten Zwischenverbindungen (z.B. PCIe) oder einer Kombination davon miteinander verbunden sein. In mindestens einer Ausführungsform sind eine oder mehrere Komponenten des Computersystems 700B miteinander unter Verwendung von Express Link (CXL) Zwischenverbindungen verbunden. Die Inferenz- und/oder Trainingslogik 615 wird verwendet, um Inferenzieren und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind, wie zuvor beispielsweise mit Bezug auf Fig. 6A-C erläutert. Einzelheiten hinsichtlich der Inferenz- und/oder Trainingslogik 615 werden nachstehend in Verbindung mit Fig. 6A-C bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 615 in dem System von Fig. 7A 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.

[0088] 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.

[0089] In mindestens einer Ausführungsform kann ein Computersystem 700A, ohne darauf beschränkt zu sein, den Prozessor 702 umfassen, der, ohne darauf beschränkt zu sein, eine oder mehrere Ausführungseinheiten 708 umfassen kann, um in Übereinstimmung mit hier beschriebenen Techniken ein Training eines maschinell lernenden Modells und/oder Inferenzieren durchzuführen. In mindestens einer Ausführungsform ist das Computersystem 700A ein Einzelprozessor-Desktop- oder Serversystem, wobei in einer anderen Ausführungsform das Computersystem 700A jedoch ein Multiprozessorsystem sein kann. In mindestens einer Ausführungsform kann der Prozessor 702, 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 der Prozessor 702 mit einem Prozessorbus 710 gekoppelt sein, der Datensignale zwischen dem Prozessor 702 und anderen Komponenten in dem Computersystem 700A übertragen kann.

[0090] In mindestens einer Ausführungsform kann der Prozessor **702**, ohne darauf beschränkt zu sein, einen internen Level-1 („L1“)-Cache-Speicher („Cache“) **704** umfassen. In mindestens einer Ausführungsform kann der Prozessor **702** einen einzelnen internen Cache oder mehrere Ebenen eines internen Caches aufweisen. In mindestens einer Ausführungsform kann sich der Cache-Speicher extern zu dem Prozessor **702** 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 **706** verschiedene Arten von Daten in verschiedenen Registern speichern, einschließlich, jedoch nicht beschränkt auf, Ganzzahlregister, Gleitkommaregister, Zustandsregister und einem Befehlszeigerregister.

[0091] In mindestens einer Ausführungsform befindet sich die Ausführungseinheit **708**, jedoch ohne darauf beschränkt zu sein, die Logik zum Ausführen von Ganzzahl- und Gleitkommaoperationen umfasst, ebenfalls in dem Prozessor **702**. In mindestens einer Ausführungsform kann der Prozessor **702** auch einen Mikrocode („ucode“)-Nur-Lese-Speicher („ROM“) umfassen, der Mikrocode für bestimmte Makrobefehle speichert. In mindestens einer Ausführungsform kann die Ausführungseinheit **708** Logik zur Handhabung eines gepackten Befehlssatzes **709** umfassen. In mindestens einer Ausführungsform können, indem der gepackte Befehlssatz **709** in den Befehlssatz eines Universalprozessors 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 Universalprozessor **702** durchgeführt werden. In einer oder mehreren Ausführungsformen können viele Multimedia-Anwendungen beschleunigt und effizienter ausgeführt werden, indem die volle Breite eines Datenbusses des Prozessors für die Ausführung von Operationen auf gepackten Daten verwendet wird, welches die Notwendigkeit eliminieren kann, kleinere Dateneinheiten über den Datenbus des Prozessors zu übertragen, um eine oder mehrere Operationen jeweils Datenelement für Datenelement durchzuführen.

[0092] In mindestens einer Ausführungsform kann die Ausführungseinheit **708** auch in Mikrocontrollern, eingebetteten Prozessoren, Graphikvorrichtungen, DSPs und anderen Arten von Logikschaltungen verwendet werden. In mindestens einer Ausführungsform kann das Computersystem **700A**, ohne darauf beschränkt zu sein, einen Speicher **720** umfassen. In mindestens einer Ausführungsform kann der Speicher **720** als eine Dynamic Random Access Memory („DRAM“)-Vorrichtung, als eine Static Random Access Memory („SRAM“)-Vorrichtung, als eine Flash-Speichervorrichtung oder eine andere Speichervorrichtung implementiert sein. In mindestens einer Ausführungsform kann der Speicher **720** eine oder mehrere Anweisungen **719** und/oder Daten **721** speichern, die durch Datensignale repräsentiert werden, die von dem Prozessor **702** ausgeführt werden können.

[0093] In mindestens einer Ausführungsform kann ein Systemlogikchip mit dem Prozessorbus **710** und dem Speicher **720** gekoppelt sein. In mindestens einer Ausführungsform kann der Systemlogikchip, ohne darauf beschränkt zu sein, einen Speichercontroller-Hub („MCH“) **716** umfassen, und der Prozessor **702** kann über den Prozessorbus **710** mit dem MCH **716** kommunizieren. In mindestens einer Ausführungsform kann der MCH **716** einen Speicherpfad **718** mit hoher Bandbreite zu dem Speicher **720** für Befehls- und Datenspeicherung sowie zur Speicherung von Graphikbefehlen, Daten und Texturen bereitstellen. In mindestens einer Ausführungsform kann der MCH **716** Datensignale zwischen dem Prozessor **702**, dem Speicher **720** und anderen Komponenten in dem Computersystem **700A** leiten und Datensignale zwischen dem Prozessorbus **710**, dem Speicher **720** und einer System-E/A-Schnittstelle **722** überbrücken. In mindestens einer Ausführungsform kann ein Systemlogikchip einen Graphikport zur Kopplung an einen Graphik-Controller bereitstellen. In mindestens einer Ausführungsform kann der MCH **716** über einen Speicherpfad **718** mit hoher Bandbreite mit dem Speicher **720** gekoppelt sein, und eine Graphik-/Video-Karte **718** kann über eine Accelerated Graphics Port („AGP“) Zwischenverbindung **714** mit dem MCH **716** gekoppelt sein.

[0094] In mindestens einer Ausführungsform kann das Computersystem **700A** die System-E/A-Schnittstelle **722** als einen proprietären Hub-Schnittstellenbus verwenden, um den MCH **716** mit einem E/A-Controller-Hub („ICH“) **730** zu verbinden. In mindestens einer Ausführungsform kann der ICH **730** 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 dem Speicher **720**, einem Chipsatz und dem Prozessor **702** umfassen. Beispiele können, ohne darauf beschränkt zu sein, einen Audio-Controller **729**, einen Firmware-Hub („Flash-BIOS“) **728**, einen drahtlosen Transceiver **726**, eine Datenspeicherung **724**, einen Legacy-E/A-Controller **723** mit Benutzereingabe und Tastaturschnittstellen **725**, einen seriellen Erweiterungsport **727**, wie beispielsweise einen Universal Serial Bus („USB“) Port, und ein Netzwerk-Controller **734** sein. Die Datenspeicherung **724** kann ein Festplattenlaufwerk, ein Diskettenlaufwerk, eine CD-ROM-Vorrichtung, eine Flash-Speichervorrichtung oder eine andere Massenspeichervorrichtung umfassen.

[0095] Fig. 7B ist ein Blockdiagramm, das eine elektronische Vorrichtung 700B zur Verwendung eines Prozessors 710 darstellt, um eine intelligente Steuerung des hier beschriebenen mobilen Rechenzentrums mit einem mobilen Rechenzentrumkühlsystem zu unterstützen, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann die elektronische Vorrichtung 700B 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. In mindestens einer Ausführungsform kann die beispielhafte elektronische Vorrichtung 700B eine oder mehrere Komponenten 328, 332 (von Fig. 3D) aufnehmen, um Verarbeitungsaspekte für das mobile Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem zu unterstützen.

[0096] In mindestens einer Ausführungsform kann die elektronische Vorrichtung 700, ohne darauf beschränkt zu sein, den Prozessor 710 umfassen, der kommunikativ mit einer beliebigen geeigneten Anzahl oder Art von Komponenten, Peripheriegeräten, Modulen oder Vorrichtungen gekoppelt ist. In mindestens einer Ausführungsform ist der Prozessor 710 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. 7B ein System, welches miteinander verbundene Hardware-Vorrichtungen oder „Chips“ umfasst, während in anderen Ausführungsformen Fig. 7B ein beispielhaftes System auf einem Chip („SoC“) darstellen kann. In mindestens einer Ausführungsform können die in Fig. 7B veranschaulichten 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. 7B unter Verwendung von Compute Express Link (CXL)-Zwischenverbindungen miteinander verbunden.

[0097] In mindestens einer Ausführungsform kann Fig. 7B eine Anzeige 724, einen berührungsempfindlichen Bildschirm bzw. Touchscreen 725, ein berührungsempfindliches Feld bzw. Touchpad 730, eine Nahfeld-Kommunikationseinheit (Near Field Communications Unit; „NFC“) 745, einen Sensor-Hub 740, einen Thermosensor 746, einen Express-Chipsatz („EC“) 735, ein Trusted Platform Module („TPM“) 738, BIOS/Firmware/Flash-Speicher („BIOS, FW-Flash“) 722, einen DSP 760, ein Laufwerk („SSD oder HDD“) 712, wie beispielsweise eine Solid State Disk („SSD“) oder eine Festplatte („HDD“), eine Wireless Local Area Network Unit („WLAN“) 750, eine Bluetooth-Einheit 752, eine Wireless Wide Area Network Unit („WWAN“) 756, ein Global Positioning System (GPS) Einheit 755, eine Kamera („USB 3.0 Camera“) 754, wie beispielsweise eine USB 3.0-Kamera, oder eine Low Power Double Data Rate („LPDDR“)-Speichereinheit („LPDDR3“) 715, die beispielsweise in einem LPDDR3-Standard implementiert ist, umfassen. Diese Komponenten können jeweils in beliebig geeigneter Art und Weise implementiert sein.

[0098] In mindestens einer Ausführungsform können andere Komponenten durch die hier beschriebenen Komponenten kommunikativ mit dem Prozessor 710 gekoppelt sein. In mindestens einer Ausführungsform können ein Beschleunigungssensor 741, ein Umgebungslichtsensor (Ambient Light Sensor; „ALS“) 742, ein Kompass 743 und ein Gyroskop 744 kommunikativ mit dem Sensor-Hub 740 gekoppelt sein. In mindestens einer Ausführungsform können ein Thermosensor 739, ein Lüfter 737, eine Tastatur 736 und ein Touchpad 730 kommunikativ mit dem EC 735 gekoppelt sein. In mindestens einer Ausführungsform können Lautsprecher 763, Kopfhörer 764 und ein Mikrofon („mic“) 765 kommunikativ mit einer Audioeinheit („Audio-Codec und Class D-Amp“) 762 gekoppelt sein, die wiederum kommunikativ mit dem DSP 760 gekoppelt sein kann. In mindestens einer Ausführungsform kann eine Audioeinheit 762, beispielsweise und ohne darauf beschränkt zu sein, einen Audio-Codierer/Decodierer („Codec“) und einen Klasse-D-Verstärker umfassen. In mindestens einer Ausführungsform kann die SIM-Karte („SIM“) 757 kommunikativ mit der WWAN-Einheit 756 gekoppelt sein. In mindestens einer Ausführungsform können Komponenten, wie beispielsweise die WLAN-Einheit 750 und die Bluetooth-Einheit 752 sowie auch die WWAN-Einheit 756, in einem Formfaktor der nächsten Generation (Next Generation Form Faktor; „NGFF“) implementiert sein.

[0099] Die Inferenz- und/oder Trainingslogik 615 wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten zu der Inferenz- und/oder Trainingslogik 615 werden hier in Verbindung mit Fig. 6B und/oder 6C bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik x615 in dem System von Fig. 7B 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.

[0100] Fig. 7C veranschaulicht ein Computersystem 700C gemäß mindestens einer Ausführungsform, um die intelligente Steuerung des hier beschriebenen mobilen Rechenzentrums mit einem mobilen Rechenzentrumkühlsystem zu unterstützen und/oder zu ermöglichen. In mindestens einer Ausführungsform umfasst das Computersystem 700C, ohne darauf beschränkt zu sein, einen Computer 771 und einen USB-Stick 770. In mindestens einer Ausführungsform kann der Computer 771, ohne darauf beschränkt zu sein, eine beliebige Anzahl und Art eines(von) Prozessor(en) (nicht gezeigt) und einen Speicher (nicht gezeigt) umfassen. In mindestens einer Ausführungsform umfasst der Computer 771, ohne darauf beschränkt zu sein, einen Server, eine Cloud-Instanz, einen Laptop und einen Desktop-Computer.

[0101] In mindestens einer Ausführungsform umfasst der USB-Stick 772, ohne darauf beschränkt zu sein, eine Verarbeitungseinheit 772, eine USB-Schnittstelle 774 und eine USB-Schnittstellenlogik 773. In mindestens einer Ausführungsform kann die Verarbeitungseinheit 772 ein beliebiges Befehlsausführungssystem, -vorrichtung oder -einrichtung sein, das/die in der Lage ist, Befehle auszuführen. In mindestens einer Ausführungsform kann die Verarbeitungseinheit 773, ohne darauf beschränkt zu sein, eine beliebige Anzahl und Art von Verarbeitungskernen umfassen (nicht gezeigt). In mindestens einer Ausführungsform umfasst die Verarbeitungseinheit oder Kern 772 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 der Verarbeitungseinheit 772 eine Tensor-Verarbeitungseinheit („TPC“), die für die Durchführung von Inferenzoperationen des maschinellen Lernens optimiert ist. In mindestens einer Ausführungsform ist die Verarbeitungseinheit 772 eine Visionsverarbeitungseinheit („VPU“), die optimiert ist, um Maschinensicht- und Maschinenlern-Inferenzoperationen durchzuführen.

[0102] In mindestens einer Ausführungsform kann die USB-Schnittstelle 774 jede beliebige Art von USB-Verbinde r oder USB-Buchse sein. Beispielsweise ist in mindestens einer Ausführungsform die USB-Schnittstelle 774 eine USB 3.0 Typ-C Buchse für Daten und Leistung. In mindestens einer Ausführungsform ist die USB-Schnittstelle 774 ein USB 3.0 Typ-A-Verbinde r. In mindestens einer Ausführungsform kann die USB-Schnittstellenlogik 773 jede beliebige Menge und Art von Logik umfassen, die es der Verarbeitungseinheit 772 ermöglicht, über den USB-Verbinde r 774 mit Vorrichtungen (z.B. dem Computer 771) zu kommunizieren.

[0103] Die Inferenz- und/oder Trainingslogik 615, wie mit Bezug auf Fig. 6B und Fig. 6C erläutert, wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenz- und/oder Trainingslogik 615 werden nachstehend in Verbindung mit Fig. 6B und/oder 6C bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik 615 im System von Fig. 7C 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 beschriebener Anwendungsfällen neuronaler Netzwerke berechnet wurden.

[0104] Fig. 8 veranschaulicht ein weiteres beispielhaftes Computersystem 800 gemäß mindestens einer Ausführungsform, um verschiedene Prozesse und Verfahren für das mobile Rechenzentrum mit einem Rechenzentrumkühlsystem zu implementieren, das in dieser gesamten Ausführungsform beschrieben ist. In mindestens einer Ausführungsform umfasst das Computersystem 800, ohne darauf beschränkt zu sein, mindestens eine zentrale Verarbeitungseinheit („CPU“) 802, die mit einem Kommunikationsbus 810 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-Kommunikationsprotokollen implementiert ist. In mindestens einer Ausführungsform umfasst das Computersystem 800, ohne darauf beschränkt zu sein, einen Hauptspeicher 804 und eine Steuerlogik (die z.B. als Hardware, Software oder eine Kombination davon implementiert ist), und Daten werden in dem Hauptspeicher 804 gespeichert, der die Form eines Direktzugriffsspeichers („RAM“) annehmen kann. In mindestens einer Ausführungsform stellt ein Netzwerkschnittstellen-Teilsystem („Network Interface“) 822 eine Schnittstelle zu anderen Rechenvorrichtungen und Netzwerken zum Empfangen von Daten von und Übertragen von Daten an andere Systeme mit dem Computersystem 800 bereit.

[0105] In mindestens einer Ausführungsform umfasst das Computersystem 800 in mindestens einer Ausführungsform, ohne darauf beschränkt zu sein, Eingabevorrichtungen 808, ein Parallelverarbeitungssystem 812 und Anzeigevorrichtungen 806, die unter Verwendung einer herkömmlichen Kathodenstrahlröhre („CRT“), ei-

ner Flüssigkristallanzeige („LCD“), Leuchtdioden („LED“), einer Plasmaanzeige oder anderer geeigneter Anzeigetechnologien. In mindestens einer Ausführungsform wird eine Benutzereingabe von Eingabegeräten **808** wie beispielsweise einer Tastatur, einer Maus, einem Touchpad, einem Mikrofon usw. empfangen. In mindestens einer Ausführungsform kann der vorstehenden Module auf einer einzigen Halbleiterplattform angeordnet sein, um ein Verarbeitungssystem zu bilden.

[0106] Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind, wie zuvor beispielsweise mit Bezug auf **Fig. 6A-C** erläutert. Einzelheiten hinsichtlich der Inferenz- und/oder Trainingslogik **615** werden nachstehend in Verbindung mit **Fig. 6A-C** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik **615** im System von **Fig. 8** 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 beschriebener Anwendungsfällen neuronaler Netzwerke berechnet wurden. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik **615** im System von **Fig. 8** 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 beschriebener Anwendungsfällen neuronaler Netzwerke berechnet wurden.

[0107] **Fig. 9A** veranschaulicht eine beispielhafte Architektur, in der mehrere GPUs **910-913** kommunikativ mit mehreren Mehrkern-Prozessoren **905-906** über Hochgeschwindigkeitsverbindungen **940-943** (z.B. Busse, Punkt-zu-Punkt-Zwischenverbindungen, usw.) gekoppelt ist. In einer Ausführungsform unterstützen die Hochgeschwindigkeitsverbindungen **940-943** 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.

[0108] Außerdem sind in einer Ausführungsform zwei oder mehr GPUs **910-913** über Hochgeschwindigkeitsverbindungen **929-930** miteinander verbunden, die unter Verwendung von ähnlichen oder unterschiedlichen Protokollen/Verbindungen als diejenigen implementiert sein können, die für die Hochgeschwindigkeitsverbindungen **940-943** verwendet werden. Auf ähnliche Weise können zwei oder mehrere Mehrkern-Prozessoren **905-906** über eine Hochgeschwindigkeitsverbindung **928** verbunden sein, welche symmetrische Mehrkern-Prozessor-(SMP)-Busse sein können, die mit 12 GB/s, 30 GB/s, 120 GB/s oder mehr arbeiten. Alternativ kann die gesamte Kommunikation zwischen in **Fig. 9A** gezeigten, verschiedenen Systemkomponenten unter Verwendung derselben Protokolle/Verbindungen (z.B. über eine gemeinsame Zwischenverbindungsstruktur) ausgeführt werden.

[0109] In einer Ausführungsform ist jeder Mehrkern-Prozessor **905-906** mit einem Prozessorspeicher **901-902** jeweils über die Speicher-Zwischenverbindungen **926-927** gekoppelt, und jede GPU **910-913** ist jeweils über GPU-Speicher-Zwischenverbindungen **950-953** kommunikativ mit dem GPU-Speicher **920-923** gekoppelt. Speicher-Zwischenverbindungen **926-927** und **950-953** können dieselben oder unterschiedliche Speicherzugriffstechnologien nutzen. Beispielsweise und nicht beschränkend können die Prozessorspeicher **901-902** und GPU-Speicher beispielsweise 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 **901-902** flüchtiger Speicher und ein anderer Teil nichtflüchtiger Speicher sein (z.B. unter Verwendung einer zweistufigen Speicher (2LM)-Hierarchie).

[0110] Wie nachstehend beschrieben kann, obwohl verschiedene Mehrkernprozessoren **905-906** und GPUs **910-913** physikalisch mit jeweils einem bestimmten Speicher **901-902**, **920-923** gekoppelt sein können, eine vereinheitlichte Speicherarchitektur implementiert sein, in welcher ein gleicher virtueller Systemadressraum (auch als „effektiver Adressraum“ bezeichnet) auf verschiedene physikalische Speicher verteilt ist. Beispielsweise können Prozessorspeicher **901-902** jeweils 64 GB Systemspeicheradressraum und GPU-Speicher **920-923** jeweils 32 GB Systemspeicheradressraum umfassen (was insgesamt zu **256** GB adressierbaren Speicher in diesem Beispiel führt).

[0111] Wie an anderer Stelle in dieser Offenbarung erläutert, können mindestens Strömungsraten und zugeordnete Temperaturen für eine erste Ebene eines intelligenten Lernsystems, wie beispielsweise eines neuronalen Netzwerksystems, festgelegt werden. Da die erste Ebene die vorherigen Daten darstellt, stellt sie ebenfalls eine kleinere Teilmenge der Daten dar, die verfügbar sein können, um das System durch Umtrainieren des Systems zu verbessern. Das Testen und Trainieren kann unter Verwendung der mehreren Prozesseureinheiten

parallel durchgeführt werden, so dass das intelligente Lernsystem robust ist. Eine Architektur, wie beispielsweise in **Fig. 9A**, kann verwendet werden. Wenn Konvergenz für das intelligente Lernsystem erreicht ist, wird eine Menge an Datenpunkte und die Daten in den Datenpunkten, die verwendet werden, um die Konvergenz zu bewirken, vermerkt. Die Daten und Datenpunkte können verwendet werden, um das mobile Rechenzentrum mit einem mobile Rechenzentrumskühlsystem zu steuern, wie beispielsweise mit Bezugnahme auf **Fig. 2-5** beschrieben.

[0112] **Fig. 9B** veranschaulicht zusätzliche Einzelheiten für eine Zwischenverbindung zwischen einem Mehrkern-Prozessor **907** und einem Graphikbeschleunigungsmodul **946** gemäß einer beispielhaften Ausführungsform. Das Graphikbeschleunigungsmodul **946** kann einen oder mehrere GPU-Chips umfassen, die auf einer Verdrahtungskarte integriert sind, welche über eine Hochgeschwindigkeitsverbindung **940** (z.B. ein PCI-Bus, NVLink usw.) mit dem Prozessor **907** gekoppelt ist. Alternativ kann das Graphikbeschleunigungsmodul **946** auf einem Gehäuse oder Chip mit dem Prozessor **907** integriert sein.

[0113] In mindestens einer Ausführungsform umfasst der Prozessor **907** mehrere Kernen **960A-960D** jeweils mit einem Translation-Lookaside-Puffer („TLB“) **961A-961D** und einem oder mehrere Caches **962A-962D**. In mindestens einer Ausführungsform können Kerne **960A-960D** verschiedene andere Komponenten zur Ausführung von Anweisungen und zur Verarbeitung von Daten umfassen, die nicht veranschaulicht sind. In mindestens einer Ausführungsform können die Caches **962A-962D** Level-1 (L1)- und Level-2 (L2)-Caches umfassen. Außerdem können ein oder mehrere gemeinsam genutzte Caches **956** in den Caches **962A-962D** umfasst sein und von Sätzen von Kernen **960A-960D** gemeinsam genutzt werden. In mindestens einer Ausführungsform umfasst der Prozessors **907** 24 Kerne jeweils mit seinem eigenen L1-Cache, zwölf gemeinsam genutzte L2-Caches und zwölf gemeinsam genutzte L3-Caches. In dieser Ausführungsform werden ein oder mehrere L2- und L3-Caches von zwei benachbarten Kernen gemeinsam genutzt. Der Prozessor **907** und das Graphikbeschleunigungsmodul **946** sind mit dem Systemspeicher **914** verbunden, der die Prozessorspeicher **901-902** von **Fig. 9A** umfassen kann.

[0114] Kohärenz wird für in verschiedenen Caches **962A-962D**, **956** und dem Systemspeicher **914** gespeicherte Daten und Anweisungen über Inter-Kern-Kommunikation über einen Kohärenzbus **964** aufrechterhalten. In mindestens einer Ausführungsform kann jedem Cache eine Cache-Kohärenz-Logik/Schaltung zugeordnet sein, um als Antwort auf erfasste Lese- oder Schreiboperationen auf bestimmte Cache-Zeilen mit dem Kohärenzbus **964** zu kommunizieren. In einer Implementierung ist ein Cache-Beobachtungsprotokoll über den Kohärenzbus **964** implementiert, um Cache-Zugriffe zu beobachten.

[0115] In einer Ausführungsform koppelt eine Proxy-Schaltung (PROXY) **925** das Graphikbeschleunigungsmodul **946** kommunikativ mit dem Kohärenzbus **964**, so dass das Graphikbeschleunigungsmodul **946** als ein Peer der Kerne **960A-960D** an einem Cache-Kohärenzprotokoll teilnehmen kann. Insbesondere stellt eine Schnittstelle (INTF) **935** Konnektivität zur Proxy-Schaltung **925** über die Hochgeschwindigkeitsverbindung **940** (z.B. eine PCIe-Bus, NVLink usw.) und eine Schnittstelle (INTF) **937** verbindet das Graphikbeschleunigungsmodul **946** mit der Hochgeschwindigkeitsverbindung **940**.

[0116] In einer Implementierung stellt eine Beschleuniger-Integrationsschaltung **936** Cache-Verwaltungs-, Speicherzugriffs-, Kontextverwaltungs- und Unterbrechungsverwaltungsdienste im Namen mehrerer Graphikverarbeitungs-Engines **931-932**, N des Graphikbeschleunigungsmoduls **946** bereit. Die Graphikverarbeitungs-Engines **931-932**, N können jeweils eine separate Graphikverarbeitungseinheit (GPU) umfassen. Alternativ können die Graphikverarbeitungs-Engines **931-932**, 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 **946** eine GPU mit mehreren Graphikverarbeitungs-Engines **931-932**, N sein, oder die Graphikverarbeitungs-Engines **931-932**, N können individuelle GPUs sein, die in bzw. auf einem gemeinsamen Gehäuse, einer gemeinsamen Verdrahtungskarte oder einem gemeinsamen Chip integriert sind. Wie es der Fall ist, kann die obige Bestimmung für den Rekonstruktionsparameter und den Rekonstruktionsalgorithmus in GPUs **931-N** von **Fig. 9B** durchgeführt werden.

[0117] In einer Ausführungsform umfasst die Beschleuniger-Integrationsschaltung **936** eine Speicherverwaltungseinheit (MMU) **939** zum Durchführen verschiedener Speicherverwaltungsfunktionen, wie beispielsweise Speicherübersetzungen von virtuell zu physikalisch (auch als Speicherübersetzungen von effektiv zu real bezeichnet) und Speicherzugriffsprotokolle zum Zugreifen auf den Systemspeicher **914**. Die MMU **939** 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 **938** Be-

fehle und Daten für effizienten Zugriff durch die Graphikverarbeitungs-Engines **931-932**, N. In einer Ausführungsform werden die in dem Cache **938** und den Graphikspeichern (GFX MEM) **933-934**, M gespeicherten Daten mit Kerncaches **962A-962D**, **956** und dem Systemspeicher **914** kohärent gehalten. Wie oben erwähnt, kann dies über die Proxy-Schaltung **925** im Namen des Caches **938** und der Speicher **933-934**, M (z.B. Senden von Aktualisierungen an den Cache **938** mit Bezug auf Modifikationen/Zugriffe auf Cache-Zeilen auf Prozessor-Caches **962A-962D**, **956** und Empfangen von Aktualisierungen vom Cache **938**) erreicht werden.

[0118] Ein Satz von Registern **945** speichert Kontextdaten für Threads, die von den Graphikverarbeitungs-Engines **931-932**, N ausgeführt werden, und eine Kontextverwaltungsschaltung **948** verwaltet Thread-Kontexte. In mindestens einer Ausführungsform kann die Kontextverwaltungsschaltung **948** Speicher- 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). In mindestens einer Ausführungsform kann die Kontextverwaltungsschaltung **948** 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) **947** von Systemvorrichtungen empfangene Unterbrechungen.

[0119] In einer Implementierung werden virtuelle/effektive Adressen von einer Graphikverarbeitungs-Engine **931** in reale/physikalische Adressen im Systemspeicher **914** durch die MMU **939** übersetzt. Eine Ausführungsform der Beschleunigerintegrationsschaltung **936** unterstützt mehrere (z.B. 4, 8, 16) Graphikbeschleunigermodule **946** und/oder andere Beschleunigungsvorrichtungen. Das Graphikbeschleunigermodul **946** kann einer einzelnen Anwendung fest zugeordnet sein, die auf dem Prozessor **907** ausgeführt wird, oder kann zwischen mehreren Anwendungen gemeinsam genutzt werden. In einer Ausführungsform wird eine virtualisierte Graphikausführungsumgebung präsentiert, in welcher Ressourcen der Graphikverarbeitungs-Engines **931-932**, 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.

[0120] In mindestens einer Ausführungsform fungiert die Beschleuniger-Integrationsschaltung **936** als eine Brücke zu einem System für das Graphikbeschleunigungsmodul **946** und stellt Adressübersetzungs- und Systemspeicher-Cache-Dienste bereit. Außerdem kann die Beschleuniger-Integrationsschaltung **936** Virtualisierungseinrichtungen für einen Host-Prozessor bereitstellen, um die Virtualisierung von Graphikverarbeitungs-Engines **931-932**, N, Interrupts und der Speicherverwaltung zu verwalten.

[0121] Weil Hardwareressourcen der Graphikverarbeitungs-Engines **931-932**, N explizit auf einen realen Adressraum abgebildet werden, der von dem Host-Prozessor **907** gesehen wird, kann jeder Host-Prozessor diese Ressourcen unter Verwendung eines effektiven Adresswerts direkt adressieren. Eine Funktion der Beschleunigerintegrationsschaltung **936** ist in einer Ausführungsform die physikalische Trennung der Graphikverarbeitungs-Engines **931**, **932**, N, so dass sie einem System als unabhängige Einheiten erscheinen.

[0122] In mindestens einer Ausführungsform sind ein oder mehrere Graphikspeicher **933-934**, M jeweils mit jeder der Graphikverarbeitungs-Engines **931-932**, N gekoppelt. Die Graphikspeicher **933-934**, M speichern Anweisungen und Daten, die von jeder der Graphikverarbeitungs-Engines **931-932**, N verarbeitet werden. Die Graphikspeicher **933-934**, M können flüchtige Speicher, wie beispielsweise DRAMs (einschließlich gestapelter DRAMs), GDDR-Speicher (z.B. GDDR5, GDDR6) oder HBM sein, und/oder können nichtflüchtige Speicher wie beispielsweise 3D XPoint oder Nano-Ram sein.

[0123] In einer Ausführungsform werden, um den Datenverkehr über die Hochgeschwindigkeitsverbindung **940** zu verringern, Biasing-Verfahren verwendet, um sicherzustellen, dass Daten, die in den Graphikspeichern **933-934**, M gespeichert sind, Daten sind, welche am häufigsten von den Graphikverarbeitungs-Engines **931-932**, N verwendet werden und vorzugsweise nicht (wenigstens nicht häufig) von den Kernen **960A-960D** verwendet werden. Auf ähnliche Weise versucht in mindestens einer Ausführungsform ein Biasing-Mechanismus, die von den Kernen (und nicht von den Graphikverarbeitungs-Engines **931**, **932**, N) benötigten Daten in den Caches **962A-962D**, **956** von Kernen und dem Systemspeicher **914** zu halten.

[0124] Fig. 9C veranschaulicht eine weitere beispielhafte Ausführungsform, bei welcher die Beschleunigerintegrationsschaltung **936** innerhalb des Prozessors **907** integriert ist, um eine intelligente Steuerung des mo-

bilien Rechenzentrums mit einem mobilen Rechenzentrumkühlsystem hier gemäß mindestens einer Ausführungsform der Offenbarung zu ermöglichen und/oder zu unterstützen. In mindestens dieser Ausführungsform kommunizieren Graphikverarbeitungs-Engines **931-932**, N direkt über die Hochgeschwindigkeitsverbindung **940** mit der Beschleunigerintegrationsschaltung **936** über die Schnittstelle **937** und die Schnittstelle **935** (die wiederum eine beliebige Form von Bus- oder Schnittstellenprotokoll nutzen können). Die Beschleuniger-Integrationsschaltung **936** kann dieselben Operationen wie diejenigen durchführen, die in Bezug auf **Fig. 9B** beschrieben sind, jedoch potenziell mit einem höheren Durchsatz aufgrund ihrer unmittelbaren Nähe zum Kohärenzbus **964** und den Caches **962A-962D**, **956**. Mindestens 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 **936** gesteuert werden, und Programmiermodelle, die von dem Graphikbeschleunigungsmodul **946** gesteuert werden.

[0125] In mindestens einer Ausführungsform sind die Graphikverarbeitungs-Engines **931-932**, N für eine einzelne Anwendung oder einen einzelnen Prozess unter einem einzigen Betriebssystem fest zugeordnet. In mindestens einer Ausführungsform kann eine einzelne Anwendung andere Anwendungsanforderungen an die Graphikverarbeitungs-Engines **931-932**, N leiten, um eine Virtualisierung innerhalb einer VM/Partition bereitzustellen.

[0126] In mindestens einer Ausführungsform können die Graphikverarbeitungs-Engines **931-932**, N von mehreren VM-/Anwendungs-Partitionen gemeinsam genutzt werden. In mindestens einer Ausführungsform können gemeinsam genutzte Modelle einen Systemhypervisor verwenden, um Graphikverarbeitungs-Engines **931-932**, N zu virtualisieren, um einen Zugriff durch jedes Betriebssystem zu ermöglichen. Für Systeme mit einer einzigen Partition ohne einen Hypervisor sind die Graphikverarbeitungs-Engines **931-932**, N im Besitz eines Betriebssystems. In mindestens einer Ausführungsform kann ein Betriebssystem die Graphikverarbeitungs-Engines **931-932**, N virtualisieren, um einen Zugriff auf jeden Prozess oder jede Anwendung zu ermöglichen.

[0127] In mindestens einer Ausführungsform wählt das Graphikbeschleunigungsmodul **946** oder eine einzelne Graphikverarbeitungs-Engine **931-932**, N ein Prozesselement unter Verwendung eines Prozesshandle aus. In einer Ausführungsform werden Prozesselemente in dem Systemspeicher **914** 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 **931-932**, 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.

[0128] **Fig. 9D** veranschaulicht ein beispielhaftes Beschleunigerintegrations-Slice **990**, um eine intelligente Steuerung des mobilen Rechenzentrums mit einem mobilen Rechenzentrumkühlsystem hier gemäß mindestens einer Ausführungsform der Offenbarung zu ermöglichen und/oder zu unterstützen. Wie hier verwendet, umfasst ein „Slice“ einen bestimmten Teil der Verarbeitungsressourcen der Beschleunigerintegrationsschaltung **936**. Ein anwendungseffektiver Adressraum **982** innerhalb des Systemspeichers **914** speichert Prozesselemente **983**. In einer Ausführungsform werden die Prozesselemente **983** als Antwort auf GPU-Aufrufe **981** aus den auf dem Prozessor **907** ausgeführten Anwendungen **980** gespeichert. Ein Prozesselement **983** enthält den Prozesszustand für die entsprechende Anwendung **980**. Ein in dem Prozesselement **983** enthaltener Arbeitsdeskriptor (WD) **984** 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 **984** ein Zeiger auf eine Auftragsanforderungswarteschlange in dem Adressraum **982** einer Anwendung.

[0129] Das Graphikbeschleunigungsmodul **946** und/oder einzelne Graphikverarbeitungs-Engines **931-932**, 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 eines WD **984** an ein Graphikbeschleunigungsmodul **946** zum Starten eines Auftrags in einer virtualisierten Umgebung enthalten sein.

[0130] In mindestens einer Ausführungsform ist ein dediziertes Prozessprogrammierungsmodell implementierungsspezifisch. In mindestens einer Ausführungsform besitzt in diesem Modell ein einzelner Prozess das Graphikbeschleunigungsmodul **946** oder eine einzelne Graphikverarbeitungs-Engine **931**. Weil das Graphik-

beschleunigungsmodul **946** einem einzelnen Prozess gehört, initialisiert ein Hypervisor die Beschleunigerintegrationsschaltung **936** für eine besitzende Partition und ein Betriebssystem initialisiert die Beschleunigerintegrationsschaltung **936** für einen besitzenden Prozess, wenn das Graphikbeschleunigungsmodul **946** zugewiesen wird.

[0131] Im Betrieb ruft eine WD-Abrufeinheit **991** in dem Beschleunigerintegrations-Slice **990** den nächsten WD **984** ab, welcher eine Angabe der von einer oder mehreren Graphikverarbeitungs-Engines des Graphikbeschleunigungsmoduls **946** zu verrichtenden Arbeiten umfasst. Daten aus dem WD **984** können in Registern **945** gespeichert und von der MMU **939**, der Unterbrechungsverwaltungsschaltung **947** und/oder der Kontextverwaltungsverwaltungsschaltung (CONTEXT MGMT) **948** wie veranschaulicht verwendet werden. Eine Ausführungsform der MMU **939** beinhaltet beispielsweise eine Segment-/Seiten-Durchlauf-Schaltungsanordnung zum Zugreifen auf Segment-/Seitentabellen **986** innerhalb eines virtuellen Adressraums **985** des Betriebssystems (OS). die Unterbrechungsverwaltungsschaltung **947** Unterbrechungsereignisse **992** verarbeiten, die von dem Graphikbeschleunigungsmodul **946** empfangen werden. Beim Durchführung von Graphikoperationen wird eine von einer Graphikverarbeitungs-Engine **931-932**, N erzeugte effektive Adresse **993** von der MMU **939** in eine reale Adresse übersetzt.

[0132] In einer Ausführungsform wird ein und derselbe Satz von Registern **945** für jede Graphikverarbeitungs-Engine **931-932**, N und/oder das Graphikbeschleunigungs-Modul **946** dupliziert und kann von einem Hypervisor oder Betriebssystem initialisiert werden. Jedes dieser duplizierten Register kann in einem Beschleunigerintegrations-Slice **990** umfasst sein. Beispielhafte Register, die von einem Hypervisor initialisiert werden können, werden in Tabelle 1 gezeigt.

Tabelle 1 - Hypervisor-initialisierte Register

Register	Beschreibung
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

[0133] Beispielhafte Register, die von einem Betriebssystem initialisiert werden können, werden in Tabelle 2 gezeigt.

Tabelle 2 - Betriebssystem-initialisierte Register

Register	Beschreibung
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

[0134] In einer Ausführungsform ist jeder WD **984** spezifisch für ein bestimmtes Graphikbeschleunigungsmodul **946** und/oder Graphikverarbeitungs-Engines **931**, **932**, N. Er enthält alle Informationen, die von einer Graphikverarbeitungs-Engine **931-932**, 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.

[0135] Fig. 9E veranschaulicht zusätzliche Einzelheiten für eine beispielhafte Ausführungsform eines gemeinsam genutzten Modells. Diese Ausführungsform umfasst einen Hypervisor-Real-Adressraum **998**, in welchem eine Prozesselementliste **999** gespeichert ist. Der Hypervisor-Real-Adressraum **998** ist über einen Hypervisor **996** zugänglich, welcher Graphikbeschleunigungsmodul-Engines für das Betriebssystem **995** virtualisiert.

[0136] In mindestens einer Ausführungsform ermöglichen gemeinsam genutzte Programmiermodelle die Verwendung eines Graphikbeschleunigungsmoduls **946** für alle oder eine Teilmenge von Prozessen aus allen oder einer Teilmenge von Partitionen in einem System. Es gibt zwei Programmiermodelle, bei denen das Graphikbeschleunigungsmodul **946** von mehreren Prozessen und Partitionen gemeinsam genutzt wird: zeitsliced gemeinsam genutzt und graphikorientiert gemeinsam genutzt.

[0137] In diesem Modell besitzt der Systemhypervisor **996** das Graphikbeschleunigungsmodul **946** und stellt seine Funktion allen Betriebssystemen **995** zur Verfügung. Damit ein Graphikbeschleunigungsmodul **946** eine Virtualisierung durch den Systemhypervisor **996** unterstützen kann, kann das Graphikbeschleunigungsmodul **946** wie folgt eingesetzt werden: 1) Die Auftragsanforderung einer Anwendung muss autonom sein (d.h. der Zustand braucht zwischen Aufträgen nicht aufrechterhalten zu werden), oder das Graphikbeschleunigungsmodul **946** muss einen Mechanismus zum Speichern und Wiederherstellen von Kontext bereitstellen. 2) Das Graphikbeschleunigungsmodul **946** garantiert, dass die Auftragsanforderung einer Anwendung in einer bestimmten Zeitspanne abgeschlossen wird, einschließlich etwaiger Übersetzungsfehler, oder das Graphikbeschleunigungsmodul **946** bietet eine Möglichkeit, der Verarbeitung eines Auftrags zuvorzukommen. 3) Dem Graphikbeschleunigungsmodul **946** muss Fairness zwischen Prozessen garantiert werden, wenn es in einem gezielten, gemeinsam genutzten Programmiermodell arbeitet.

[0138] In mindestens einer Ausführungsform muss die Anwendung **980** einen Systemaufruf des Betriebssystems **995** mit einem Typ des Graphikbeschleunigungsmoduls **946**, 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 **946** eine Soll-Beschleunigungsfunktion für einen Systemaufruf. In mindestens einer Ausführungsform kann der Typ des Graphikbeschleunigungsmoduls **946** ein systemspezifischer Wert sein. In mindestens einer Ausführungsform ist der WD speziell für das Graphikbeschleunigungsmodul **946** formatiert und kann in einer Form eines Befehls des Graphikbeschleunigungsmoduls **946**, eines effektiven Adresszeigers auf eine benutzerdefinierte Struktur, eines effektiven Adresszeigers auf eine Befehlswarteschlange oder einer beliebigen anderen Datenstruktur zur Beschreibung der von dem Graphikbeschleunigungsmodul **946** 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. In mindestens einer Ausführungsform kann, falls die Implementierungen der Beschleunigerintegrationsschaltung **936** und des Graphikbeschleunigungsmoduls **946** ein Benutzerberechtigungsmasken-Übersteuerungsregister (User Authority Mask Override Register; „UAMOR“) nicht unterstützen, ein Betriebssystem einen aktuellen UAMOR-Wert auf einen AMR-Wert anwenden, bevor es einen AMR in einem Hypervisor-Aufruf übergibt. Der Hypervisor **996** kann optional einen aktuellen Berechtigungsmasken-Übersteuerungsregister (Authority Mask Override Register; AMOR)-Wert anwenden, bevor er ein AMR in das Prozesselement **983** platziert. In mindestens einer Ausführungsform ist CSRP eines von Registern **945**, die eine effektive Adresse eines Bereichs im Adressraum **982** einer Anwendung für das Graphikbeschleunigungsmodul **946** zum Speichern und Wiederherstellen des Kontextzustands enthalten. In mindestens einer Ausführungsform ist dieser Zeiger 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.

[0139] Bei Empfangen eines Systemaufrufs kann das Betriebssystem **995** verifizieren, dass die Anwendung **980** registriert ist und die Berechtigung zur Verwendung des Graphikbeschleunigungsmoduls **946** erhalten hat. In mindestens einer Ausführungsform ruft das Betriebssystem **995** dann den Hypervisor **996** mit den in Tabelle 3 gezeigten Informationen auf.

Tabelle 3 - Parameter für Hypervisor-Aufruf durch Betriebssystem

Parameter	Beschreibung
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)

[0140] In mindestens einer Ausführungsform verifiziert bei Empfangen eines Hypervisoraufrufs der Hypervisor **996**, dass das Betriebssystem **995** registriert hat und berechtigt ist, das Graphikbeschleunigungsmodul **946** zu verwenden. In mindestens einer Ausführungsform stellt der Hypervisor **996** dann das Prozesselement **983** in eine verknüpfte Liste von Prozesselementen für einen Typ eines entsprechenden Graphikbeschleunigungsmodul **946** ein. In mindestens einer Ausführungsform kann ein Prozesselement in Tabelle 4 gezeigte Informationen umfassen.

Tabelle 4 - Informationen zu Prozesselementen

Element	Beschreibung
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)

[0141] In mindestens einer Ausführungsform initialisiert der Hypervisor mehrere Register **945** des Beschleunigerintegrations-Slice **990**.

[0142] Wie in **Fig. 9F** 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 **901-902** und GPU-Speicher **920-923** verwendet wird. In dieser Implementierung verwenden Operationen, die auf GPUs **920-923** ausgeführt werden, ein und denselben virtuellen/effektiven Speicheradressraum, um auf die Prozessorspeicher **901-902** zuzugreifen und umgekehrt, wodurch die Programmierbarkeit vereinfacht wird. In mindestens einer Ausführungsform ist ein erster Abschnitt eines virtuellen/effektiven Adressraums dem Prozessorspeicher **901**, ein zweiter Abschnitt dem zweiten Prozessorspeicher **902**, ein dritter Abschnitt dem GPU-Speicher **920** 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 **901-902** und GPU-Speicher **920-923** hinweg verteilt, so dass jeder Prozessor oder jede GPU auf jeden physikalischen Speicher zugreifen kann, wobei eine virtuelle Adresse auf diesen Speicher abgebildet wird.

[0143] In einer Ausführungsform stellen die Bias/Kohärenzverwaltungs-Schaltungen **994A-994E** innerhalb einer oder mehrerer MMUs **939A-939E** die Cache-Kohärenz zwischen Caches eines oder mehrerer Host-Prozessoren (z.B. **905**) und den GPUs **910-913** sicher und implementieren Biasing-Techniken, die physikalische Speicher angeben, in welchen bestimmte Arten von Daten gespeichert werden sollten. Während in **Fig. 9F** mehrere Instanzen der Bias-/Kohärenzverwaltungsschaltungen **994A-994E** veranschaulicht sind, die Bias-/Kohärenzschaltungen innerhalb einer MMU eines oder mehrerer Hostprozessoren **905** und/oder innerhalb der Beschleunigerintegrationsschaltung **936** implementiert sein.

[0144] Eine Ausführungsform ermöglicht GPU-Speichern **920-923** 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 die Möglichkeit, auf den an die GPU gebundenen Speicher **920-923** als Systemspeicher ohne lästigen Cachekohärenz-Overhead zuzugreifen, eine vorteilhafte Betriebsumgebung für GPU-Entlastung. Diese Anordnung ermöglicht Software des Host-Prozessors **905**, Operanden einzurichten und ohne den Aufwand traditioneller E/A-DMA-Datenkopien auf Rechenergebnisse zuzugreifen. In mindestens einer Ausführungsform beinhalten derartige traditionelle Kopien Treiberaufrufe, Unterbrechungen bzw. Interrupts und Memory Mapped E/A (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 **920-923** ohne Cache-Kohärenz-Overheads zuzugreifen, entscheidend für die Ausführungszeit einer entlasteten Berechnung sein. In mindestens einer Ausführungsform kann in Fällen mit wesentlichem Streaming-Speicherschreibe-Datenaufkommen beispielsweise der Cache-Kohärenz-Overhead eine effektive Schreibbandbreite, die von einer GPU **910-913** 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.

[0145] In mindestens einer Ausführungsform wird die Auswahl von GPU-Bias und Host-Prozessor-Bias durch eine Bias-Tracker-Datenstruktur gesteuert. Eine Bias-Tabelle kann verwendet werden, die eine Seitengranularstruktur sein kann (in mindestens einer Ausführungsform kann dies auf eine Granularität einer Speicherseite gesteuert werden), 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 **920-923** mit oder ohne einen Bias-Cache in einer GPU **910-913** implementiert sein (z.B. um häufig/kürzlich verwendete Einträge einer Bias-Tabelle zwischenspeichern). Alternativ kann in mindestens einer Ausführungsform eine gesamte Bias-Tabelle innerhalb einer GPU aufrechterhalten werden.

[0146] In mindestens einer Ausführungsform wird auf einen Bias-Tabelleneintrag zugegriffen, der jedem Zugriff auf den GPU-gebundenen Speicher **920-923** zugeordnet ist, bevor tatsächlich auf einen GPU-Speicher zugegriffen wird, welcher die folgenden Operationen veranlasst. Zunächst werden lokale Anfragen von einer GPU **910-913**, die ihre Seite in GPU-Bias finden, direkt an einen entsprechenden GPU-Speicher **920-923** weitergeleitet. Lokale Anfragen von einer GPU, die ihre Seite in Host-Bias finden, werden an den Prozessor **905** weitergeleitet (z.B. über eine Hochgeschwindigkeitsverbindung, wie oben beschrieben). In mindestens einer Ausführungsform beenden Anfragen von dem Prozessor **905**, 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 eine GPU **910-913** 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.

[0147] Ein Mechanismus zum Ändern des Bias-Zustands setzt einen API-Aufruf (z.B. OpenCL) ein, 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 **905** zur Bias der GPU verwendet, nicht jedoch für einen entgegengesetzten Übergang.

[0148] In einer Ausführungsform wird Cache-Kohärenz durch vorübergehendes Rendern von Seiten mit GPU-Bias, die von dem Hostprozessor **905** nicht zwischengespeichert werden können, aufrechterhalten. Um auf diese Seiten zuzugreifen, kann der Prozessor **905** Zugriff von der GPU **910** anfordern, welche den Zugriff sofort gewähren kann oder auch nicht. Um die Kommunikation zwischen Prozessor **905** und GPU **910** zu verringern,

ist es somit vorteilhaft, sicherzustellen, dass Seiten mit GPU-Bias diejenigen sind, die von einer GPU, jedoch nicht von dem Hostprozessor **905** benötigt werden und umgekehrt.

[0149] Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um eine oder mehrere Ausführungsformen durchzuführen. Einzelheiten hinsichtlich der Inferenz- und/oder Trainingslogik **615** werden nachstehend in Verbindung mit **Fig. 6B** und/oder **6C** bereitgestellt

[0150] **Fig. 10A** veranschaulicht beispielhaft integrierte Schaltungen und zugehörige Graphikprozessoren, die unter Verwendung von einem oder mehreren IP-Kernen hergestellt sein können, gemäß verschiedenen hier beschriebenen Ausführungsformen, um das mobile Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem zu unterstützen oder zu ermöglichen. Zusätzlich zu dem, was veranschaulicht ist, können andere Logik und Schaltungen in mindestens einer Ausführungsform umfasst sein, einschließlich zusätzlicher Graphikprozessoren/Kerne, peripherer Schnittstellen-Controllern oder universeller Prozessorkerne.

[0151] **Fig. 10A** ist ein Blockdiagramm, das ein beispielhaftes System auf einer integrierten Chip-Schaltung **1000A** veranschaulicht, die aus einem oder mehreren IP-Kernen hergestellt sein kann, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform umfasst die integrierte Schaltung **1000A** einen oder mehrere Anwendungsprozessor(en) **1005** (z.B. CPUs), mindestens einen Graphikprozessor **1010** und kann zusätzlich einen Bildprozessor **1015** und/oder einen Videoprozessor **1020** umfassen, von welchen jeder ein modularer IP-Kern sein kann. In mindestens einer Ausführungsform umfasst die integrierte Schaltung **1000A** Peripherie- oder Buslogik mit einem USB-Controller **1025**, einem UART-Controller **1030**, einem SPI/SDIO-Controller **1035** und einem I²S/I²C-Controller **1040**. In mindestens einer Ausführungsform kann die integrierte Schaltung **1000A** eine Anzeigevorrichtung **1045** umfassen, die mit einem oder mehreren eines High-Definition-Multimedia-Schnittstellen-(HDMI)-Controllern **1050** und einer Anzeigeschnittstelle **1055** für ein Mobile Industry Processor Interface (MIPI) gekoppelt ist. In mindestens einer Ausführungsform kann eine Speicherung durch ein Flash-Speicher-Teilsystem **1060** einschließlich eines Flash-Speichers und eines Flash-Speicher-Controllers bereitgestellt sein. In mindestens einer Ausführungsform kann eine Speicherschnittstelle über einen Speicher-Controller **1065** für Zugriff auf SDRAM- oder SRAM-Speichervorrichtungen bereitgestellt sein. In mindestens einer Ausführungsform umfassen einige integrierte Schaltungen zusätzlich eine eingebettete Sicherheits-Engine **1070**.

[0152] Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenz- und/oder Trainingslogik **615** werden hier in Verbindung mit **Fig. 6B** und/oder **6C** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik **615** in der integrierten Schaltung **1000A** 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.

[0153] **Fig. 10B-10C** veranschaulichen beispielhafte integrierte Schaltungen und zugeordnete Graphikprozessoren, die unter Verwendung eines oder mehrerer IP-Kerne hergestellt sein können, gemäß verschiedenen der hier beschriebenen Ausführungsformen, um das mobile Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem zu unterstützen und oder zu ermöglichen. Zusätzlich zu dem, was veranschaulicht ist, können andere Logik und Schaltungen in mindestens einer Ausführungsform umfasst sein, einschließlich zusätzlicher Graphikprozessoren/Kerne, peripherer Schnittstellen-Controllern oder universeller Prozessorkerne.

[0154] **Fig. 10B-10C** sind Blockdiagramme, die beispielhafte Graphikprozessoren zur Verwendung innerhalb eines SoC veranschaulichen, gemäß hier beschriebenen Ausführungsformen, um das mobile Rechenzentrum mit einem mobilen Rechenzentrumkühlsystem zu unterstützen oder zu ermöglichen. In einem Beispiel kann der Graphikprozessor bei der intelligenten Steuerung des mobilen Rechenzentrums mit einem mobilen Rechenzentrumkühlsystem verwendet werden, weil existierende Math-Engines imstande sind, Multi-level neuronale Netzwerke schneller zu verarbeiten. **Fig. 10B** veranschaulicht einen beispielhaften Graphikprozessor **1010** eines Systems auf einer integrierten Chip-Schaltung, das unter Verwendung eines oder mehrerer IP-Kerne hergestellt sein kann, gemäß mindestens einer Ausführungsform. **Fig. 10B** veranschaulicht einen zusätzlichen beispielhaften Graphikprozessor **1040** 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 **1010** von **Fig. 10A** ein Graphikprozessorkern mit niedrigem Stromverbrauch. In mindestens einer Ausführungsform ist der Graphikprozessor **1040** von **Fig. 10C**

ein leistungsfähigerer Graphikprozessorkern. In mindestens einer Ausführungsform kann jeder der Graphikprozessoren **1010**, **1040** eine Variante des Graphikprozessors **1010** von **Fig. 10A** sein.

[0155] In mindestens einer Ausführungsform umfasst der Graphikprozessor **1010** einen Vertex-Prozessor **1005** und einen oder mehrere Fragment-Prozessoren **1015A**, **1015B**, **1015C**, **1015D** bis **1015N-1** und **1015N**). In mindestens einer Ausführungsform kann der Graphikprozessor **1010** verschiedene Shader-Programme über separate Logik ausführen, so dass der Vertex-Prozessor **1005** für die Ausführung von Operationen für Vertex-Shader-Programme optimiert ist, während ein oder mehrere Fragment-Prozessoren **1015A-1015N** Fragment (z.B. Pixel)-Shading-Operationen für Fragment- oder Pixel-Shader-Programme ausführen. In mindestens einer Ausführungsform führt der Vertex-Prozessor **1005** eine Vertex-Verarbeitungsphase einer 3D-Graphikpipeline durch und erzeugt Primitive und Vertex-Daten. In mindestens einer Ausführungsform verwenden der(die) Fragment-Prozessor(en) **1015A-1015N** Primitive- und Vertex-Daten, die von dem Vertex-Prozessor **1005** 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) **1015A-1015N** optimiert, Fragment-Shader-Programme auszuführen, wie sie in einer OpenGL-API vorgesehen sind, welche verwendet werden können, ähnliche Operationen wie ein Pixel-Shader-Programm durchzuführen, wie sie in einer Direct 3D-API vorgesehen sind.

[0156] In mindestens einer Ausführungsform umfasst der Graphikprozessor **1010** zusätzlich eine oder mehrere Speicherverwaltungseinheiten (MMUs) **1020A-1020B**, Cache(s) **1025A-1025B** und Schaltungszwischenverbindungen **1030A-1030B**. In mindestens einer Ausführungsform stellen eine oder mehrere MMU(s) **1020A-1020B** ein Adressmapping von virtuell nach physikalisch für den Graphikprozessor **1010** bereit, einschließlich des Vertex-Prozessors **1005** und/oder des/der Fragment-Prozessoren **1015A-1015N**, die auf Vertex- oder Bild-/Texturdaten Bezug nehmen können, die in Speicher gespeichert sind, zusätzlich zu Vertex- oder Bild-/Textur-Daten, die in einem oder mehreren Cache(s) **1025A-1025B** gespeichert sind. In mindestens einer Ausführungsform können eine oder mehrere MMU(s) **1020A-1020B** mit anderen MMU(s) innerhalb des Systems synchronisiert werden, einschließlich einer oder mehrerer MMU(s), die einem oder mehreren Anwendungsprozessoren **1005**, Bildprozessoren **1015** und/oder Videoprocessoren **1012** von **Fig. 10A** zugeordnet sind, so dass jeder Prozessor **1005-1012** an einem gemeinsam genutzten oder vereinigten virtuellen Speichersystem partizipieren kann. In mindestens einer Ausführungsform ermöglichen eine oder mehrere Schaltungszwischenverbindungen **1030A-1030B** dem Graphikprozessor **1010**, sich mit anderen IP-Kernen innerhalb des SoC entweder über einen internen Bus des SoC oder über eine direkte Verbindung schnittstellenmäßig zu verbinden.

[0157] In mindestens einer Ausführungsform umfasst der Graphikprozessor **1010** eine oder mehrere MMUs **1020A-1020B**, Cache(s) **1025A-1025B** und Schaltungszwischenverbindungen **1030A-1030B** des Graphikprozessors **1010** von **Fig. 10A**. In mindestens einer Ausführungsform umfasst der Graphikprozessor **1040** einen oder mehrere Shader-Kerne **1055A-1055N** (z.B. **1055A**, **1055B**, **1055C**, **1055D**, **1055E**, **1055F** bis **1055N-1** und **1055N**), wie in **Fig. 10B** gezeigt, welche eine vereinheitlichte Shader-Kern-Architektur bereitstellen, in welcher ein einzelner Kern oder Typ oder Kern alle Arten von programmierbarem Shader-Code ausführen kann, einschließlich Shader-Programmcode zur Implementierung von Vertex-Shadern, Fragment-Shadern und/oder Rechen-Shadern. In mindestens einer Ausführungsform kann eine Anzahl von Shader-Kernen variieren. In mindestens einer Ausführungsform umfasst der Graphikprozessor **1040** einen Inter-Core-Task-Manager **1045**, der als ein Thread-Dispatcher fungiert, um Ausführung-Threads an einen oder mehrere Shader-Kerne **1055A-1055N** zu senden, und eine Kachelungs-Einheit **1058**, um Kachelungs-Operationen für kachelbasiertes Rendering zu beschleunigen, bei welchen 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.

[0158] Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenz- und/oder Trainingslogik **615** werden hier in Verbindung mit **Fig. 6B** und/oder **6C** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik **615** in der integrierten Schaltung **10A** und/oder **10B** verwendet werden, um Operationen zu inferenzieren oder vorherzusagen, 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.

[0159] **Fig. 10D-10E** veranschaulichen eine zusätzliche beispielhafte Graphikprozessorlogik gemäß hier beschriebenen Ausführungsformen, um das mobile Rechenzentrum mit einem mobilen Rechenzentrumkühlsys-

tem zu unterstützen und oder zu ermöglichen. **Fig. 10D** veranschaulicht einen Graphikkern **1000D**, der in mindestens einer Ausführungsform in dem Graphikprozessor **1000D** von **Fig. 10A** umfasst sein kann und in mindestens einer Ausführungsform ein vereinheitlichter Shader-Kern **1055A-1055N**, wie in **Fig. 10C**, in mindestens einer Ausführungsform sein kann. **Fig. 10B** veranschaulicht eine hochparallele universelle Graphikverarbeitungseinheit **1030**, die für den Einsatz auf einem Multi-Chip-Modul in mindestens einer Ausführungsform geeignet ist.

[0160] In mindestens einer Ausführungsform kann der Graphikkern **1000D** mehrere Slices **1001A-1001N** oder Partitionen für jeden Kern umfassen, und ein Graphikprozessor kann mehrere Instanzen des Graphikkerns **1000D** umfassen. Slices **1001A-1001N** können eine Unterstützungslogik umfassen, die einen lokalen Befehls-Cache **1004A-1004N**, einen Thread-Planer **1006A-1006N**, einen Thread-Dispatcher **1008A-1008N** und einen Satz von Registern **1010A-1010N** umfasst. In mindestens einer Ausführungsform können die Slices **1001A-1001N** einen Satz zusätzlicher Funktionseinheiten (AFUs **1012A-1012N**), Gleitkommaeinheiten (FPU **1014A-1014N**), Ganzzahl-Arithmetiklogikeinheiten (ALUs **1016A-1016N**), Adressberechnungseinheiten (ACU **1013A-1013N**), doppelt genaue Gleitkommaeinheiten (DPFPU **1015A-1015N**) und Matrixverarbeitungseinheiten (MPU **1017A-1017N**) umfassen.

[0161] In mindestens einer Ausführungsform können FPUs **1014A-1014N** einfach genaue (32-Bit) und halb genaue (16-Bit) Gleitkomma-Operationen durchführen, während DPFPU **1015A-1015N** doppelt genaue (64-Bit) Gleitkomma-Operationen durchführen. In mindestens einer Ausführungsform können die ALUs **1016A-1016N** Ganzzahl-Operationen mit variabler Genauigkeit und 8-Bit-, 16-Bit- und 32-Bit-Genauigkeit durchführen und für Operationen mit gemischter Genauigkeit konfiguriert sein. In mindestens einer Ausführungsform können die MPUs **1017A-1017N** auch für Matrix-Operationen mit gemischter Genauigkeit konfiguriert sein, einschließlich halb genauer Gleitkomma- und 8-Bit Ganzzahl-Operationen. In mindestens einer Ausführungsform können die MPUs **1017-1017N** eine Vielzahl von Matrix-Operationen durchführen, um die Frameworks von Anwendungen für maschinelles Lernen zu beschleunigen, einschließlich des Ermöglichens einer Unterstützung für beschleunigte allgemeine Matrix-zu-Matrix Multiplikation (GEMM). In mindestens einer Ausführungsform können die AFUs **1012A-1012N** zusätzliche logische Operationen durchführen, die nicht von Gleitkomma- oder Integer-Einheiten unterstützt werden, die trigonometrische Operationen (z.B. Sinus, Cosinus usw.) umfassen.

[0162] Wie an anderer Stelle in dieser Offenbarung erläutert, kann die Inferenz- und/oder Trainingslogik **615** (referenziert mindestens in **Fig. 6B**, **Fig. 6C**) verwendet werden, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenz- und/oder Trainingslogik **615** werden nachstehend in Verbindung mit **Fig. 6B** und/oder **6C** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik **615** im Graphikkern **1000D** 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.

[0163] **Fig. 11A** ist ein Blockdiagramm, das ein Computersystem **1100A** gemäß mindestens einer Ausführungsform darstellt. In mindestens einer Ausführungsform umfasst das Computersystem **1100A** ein Verarbeitungsteilsystem **1101** mit einem oder mehreren Prozessoren **1102** und einem Systemspeicher **1104**, der über einen Zwischenverbindungspfad kommuniziert, der einen Speicher-Hub **1105** umfassen kann. In mindestens einer Ausführungsform kann der Speicher-Hub **1105** eine separate Komponente innerhalb einer Chipsatz-Komponente sein oder in einen oder mehrere Prozessoren **1102** integriert sein. In mindestens einer Ausführungsform ist der Speicher-Hub **1105** über eine Kommunikationsverbindung **1106** mit einem E/A-Teilsystem **1111** gekoppelt. In mindestens einer Ausführungsform umfasst das E/A-Teilsystem **1111** einen E/A-Hub **1107**, der dem Computersystem **1100A** ermöglichen kann, Eingaben von einer oder mehreren Eingabevorrichtungen **1108** zu empfangen. In mindestens einer Ausführungsform kann der E/A-Hub **1107** einen Anzeigen-Controller ermöglichen, der in einem oder mehreren Prozessoren **1102** umfasst sein kann, um Ausgaben an eine oder mehrere Anzeigevorrichtungen **1110A** bereitzustellen. In mindestens einer Ausführungsform können eine oder mehrere mit dem E/A-Hub **1107** gekoppelte Anzeigevorrichtungen **1110A** eine lokale, interne oder eingebettete Anzeigevorrichtung umfassen.

[0164] In mindestens einer Ausführungsform umfasst das Verarbeitungsteilsystem **1101** einen oder mehrere Parallelprozessoren **1112**, die über einen Bus oder eine andere Kommunikationsverbindung **1113** mit dem Speicher-Hub **1105** gekoppelt sind. In mindestens einer Ausführungsform kann die Kommunikationsverbindung **1113** eine von einer beliebigen Anzahl von auf Standards basierenden Kommunikation-Verbindungstechnologien umfassen.

nologien oder - Verbindungsprotokollen sein, wie beispielsweise, jedoch nicht beschränkt auf, PCI Express, oder kann eine händlerspezifische Kommunikationsschnittstelle oder Kommunikationsstruktur sein. In mindestens einer Ausführungsform bilden ein oder mehrere Parallelprozessoren **1112** ein rechnerisch fokussiertes Parallel- oder Vektor-Verarbeitungssystem, das eine große Anzahl von Verarbeitungskernen und/oder Verarbeitungsclustern umfassen kann, wie beispielsweise einen Prozessor mit vielen integrierten Kernen (Many Integrated Core; MIC). In mindestens einer Ausführungsform bilden ein oder mehrere Parallelprozessoren **1112** ein Graphikverarbeitungs-Teilsystem, das Pixel an eine oder mehrere Anzeigevorrichtungen **1110A** ausgeben kann, die über den E/A-Hub **1107** gekoppelt sind. In mindestens einer Ausführungsform können ein oder mehrere Parallelprozessoren **1112** auch einen Anzeigen-Controller und eine Anzeigeschnittstelle (nicht gezeigt) umfassen, um eine direkte Verbindung zu einer oder mehreren Anzeigevorrichtungen **1110B** zu ermöglichen.

[0165] In mindestens einer Ausführungsform kann sich eine Systemspeichereinheit **1114** mit dem E/A-Hub **1107** verbinden, um einen Speichermechanismus für das Computersystem **1100A** bereitzustellen. In mindestens einer Ausführungsform kann ein E/A-Schalter **1116** verwendet werden, um einen Schnittstellenmechanismus bereitzustellen, um Verbindungen zwischen dem E/A-Hub **1107** und anderen Komponenten, wie beispielsweise einem Netzwerkadapter **1118** und/oder einem drahtlosen Netzwerkadapter **1119**, der in die Plattform integriert sein kann, und verschiedenen anderen Vorrichtungen, die über eine oder mehrere Add-In-Vorrichtungen **1112** hinzugefügt sein können, zu ermöglichen. In mindestens einer Ausführungsform kann der Netzwerkadapter **1118** ein Ethernet-Adapter oder ein anderer verdrahteter Netzwerkadapter sein. In mindestens einer Ausführungsform kann der drahtlose Netzwerkadapter **1119** ein oder mehrere Wi-Fi, Bluetooth-, Nahfeldkommunikation(NFC)- oder andere Netzwerkgeräte umfassen, die ein oder mehrere drahtlose Funk-einrichtungen umfassen.

[0166] In mindestens einer Ausführungsform kann das Computersystem **1100A** andere Komponenten umfassen, die nicht explizit gezeigt sind, einschließlich USB- oder andere Portverbindungen, optische Speicherlaufwerke, Videoaufnahmegeräte und dergleichen, die auch mit dem E/A-Hub **1107** verbunden sein können. In mindestens einer Ausführungsform können Kommunikationspfade, die verschiedene Komponenten in **Fig. 11A** 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. NVLink-Hochgeschwindigkeitsverbindungen oder Zwischenverbindungsprotokolle, implementiert sein.

[0167] In mindestens einer Ausführungsform umfassen ein oder mehrere Parallelprozessoren **1112** für Graphik- und Videoverarbeitung optimierte Schaltungen, die beispielsweise Videoausgabeschaltungen umfassen und eine Graphikverarbeitungseinheit (GPU) bilden. In mindestens einer Ausführungsform umfassen ein oder mehrere Parallelprozessoren **1112** für allgemeine Verarbeitung optimierte Schaltungen. In mindestens einer Ausführungsform können Komponenten des Computersystems **1100A** mit einem oder mehreren anderen Systemelementen auf einer einzigen integrierten Schaltung integriert sein. Beispielsweise können in mindestens einer Ausführungsform ein oder mehrere Parallelprozessoren **1112**, der Speicher-Hub **1105**, der bzw. die Prozessoren **1102** und der E/A-Hub **1107** in eine System-on-Chip (SoC)-integrierte Schaltung integriert sein. In mindestens einer Ausführungsform können Komponenten des Computersystems **1100A** 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 des Computersystems **1100A** in ein Multi-Chip-Modul (MCM) integriert sein, welches mit anderen Multi-Chip-Modulen zu einem modularen Computersystem verbunden sein kann.

[0168] Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenz- und/oder Trainingslogik **615** werden hier in Verbindung mit **Fig. 6B** und/oder **6C** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik **615** in dem System **Fig. 11** 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.

PROZESSOREN

[0169] **Fig. 11B** veranschaulicht einen Parallelprozessor **1100B** gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform können verschiedene Komponenten des Parallelprozessors **1100B** 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 der veranschaulichte Parallelprozessor **1100B** eine Variante eines oder mehrerer in **Fig. 11B** gezeigter Parallelprozessoren **1112** gemäß einer beispielhaften Ausführungsform.

[0170] In mindestens einer Ausführungsform umfasst der Parallelprozessor **1100B** eine Parallelverarbeitungseinheit **1102**. In mindestens einer Ausführungsform umfasst die Parallelverarbeitungseinheit **1102** eine E/A-Einheit **1104**, die eine Kommunikation mit anderen Vorrichtungen ermöglicht, einschließlich anderer Instanzen der Parallelverarbeitungseinheit **1102**. In mindestens einer Ausführungsform kann die E/A-Einheit **1104** direkt mit anderen Vorrichtungen verbunden sein. In mindestens einer Ausführungsform verbindet sich die E/A-Einheit **1104** mit anderen Vorrichtungen unter Verwendung eines Hubs oder einer Schalter-Schnittstelle, wie beispielsweise dem Speicher-Hub **1105**. In mindestens einer Ausführungsform bilden Verbindungen zwischen dem Speicher-Hub **1105** und der E/A-Einheit **1104** eine Kommunikationsverbindung **1113**. In mindestens einer Ausführungsform verbindet sich die E/A-Einheit **1104** mit einer Host-Schnittstelle **1106** und einer Speicherquerschene **1116**, wobei die Host-Schnittstelle **1106** Befehle empfängt, die zur Durchführung von Verarbeitungsoperationen bestimmt sind, und die Speicherquerschene **1116** Befehle empfängt, die zur Durchführung von Speicheroperationen bestimmt sind.

[0171] In mindestens einer Ausführungsform kann dann, wenn die Host-Schnittstelle **1106** über die E/A-Einheit **1104** einen Befehlspuffer empfängt, die Host-Schnittstelle **1106** Arbeitsoperationen anweisen, um diese Befehle an einem Frontend **1108** auszuführen. In mindestens einer Ausführungsform koppelt das Frontend **1108** mit einem Planer **1110**, der konfiguriert ist, um Befehle oder andere Arbeitselemente an eine Verarbeitungsclusteranordnung **1112** zu verteilen. In mindestens einer Ausführungsform stellt der Planer **1110** sicher, dass die Clusteranordnung **1112** richtig konfiguriert ist und sich in einem gültigen Zustand befindet, bevor Aufgaben an eine Verarbeitungsclusteranordnung **1112** verteilt werden. In mindestens einer Ausführungsform ist der Planer **1110** über Firmware-Logik implementiert, die auf einem Mikrocontroller ausgeführt wird. In mindestens einer Ausführungsform ist der von dem Mikrocontroller implementierte Planer **1110** konfigurierbar, um komplexe Planungs- und Arbeitsverteilungsoperationen mit grober und feiner Granularität durchzuführen, was eine schnelle Vorwegnahme und Kontextumschaltung von Threads ermöglicht, die auf der Verarbeitungsanordnung **1112** ausgeführt werden. In mindestens einer Ausführungsform kann die Host-Software Arbeitslasten für die Planung auf der Verarbeitungsanordnung **1112** über eine von mehreren Graphikverarbeitungstürklingen nachweisen. In mindestens einer Ausführungsform können dann Arbeitslasten über die Verarbeitungsanordnung **1112** hinweg durch die Logik des Planers **1110** innerhalb eines Mikrocontrollers einschließlich des Planers **1110** automatisch verteilt werden.

[0172] In mindestens einer Ausführungsform kann die Verarbeitungsclusteranordnung **1112** bis zu „N“-Verarbeitungscluster (z.B. Cluster **1114A**, Cluster **1114B**, bis Cluster **1114N**) umfassen. In mindestens einer Ausführungsform kann jeder Cluster **1114A-1114N** der Verarbeitungsclusteranordnung **1112** eine große Anzahl nebenläufigen Threads ausführen. In mindestens einer Ausführungsform kann der Planer **1110** den Clustern **1114A-1114N** der Verarbeitungsclusteranordnung **1112** unter Verwendung verschiedener Planungs- und/oder Arbeitsverteilungs-Algorithmen 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 dem Planer **1110** dynamisch gehandhabt werden, oder kann teilweise durch Compilerlogik während der Kompilierung der Programmlogik, die für die Ausführung durch die Verarbeitung der Clusteranordnung **1112** konfiguriert ist, unterstützt werden. In mindestens einer Ausführungsform können verschiedene Cluster **1114A-1114N** der Verarbeitungsclusteranordnung **1112** zur Verarbeitung verschiedener Arten von Programmen oder zur Durchführung verschiedener Arten von Berechnungen zugeteilt sein.

[0173] In mindestens einer Ausführungsform kann die Verarbeitungsclusteranordnung **1112** konfiguriert sein, um verschiedene Arten von Parallelverarbeitungsoperationen durchzuführen. In mindestens einer Ausführungsform ist die Verarbeitungsclusteranordnung **1112** konfiguriert, um allgemeine parallele Rechenoperationen durchzuführen. In mindestens einer Ausführungsform kann die Verarbeitungsclusteranordnung **1112** Logik umfassen, um Verarbeitungsaufgaben einschließlich eines Filterns von Video- und/oder Audiodaten, Durchführens von Modellierungsoperationen, einschließlich physikalischer Operationen, und Durchführens von Datentransformationen auszuführen.

[0174] In mindestens einer Ausführungsform ist die Verarbeitungsclusteranordnung **1112** konfiguriert, um parallele Graphikverarbeitungsoperationen durchzuführen. In mindestens einer Ausführungsform kann die Verarbeitungsclusteranordnung **1112** zusätzliche Logik umfassen, um die Ausführung solcher Graphikverarbeitungsoperationen zu unterstützen, einschließlich einer, jedoch nicht beschränkt auf, Texturabtastrlogik zur

Durchführung von Texturoperationen, sowie einer Tessellierungslogik und anderer Vertex-Verarbeitungslogik. In mindestens einer Ausführungsform kann die Verarbeitungsclusteranordnung **1112** konfiguriert sein, um grafikverarbeitungsbezogene Shader-Programme auszuführen, wie beispielsweise Vertex-Shader, Tessellation-Shader, Geometrie-Shader und Pixel-Shader. In mindestens einer Ausführungsform kann die Parallelverarbeitungseinheit **1102** Daten aus dem Systemspeicher über die E/A-Einheit **1104** zur Verarbeitung übertragen. In mindestens einer Ausführungsform können während der Verarbeitung übertragene Daten während der Verarbeitung in einem On-Chip-Speicher (z.B. Speicher des Parallelprozessors **1122**) gespeichert und anschließend in den Systemspeicher zurückgeschrieben werden.

[0175] In mindestens einer Ausführungsform kann dann, wenn die Parallelverarbeitungseinheit **1102** zur Durchführung von Graphikverarbeitung verwendet wird, der Planer **1110** konfiguriert sein, um eine Verarbeitungsarbeitslast in etwa gleich große Aufgaben aufzuteilen, um die Verteilung von Graphikverarbeitungsoperationen auf mehrere Cluster **1114A-1114N** der Verarbeitungsclusteranordnung **1112** besser zu ermöglichen. In mindestens einer Ausführungsform können Abschnitte der Verarbeitungsclusteranordnung **1112** konfiguriert sein, um verschiedene Arten von Verarbeitung durchzuführen. In mindestens einer Ausführungsform kann ein erster Abschnitt konfiguriert sein, um Vertex-Shading und Topologieerzeugung durchzuführen, ein zweiter Abschnitt kann konfiguriert sein, um Tessellation und Geometrie-Shading durchzuführen, und kann ein dritter Abschnitt konfiguriert sein, um Pixel-Shading oder andere Bildschirmraumoperationen durchzuführen, um ein gerendertes Bild für die Anzeige zu erzeugen, wenn eine Simulation einer Ventilsteuerung für das mobile Rechenzentrum mit einem mobilen Kühlsystem erforderlich ist. In mindestens einer Ausführungsform können Zwischendaten, die von einem oder mehreren der Cluster **1114A-1114N** erzeugt wurden, in Puffern gespeichert werden, so dass Zwischendaten zwischen den Clustern **1114A-1114N** zur weiteren Verarbeitung übertragen werden können.

[0176] In mindestens einer Ausführungsform kann die Verarbeitungsclusteranordnung **1112** Verarbeitungsaufgaben empfangen, die über den Planer **1110** auszuführen sind, der von dem Frontend **1108** 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 der Planer **1110** konfiguriert sein, um Indizes zu holen, die Aufgaben entsprechen, oder kann Indizes von dem Frontend **1108** empfangen. In mindestens einer Ausführungsform kann das Frontend **1108** konfiguriert sein, um sicherzustellen, dass die Verarbeitungsclusteranordnung **1112** in einen gültigen Zustand konfiguriert ist, bevor eine durch eingehende Befehlspuffer (z.B. Stapelpuffer, Schiebepuffer usw.) spezifizierte Arbeitslast initiiert wird.

[0177] In mindestens einer Ausführungsform kann jede von einer oder mehreren Instanzen der Parallelverarbeitungseinheit **1102** mit dem Parallelprozessorspeicher **1122** gekoppelt sein. In mindestens einer Ausführungsform kann auf den Parallelprozessorspeicher **1122** über die Speicherquerschienen **1116** zugegriffen werden, die Speicheranforderungen von der Verarbeitungsclusteranordnung **1112** sowie der E/A-Einheit **1104** empfangen kann. In mindestens einer Ausführungsform kann die Speicherquerschienen **1116** über eine Speicherschnittstelle **1118** auf den Parallelprozessorspeicher **1122** zugreifen. In mindestens einer Ausführungsform kann die Speicherschnittstelle **1118** mehrere Partitionierungseinheiten (z.B. Partitionierungseinheit **1120A**, Partitionierungseinheit **1120B** bis Partitionierungseinheit **1120N**) umfassen, die jeweils mit einem Abschnitt (z.B. der Speichereinheit) des Parallelprozessorspeichers **1122** koppeln können. In mindestens einer Ausführungsform ist eine Anzahl von Partitionierungseinheiten **1120A-1122N** konfiguriert, um gleich einer Anzahl von Speichereinheiten zu sein, so dass eine erste Partitionierungseinheit **1120A** eine entsprechende erste Speichereinheit **1124A** aufweist, eine zweite Partitionierungseinheit **1120B** eine entsprechende Speichereinheit **1124B** aufweist und eine N-te Partitionierungseinheit **1120N** eine entsprechende N-te Speichereinheit **1124N** aufweist. In mindestens einer Ausführungsform kann eine Anzahl von Partitionierungseinheiten **1120A-1120N** nicht gleich einer Anzahl von Speichervorrichtungen sein.

[0178] In mindestens einer Ausführungsform können die Speichereinheiten **1124A-1124N** verschiedene Arten von Speichervorrichtungen umfassen, einschließlich dynamischen Direktzugriffspeichers (DRAM) oder Graphik-Direktzugriffspeichers, wie beispielsweise synchronen Graphik-Direktzugriffspeichers (SGRAM), einschließlich Graphikspeichers mit doppelter Datenrate (GDDR). In mindestens einer Ausführungsform können die Speichereinheiten **1124A-1124N** 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 **1124A-1124N** hinweg gespeichert sein, so dass die Partitionseinheiten **1120A-1120N** Abschnitte jedes Renderziels parallel schreiben können, um verfügbare Bandbreite des Parallelprozessorspeichers **1122** effizient zu

nutzen. In mindestens einer Ausführungsform kann eine lokale Instanz des Parallelprozessorspeichers **1122** zugunsten eines vereinheitlichten Speicherdesigns, das Systemspeicher in Verbindung mit lokalem Cache-Speicher nutzt, ausgeschlossen sein.

[0179] In mindestens einer Ausführungsform kann jeder beliebige der Cluster **1114A-1114N** der Verarbeitungsclusteranordnung **1112** Daten verarbeiten, die in eine beliebige der Speichereinheiten **1124A-1124N** in dem Parallelprozessorspeicher **1122** geschrieben werden. In mindestens einer Ausführungsform kann die Speicherquerschleife **1116** konfiguriert sein, um eine Ausgabe jedes Clusters **1114A-1114N** an eine beliebige Partitionseinheit **1112A-1112N** oder an einen anderen Cluster **1114A-1114N** zu übertragen, der zusätzliche Verarbeitungsoperationen an einer Ausgabe durchführen kann. In mindestens einer Ausführungsform kann jeder Cluster **1114A-1114N** mit der Speicherschnittstelle **1118** über die Speicherquerschleife **1116** kommunizieren, um aus verschiedenen externen Speichervorrichtungen zu lesen oder in diese zu schreiben. In mindestens einer Ausführungsform weist die Speicherquerschleife **1116** zur Kommunikation mit der E/A-Einheit **1104** Verbindung zu der Speicherschnittstelle **1118** auf, sowie eine Verbindung zu einer lokalen Instanz des Parallelprozessorspeichers **1122**, so dass Verarbeitungseinheiten innerhalb verschiedener Verarbeitungscluster **1114A-1114N** mit Systemspeicher oder anderem Speicher kommunizieren können, der nicht lokal zu der Parallelverarbeitungseinheit **1102** ist. In mindestens einer Ausführungsform kann die Speicherquerschleife **1116** virtuelle Kanäle verwenden, um Ströme von Datenaufkommen zwischen den Clustern **1114A-1114N** und den Partitionierungseinheiten **1120A-1120N** zu trennen.

[0180] In mindestens einer Ausführungsform können mehrere Instanzen der Parallelverarbeitungseinheit **1102** auf einer einzigen Erweiterungskarte bereitgestellt sein oder können mehrere Erweiterungskarten miteinander verbunden sein. In mindestens einer Ausführungsform können verschiedene Instanzen der Parallelverarbeitungseinheit **1102** 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. In mindestens einer Ausführungsform können einige Instanzen der Parallelverarbeitungseinheit **1102** relativ zu anderen Instanzen Gleitkommaeinheiten mit höherer Genauigkeit umfassen. In mindestens einer Ausführungsform können Systeme, die eine oder mehrere Instanzen der Parallelverarbeitungseinheit **1102** oder des Parallelprozessors **1100B** 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.

[0181] Fig. **11C** ist ein Blockdiagramm einer Partitionierungseinheit **1120** gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist die Partitionierungseinheit **1120** eine Instanz einer der Partitionierungseinheiten **1120A-1120N** aus Fig. **11B**. In mindestens einer Ausführungsform umfasst die Partitionierungseinheit **1120** einen L2-Cache **1121**, ein Frame-Buffer-Interface bzw. eine Einzelbildpufferschnittstelle **1125** und eine Rasteroperationseinheit „ROP“ **1126**. Der L2-Cache **1121** ist ein Lese-/Schreib-Cache, der konfiguriert ist, um Lade- und Speicheroperationen durchzuführen, die von der Speicherquerschleife **1116** und der ROP **1126** empfangen wurden. In mindestens einer Ausführungsform werden Lesefehler und dringende Rückschreibanforderungen von dem L2-Cache **1121** an die Einzelbildpufferschnittstelle **1125** zur Verarbeitung ausgegeben. In mindestens einer Ausführungsform können Aktualisierungen über die Einzelbildpufferschnittstelle **1125** zur Verarbeitung auch an einen Einzelbildpuffer gesendet werden. In mindestens einer Ausführungsform ist die Einzelbildpufferschnittstelle **1125** mit einer der Speichereinheiten in dem Parallelprozessorspeicher, wie beispielsweise den Speichereinheiten **1124A-2224N** aus Fig. **11B** (z.B. innerhalb des Parallelprozessorspeichers **1122**) verbunden.

[0182] In mindestens einer Ausführungsform ist die ROP **1126** eine Verarbeitungseinheit, die Rasteroperationen wie beispielsweise Schablone, Z-Test, Blending und so weiter durchführt. In mindestens einer Ausführungsform gibt die ROP **1126** dann verarbeitete Graphikdaten aus, die in den Graphikspeicher gespeichert werden. In mindestens einer Ausführungsform umfasst die ROP **1126** 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. Komprimierungslogik, die von der ROP **1126** durchgeführt wird, kann basierend auf statistischen Merkmalen der zu komprimierenden Daten variieren. In mindestens einer Ausführungsform wird eine Delta-Farbkompensation auf Tiefen- und Farbdaten auf kachelweiser Grundlage durchgeführt.

[0183] In mindestens einer Ausführungsform ist die ROP **1126** in jedem Verarbeitungscluster (z.B. Cluster **1114A-1114N** aus Fig. **11B**) anstelle von in der Partitionierungseinheit **1120** umfasst. In mindestens einer Aus-

föhrungsform werden Lese- und Schreibanforderungen für Pixeldaten über die Speicherquerschiene **1116** anstelle von Pixelfragmentdaten übertragen. In mindestens einer Ausführungsform können verarbeitete Graphikdaten auf einer Anzeigevorrichtung, wie beispielsweise einer oder mehreren Anzeigevorrichtung(en) **1110** von **Fig. 11B** angezeigt werden, zur weiteren Verarbeitung durch den/die Prozessor(en) **1102** weitergeleitet werden, oder zur weiteren Verarbeitung durch eine der Verarbeitungsentitäten innerhalb des Parallelprozessors **1100B** aus **Fig. 11B** weitergeleitet werden.

[0184] **Fig. 11D** ist ein Blockdiagramm eines Verarbeitungsclusters **1114** innerhalb einer Parallelverarbeitungseinheit gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist ein Verarbeitungscluster eine Instanz eines der Verarbeitungscluster **1114A-1114N** von **Fig. 11B**. In mindestens einer Ausführungsform können ein oder mehrere der Verarbeitungscluster **1114** konfiguriert sein, um viele Threads parallel auszuführen, wobei sich „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 synchronisierten Threads zu unterstützen, wobei eine gemeinsame Befehlseinheit verwendet wird, die konfiguriert ist, um Anweisungen an eine Anzahl von Verarbeitungs-Engines innerhalb jedes einzelnen der Verarbeitungscluster auszugeben.

[0185] In mindestens einer Ausführungsform kann der Betrieb des Verarbeitungsclusters **1114** über einen Pipeline-Verwalter **1132** gesteuert werden, der Verarbeitungsaufgaben an SIMT-Parallelprozessoren verteilt. In mindestens einer Ausführungsform empfängt der Pipeline-Verwalter **1132** Anweisungen von dem Planer **1110** von **Fig. 11B** und verwaltet die Ausführung dieser Anweisungen über einen Graphik-Multiprozessor **1134** und/oder eine Textureinheit **1136**. In mindestens einer Ausführungsform ist der Graphik-Multiprozessor **1134** eine beispielhafte Instanz eines SIMT-Parallelprozessors. In mindestens einer Ausführungsform können jedoch verschiedene Arten von SIMT-Parallelprozessoren unterschiedlicher Architekturen innerhalb des Verarbeitungsclusters **1114** umfasst sein. In mindestens einer Ausführungsform können eine oder mehrere Instanzen des Graphik-Multiprozessors **1134** in einem Verarbeitungscluster **1114** umfasst sein. In mindestens einer Ausführungsform kann der Graphik-Multiprozessor **1134** Daten verarbeiten und kann eine Datenquerschiene **1140** verwendet, um verarbeitete Daten an eines von mehreren möglichen Zielen, einschließlich anderer Shader-Einheiten, zu verteilen. In mindestens einer Ausführungsform kann der Pipeline-Verwalter **1132** die Verteilung der verarbeiteten Daten erleichtern, indem er Ziele für verarbeitete Daten spezifiziert, die über die Datenquerschiene **1140** zu verteilen sind.

[0186] In mindestens einer Ausführungsform kann jeder Graphik-Multiprozessor **1134** innerhalb des Verarbeitungsclusters **1114** 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 eine Mehrzahl von 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.

[0187] In mindestens einer Ausführungsform bilden an den Verarbeitungscluster **1114** 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 **1134** zugeordnet sein. In mindestens einer Ausführungsform kann eine Thread-Gruppe weniger Threads umfassen als eine Anzahl von Verarbeitungs-Engines innerhalb des Graphik-Multiprozessors **1134**. In mindestens einer Ausführungsform können, 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 des Graphik-Multiprozessors **1134** umfassen. In mindestens einer Ausführungsform kann, wenn eine Thread-Gruppe mehr Threads als Verarbeitungs-Engines innerhalb des Graphik-Multiprozessors **1134** um-

fasst, die Verarbeitung über aufeinanderfolgende Taktzyklen durchgeführt werden. In mindestens einer Ausführungsform können mehrere Thread-Gruppen nebenläufig auf einem Graphik-Multiprozessor **1134** ausgeführt werden.

[0188] In mindestens einer Ausführungsform umfasst der Graphik-Multiprozessor **1134** einen internen Cache-Speicher, um Lade- und Speicheroperationen durchzuführen. In mindestens einer Ausführungsform kann der Graphik-Multiprozessor **1134** auf einen internen Cache verzichten und einen Cache-Speicher (z.B. den L1-Cache **1148**) innerhalb des Verarbeitungsclusters **1114** verwenden. In mindestens einer Ausführungsform weist jeder Graphik-Multiprozessor **1134** auch Zugriff auf L2-Caches innerhalb von Partitionseinheiten auf (z.B. den Partitionseinheiten **1120A-1120N** von **Fig. 11B**), die unter allen Verarbeitungsclustern **1114** gemeinsam genutzt werden und zum Übertragen von Daten zwischen Threads verwendet werden können. In mindestens einer Ausführungsform kann der Graphik-Multiprozessor **1134** 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 der Parallelverarbeitungseinheit **1102** liegt, als globaler Speicher verwendet werden. In mindestens einer Ausführungsform umfasst der Verarbeitungscluster **1114** mehrere Instanzen des Graphik-Multiprozessors **1134**, der gemeinsame Anweisungen und Daten austauschen kann, welche in dem L1-Cache **1148** gespeichert sein können.

[0189] In mindestens einer Ausführungsform kann jeder Verarbeitungscluster **1114** eine Speichermanagementeinheit („MMU“) **1145** umfassen, die konfiguriert ist, um virtuelle Adressen auf physikalische Adressen abzubilden. In mindestens einer Ausführungsform können sich eine oder mehrere Instanzen der MMU **1145** innerhalb der Speicherschnittstelle **1118** von **Fig. 11B** befinden. In mindestens einer Ausführungsform umfasst die MMU **1145** einen Satz von Seitentabelleneinträgen (Page Table Entries; PTEs), die verwendet werden, um eine virtuelle Adresse auf eine physikalische Adresse einer Kachel und in mindestens einer Ausführungsform einen Cache-Zeilenindex abzubilden. In mindestens einer Ausführungsform kann die MMU **1145** Adressübersetzungs-Lookaside-Puffer (TLB) oder Caches umfassen, die sich in dem Graphik-Multiprozessor **1134** oder dem L1-Cache oder in dem Verarbeitungscluster **1114** befinden können. In mindestens einer Ausführungsform wird eine physikalische Adresse verarbeitet, um den Oberflächendatenzugriff 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.

[0190] In mindestens einer Ausführungsform kann ein Verarbeitungscluster **1114** derart konfiguriert sein, dass jeder Graphik-Multiprozessor **1134** mit einer Textureinheit **1136** 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-L1-Cache (nicht gezeigt) oder aus einem L1-Cache innerhalb des Graphik-Multiprozessors **1134** gelesen und bei Bedarf aus einem L2-Cache, einem lokalen Parallelprozessorspeicher oder einem Systemspeicher geholt. In mindestens einer Ausführungsform gibt jeder Graphik-Multiprozessor **1134** verarbeitete Aufgaben an eine Datenquerschienen **1140** aus, um die verarbeitete(n) Aufgabe(n) einem anderen Verarbeitungscluster **1114** zur weiteren Verarbeitung bereitzustellen oder um die verarbeitete Aufgabe über die Speicherquerschienen **1116** in einem L2-Cache, lokalen Parallelprozessorspeicher oder Systemspeicher zu speichern. In mindestens einer Ausführungsform ist eine preROP **1142** (Pre-Raster Operations Unit) bzw. Vorrasteroperationseinheit konfiguriert, um Daten von dem Graphik-Multiprozessor **1134** zu empfangen und Daten an ROP-Einheiten zu leiten, welche mit Partitionierungseinheiten wie hier beschrieben angeordnet sein können (z.B. die Partitionierungseinheiten **1120A-1120N** von **Fig. 11B**). In mindestens einer Ausführungsform kann die PreROP-Einheit **1142** Optimierungen zur Farbmischung durchführen, Pixelfarbdaten zu organisieren und Adressübersetzungen durchzuführen.

[0191] Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten zu der Inferenz- und/oder Trainingslogik **615** werden nachstehend in Verbindung mit **Fig. 6B** und/oder **6C** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik **615** in dem Graphikverarbeitungscluster **1114** 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.

[0192] **Fig. 11D** zeigt einen Graphik-Multiprozessor **1134** gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform koppelt der Graphik-Multiprozessor **1134** mit dem Pipeline-Verwalter **1132** des

Verarbeitungsclusters **1114**. In mindestens einer Ausführungsform weist der Graphik-Multiprozessor **1134** eine Ausführungspipeline auf, die einen Anweisungs- -Cache **1152**, eine Anweisungseinheit **1154**, eine Adresszuordnungseinheit **1156**, eine Registerdatei **1158**, einen oder mehrere Vielzweck-Graphikverarbeitungseinheit GPGPU-Kerne **1162** und eine oder mehrere Lade-/Speicher-Einheiten **1166** umfasst, ohne darauf beschränkt zu sein. Die GPGPU-Kerne **1162** und die Lade-/Speicher-Einheiten **1166** sind mit dem Cache-Speicher **1172** und dem gemeinsam genutzten Speicher **1170** über eine Speicher- und Cache-Zwischenverbindung **1168** gekoppelt.

[0193] In mindestens einer Ausführungsform empfängt der Anweisungs-Cache **1152** einen Strom von Anweisungen, die von dem Pipeline-Verwalter **1132** auszuführen sind. In mindestens einer Ausführungsform werden Anweisungen in dem Anweisungs-Cache **1152** zwischengespeichert und zur Ausführung durch die Befehlseinheit **1154** bereitgestellt. In mindestens einer Ausführungsform kann die Befehlseinheit **1154** Anweisungen als Thread-Gruppen (z.B. Warps) senden, wobei jeder Thread der Thread-Gruppe einer anderen Ausführungseinheit innerhalb des GPGPU-Kerns **1162** 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 vereinheitlichten Adressraums spezifiziert. In mindestens einer Ausführungsform kann die Adresszuordnungseinheit **1156** verwendet werden, um Adressen in einem vereinheitlichten Adressraum in eine eindeutige Speicheradresse zu übersetzen, auf von den Lade-/Speicher-Einheiten **1166** zugegriffen werden kann.

[0194] In mindestens einer Ausführungsform stellt die Registerdatei **1158** einen Satz von Registern für funktionale Einheiten bzw. Funktionseinheiten des Graphik-Multiprozessors **1134** bereit. In mindestens einer Ausführungsform stellt die Registerdatei **1158** einen temporären Speicher für Operanden bereit, die mit Datenpfaden von Funktionseinheiten (z.B. den GPGPU-Kernen **1162**, den Lade-/Speicher-Einheiten **1166**) des Graphik-Multiprozessors **1134** verbunden sind. In mindestens einer Ausführungsform ist die Registerdatei **1158** derart auf jede der Funktionseinheiten aufgeteilt, dass jeder Funktionseinheit ein dedizierter Abschnitt der Registerdatei **1158** zugeordnet ist. In mindestens einer Ausführungsform ist die Registerdatei **1158** auf zwischen verschiedenen Ketten bzw. Warps aufgeteilt, die von dem den Graphik-Multiprozessor **1134** ausgeführt werden.

[0195] In mindestens einer Ausführungsform können die GPGPU-Kerne **1162** jeweils Gleitkommaeinheiten (FPUs) und/oder Ganzzahlarithmetik-Logikeinheiten (ALUs) umfassen, die zum Ausführen von Anweisungen des Graphik-Multiprozessors **1134** verwendet werden. Die GPGPU-Kerne **1162** können in der Architektur ähnlich sein oder sich in der Architektur unterscheiden. In mindestens einer Ausführungsform umfasst ein erster Abschnitt der GPGPU-Kerne **1162** 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 der Graphik-Multiprozessor **1134** zusätzlich eine oder mehrere Einheiten mit fester Funktion oder Spezialfunktion zum Durchführen bestimmter Funktionen wie beispielsweise Kopierrechteck- oder Pixelblendingoperationen umfassen. In mindestens einer Ausführungsform kann bzw. können ein oder mehrere GPGPU-Kerne **1162** auch Logik für eine feste oder spezielle Funktion umfassen.

[0196] In mindestens einer Ausführungsform umfassen die GPGPU-Kerne **1162** eine SIMD-Logik, die in der Lage ist, eine einzelne Anweisung auf bzw. für mehrere Datensätze durchzuführen. In mindestens einer Ausführungsform können GPGPU-Kerne **1162** 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 SMT-Architekturen geschrieben und kompiliert wurden. In mindestens einer Ausführungsform können mehrere Threads eines für ein SMT-Ausführungsmodell konfigurierten Programms über eine einzelne SIMD-Anweisung ausgeführt werden. Beispielsweise können in mindestens einer Ausführungsform acht SMT-Threads, die gleiche oder ähnliche Operationen durchführen, über eine einzige SIMD8-Logikeinheit parallel ausgeführt werden.

[0197] In mindestens einer Ausführungsform ist die Speicher- und Cache-Zwischenverbindung **1168** ein Zwischenverbindungsnetzwerk, das jede Funktionseinheit des Graphik-Multiprozessors **1134** mit der Registerdatei **1158** und dem gemeinsamen Speicher **1170** verbindet. In mindestens einer Ausführungsform ist die Speicher- und Cache-Zwischenverbindung **1168** eine Querschienen-Zwischenverbindung, die der Lade-/Speicher-Einheit **1166** ermöglicht, Lade- und Speicheroperationen zwischen dem gemeinsamen Speicher **1170** und der Registerdatei **1158** zu implementieren. In mindestens einer Ausführungsform kann die Registerdatei **1158** mit

der gleichen Frequenz wie die GPGPU-Kerne **1162** arbeiten, so dass die Datenübertragung zwischen den GPGPU-Kernen **1162** und der Registerdatei **1158** eine sehr geringe Latenzzeit aufweist. In mindestens einer Ausführungsform kann der gemeinsame Speicher **1170** verwendet werden, um die Kommunikation zwischen Threads zu ermöglichen, die auf Funktionseinheiten innerhalb des Graphik-Multiprozessors **1134** ausgeführt werden. In mindestens einer Ausführungsform kann der Cache-Speicher **1172** beispielsweise als ein Daten-cache verwendet werden, um Texturdaten zwischenspeichern, die zwischen Funktionseinheiten und der Textureinheit **1136** ausgetauscht werden. In mindestens einer Ausführungsform kann der gemeinsam genutzte Speicher **1170** auch als ein programmverwalteter Cache verwendet werden. In mindestens einer Ausführungsform können Threads, die auf GPGPU-Kernen **1162** ausgeführt werden, Daten innerhalb des gemeinsam genutzten Speichers programmatisch zusätzlich zu automatisch zwischengespeicherten Daten speichern, die innerhalb des Cache-Speichers **1172** gespeichert sind.

[0198] In mindestens einer Ausführungsform ist ein Parallelprozessor oder eine GPGPU, wie hier beschrieben, kommunikativ mit Host/ProzessorKernen 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 die GPU in bzw. auf demselben Gehäuse oder Chip wie die Kerne integriert und kommunikativ mit Kernen über einen internen Prozessorbus/eine interne Zwischenverbindung (in mindestens einer Ausführungsform intern zu dem Gehäuse oder Chip) gekoppelt sein. In mindestens einer Ausführungsform können Prozessorkerne unabhängig davon, wie die GPU verbunden ist, der GPU Arbeit in Form von Sequenzen von Befehlen/Anweisungen zuweisen, die in einem Arbeitsdeskriptor umfasst sind. In mindestens einer Ausführungsform verwendet die GPU dann dedizierte Schaltkreise/Logik zum effizienten Verarbeiten dieser Befehle/Anweisungen.

[0199] Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten zu der Inferenz- und/oder Trainingslogik **615** werden nachstehend in Verbindung mit **Fig. 6B** und/oder **6C** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik **615** in dem Graphik-Multiprozessor **1134** verwendet werden, um Operationen zu inferenzieren oder vorherzusagen, 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.

[0200] **Fig. 12A** veranschaulicht ein Multi-GPU-Computersystem **1200A** gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann das Multi-GPU-Computersystem **1200A** einen Prozessor **1202** umfassen, der über einen Host-Schnittstellen-Schalter **1204** mit mehreren universellen Graphikverarbeitungseinheiten (GPGPUs) **1206A-D** gekoppelt ist. In mindestens einer Ausführungsform ist der Host-Schnittstellen-Schalter **1204** eine PCI-Express-Schaltervorrichtung, die den Prozessor **1202** an einen PCI-Express-Bus koppelt, über welchen der Prozessor **1202** mit den GPGPUs **1206A-D** kommunizieren kann. In mindestens einer Ausführungsform können sich die GPGPUs **1206A-D** über eine Anzahl von Hochgeschwindigkeits-Punkt-zu-Punkt-GPU-zu-GPU-Verbindungen **1216** zwischenverbinden. In mindestens einer Ausführungsform verbinden sich die P2P-Verbindungen **1216** über eine dedizierte GPU-Verbindung mit jeder der GPGPUs **1206A-D**. In mindestens einer Ausführungsform ermöglichen die P2P-GPU-Verbindungen **1216** eine direkte Kommunikation zwischen jeder der GPGPUs **1206A-D**, ohne eine Kommunikation über den Host-Schnittstellenbus **1204** zu erfordern, mit welchem der Prozessor **1202** verbunden ist. In mindestens einer Ausführungsform bleibt mit GPU-zu-GPU-Datenaufkommen, das auf die P2P-GPU-Verbindungen **1216** gerichtet ist, der Host-Schnittstellenbus **1204** für Systemspeicherzugriff oder zur Kommunikation mit anderen Instanzen des Multi-GPU-Computersystems **1200A**, beispielsweise, über eine oder mehrere Netzwerkvorrichtung(en) verfügbar. Während in mindestens einer Ausführungsform die GPGPUs **1206A-D** über den Host-Schnittstellen-Schalter **1204** mit dem Prozessor **1202** verbinden, umfasst in mindestens einer Ausführungsform der Prozessor **1202** direkte Unterstützung für die P2P-GPU-Verbindungen **1216** und kann direkt mit den GPGPUs **1206A-D** verbunden werden.

[0201] Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenz- und/oder Trainingslogik **615** sind nachstehend in Verbindung mit **Fig. 6B** und/oder **6C** bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik **615** in dem Multi-GPU-Computersystem **1200A** verwendet werden, um Operationen zu inferenzieren oder vorherzusagen, 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.

[0202] Fig. 12B ist ein Blockdiagramm eines Graphikprozessors **1200B** gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform umfasst der Graphikprozessor **1200B** eine Ring-Zwischenverbindung **1202**, ein Pipeline-Frontend **1204**, eine Medien-Engine **1237** und Graphikkern(e) **1280A-1280N**. In mindestens einer Ausführungsform koppelt die Ring-Zwischenverbindung **1202** den Graphikprozessor **1200B** an andere Verarbeitungseinheiten, einschließlich anderer Graphikprozessoren oder einen oder mehrere universelle Prozessorkerne. In mindestens einer Ausführungsform ist der Graphikprozessor **1200B** einer von vielen Prozessoren, die innerhalb eines Multicore-Verarbeitungssystems integriert sind.

[0203] In mindestens einer Ausführungsform empfängt der Graphikprozessor **1200B** Stapel von Befehlen über die Ring-Zwischenverbindung **1202**. In mindestens einer Ausführungsform werden eingehende Befehle von einem Befehlsstreamer **1203** in dem Pipeline-Frontend **1204** interpretiert. In mindestens einer Ausführungsform umfasst der Graphikprozessor **1200B** eine skalierbare Ausführungslogik zur Durchführung von 3D-Geometrieverarbeitung und Medienverarbeitung über den/die Graphikkern(e) **1280A-1280N**. In mindestens einer Ausführungsform liefert der Befehlsstreamer **1203** für 3D-Geometrieverarbeitungsbefehle Befehle an die Geometrie-Pipeline **1236**. In mindestens einer Ausführungsform liefert der Befehlsstreamer **1203** für mindestens einige Befehle zur Medienverarbeitung Befehle an ein Video-Frontend **1234**, das mit einer Medien-Engine **1237** gekoppelt ist. In mindestens einer Ausführungsform umfasst die Medien-Engine **1237** eine Videoqualitäts-Engine (VQE) **1230** zur Video- und Bild-Nachbearbeitung und eine Multiformat-Codier-/Decodier-Engine (MFX) **1233** zur hardwarebeschleunigten Mediendatencodierung und -decodierung. In mindestens einer Ausführungsform erzeugen die Geometrie-Pipeline **1236** und die Medien-Engine **1237** jeweils Ausführungsthreads für Thread-Ausführungsressourcen, die von mindestens einem Graphikkern **1280A** bereitgestellt werden.

[0204] In mindestens einer Ausführungsform umfasst der Graphikprozessor **1200B** skalierbare Thread-Ausführungsressourcen mit modularen Kernen **1280A-1280N** (manchmal als Kern-Slices bezeichnet), die jeweils mehrere Unterkerne **1250A-550N**, **1260A-1260N** (manchmal als Kern-Subslices bezeichnet) aufweisen. In mindestens einer Ausführungsform kann der Graphikprozessor **1200B** eine beliebige Anzahl von Graphikkernen **1280A bis 1280N** aufweisen. In mindestens einer Ausführungsform umfasst der Graphikprozessor **1200B** einen Graphikkern **1280A** mit mindestens einem ersten Unterkern **1250A** und einem zweiten Unterkern **1260A**. In mindestens einer Ausführungsform ist der Graphikprozessor **1200B** ein stromsparender Prozessor mit einem einzigen Unterkern (z.B. **1250A**). In mindestens einer Ausführungsform umfasst der Graphikprozessor **1200B** mehrere Graphikkern(e) **1280A-1280N**, die jeweils einen Satz von ersten Unterkernen **1250A-1250N** und einen Satz von zweiten Unterkernen **1260A-1260N** umfassen. In mindestens einer Ausführungsform umfasst jeder Unterkern in den ersten Unterkernen **1250A-1250N** mindestens einen ersten Satz von Ausführungseinheiten **1252A-1252N** und Medien/Textur-Abtastern **1254A-1254N**. In mindestens einer Ausführungsform umfasst jeder Unterkern in den zweiten Unterkernen **1260A-1260N** mindestens einen zweiten Satz von Ausführungseinheiten **1262A-1262N** und Abtastern **1264A-1264N**. In mindestens einer Ausführungsform teilt sich jeder Unterkern **1250A-1250N**, **1260A-1260N** einen Satz gemeinsamer Ressourcen **1270A-1270N**. In mindestens einer Ausführungsform umfassen gemeinsam genutzte Ressourcen gemeinsam genutzten Cache-Speicher und Pixeloperationslogik.

[0205] Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten zu der Inferenz- und/oder Trainingslogik **615** werden nachstehend in Verbindung mit Fig. 6B und/oder 6C bereitgestellt. In mindestens einer Ausführungsform kann die Inferenz- und/oder Trainingslogik **615** in dem Graphikprozessor **1200B** 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.

[0206] Fig. 13 ist ein Blockdiagramm, das Mikroarchitektur für einen Prozessor **1300**, der Logikschaltungen zum Ausführen von Anweisungen umfassen kann, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann der Prozessor **1300** Anweisungen ausführen, einschließlich von x86-Befehlen, ARM-Befehlen, speziellen Anweisungen für anwendungsspezifische integrierte Schaltungen (ASICs) usw. In mindestens einer Ausführungsform kann der Prozessor **1300** Register umfassen zum Speichern gepackter Daten umfassen, wie beispielsweise 64-Bit breite MMX™-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 ge-

packten 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 umfasst. In mindestens einer Ausführungsform kann der Prozessor **1300** Anweisungen ausführen, um maschinelles Lernen oder Algorithmen für tiefes Lernen, Training oder Inferenzen zu beschleunigen.

[0207] In mindestens einer Ausführungsform umfasst der Prozessor **1300** ein In-Reihenfolge-Frontend („Frontend“) **1301**, um auszuführende Anweisungen zu holen und Anweisungen vorzubereiten, die später in der Prozessorpipeline zu verwenden sind. In mindestens einer Ausführungsform kann das Frontend **1301** mehrere Einheiten umfassen. In mindestens einer Ausführungsform holt ein Anweisungs-Vorabrufer bzw. -Prefetcher **1326** Anweisungen aus dem Speicher und leitet Anweisungen an einen Anweisungsdecodierer **1328** weiter, der wiederum Anweisungen decodiert oder interpretiert. In mindestens einer Ausführungsform decodiert der Anweisungsdecodierer **1328** eine empfangene Anweisung in eine oder mehrere Operationen, die als „Mikroanweisungen“ oder „Mikrooperationen“ bezeichnet werden (die auch als „Mikro-Ops“ oder „Ups“ bezeichnet werden), die eine Maschine ausführen kann. In mindestens einer Ausführungsform parst bzw. analysiert der Anweisungsdecodierer **1328** 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 **1330** decodierte Ups zu programmgeordneten Sequenzen oder Spuren in einer Uop-Warteschlange **1334** zur Ausführung zusammenstellen. In mindestens einer Ausführungsform stellt dann, wenn der Trace-Cache **1330** auf eine komplexe Anweisung trifft, ein Mikrocode-ROM **1332** die für den Abschluss der Operation erforderlichen Ups bereit.

[0208] In mindestens einer Ausführungsform können einige Anweisungen in eine einzige Mikro-Op umgewandelt werden, wohingegen andere mehrere Mikro-Operationen benötigen, um die volle Operation abzuschließen.

[0209] In mindestens einer Ausführungsform kann dann, wenn mehr als vier Mikro-Ops benötigt werden, um eine Anweisung abzuschließen, der Anweisungsdecodierer **1328** auf den Mikrocode-ROM **1332** zugreifen, um die Anweisung auszuführen. In mindestens einer Ausführungsform kann eine Anweisung zur Verarbeitung an dem Anweisungsdecodierer **1328** in eine kleine Anzahl von Mikro-Ops decodiert werden. In mindestens einer Ausführungsform kann eine Anweisung in dem Mikrocode ROM **1332** gespeichert werden, falls eine Anzahl von Mikro-Operationen zur Durchführung der Operation erforderlich sein sollte. In mindestens einer Ausführungsform bezieht sich der Trace-Cache **1330** auf eine programmierbare Eintrittspunkt-Logikanordnung („PLA“), um einen korrekten Mikroanweisungszeiger zum Lesen von Mikrocodesequenzen zu bestimmen, um eine oder mehrere Anweisungen von dem Mikrocode ROM **1332** abzuschließen, gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform beendet der Mikrocode-ROM **1332** die Sequenzierung von Mikro-Ops für eine Anweisung, wobei das Frontend **1301** der Maschine das Abholen von Mikro-Ops aus dem Trace-Cache **1330** wieder aufnehmen kann.

[0210] In mindestens einer Ausführungsform kann eine Außer-Betrieb-Ausführungs-Engine („Out-of-Order-Engine“) **1303** Anweisungen zur 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. Die Out-of-Order-Ausführungs-Engine **1303** umfasst, ohne darauf beschränkt zu sein, einen Allokator/Register-Umbenener **1340**, eine Speicher-Uop-Warteschlange **1342**, eine Ganzzahl/Gleitkomma-Uop-Warteschlange **1344**, einen Speicher-Planer **1346**, einen Schnell-Planer **1302**, einen Langsam/Universal-Gleitkomma-Planer („Langsam/Universal-FP-Planer“) **1304** und einen einfachen Gleitkomma-Planer („Einfach-FP-Planer“) **1306**. In mindestens einer Ausführungsform werden der Schnell-Planer **1302**, der Langsam/Universal-Gleitkomma-Planer **1304** und der Einfach-Gleitkomma-Planer **1306** hier auch kollektiv als „Uop-Planer **1302, 1304, 1306**“ bezeichnet. In mindestens einer Ausführungsform teilt der Allokator/Register-Umbenener **1340** Engine- bzw. Maschinenpuffer und Ressourcen zu, die jeder Uop zur Ausführung benötigt. In mindestens einer Ausführungsform benennt der Allokator/Register-Umbenener **1340** Logikregister auf Einträge in einer Registerdatei um. In mindestens einer Ausführungsform teilt der Allokator/Register-Umbenener **1340** auch einen Eintrag für jeden Uop in einer von zwei Uop-Warteschlangen, der Speicher-Uop-Warteschlange **1342** für Speicheroperationen und der Ganzzahl/Gleitkomma-Uop-Warteschlange **1344** für Nichtspeicheroperationen, vor dem Speicher-Planer **1346** und den Uop-Planern **1302, 1304, 1306** zu. In mindestens einer Ausführungsform bestimmen die Uop-Planer **1302, 1304, 1306**, wann ein Uop ausführbar ist, basierend auf der Bereitschaft ihrer abhängigen Eingangsregister-Operandenquellen und der Verfügbarkeit von Ausführungs-

ressourcen, die Uops benötigen, um ihre Operation abzuschließen. In mindestens einer Ausführungsform kann der Schnell-Planer **1302** auf jeder Hälfte eines Haupttaktzyklus planen, während der Langsam/Universal-Gleitkomma-Planer **1304** und der Einfach-Gleitkomma-Planer **1306** einmal pro Hauptprozessortaktzyklus planen können. In mindestens einer Ausführungsform arbitrieren die Uop-Planer **1302**, **1304**, **1306** für Sende-Ports, um Uops zur Ausführung einzuplanen.

[0211] In mindestens einer Ausführungsform umfasst ein Ausführungsblock **1311**, ohne darauf beschränkt zu sein, ein Ganzzahl-Registerdatei/Umgehungs-Netzwerk **1308**, ein Gleitkomma-Registerdatei/Umgehungs-Netzwerk („FP-Registerdatei/Umgehungsnetzwerk“) **1310**, Adresserzeugungseinheiten („AGUs“) **1312** und **1314**, schnelle Arithmetiklogikeinheiten (ALUs) („schnelle ALUs“) **1316** und **1318**, eine langsame Arithmetiklogikeinheit („langsame ALU“) **1312**, eine Gleitkomma-ALU („FP“) **1322** und eine Gleitkomma-Bewegungseinheit („FP Bewegung“) **1324**. In mindestens einer Ausführungsform werden das Ganzzahl-Registerdatei/Umgehungs-Netzwerk **1308** und das Gleitkomma-Registerdatei/Umgehungs-Netzwerk **1310** hier auch als „Registerdateien **1308**, **1310**“ bezeichnet. In mindestens einer Ausführungsform werden die AGUs **1312** und **1314**, die schnellen ALUs **1316** und **1318**, die langsame ALU **1312**, die Gleitkomma-ALU **1322** und die Gleitkomma-Bewegungseinheit **1324** auch als „Ausführungseinheiten **1312**, **1314**, **1316**, **1318**, **1312**, **1322** und **1324**“ bezeichnet. In mindestens einer Ausführungsform kann der Ausführungsblock **1311**, ohne darauf beschränkt zu sein, eine beliebige Anzahl (einschließlich Null) und Art von Registerdateien, Umgehungsnetzwerken, Adresserzeugungseinheiten und Ausführungseinheiten in beliebiger Kombination umfassen.

[0212] In mindestens einer Ausführungsform können Registerdateien **1308**, **1310** zwischen den Uop-Planern **1302**, **1304**, **1306** und den Ausführungseinheiten **1312**, **1314**, **1316**, **1318**, **1312**, **1322** und **1324** angeordnet sein. In mindestens einer Ausführungsform führt das Ganzzahl-Registerdatei/Umgehungs-Netzwerk **1308** Ganzzahl-Operationen aus. In mindestens einer Ausführungsform führt das Gleitkomma-Registerdatei/Umgehungs-Netzwerk **1310** Gleitkommaoperationen durch. In mindestens einer Ausführungsform kann jede der Registerdateien **1308**, **1310**, ohne darauf beschränkt zu sein, ein Umgehungs-Netzwerk umfassen, das gerade abgeschlossene Ergebnisse, die noch nicht in die Registerdatei geschrieben wurden, umgehen oder an neue abhängige Uops weiterleiten kann. In mindestens einer Ausführungsform können die Registerdateien **1308**, **1310** Daten miteinander kommunizieren. In mindestens einer Ausführungsform kann das Ganzzahl-Registerdatei/Umgehungs-Netzwerk **1308**, 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 das Gleitkomma-Registerdatei/Umgehungs-Netzwerk **1310**, ohne darauf beschränkt zu sein, 128-Bit breite Einträge umfassen, weil Gleitkommaanweisungen typischerweise Operanden von 64 bis 128 Bit Breite aufweisen.

[0213] In mindestens einer Ausführungsform können die Ausführungseinheiten **1312**, **1314**, **1316**, **1318**, **1312**, **1322**, **1324** Anweisungen ausführen. In mindestens einer Ausführungsform speichern die Registerdateien **1308**, **1310** Ganzzahl- und Gleitkomma-Daten-Operandenwerte, die Mikroanweisungen ausführen müssen. In mindestens einer Ausführungsform kann der Prozessor **1300**, ohne darauf beschränkt zu sein, eine beliebige Anzahl und Kombination von Ausführungseinheiten **1312**, **1314**, **1316**, **1318**, **1312**, **1322**, **1324** umfassen. In mindestens einer Ausführungsform können die Gleitkomma-ALU **1322** und die Gleitkomma-Bewegungseinheit **1324** Gleitkomma-, MMX-, SIMD-, AVX- und SSE-Operationen oder andere Operationen ausführen, einschließlich spezieller Anweisungen für maschinelles Lernen. In mindestens einer Ausführungsform kann die Gleitkomma ALU **1322**, 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 **1316**, **1318** übergeben werden. In mindestens einer Ausführungsform können schnelle ALUS **1316**, **1318** schnelle Operationen mit einer effektiven Latenz von einem halben Taktzyklus ausführen. In mindestens einer Ausführungsform gehen die komplexesten Ganzzahl-Operationen auf die langsame ALU **1320** über, da die langsame ALU **1320**, ohne darauf beschränkt zu sein, eine Ganzzahl-Ausführungshardware für Operationen mit langer Latenz umfassen kann, wie beispielsweise ein Multiplikator, ein Schieber, Flag- bzw. Kennzeichenbit-Logik und Zweigverarbeitung. In mindestens einer Ausführungsform können Speicher-Lade-/Speicher-Operationen durch AGUS **1312**, **1314** ausgeführt werden. In mindestens einer Ausführungsform können die schnelle ALU **1316**, die schnelle ALU **1318** und die langsame ALU **1312** Ganzzahl-Operationen an 64-Bit-Datenoperanden durchführen. In mindestens einer Ausführungsform können die schnelle ALU **1316**, die schnelle ALU **1318** und die langsame ALU **1320** implementiert sein, um eine Mehrzahl von Datenbitgrößen einschließlich sechzehn, zweiunddreißig, 128, 326 usw. zu unterstützen. In mindestens einer Ausführungsform können die Gleitkomma-ALU **1322** und die Gleitkomma-Bewegungseinheit **1324** implementiert sein, um einen Bereich von Operanden mit Bits unterschiedlicher Breite zu unterstützen. In mindestens einer Ausführungsform können die Gleitkomma-ALU **1322**

und die Gleitkomma-Bewegungseinheit **1324** an 128-Bit breite gepackte Datenoperanden in Verbindung mit SIMD- und Multimedia-Anweisungen arbeiten.

[0214] In mindestens einer Ausführungsform versenden die Uop-Planer **1302**, **1304**, **1306** abhängige Operationen, bevor die Ausführung einer Elternlast beendet ist. In mindestens einer Ausführungsform kann, da Uops spekulativ geplant und in dem Prozessor **1300** ausgeführt werden können, der Prozessor **1300** auch eine Logik zur Handhabung von Speicherfehlzugriffen umfassen. In mindestens einer Ausführungsform kann es 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.

[0215] In mindestens einer Ausführungsform kann sich „Register“ auf On-Board-Prozessorspeicherplätze 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 Programmiers) 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 zugeteilte 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.

[0216] Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten zu der Inferenz- und/oder Trainingslogik **615** werden nachstehend in Verbindung mit **Fig. 6B** und/oder **6C** bereitgestellt. In mindestens einer Ausführungsform können Abschnitte oder die gesamte Inferenz- und/oder Trainingslogik **615** in einem Ausführungsblock **1311** und andere Speicher oder Register integriert sein, die gezeigt oder nicht gezeigt sind. In mindestens einer Ausführungsform können hier beschriebene Trainings- und/oder Inferenztechniken eine oder mehrere der in dem Ausführungsblock **1311** veranschaulichten ALUs verwenden. Außerdem können Gewichtparameter in On-Chip- oder Off-Chip-Speicher und/oder in Registern (gezeigt oder nicht gezeigt) gespeichert sein, die ALUs des Ausführungsblocks **1311** konfigurieren, um einen oder mehrere Algorithmen maschinellen Lernens, neuronale Netzwerkarchitekturen, Anwendungsfälle oder hier beschriebene Trainings-techniken auszuführen.

[0217] **Fig. 14** veranschaulicht einen tief lernenden bzw. Deep-Learning-Anwendungsprozessor **1400** gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform verwendet der Deep-Learning-Anwendungsprozessor **1400** Anweisungen, die dann, wenn sie durch den Deep-Learning-Anwendungsprozessor **1400** ausgeführt werden, bewirken, dass der Deep-Learning-Anwendungsprozessor **1400** einige oder alle der in dieser Offenbarung beschriebenen Prozesse und Techniken ausführt. In mindestens einer Ausführungsform ist der Deep-Learning-Anwendungsprozessor **1400** eine anwendungsspezifische integrierte Schaltung (ASIC). In mindestens einer Ausführungsform führt der Anwendungsprozessor **1400** 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 **1400**, ohne darauf beschränkt zu sein, Verarbeitungcluster 1410(1)-1410(12), Inter-Chip-Verbindungen („ICLs“) 1420(1)-1420(12), Inter-Chip-Controller („ICCs“) 1430(1)-1430(2), Speicher-Controller („Mem Ctrlrs“) 1442(1)-1442(4), eine physikalische Schicht von Speicher mit hoher Bandbreite („HBM PHY“) 1444(1)-1444(4), eine Management-Controller-Zentralverarbeitungseinheit („Management-Controller-CPU“) **1450**, eine serielle Peripheriegeräte-Schnittstelle, eine integrierte Inter-Schaltung und einen universellen Eingabe-/Ausgabeblock („SPI, I²C, GPIO“), einen Peripheriekomponenten-Zwischenverbindungs-Express-Controller und einen Direktspeicherzugriffsblock („PCIe Controller und DMA“) **1470**, sowie einen sechzehnkanaligen Peripheriekomponenten-Zwischenverbindungs-Express-Port („PCI Express × 16“) **1480**.

[0218] In mindestens einer Ausführungsform können Verarbeitungscluster **1410** Deep-Learning-Operationen durchführen, einschließlich Inferenz- oder Vorhersage-Operationen basierend auf Gewichtsparametern, die mit einer oder mehreren Trainingstechniken berechnet wurden, einschließlich der hier beschriebenen. In mindestens einer Ausführungsform kann jeder Verarbeitungscluster **1410**, ohne darauf beschränkt zu sein, eine beliebige Anzahl und Art von Prozessoren umfassen. In mindestens einer Ausführungsform kann der Deep-Learning-Anwendungsprozessor **1400** eine beliebige Anzahl und Art von Verarbeitungsclustern **1400** umfassen. In mindestens einer Ausführungsform sind Inter-Chip-Verbindungen **1420** bidirektional. In mindestens einer Ausführungsform ermöglichen Inter-Chip-Verbindungen **1420** und Inter-Chip-Controller **1430** mehreren Deep-Learning-Anwendungs-Prozessoren **1400**, 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 der Deep-Learning-Anwendungsprozessor **1400** eine beliebige Anzahl (einschließlich Null) und Art von ICLs **1420** und ICCs **1430** umfassen.

[0219] In mindestens einer Ausführungsform stellen die HBM2s **1440** insgesamt 32 Gigabyte (GB) Speicher zur Verfügung. Der HBM2 **1440(i)** ist sowohl der Speichersteuerung **1442(i)** als auch der HBM PHY **1444(i)** zugeordnet. In mindestens einer Ausführungsform kann eine beliebige Anzahl von HBM2s **1440** eine beliebige Art und Gesamtmenge an Speicher mit hoher Bandbreite bereitstellen und kann einer beliebigen Anzahl (einschließlich Null) und Art von Speicher-Controllern **1442** und HBM PHYs **1444** zugeordnet sein. In mindestens einer Ausführungsform können SPI, I²C, GPIO **1460**, PCIe-Controller und DMA **1470** und/oder PCIe **1480** durch eine beliebige Anzahl und Art von Blöcken ersetzt sein, die eine beliebige Anzahl und Art von Kommunikationsstandards in beliebiger technisch machbarer Weise ermöglichen.

[0220] Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenz- und/oder Trainingslogik **615** werden nachstehend in Verbindung mit **Fig. 6B** und/oder **6C** 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 **1400** bereitgestellt werden. In mindestens einer Ausführungsform wird der Deep-Learning-Anwendungsprozessor **1400** verwendet, um Informationen basierend auf einem trainierten Modell maschinellen Lernens (z.B. ein neuronales Netzwerk), das von einem anderen Prozessor oder System oder von dem Deep-Learning-Anwendungsprozessor **1400** trainiert wurde, abzuleiten oder vorherzusagen. In mindestens einer Ausführungsform kann der Prozessor **3300** verwendet werden, um einen oder mehrere der hier beschriebenen Anwendungsfälle neuronaler Netzwerke durchzuführen.

[0221] **Fig. 15** ist ein Blockdiagramm eines neuromorphen Prozessors **1500** gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann der neuromorphe Prozessor **1500** ein oder mehrere Eingaben von zu dem neuromorphen Prozessor **1500** externen Quellen empfangen. In mindestens einer Ausführungsform können diese Eingaben an ein oder mehrere Neuronen **1502** innerhalb des neuromorphen Prozessors **1500** übertragen werden. In mindestens einer Ausführungsform können die Neuronen **1502** und ihre Komponenten unter Verwendung von Schaltungen oder Logik, einschließlich einer oder mehrerer Arithmetik-Logikeinheiten (ALUs), implementiert sein. In mindestens einer Ausführungsform kann der neuromorphe Prozessor **1500**, ohne darauf beschränkt zu sein, Tausende oder Millionen von Instanzen von Neuronen **1502** umfassen, wobei jedoch eine beliebige Anzahl von Neuronen **1502** verwendet werden kann. In mindestens einer Ausführungsform kann jede Instanz des Neurons **1502** einen Neuroneneingang **1504** und einen Neuronenausgang **1506** umfassen. In mindestens einer Ausführungsform können Neuronen **1502** Ausgaben erzeugen, die Eingänge anderer Instanzen der Neuronen **1502** übertragen werden können. In mindestens einer Ausführungsform können die Neuronen-Eingänge **1504** und die Neuronen-Ausgänge **1506** über Synapsen **1508** miteinander verbunden sein.

[0222] In mindestens einer Ausführungsform können die Neuronen **1502** und die Synapsen **1508** derart miteinander verbunden sein, dass der neuromorphe Prozessor **1500** zum Verarbeiten oder Analysieren der von dem neuromorphen Prozessor **1500** empfangenen Informationen eingesetzt wird. In mindestens einer Ausführungsform können die Neuronen **1502** einen Ausgangsimpuls (oder „Feuern“ oder „Spike“) senden, wenn über den Neuroneneingang **1504** empfangenen Eingaben einen Schwellenwert überschreiten. In mindestens einer Ausführungsform können die Neuronen **1502** Signale, die an den Neuroneneingängen **1504** empfangen wurden, summieren oder integrieren. In mindestens einer Ausführungsform können in mindestens einer Ausführungsform Neuronen **1502** als lecke Integrations- und Feuerneuronen implementiert sein, wobei dann, wenn eine Summe (als ein „Membranpotential“ bezeichnet) einen Schwellenwert überschreitet, ein Neuron **1502**

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 Feuervneuron Signale, die an den Neuroneneingängen **1504** 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 Feuervneuron feuern, falls mehrere Eingangssignale an den Neuroneneingängen **1504** schnell genug empfangen werden, um einen Schwellenwert zu überschreiten (in mindestens einer Ausführungsform ist dies bevor ein Membranpotential zu niedrig zum Feuervn wird). In mindestens einer Ausführungsform können Neuronen **1502** 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 kann eine beliebige andere geeignete Übertragungsfunktion verwendet werden. Ferner können in mindestens einer Ausführungsform, ohne darauf beschränkt zu sein, die Neuronen **1502** Komparatorschaltungen oder Logik umfassen, die einen Ausgangsspike an dem Neuronenausgang **1506** erzeugen, wenn ein Ergebnis der Anwendung einer Übertragungsfunktion auf den Neuroneneingang **1504** einen Schwellenwert überschreitet. In mindestens einer Ausführungsform kann, nachdem das Neuron **1502** 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, das Neuron **1502** nach einer geeigneten Zeitspanne (oder Refraktärperiode) den Normalbetrieb wieder aufnehmen.

[0223] In mindestens einer Ausführungsform können die Neuronen **1502** durch Synapsen **1508** miteinander verbunden sein. In mindestens einer Ausführungsform können die Synapsen **1508** betrieben werden, um Signale von einem Ausgang eines ersten Neurons **1502** zu einem Eingang eines zweiten Neurons **1502** zu übertragen. In mindestens einer Ausführungsform können Neuronen **1502** Informationen über mehr als eine Instanz der Synapse **1508** übertragen. In mindestens einer Ausführungsform können ein oder mehrere Instanzen des Neuronenausgangs **1506** über eine Instanz der Synapse **1508** mit einer Instanz des Neuroneneingangs **1504** im gleichen Neuron **1502** verbunden sein. In mindestens einer Ausführungsform kann eine Instanz des Neurons **1502**, die eine Ausgabe erzeugt, die über eine Instanz der Synapse **1508** übertragen werden soll, als „präsynaptisches Neuron“ in Bezug auf diese Instanz der Synapse **1508** bezeichnet werden. In mindestens einer Ausführungsform kann eine Instanz des Neurons **1502**, die eine über eine Instanz der Synapse **1508** übertragene Eingabe empfängt, in Bezug auf diese Instanz der Synapse **1508** als „postsynaptisches Neuron“ bezeichnet sein. Weil eine Instanz des Neurons **1502** Eingaben von einer oder mehreren Instanzen der Synapse **1508** empfangen kann, und auch Ausgaben über einen oder mehrere Instanzen der Synapse **1508** übertragen kann, kann eine einzelne Instanz des Neurons **1502** daher sowohl ein „präsynaptisches Neuron“ als auch ein „postsynaptisches Neuron“ in Bezug auf verschiedene Instanzen von Synapsen **1508** in mindestens einer Ausführungsform sein.

[0224] In mindestens einer Ausführungsform können die Neuronen **1502** in einer oder mehreren Schichten organisiert sein. Jede Instanz des Neurons **1502** kann einen Neuronenausgang **1506** aufweisen, der sich durch eine oder mehrere Synapsen **1508** zu einem oder mehreren Neuroneneingängen **1504** ausbreiten kann. In mindestens einer Ausführungsform können Neuronenausgänge **1506** 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 die Schicht **1510** als eine „Feed-Forward-Schicht“ bezeichnet sein. In mindestens einer Ausführungsform kann sich jede Instanz des Neurons **1502** in einer Instanz der ersten Schicht **1510** zu jeder Instanz des Neurons **1502** in der zweiten Schicht **1512** ausbreiten. In mindestens einer Ausführungsform kann die erste Schicht **1510** als „vollständig verbundene Feed-Forward-Schicht“ bezeichnet sein. In mindestens einer Ausführungsform kann sich jede Instanz des Neurons **1502** in einer Instanz der zweiten Schicht **1512** auf weniger als alle Instanzen des Neurons **1502** in einer dritten Schicht **1514** verteilen. In mindestens einer Ausführungsform kann die zweite Schicht **1512** als „spärlich verbundene Feed-Forward-Schicht“ bezeichnet sein. In mindestens einer Ausführungsform können sich Neuronen **1502** in der zweiten Schicht **1512** zu Neuronen **1502** in mehreren anderen Schichten ausbreiten, einschließlich zu Neuronen **1502** in der (gleichen) zweiten Schicht **1512**. In mindestens einer Ausführungsform kann die zweite Schicht **1512** als „rekurrente Schicht“ bezeichnet sein. Der neuromorphe Prozessor **1500** 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.

[0225] In mindestens einer Ausführungsform kann der neuromorphe Prozessor **1500**, ohne darauf beschränkt zu sein, eine rekonfigurierbare Zwischenverbindungsarchitektur oder dedizierte fest verdrahtete Zwischenverbindungen umfassen, um die Synapse **1508** mit den Neuronen **1502** zu verbinden. In mindestens einer Aus-

führungsform kann der neuromorphe Prozessor **1500**, ohne darauf beschränkt zu sein, eine Schaltung oder Logik umfassen, die ermöglicht, Synapsen bei Bedarf verschiedenen Neuronen **1502** basierend auf neuronaler Netzwerktopologie und Neuronen-Fan-In/Out zuzuordnen. In mindestens einer Ausführungsform können in mindestens einer Ausführungsform Synapsen **1508** mit Neuronen **1502** 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.

[0226] Fig. **16A** ist ein Blockdiagramm eines Verarbeitungssystems gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform umfasst das System **1600A** einen oder mehrere Prozessoren **1602** und einen oder mehrere Graphikprozessoren **1608** und kann ein Desktop-System mit einem einzelnen Prozessor, ein Multiprozessor-Workstation-System oder ein Serversystem mit einer großen Anzahl von Prozessoren **1602** oder Prozessorkernen **1607** sein. In mindestens einer Ausführungsform ist das System **1600A** eine Verarbeitungsplattform, die in eine System-on-a-Chip (SoC)-integrierte Schaltung zur Verwendung in mobilen, tragbaren oder eingebetteten Geräten integriert ist.

[0227] In mindestens einer Ausführungsform kann das System **1600A** eine Spielkonsole, einschließlich einer Spiel- und Medienkonsole, eine mobile Spielkonsole, eine tragbare Spielkonsole oder eine Online-Spielkonsole umfassen oder in diese innerhalb einer serverbasierten Spieleplattform integriert sein. In mindestens einer Ausführungsform ist das System **1600A** ein Mobiltelefon, Smartphone, Tablet-Computergerät oder mobiles Internetgerät. In mindestens einer Ausführungsform kann das Verarbeitungssystem **1600A** auch ein tragbares Gerät, wie beispielsweise ein tragbares Smart Watch-Gerät, intelligentes Brillengerät, Gerät für erweiterte Realität oder Gerät für virtuelle Realität umfassen, mit ihm koppeln oder in dieses integriert sein. In mindestens einer Ausführungsform ist das Verarbeitungssystem **1600A** ein Fernseh- oder Set-Top-Box-Gerät mit einem oder mehreren Prozessoren **1602** und einer von einem oder mehreren Graphikprozessoren **1608** erzeugten Graphikchnittstelle.

[0228] In mindestens einer Ausführungsform umfassen ein oder mehrere Prozessoren **1602** jeweils einen oder mehrere Prozessorkerne **1607** zum Verarbeiten von Anweisungen, die dann, wenn sie ausgeführt werden, Operationen für ein System- und Benutzersoftware ausführen. In mindestens einer Ausführungsform ist jeder eines oder mehrerer Prozessorkerne **1607** konfiguriert, um einen bestimmten Anweisungssatz **1609** zu verarbeiten. In mindestens einer Ausführungsform kann der Anweisungssatz **1609** Complex Instruction Set Computing (CISC), Reduced Instruction Set Computing (RISC) oder das Rechnen über ein Very Long Instruction Word (VLIW) erleichtern. In mindestens einer Ausführungsform können die Prozessorkerne **1607** jeweils einen anderen Anweisungssatz **1609** verarbeiten, die Anweisungen zum Erleichtern der Emulation anderer Anweisungssätze umfassen kann. In mindestens einer Ausführungsform kann der Prozessorkern **1607** auch andere Verarbeitungsvorrichtungen, wie beispielsweise einen digitalen Signalprozessor (DSP), umfassen.

[0229] In mindestens einer Ausführungsform umfasst der Prozessor **1602** den Cache-Speicher **1604**. In mindestens einer Ausführungsform kann der Prozessor **1602** einen einzelnen internen Cache oder mehrere Ebenen internen Caches aufweisen. In mindestens einer Ausführungsform wird der Cache-Speicher von verschiedenen Komponenten des Prozessors **1602** gemeinsam genutzt. In mindestens einer Ausführungsform verwendet der Prozessor **1602** auch einen externen Cache (z.B. einen Level 3 (L3)-Cache oder einen Last Level Cache (LLC)) (nicht gezeigt), welcher unter Verwendung bekannter Cache-Kohärenztechniken von den Prozessorkernen **1607** gemeinsam genutzt werden kann. In mindestens einer Ausführungsform ist die Registerdatei **1606** zusätzlich in dem Prozessor **1602** enthalten, welcher verschiedene Registertypen zum Speichern verschiedener Datentypen umfassen kann (z.B. ein Ganzzahlregister, ein Gleitkommaregister, ein Zustandsregister und ein Befehlszeigerregister). In mindestens einer Ausführungsform kann die Registerdatei **1606** Universalregister oder andere Register umfassen.

[0230] In mindestens einer Ausführungsform sind ein oder mehrere Prozessor(en) **1602** mit einem oder mehreren Schnittstellenbus(en) **1610** gekoppelt, um Kommunikationssignale wie Adressen, Daten oder Steuersignale zwischen dem Prozessor **1602** und anderen Komponenten in dem System **1600A** zu übertragen. In mindestens einer Ausführungsform kann, in einer Ausführungsform, ein Schnittstellenbus **1610** ein Prozessorbuss sein, wie beispielsweise eine Version eines Direct Media Interface (DMI)-Busses. In mindestens einer Ausführungsform ist die Schnittstelle **1610** nicht auf einen DMI-Bus beschränkt und kann einen oder mehrere Peripheral Connect Zwischenverbindung-Busse (z.B. PCI, PCI Express), Speicherbusse oder andere Arten von Schnittstellenbussen umfassen. In mindestens einer Ausführungsform umfassen der/die Prozessor(en) **1602** eine integrierte Speicher-Controller **1616** und einen Plattform-Controller-Hub **1630**. In mindestens einer Ausführungsform erleichtert der Speicher-Controller **1616** die Kommunikation zwischen einer Speichervorrichtung

und anderen Komponenten des Systems **1600A**, während der Plattform-Controller-Hub (PCH) **1630** Verbindungen zu E/A-Vorrichtungen über einen lokalen E/A-Bus bereitstellt.

[0231] In mindestens einer Ausführungsform kann die Speichervorrichtung **1620** eine dynamische Direktzugriffsspeicher (DRAM)-Vorrichtung, eine statische Direktzugriffsspeicher (SRAM)-Vorrichtung, eine Flash-Speicher-Vorrichtung, eine Phasenänderungs-Speichervorrichtung oder irgendeine andere Speichervorrichtung mit geeigneter Leistung sein, um als Prozessspeicher zu dienen. In mindestens einer Ausführungsform kann die Speichervorrichtung **1620** als Systemspeicher für das System **1600A** arbeiten, um Daten **1622** und Anweisungen **1621** zu speichern, die verwendet werden, wenn ein oder mehrere Prozessoren **1602** eine Anwendung oder einen Prozess ausführen. In mindestens einer Ausführungsform koppelt der Speichercontroller **1616** darüber hinaus mit einem, in mindestens einer Ausführungsform, optionalen externen Graphikprozessor **1612**, welcher mit einem oder mehreren Graphikprozessoren **1608** in den Prozessoren **1602** kommunizieren kann, um Graphik- und Medienoperationen durchzuführen. In mindestens einer Ausführungsform kann eine Anzeigevorrichtung **1611** eine Verbindung mit dem/den Prozessor(en) **1602** herstellen. In mindestens einer Ausführungsform kann die Anzeigevorrichtung **1611** eine oder mehrere einer internen Anzeigevorrichtung umfassen, wie in einer mobilen elektronischen Vorrichtung oder einer Laptopvorrichtung oder einer externen Anzeigevorrichtung, die über eine Anzeigeschnittstelle (z.B. DisplayPort usw.) angeschlossen ist. In mindestens einer Ausführungsform kann die Anzeigevorrichtung **1611** eine am kopfangebrachte Anzeige (HMD) umfassen, wie beispielsweise eine stereoskopische Anzeigevorrichtung zur Verwendung in Virtual-Reality-Anwendungen (VR) oder Augmented-Reality-Anwendungen (AR).

[0232] In mindestens einer Ausführungsform ermöglicht der Plattform-Controller-Hub **1630** Peripheriegeräten, sich mit der Speichervorrichtung **1620** und dem Prozessor **1602** über einen Hochgeschwindigkeits-E/A-Bus zu verbinden. In mindestens einer Ausführungsform umfassen E/A-Peripheriegeräte unter anderem, sind jedoch nicht beschränkt auf, einen Audio-Controller **1646**, einen Netzwerk-Controller **1634**, eine Firmware-Schnittstelle **1611**, einen drahtlosen Sender-Empfänger **1626**, Berührungssensoren **1625**, eine Datenspeichervorrichtung **1624** (z.B. eine Festplatte, einen Flash-Speicher usw.). In mindestens einer Ausführungsform kann die Datenspeichervorrichtung **1624** über eine Speicherschnittstelle (z.B. SATA) oder über einen Peripheriebus, wie beispielsweise einen Peripheral Component Interconnection Bus (z.B. PCI, PCI Express) verbinden. In mindestens einer Ausführungsform können die Berührungssensoren **1625** Sensoren eines berührungsempfindlichen Bildschirms, Drucksensoren oder Fingerabdrucksensoren umfassen. In mindestens einer Ausführungsform kann der drahtlose Sender-Empfänger **1626** ein Wi-Fi-Sender-Empfänger, ein Bluetooth-Sender-Empfänger oder ein Mobilfunk-Sender-Empfänger wie beispielsweise ein 3G-, 4G- oder Long Term Evolution (LTE)-Transceiver sein. In mindestens einer Ausführungsform ermöglicht die Firmware-Schnittstelle **1611** die Kommunikation mit der System-Firmware und kann beispielsweise eine vereinheitlichte erweiterbare Firmware-Schnittstelle (UEFI) sein. In mindestens einer Ausführungsform kann der Netzwerk-Controller **1634** eine Netzwerkverbindung zu einem verdrahteten Netzwerk aktivieren.

[0233] In mindestens einer Ausführungsform koppelt einen Hochleistungs-Netzwerk-Controller (nicht gezeigt) mit dem Schnittstellenbus **1610**. In mindestens einer Ausführungsform ist der Audio-Controller **1646** ein mehrkanaliger hochauflösenden Audio-Controller. In mindestens einer Ausführungsform umfasst das System **1600A** einen Legacy-E/A-Controller **1640** zum Koppeln von Legacy-Geräten (z.B. Personal System 2 (PS/2)) mit dem System. In mindestens einer Ausführungsform kann der Plattform-Controller-Hub **1630** auch mit einer oder mehreren Universal Serial Bus (USB)-Controllern **1642** verbinden, um Eingabegeräte, wie beispielsweise Tastatur- und Maus-Kombinationen **1643**, eine Kamera **1644** oder andere USB-Eingabegeräte, zu verbinden.

[0234] In mindestens einer Ausführungsform kann eine Instanz des Speichercontroller **1616** und des Plattform-Controller-Hub **1630** in einem diskreten externen Graphikprozessor, wie beispielsweise dem externen Graphikprozessor **1612**, integriert sein. In mindestens einer Ausführungsform können der Plattform-Controller-Hub **1630** und/oder der Speicher-Controller **1616** extern zu einem oder mehreren Prozessoren **1602** sein. In mindestens einer Ausführungsform kann das System **1600A** einen externen Speicher-Controller **1616** und einen Plattform-Controller-Hub **1630** umfassen, die als ein Speicher-Controller-Hub und ein peripherer Controller-Hub innerhalb eines System-Chipsatzes, der mit dem bzw. den Prozessor(en) **1602** in Verbindung steht, konfiguriert sein können.

[0235] Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenz- und/oder Trainingslogik **615** werden nachstehend in Verbindung mit Fig. 6B und/oder 6C bereitgestellt. In mindestens einer Ausführungsform können Abschnitte oder die gesamte Inferenz- und/oder Trainingslogik **615** in dem Graphikprozessor **1600A** integriert sein. In mindestens einer Ausführungsform können

hier beschriebene Trainings- und/oder Inferenz-Techniken eine oder mehrere der in einem Graphikprozessor verkörpert ALUs verwenden. Außerdem können in mindestens einer Ausführungsform hier beschriebene Inferenz- und/oder Trainingsoperationen unter Verwendung einer anderen Logik als die in **Fig. 6B** oder **Fig. 6C** veranschaulichte Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtsparameter in einem On-Chip- oder Off-Chip-Speicher und/oder Registern (gezeigt oder nicht gezeigt) gespeichert werden, die ALUs des Graphikprozessors **1600A** konfigurieren, um einen oder mehrere Algorithmen maschinellen Lernens, neuronale Netzwerkarchitekturen, Anwendungsfälle oder Trainingstechniken auszuführen, die hier beschrieben sind.

[0236] IG. 16B ist ein Blockdiagramm eines Prozessors **1600B** mit einem oder mehreren Prozessorkernen **1602A-1602N**, einen integrierten Speicher-Controller **1614** und einem integrierten Graphikprozessor **1608** gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform kann der Prozessor **1600B** zusätzliche Kerne bis hin zu dem und einschließlich des zusätzlichen Kerns **1602N** umfassen, die durch gestrichelte, linierte Kästen dargestellt sind. In mindestens einer Ausführungsform umfasst jeder der Prozessorkerne **1602A-1602N** eine oder mehrere interne Cache-Einheiten **1604A-1604N**. In mindestens einer Ausführungsform weist jeder Prozessorkern auch Zugriff auf eine oder mehrere gemeinsam genutzte Cache-Einheiten **1606** auf.

[0237] In mindestens einer Ausführungsform stellen die internen Cache-Einheiten **1604A-1604N** und die gemeinsam genutzten Cache-Einheiten **1606** eine Cache-Speicherhierarchie innerhalb des Prozessors **1600B** dar. In mindestens einer Ausführungsform können die Cache-Speichereinheiten **1604A-1604N** mindestens eine Befehls- und Daten-Cache-Ebene innerhalb jedes Prozessorkerns und eine oder mehrere Ebenen gemeinsam genutzten Mid-Level-Caches, wie beispielsweise ein Level 2 (L2), ein Level 3 (L3), ein Level 4 (L4) oder andere Cache-Ebenen, umfassen, wobei eine höchste Cache-Ebene vor externem Speicher als eine LLC klassifiziert ist. In mindestens einer Ausführungsform behält die Cache-Kohärenzlogik Kohärenz zwischen den verschiedenen Cache-Einheiten **1606** und **1604A-1604N** bei.

[0238] In mindestens einer Ausführungsform kann der Prozessor **1600B** auch einen Satz einer oder mehrerer Bus-Controller-Einheiten **1616** und einen System-Agent-Kern **1610** umfassen. In mindestens einer Ausführungsform verwalten eine oder mehrere Bus-Controller-Einheiten **1616** einen Satz von Peripheriebussen, wie beispielsweise einen oder mehrere PCI- oder PCI-Express-Busse. In mindestens einer Ausführungsform stellt der System-Agent-Kern **1610** Managementfunktionen für verschiedene Prozessorkomponenten zur Verfügung. In mindestens einer Ausführungsform umfasst der System-Agent-Kern **1610** eine oder mehrere integrierte Speichercontroller **1614**, um den Zugriff auf verschiedene externe Speichervorrichtungen (nicht gezeigt) zu verwalten.

[0239] In mindestens einer Ausführungsform umfassen ein oder mehrere der Prozessorkerne **1602A-1602N** Unterstützung für gleichzeitiges Multithreading. In mindestens einer Ausführungsform umfasst der System-Agent-Kern **1610** Komponenten zum Koordinieren und Betreiben der Kerne **1602A-1602N** während der Multithreading-Verarbeitung. In mindestens einer Ausführungsform kann der System-Agent-Kern **1610** zusätzlich eine Leistungssteuereinheit (PCU) umfassen, die Logik und Komponenten zum Regeln eines oder mehrerer Leistungszustände der Prozessorkerne **1602A-1602N** und des Graphikprozessors **1608** umfasst.

[0240] In mindestens einer Ausführungsform umfasst der Prozessor **1600B** zusätzlich den Graphikprozessor **1608**, um Graphikverarbeitungsoperationen auszuführen. In mindestens einer Ausführungsform koppelt der Graphikprozessor **1608** mit den gemeinsam genutzten Cache-Einheiten **1606** und dem System-Agent-Kern **1610**, einschließlich einer oder mehrerer integrierter Speichercontroller **1614**. In mindestens einer Ausführungsform umfasst der System-Agent-Kern **1610** auch einen Anzeige-Controller **1611**, um die Graphikprozessorausgabe auf ein oder mehrere gekoppelte Anzeigen zu treiben. In mindestens einer Ausführungsform kann der Anzeige-Controller **1611** auch ein separates Modul sein, das über mindestens eine Zwischenverbindung mit dem Graphikprozessor **1608** gekoppelt ist oder in den Graphikprozessor **1608** integriert sein kann.

[0241] In mindestens einer Ausführungsform wird eine ringbasierte Zwischenverbindungseinheit **1612** verwendet, interne Komponenten des Prozessors **1600B** zu koppeln. In mindestens einer Ausführungsform kann eine alternative Verbindungseinheit verwendet werden, wie beispielsweise eine Punkt-zu-Punkt-Zwischenverbindung, eine geschaltete Zwischenverbindung oder andere Techniken. In mindestens einer Ausführungsform koppelt der Graphikprozessor **1608** über eine E/A-Verbindung **1613** mit der Ring-Zwischenverbindung **1612**.

[0242] In mindestens einer Ausführungsform stellt die E/A-Verbindung **1613** mindestens eine von mehreren Varianten von E/A-Zwischenverbindungen dar, einschließlich einer E/A-Zwischenverbindung auf dem Ge-

häuser, welche die Kommunikation zwischen verschiedenen Prozessorkomponenten und einem eingebetteten Hochleistungsspeichermodule **1618**, wie beispielsweise einem eDRAM-Modul, erleichtert. In mindestens einer Ausführungsform verwenden jeder der Prozessorkerne **1602A-1602N** und der Graphikprozessor **1608** eingebettete Speichermodule **1618** als einen gemeinsam genutzten Cache letzter Ebene bzw. Last Level Cache.

[0243] In mindestens einer Ausführungsform sind die Prozessorkerne **1602A-1602N** homogene Kerne, die eine gemeinsame Anweisungssatzarchitektur ausführen. In mindestens einer Ausführungsform sind die Prozessorkerne **1602A-1602N** heterogen in Bezug auf die Anweisungssatzarchitektur (Instruction Set Architecture; ISA), wobei einer oder mehrere der Prozessorkerne **1602A-1602N** einen gemeinsamen Befehlssatz ausführen, während ein oder mehrere andere Kerne der Prozessorkerne **1602A-1602N** eine Teilmenge eines gemeinsamen Befehlssatzes oder einen bzw. eines anderen Befehlssatzes ausführen. In mindestens einer Ausführungsform sind die Prozessorkerne **1602A-1602N** in Bezug auf Mikroarchitektur heterogen, wobei ein oder mehrere Kerne mit einem relativ höheren Stromverbrauch mit einem oder mehreren Leistungskernen mit einem niedrigeren Stromverbrauch koppeln. In mindestens einer Ausführungsform kann der Prozessor **1600B** auf einem oder mehreren Chips oder als eine in einem SoC integrierte Schaltung implementiert sein.

[0244] Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenz- und/oder Trainingslogik **615** werden nachstehend in Verbindung mit **Fig. 6B** und/oder **6C** bereitgestellt. In mindestens einer Ausführungsform können Abschnitte der oder die gesamte Inferenz- und/oder Trainingslogik **615** in dem Prozessor **1600B** integriert sein. Beispielsweise können in mindestens einer Ausführungsform hier beschriebene Trainings- und/oder Inferenz-Techniken eine oder mehrere ALUs verwenden, die in dem Graphikprozessor **1612**, Graphikkern(en) **1602a-1602N**, gemeinsam genutzter Funktionslogik oder einer anderen Komponenten in **Fig. 16** verkörpert sind. Darüber hinaus können in mindestens einer Ausführungsform hier beschriebene Inferenz- und/oder Trainingsoperationen mit einer anderen Logik als der in den **Fig. 6B** oder **Fig. 6C** veranschaulichten Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtsparameter in einem On-Chip- oder Off-Chip-Speicher und/oder Registern (gezeigt oder nicht gezeigt) gespeichert werden, die ALUs des Graphikprozessors **1600B** konfigurieren, um einen oder mehrere Algorithmen maschinellen Lernens, neuronale Netzwerkarchitekturen, Anwendungsfälle oder die hier beschriebenen Trainingstechniken auszuführen.

[0245] **Fig. 16C** ist ein Blockdiagramm von Hardwarelogik eines Graphikprozessorkerns **1600C** gemäß mindestens einer hier beschriebenen Ausführungsform. In mindestens einer Ausführungsform kann der Graphikprozessorkern **1600C**, manchmal auch als ein Kern-Slice bezeichnet, ein oder mehrere Graphikkern(e) innerhalb eines modularen Graphikprozessors sein. In mindestens einer Ausführungsform ist der Graphikprozessorkern **1600C** beispielhaft für ein Graphikkern-Slice, und einen Graphikprozessor, wie hier beschrieben, kann mehrere Graphikkern-Slices basierend auf einer Sollleistung und Leistungshüllen umfassen. In mindestens einer Ausführungsform kann jeder Graphikkern **1600C** einen Festfunktionsblock **1630** umfassen, der mit mehreren, auch als Sub-Slices bezeichneten Unterkernen **1601A-1601F** gekoppelt ist, die modulare Blöcke von Allzweck- und Festfunktionslogik umfassen.

[0246] In mindestens einer Ausführungsform umfasst der Festfunktionsblock **1630** eine Geometrie- und Festfunktions-Pipeline **1636**, die von allen Unterkernen in dem Graphikprozessor **1600C** 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 **1636** eine 3D-Festfunktions-Pipeline, eine Video-Frontend-Einheit, einen Thread-Spawner und Thread-Abfertiger sowie einen Unified Return Buffer Manager, der vereinheitlichte Rückgabepuffer verwaltet.

[0247] In mindestens einer Ausführungsform umfasst der Festfunktionsblock **1630** auch eine Graphik-SoC-Schnittstelle **1631**, einen Graphik-Mikrocontroller **1638** und eine Medien-Pipeline **1639**. In mindestens einer Ausführungsform stellt die feste, Graphik-SoC-Schnittstelle **1637** eine Schnittstelle zwischen dem Graphikkern **1600C** und anderen Prozessorkernen innerhalb einer integrierten SoC-Schaltung bereit. In mindestens einer Ausführungsform ist der Graphik-Mikrocontroller **1638** ein programmierbarer Subprozessor, der konfiguriert sein kann, um verschiedene Funktionen des Graphikprozessors **1600C** zu verwalten, einschließlich der Thread-Versendung, des Scheduling und der Präemption. In mindestens einer Ausführungsform umfasst die Medien-Pipeline **1639** Logik zur Erleichterung der Decodierung, Codierung, Vorverarbeitung und/oder Nachbearbeitung von Multimedia-Daten, einschließlich Bild- und Videodaten. In mindestens einer Ausführungsform implementiert die Medien-Pipeline **1639** Medienoperationen über Anforderungen an Berechnungs- oder Abtast-Logik innerhalb der Unterkerne **1601-1601F**.

[0248] In mindestens einer Ausführungsform ermöglicht die SoC-Schnittstelle **1637** dem Graphikkern **1600C**, mit universellen 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 die SoC-Schnittstelle **1637** auch Kommunikation mit Vorrichtungen mit fester Funktion innerhalb eines SoC ermöglichen, wie beispielsweise Kameraabbildungspipelines, und ermöglicht die Verwendung und/oder Implementierung globaler Speicher-Atome, die zwischen dem Graphikkern **1600C** und CPUs innerhalb eines SoC gemeinsam genutzt werden können. In mindestens einer Ausführungsform kann die SoC-Schnittstelle **1637** auch Energiemanagementsteuerungen für den Graphikkern **1600C** implementieren und eine Schnittstelle zwischen einer Taktdomäne des Graphikkerns **1600C** und anderen Takt-domänen innerhalb eines SoC ermöglichen. In mindestens einer Ausführungsform ermöglicht die SoC-Schnittstelle **1637** 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 Graphikkerns innerhalb eines Graphikprozessors bereitzustellen. In mindestens einer Ausführungsform können Befehle und Anweisungen an die Medien-Pipeline **1639**, wenn Medienoperationen durchzuführen sind, oder an eine Geometrie- und Festfunktions-Pipeline (z.B. Geometrie- und Festfunktions-Pipeline **1636**, Geometrie- und Festfunktions-Pipeline **1614**), wenn Graphikverarbeitungsoperationen durchzuführen sind, gesendet werden.

[0249] In mindestens einer Ausführungsform kann der Graphik-Mikrocontroller **1638** konfiguriert sein, um verschiedene Planungs- und Managementaufgaben für den Graphikkern **1600C** auszuführen. In mindestens einer Ausführungsform kann der Graphik-Mikrocontroller **1638** eine Graphik- und/oder Rechen-Arbeitslast-Planung auf verschiedenen parallelen Graphik-Engines innerhalb der Ausführungseinheit (EU)-Anordnungen **1602A-1602F**, **1604A-1604F** innerhalb der Unterkerne **1601A-1601F** durchführen. In mindestens einer Ausführungsform kann Host-Software, die auf einem CPU-Kern eines SoC mit dem Graphikkern **1600C** ausgeführt wird, Arbeitslasten einer von mehreren Graphikprozessor-Doorbells übergeben, welche einen Planungsvorgang auf einer geeigneten Graphik-Engine auslösen. 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 der Graphik-Mikrocontroller **1638** auch Zustände mit geringer Stromaufnahme oder Leerlaufzustände für den Graphikkern **1600C** ermöglichen, die dem Graphikkern **1600C** die Möglichkeit geben, Register innerhalb des Graphikkerns **1600C** über Zustandsübergänge mit geringem Stromverbrauch unabhängig von einem Betriebssystem und/oder einer Graphiktreibersoftware auf einem System zu speichern und wiederherzustellen.

[0250] In mindestens einer Ausführungsform kann der Graphikkern **1600C** mehr als oder weniger als die veranschaulichten Unterkerne **1601A-1601F** und bis hin zu N modularen Unterkernen aufweisen. Für jeden Satz von N-Unterkernen kann der Graphikkern **1600C** in mindestens einer Ausführungsform auch eine gemeinsam genutzte Funktionslogik **1610**, einen gemeinsam genutzten und/oder Cache-Speicher **1612**, eine Geometrie/Festfunktions-Pipeline **1614** sowie zusätzliche Festfunktionslogik **1616** zur Beschleunigung verschiedener Graphik- und Rechenverarbeitungsoperationen umfassen. In mindestens einer Ausführungsform kann die gemeinsam genutzte Funktionslogik **1610** Logikeinheiten (z.B. Abtaster-, Mathematik- und/oder Inter-Thread-Kommunikationslogik) umfassen, die von allen N Unterkernen innerhalb des Graphikkerns **1600C** gemeinsam genutzt werden können. In mindestens einer Ausführungsform kann der gemeinsam genutzte und/oder Cache-Speicher **1612** ein Last-Level-Cache für die N Unterkerne **1601A-1601F** innerhalb des Graphikkerns **1600C** sein und kann ebenfalls als gemeinsam genutzter Speicher dienen, auf den mehrere Unterkerne zugreifen können. In mindestens einer Ausführungsform kann die Geometrie/Festfunktions-Pipeline **1614** anstelle der Geometrie/Festfunktions-Pipeline **1636** innerhalb des Festfunktionsblocks **1630** aufgenommen sein und kann gleiche oder ähnliche Logikeinheiten umfassen.

[0251] In mindestens einer Ausführungsform umfasst der Graphikkern **1600C** zusätzliche Festfunktionslogik **1616**, die verschiedene Festfunktionsbeschleunigungslogik zur Verwendung durch den Graphikkern **1600C** umfassen kann. In mindestens einer Ausführungsform umfasst die zusätzliche Festfunktionslogik in Nur-Positions-Shading. Bei dem Nur-Positions-Shading existieren mindestens zwei Geometrie-Pipelines, während in einer Vollgeometrie-Pipeline innerhalb der Geometrie/Festfunktions-Pipeline **1616**, **1636**, und eine Cull-Pipeline, welche eine zusätzliche Geometrie-Pipeline ist, welche innerhalb der zusätzlichen Festfunktionslogik **1616** 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 ver-

bergen, so dass das Shading in einigen Fällen früher abgeschlossen werden kann. In mindestens einer Ausführungsform kann die Cull-Pipeline-Logik innerhalb der zusätzlichen Festfunktionslogik **1616** Positions-Shader parallel zu einer Hauptanwendung ausführen und erzeugt kritische Ergebnisse schneller als eine Vollpipeline, da die Cull-Pipeline das Attribut der Position von Vertices holt und schattiert, ohne eine Rasterisierung und ein Rendering von Pixeln in einem Einzelbild-Puffer durchzuführen. In mindestens einer Ausführungsform kann die 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.

[0252] In mindestens einer Ausführungsform kann die zusätzliche Festfunktionslogik **1616** auch eine maschinell lernende Beschleunigungslogik, wie beispielsweise eine Festfunktion-Matrix-Multiplikationslogik, für Implementierungen einschließlich Optimierungen für das Training oder das Inferenzieren des maschinellen Lernens umfassen.

[0253] In mindestens einer Ausführungsform ist innerhalb jedes Graphik-Unterkerns **1601A-1601F** ein Satz von Ausführungsressourcen umfasst, 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-Unterkerne **1601A-1601F** mehrere EU-Arrays **1602A-1602F**, **1604A-1604F**, Thread-Abfertigungs- und Inter-Thread-Kommunikationslogik **1603A-1603F**, einen 3D-Abtaster **1605A-1605F**, einen Medien-Abtaster **1606A-1606F**, einen Shaderprozessor **1607A-1607F** und einen gemeinsam genutzten lokalen Speicher (SLM) **1608A-1608F**. Die EU-Arrays **1602A-1602F**, **1604A-1604F** umfassen jeweils mehrere Ausführungseinheiten, welche universelle 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 **1603A-1603F** lokale Thread-Abfertigungs- und Thread-Steuer-Operationen für Ausführungseinheiten innerhalb eines Unterkerns durch und erleichtern die Kommunikation zwischen Threads, die auf Ausführungseinheiten eines Unterkerns ausgeführt werden. In mindestens einer Ausführungsform kann der 3D-Abtaster **1605A-1605F** Texturen oder andere 3D-Graphikdaten in den Speicher einlesen. In mindestens einer Ausführungsform kann der 3D-Abtaster Texturdaten unterschiedlich lesen, basierend auf einem konfigurierten Abtastzustand und einem Texturformat, das einer bestimmten Textur zugeordnet ist. In mindestens einer Ausführungsform kann der Medien-Abtaster **1606A-1606F** ähnliche Leseoperationen durchführen, die auf einem Typ und einem Format basieren, der/das Mediendaten zugeordnet ist. In mindestens einer Ausführungsform kann jeder Graphik-Unterkern **1601A-1601F** abwechselnd einen vereinheitlichten 3D- und Medien-Abtaster umfassen. In mindestens einer Ausführungsform können Threads, die auf Ausführungseinheiten innerhalb jedes der Unterkerne **1601A-1601F** ausgeführt werden, den gemeinsam genutzten lokalen Speicher **1608A-1608F** innerhalb jedes Unterkerns verwenden, um Threads, die innerhalb einer Thread-Gruppe ausgeführt werden, die Ausführung unter Verwendung eines gemeinsamen Pools von On-Chip-Speicher zu ermöglichen.

[0254] Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenz- und/oder Trainingslogik **615** werden nachstehend in Verbindung mit **Fig. 6B** und/oder **6C** bereitgestellt. In mindestens einer Ausführungsform können Abschnitte der oder die gesamte Inferenz- und/oder Trainingslogik **615** in den Graphikprozessor **1610** integriert sein. Beispielsweise können in mindestens einer Ausführungsform hier beschriebene Trainings- und/oder Inferenz-Techniken eine oder mehrere der ALUs verwenden, die in der Graphikprozessor **1612**, dem Graphik-Mikrocontroller **1638**, den Geometrie- & und Festfunktions-Pipelines **1614** und **1636** oder einer anderen Logik in **Fig. 16B** verkörpert sind. Außerdem können in mindestens einer Ausführungsform die hier beschriebenen Inferenz- und/oder Trainingsoperationen mit einer anderen Logik als der in **Fig. 6B** oder **Fig. 6C** veranschaulichten Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtsparameter in On-Chip- oder Off-Chip-Speicher und/oder Registern (gezeigt oder nicht gezeigt) gespeichert werden, die ALUs des Graphikprozessors **1600C** konfigurieren, um einen oder mehrere Algorithmen maschinellen Lernens, neuronale Netzwerkarchitekturen, Anwendungsfälle oder die hier beschriebenen Trainingstechniken auszuführen.

[0255] **IGUREN 16A** und **16B** veranschaulichen eine Threadausführungslogik **1600D** einschließlich einer Anordnung von Verarbeitungselementen eines Graphikprozessorkerns gemäß mindestens einer Ausführungsform. **Fig. 16D** veranschaulicht mindestens eine Ausführungsform, bei welcher die Threadausführungslogik

1600D verwendet wird. **Fig. 16E** veranschaulicht interne Einzelheiten einer Ausführungseinheit gemäß mindestens einer Ausführungsform.

[0256] Wie in **Fig. 16D** veranschaulicht, umfasst die Thread-Ausführungslogik **1600D** in mindestens einer Ausführungsform einen Shader-Prozessor **1602**, einen Thread-Abfertiger **1604**, einen Anweisungs-Cache **1606**, eine Anordnung skalierbarer Ausführungseinheiten mit einer Mehrzahl von Ausführungseinheiten **1608A-1608N**, Abtaster bzw. Sampler **1610**, einen Daten-Cache **1612** und einen Datenport **1614**. In mindestens einer Ausführungsform kann eine Anordnung skalierbarer Ausführungseinheiten dynamisch skaliert werden, indem eine oder mehrere Ausführungseinheiten (z.B. eine der Ausführungseinheiten **1608A 1608B, 1608C bis 1608N-1** und **1608N**) 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 die Thread-Ausführungslogik **1600D** eine oder mehrere Verbindungen zu Speicher, wie beispielsweise Systemspeicher oder Cache-Speicher, über eine oder mehrere des Anweisungs-Caches **1606**, des Datenports **1614**, des Abtasters **1610** und der Ausführungseinheiten **1608A-1608N**. In mindestens einer Ausführungsform ist jede Ausführungseinheit (z.B. **1608A**) eine eigenständige, programmierbare, universelle 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 die Anordnung der Ausführungseinheiten **1608A-1608N** skalierbar, um eine beliebige Anzahl einzelner Ausführungseinheiten zu umfassen.

[0257] In mindestens einer Ausführungsform werden die Ausführungseinheiten **1608A-1608N** hauptsächlich zum Ausführen von Shader-Programmen verwendet. In mindestens einer Ausführungsform kann der Shader-Prozessor **1602** verschiedene Shader-Programme verarbeiten und Ausführungsthreads, die mit Shader-Programmen verknüpft sind, über einen Thread-Abfertiger **1604** versenden. In mindestens einer Ausführungsform umfasst der Thread-Abfertiger **1604** Logik, um Thread-Initiationsanforderungen von Graphik- und Medien-Pipelines zu arbitrieren und angeforderte Threads auf einer oder mehreren Ausführungseinheiten in den Ausführungseinheiten **1608A-1608N** zu instanziierten. In mindestens einer Ausführungsform kann eine Geometrie-Pipeline Vertex-, Tessellations- oder Geometrie-Shader an die Thread-Ausführungslogik zur Verarbeitung senden. In mindestens einer Ausführungsform kann der Thread-Abfertiger **1604** auch Laufzeit-Thread-Übergabe-Anforderungen von ausführenden Shader-Programmen verarbeiten.

[0258] In mindestens einer Ausführungsform unterstützen die Ausführungseinheiten **1608A-1608N** 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 **1608A-1608N**, 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 **1608A-1608N**, 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 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.

[0259] In mindestens einer Ausführungsform arbeitet jede Ausführungseinheit in den Ausführungseinheiten **1608A-1608N** an Anordnungen bzw. Arrays von Datenelementen. In mindestens einer Ausführungsform ist eine Anzahl von Datenelementen von „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 (ALUs) oder Gleitkommaeinheiten (FPUs) für einen bestimmten Graphikprozessor sein. In mindestens einer Ausführungsform unterstützen die Ausführungseinheiten **1608A-1608N** Ganzzahl- und Gleitkomma-Datentypen.

[0260] 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. In mindestens einer Ausführungsform werden, 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.

[0261] In mindestens einer Ausführungsform können eine oder mehrere Ausführungseinheiten zu einer fusionierten Ausführungseinheit **1609A-1609N** mit Thread-Steuerlogik (**1607B-1607N**) kombiniert sein, die für fusionierte Ausführungseinheiten (Execution Units; EUs) üblich 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, einen separaten SIMD-Hardware-Thread auszuführen. Die Anzahl von EUs in einer fusionierten EU-Gruppe kann gemäß verschiedenen 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 **1609A-1609N** mindestens zwei Ausführungseinheiten. In mindestens einer Ausführungsform umfasst die fusionierte Ausführungseinheit **1609A** eine erste EU **1608A**, eine zweite EU **1608B** und eine Thread-Steuerlogik **1607B**, die der ersten EU **1608A** und der zweiten EU **1608B** gemeinsam ist. In mindestens einer Ausführungsform steuert die Thread-Steuerlogik **1607A** Threads, die auf der fusionierten Graphikausführungseinheit **1609A** ausgeführt werden, welches jeder EU innerhalb der fusionierten Ausführungseinheiten **1609A-1609N** erlaubt, unter Verwendung eines gemeinsamen Anweisungszeigerregisters ausgeführt zu werden.

[0262] In mindestens einer Ausführungsform sind ein oder mehrere interne Anweisungs-Caches (z.B. **1606**) in der Thread-Ausführungslogik **1600D** umfasst, um Thread-Anweisungen für Ausführungseinheiten zwischenzuspeichern. In mindestens einer Ausführungsform sind ein oder mehrere Daten-Caches (z.B. **1612**) umfasst, um Thread-Daten während der Thread-Ausführung zwischenzuspeichern. In mindestens einer Ausführungsform ist ein Abtaster **1610** umfasst, um eine Texturabtastung für 3D-Operationen und eine Medienabtastung für Medienoperationen bereitzustellen. In mindestens einer Ausführungsform umfasst der Abtaster **1610** eine spezialisierte Textur- oder Medien-Abtastfunktionalität, um Textur- oder Mediendaten während des Abtastprozesses zu verarbeiten, bevor abgetastete Daten an eine Ausführungseinheit bereitgestellt werden.

[0263] Während der Ausführung senden in mindestens einer Ausführungsform Graphik- und Medien-Pipelines Thread-Initiationsanforderungen über Thread-Übergabe- und Sende-Logik an die Thread-Ausführungslogik **1600D**. 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 **1602** 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 **1602** dann ein von der Anwendungsprogrammierschnittstelle (API) bereitgestelltes Pixel- oder Fragment-Shader-Programm aus. In mindestens einer Ausführungsform sendet, um ein Shader-Programm auszuführen, der Shader-Prozessor **1602** Threads über den Thread-Abfertiger **1604** an eine Ausführungseinheit (z.B. **1608A**). In mindestens einer Ausführungsform verwendet der Shader-Prozessor **1602** Texturabtastlogik in dem Abtaster **1610**, 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 verwerfen ein oder mehrere Pixel aus der weiteren Verarbeitung.

[0264] In mindestens einer Ausführungsform stellt ein Datenport **1614** einen Speicherzugriffsmechanismus für die Thread-Ausführungslogik **1600D** bereit, um verarbeitete Daten zur weiteren Verarbeitung auf einer Gra-

phikprozessor-Ausgabepipeline an den Speicher auszugeben. In mindestens einer Ausführungsform umfasst oder koppelt der Datenport **1614** einen oder mehrere Cache-Speicher (z.B. Daten-Cache **1612**), um Daten für einen Speicherzugriff über einen Datenport zwischenspeichern.

[0265] Wie in **Fig. 16E** veranschaulicht, kann in mindestens einer Ausführungsform eine Graphikausführungseinheit **1608** eine Befehlsabrufeinheit **1637**, eine allgemeine Registerdateianordnung (GRF) **1624**, eine architektonische Registerdateianordnung (ARF) **1626**, einen Thread-Arbiter **1622**, eine Sendeeinheit **1630**, eine Verzweigungseinheit **1632**, einen Satz von SIMD-Gleitkommaeinheiten (FPU(s)) **1634** und in mindestens einer Ausführungsform einen Satz von dedizierten ganzzahligen SIMD ALUs **1635** umfassen. In mindestens einer Ausführungsform umfassen die GRF **1624** und die ARF **1626** einen Satz von allgemeinen Registerdateien und Architekturregisterdateien, die jedem gleichzeitigen Hardware-Thread zugeordnet sind, der in der Graphikausführungseinheit **1608** aktiv sein kann. In mindestens einer Ausführungsform wird der Architekturzustand pro Thread in der ARF **1626** beibehalten, während Daten, die während der Thread-Ausführung verwendet werden, in der GRF **1624** gespeichert werden. In mindestens einer Ausführungsform kann der Ausführungszustand jedes Threads, einschließlich von Befehlszeigern für jeden Thread, in threadspezifischen Registern in der ARF **1626** gehalten werden.

[0266] In mindestens einer Ausführungsform weist die Graphikausführungseinheit **1608** eine Architektur auf, die eine Kombination aus simultanem Multi-Threading (SMT) und feinkörnigem Interleaved Multi-Threading (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.

[0267] In mindestens einer Ausführungsform kann die Graphikausführungseinheit **1608** mehrere Anweisungen gemeinsam ausgeben, welche jeweils unterschiedliche Anweisungen sein können. In mindestens einer Ausführungsform kann der Thread-Arbiter **1622** der Graphikausführungseinheit-Threads **1608** Anweisungen an eine der Sendeeinheit **1630**, der Verzweigungseinheit **1642** oder der SIMD FPU(s) **1634** zur Ausführung senden. In mindestens einer Ausführungsform kann jeder Ausführungsthread auf 128 Universalregister innerhalb der GRF **1624** 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 KBytes innerhalb der GRF **1624** 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 eine Anzahl von Threads pro Ausführungseinheit je nach Ausführungsform variieren kann. In mindestens einer Ausführungsform, in welcher sieben Threads auf 4 Kbytes zugreifen können, kann die GRF **1624** insgesamt 28 Kbytes speichern. In mindestens einer Ausführungsform können flexible Adressierungsmodi Registern ermöglichen, gemeinsam adressiert zu werden, um breitere Register wirksam aufzubauen oder schrittförmige rechteckige Blockdatenstrukturen darzustellen.

[0268] In mindestens einer Ausführungsform werden Speicheroperationen, Abtasteroperationen und andere Systemkommunikationen mit längerer Latenz über „Sende“-Anweisungen gesendet, die durch eine Nachrichtenübergabe-Sendeeinheit **1630** ausgeführt werden. In mindestens einer Ausführungsform werden Verzweigungsanweisungen an eine dedizierte Verzweigungseinheit **1632** gesendet, um SIMD-Divergenz und eventuelle Konvergenz zu erleichtern.

[0269] In mindestens einer Ausführungsform umfasst die Ausführungseinheit **1608** eine oder mehrere SIMD-Gleitkommaeinheiten (FPU(s)) **1634** zum Durchführen von Gleitkommaoperationen. In mindestens einer Ausführungsform unterstützen die FPU(s) **1634** auch Ganzzahlberechnungen. In mindestens einer Ausführungsform können die FPU(s) **1634** 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 der FPU(s) erweiterte mathematische Fähigkeiten zur Unterstützung von transzendentalen mathematischen Funktionen und doppeltgenauen 64-Bit-Gleitkomma mit hohem Durchsatz bereit. In mindestens einer Ausführungsform ist auch ein Satz von 8-Bit Ganzzahl SIMD-ALUs **1635** vorhanden, und kann speziell optimiert sein, um Operationen im Zusammenhang mit Berechnungen zum maschinellen Lernen durchzuführen.

[0270] In mindestens einer Ausführungsform können Arrays von mehreren Instanzen der Graphikausführungseinheit **1608** in einer Graphik-Unterkerngruppierung (z.B. einer Sub-Slice) instanziiert sein. In mindestens einer Ausführungsform kann die Ausführungseinheit **1608** Anweisungen über eine Mehrzahl von Ausführungs-

kanälen hinweg ausführen. In mindestens einer Ausführungsform wird jeder auf der Graphikausführungseinheit **1608** ausgeführte Thread auf einem unterschiedlichen Kanal ausgeführt.

[0271] Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten zu der Inferenz- und/oder Trainingslogik **615** werden nachstehend in Verbindung mit **Fig. 6B** und/oder **6C** bereitgestellt. In mindestens einer Ausführungsform können Abschnitt der oder die gesamte Inferenz- und/oder Trainingslogik **615** in die Ausführungslogik **1600D** integriert sein. Darüber hinaus können in mindestens einer Ausführungsform hier beschriebene Inferenz- und/oder Trainingsoperationen mit einer anderen Logik als der in **Fig. 6B** oder **Fig. 6C** veranschaulichten Logik durchgeführt werden. In mindestens einer Ausführungsform können Gewichtsparameter in On-Chip- oder Off-Chip-Speicher und/oder Registern (gezeigt oder nicht gezeigt) gespeichert werden, die ALUs der Ausführungslogik **1600D** konfigurieren, um einen oder mehrere Algorithmen maschinellen Lernens, Lernalgorithmen, neuronale Netzwerkarchitekturen, Anwendungsfälle oder die hier beschriebenen Trainingstechniken auszuführen.

[0272] **Fig. 17A** veranschaulicht eine Parallelverarbeitungseinheit („PPU“) **1700A** gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist die PPU **1700A** mit einem maschinenlesbarem Code konfiguriert, der, falls von der PPU **1700A** ausgeführt, die PPU **1700A** veranlasst, einige oder sämtliche der Prozesse und Techniken durchzuführen, die in dieser Offenbarung durchweg beschrieben sind. In mindestens einer Ausführungsform ist die PPU **1700A** 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 parallel 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 **1700A** ausgeführt zu werden. In mindestens einer Ausführungsform ist die PPU **1700A** 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 **1700A** benutzt, um Rechnungen, wie beispielsweise Operationen der linearen Algebra und Operationen des Maschinenlernens durchzuführen. **Fig. 17A** veranschaulicht einen beispielhaften parallelen Prozessor 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.

[0273] In mindestens einer Ausführungsform sind eine oder mehrere PPUs **1700A** konfiguriert, um HPC(High Performance Computing), Rechenzentrum und Maschinenlern-Anwendungen zu beschleunigen. In mindestens einer Ausführungsform ist die PPU **1700A** konfiguriert, um 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 und personalisierte Benutzerempfehlungen und dergleichen.

[0274] In mindestens einer Ausführungsform umfasst die PPU **1700A**, ohne darauf beschränkt zu sein, eine Eingabe/Ausgabe („E/A“-) Einheit **1706**, eine Frontend-Einheit **1710**, eine Planer-Einheit **1712**, eine Arbeitsverteilungs-Einheit **1714**, einen Hub **1716**, eine Kreuzschiene („XBar“) **1720**, einen oder mehrere allgemeine Verarbeitungscluster („GPCs“) **1718** und eine oder mehrere Partitions-Einheiten („Speicherpartitions-Einheiten“) **1722**. In mindestens einer Ausführungsform ist die PPU **1700A** mit einem Host-Prozessor oder anderen PPUs **1700A** über einen oder mehrere Hochgeschwindigkeits-GPU-Zwischenverbindungen („GPU-Zwischenverbindungen“) **1708** verbunden.

[0275] In mindestens einer Ausführungsform ist die PPU **1700A** mit einem Host-Prozessor oder anderen peripheren Vorrichtungen über eine Zwischenverbindung **1702** verbunden. In mindestens einer Ausführungsform ist die PPU **1700A** mit einem lokalen Speicher verbunden, der eine oder mehrere Speichervorrichtungen („Speicher“) **1704** umfasst. In mindestens einer Ausführungsform umfassen die Speichervorrichtungen **1704**, 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.

[0276] In mindestens einer Ausführungsform kann sich die Hochgeschwindigkeits-GPU-Zwischenverbindung **1708** auf eine drahtgebundene mehrspurige Kommunikationsverbindung beziehen, die von Systemen verwendet wird, um eine oder mehrere PPU's **1700A** zu skalieren und zu umfassen, die mit einer oder mehreren zentralen Verarbeitungseinheiten („CPUs“) kombiniert sind, unterstützt Cache-Kohärenz zwischen den PPU's **1700A** und CPUs sowie CPU-Mastering. In mindestens einer Ausführungsform werden Daten und/oder Befehle mittels der Hochgeschwindigkeits-GPU-Zwischenverbindung **1708** durch den Hub **1716** an/von anderen Einheiten der PPU **1700A** übertragen, wie beispielsweise eine oder mehrere Kopier-Engines, Videocodierer, Videodecodierer, Leistungsverwaltungseinheiten und andere Komponenten, die in **Fig. 17A** nicht explizit veranschaulicht werden können.

[0277] In mindestens einer Ausführungsform ist die E/A-Einheit **1706** konfiguriert, um Kommunikationen (d.h. Befehle, Daten usw.) von einem Host-Prozessor (in **Fig. 17A** nicht gezeigt) über den Systembus **1702** zu übertragen und zu empfangen. In mindestens einer Ausführungsform kommuniziert die E/A-Einheit **1706** direkt mit dem Host-Prozessor über den Systembus **1702** oder durch eine oder mehrere Zwischenvorrichtungen, wie beispielsweise eine Speicherbrücke. In mindestens einer Ausführungsform kann die E/A-Einheit **1706** mit einem oder mehreren anderen Prozessoren, wie beispielsweise eine oder mehrere PPU's, über den Systembus **1702** kommunizieren. In mindestens einer Ausführungsform implementiert die E/A-Einheit **1706** eine Peripheral Component Interconnect Express („PCIe“)-Schnittstelle für Kommunikationen über einen PCIe-Bus und der Systembus **1702** ist ein PCIe-Bus. In mindestens einer Ausführungsform implementiert die E/A-Einheit **1706** Schnittstellen zum Kommunizieren mit externen Vorrichtungen.

[0278] In mindestens einer Ausführungsform decodiert die E/A-Einheit **1706** Pakete, die über den Systembus **1702** empfangen wurden. In mindestens einer Ausführungsform stellen mindestens einige Pakete Befehle dar, die konfiguriert sind, um die PPU **1700A** zu veranlassen, verschiedene Operationen durchzuführen. In mindestens einer Ausführungsform überträgt die E/A-Einheit **1706** decodierte Befehle an verschiedene andere Einheiten der PPU **1700A**, wie durch Befehle spezifiziert. In mindestens einer Ausführungsform werden Befehle an die Frontend-Einheit **1710** und/oder an den Hub **1716** oder andere Einheiten der PPU **1700A** übertragen, wie beispielsweise eine oder mehrere Kopier-Engines, einen Video-Codierer, einen Video-Decodierer, eine Leistungsverwaltungseinheit usw. (nicht explizit in **Fig. 17A** gezeigt). In mindestens einer Ausführungsform ist die E/A-Einheit **1706** konfiguriert, um Kommunikationen zwischen und unter verschiedenen logischen Einheiten der PPU **1700A** weiterzuleiten.

[0279] In mindestens einer Ausführungsform codiert ein von dem Host-Prozessor ausgeführtes Programm einen Befehlsstrom in einem Puffer, welcher der PPU **1700A** 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 **1700A** zugänglich ist (d.h. Lesen/Schreiben) - eine Host-Schnittstelleneinheit kann konfiguriert sein, um auf diesen Puffer in einem Systemspeicher, der mit dem Systembus **1702** verbunden ist, über Speicheranforderungen zuzugreifen, die über den Systembus **1702** durch die E/A-Einheit **1706** übertragen werden. In mindestens einer Ausführungsform schreibt ein Host-Prozessor einen Befehlsstrom in einen Puffer und überträgt dann einen Zeiger zu einem Start des Befehlsstroms an die PPU **1700A**, so dass die Frontend-Einheit **1710** Zeiger zu einem oder mehreren Befehlsströme empfängt und einen oder mehrere Ströme verwaltet, Befehle aus den Befehlsströmen liest und Befehle an verschiedene Einheiten der PPU **1700A** weiterleitet.

[0280] In mindestens einer Ausführungsform ist die Frontend-Einheit **1710** mit einer Planer-Einheit **1712** gekoppelt, die verschiedene GPCs **1718** konfiguriert, um Aufgaben zu verarbeiten, die durch einen oder mehrere Befehlsströme definiert sind. In mindestens einer Ausführungsform ist die Planer-Einheit **1712** konfiguriert, um Zustandsinformation zu verfolgen, die verschiedene Aufgaben betrifft, die von der Planer-Einheit **1712** verwaltet werden, wobei die Zustandsinformation angeben kann, welchem der GPCs **1718** 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 **1712** die Ausführung mehrerer Aufgaben auf einem oder mehreren der GPCs **1718**.

[0281] In mindestens einer Ausführungsform ist die Planer-Einheit **1712** mit einer Arbeitsverteilungs-Einheit **1714** gekoppelt, die konfiguriert ist, um Aufgaben zur Ausführung auf GPCs **1718** zu versenden. In mindestens einer Ausführungsform verfolgt die Arbeitsverteilungs-Einheit **1714** eine Anzahl von eingeplanten Aufgaben, die von der Planer-Einheit **1712** empfangen werden, und die Arbeitsverteilungs-Einheit **1714** verwaltet einen Pool für anstehende Aufgaben und einen Pool für aktive Aufgaben für jeden der GPCs **1718**. 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 **1718** verarbeitet zu werden; ein Pool für aktive Aufgaben kann eine Anzahl von Schlitzen (z.B. 4 Schlitze) für Aufgaben umfassen, die von den GPCs **1718** aktiv verarbeitet werden, so dass, wenn einer der GPCs **1718** die Ausführung einer Aufgabe abschließt, diese Aufgabe aus dem Pool für aktive Aufgaben für den GPC **1718** geräumt wird und aus dem Pool für anstehende Aufgaben ausgewählt und zur Ausführung auf dem GPC **1718** eingeplant wird. In mindestens einer Ausführungsform, wenn eine aktive Aufgabe auf dem GPC **1718** inaktiv ist, wie beispielsweise während darauf gewartet wird, dass eine Datenabhängigkeit behoben wird, dann wird die aktive Aufgabe aus dem GPC **1718** geräumt und zu dem Pool für anstehende Aufgaben zurückgeführt, während eine andere Aufgabe in diesem Pool für anstehende Aufgaben ausgewählt und zur Ausführung auf dem GPC **1718** eingeplant wird.

[0282] In mindestens einer Ausführungsform kommuniziert die Arbeitsverteilungs-Einheit **1714** mit einem oder mehreren GPCs **1718** über die Kreuzschiene bzw. XBar **1720**. In mindestens einer Ausführungsform ist die XBar **1720** ein Zwischenverbindung-Netzwerk, das viele der Einheiten der PPU **1700A** mit anderen Einheiten der PPU **1700A** koppelt und konfiguriert sein kann, um die Arbeitsverteilungs-Einheit **1714** mit einem bestimmten GPC **1718** zu koppeln. In mindestens einer Ausführungsform können eine oder mehrere andere Einheiten der PPU **1700A** ebenfalls mit der XBar **1720** über den Hub **1716** verbunden sein.

[0283] In mindestens einer Ausführungsform werden Aufgaben von der Planer-Einheit **1712** verwaltet und an einen der GPCs **1718** durch die Arbeitsverteilungs-Einheit **1714** abgefertigt. Der GPC **1718** ist konfiguriert, um die Aufgabe zu verarbeiten und Ergebnisse zu erzeugen. In mindestens einer Ausführungsform können Ergebnisse von anderen Aufgaben innerhalb des GPC **1718** konsumiert werden, an einen unterschiedlichen GPC **1718** über die XBar **1720** weitergeleitet oder im Speicher **1704** gespeichert werden. In mindestens einer Ausführungsform können Ergebnisse in den Speicher **1704** über Partitions-Einheiten **1722** geschrieben werden, die eine Speicherschnittstelle zum Lesen und Schreiben von Daten in/aus dem Speicher **1704** implementieren. In mindestens einer Ausführungsform können Ergebnisse an eine andere PPU **1704** oder CPU über die Hochgeschwindigkeit-GPU-Zwischenverbindung **1708** übertragen werden. In mindestens einer Ausführungsform umfasst die PPU, ohne darauf beschränkt zu sein, eine Anzahl U von Speicherpartitions-Einheiten **1722**, die gleich einer Anzahl von getrennten und unterschiedlichen Speichervorrichtungen **1704** ist, die mit der PPU **1700A** gekoppelt sind, wie nachstehend hier ausführlicher in Verbindung mit **Fig. 17C** beschrieben.

[0284] 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 **1700A** einzuplanen. In mindestens einer Ausführungsform werden mehrere Rechenanwendungen gleichzeitig von der PPU **1700A** ausgeführt und die PPU **1700A** 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 **1700A** zu erzeugen, und der Treiberkernel gibt Aufgaben an einen oder mehrere Streams aus, die von der PPU **1700A** 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 die 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. 17C** beschrieben.

[0285] Eine Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenz- und/oder Trainingslogik **615** werden nachstehend in Verbindung mit **Fig. 6B** und/oder **6C** 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 **1700A** bereitgestellt werden. In mindestens einer Ausführungsform wird die PPU **1700A** 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 die PPU **1700A** trainiert wurde. In mindestens einer Ausführungsform kann die PPU **1700A** verwendet werden, um ein oder mehrere hier beschriebene Anwendungsfälle eines neuronalen Netzwerks durchzuführen.

[0286] Fig. 17B veranschaulicht einen allgemeinen Verarbeitungscluster („GPC“) 1700B gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist der GPC 1700B der GPC 1718 von Fig. 17A. In mindestens einer Ausführungsform umfasst jeder GPC 1700, ohne darauf beschränkt zu sein, eine Anzahl von Hardwareeinheiten zur Verarbeitung von Aufgaben, und jeder GPC 1700B umfasst, ohne darauf beschränkt zu sein, einen Pipeline-Manager 1702, eine Vor-Raster-Operationen-Einheit („PROP“) 1704, eine Raster-Engine 1708, eine Arbeitsverteilungs-Kreuzschiene („WDX“) 1716, eine Speicherverwaltungseinheit („MMU“) 1718 und einen oder mehrere Datenverarbeitungscluster („DPCs“) 1706 und jede geeignete Kombination von Teilen.

[0287] In mindestens einer Ausführungsform wird der Betrieb des GPC 1700B durch den Pipeline-Manager 1702 gesteuert. In mindestens einer Ausführungsform verwaltet der Pipeline-Manager 1702 die Konfiguration eines oder mehrerer DPCs 1706 zur Verarbeitung von Aufgaben, die dem GPC 1700B zugeteilt sind. In mindestens einer Ausführungsform konfiguriert der Pipeline-Manager 1702 mindestens einen von einem oder mehreren DPCs 1706, um mindestens einen Abschnitt einer Graphik-Rendering-Pipeline zu implementieren. In mindestens einer Ausführungsform ist ein DPC 1706 konfiguriert, um ein Vertex-Shader-Programm auf einem programmierbaren Streaming-Multiprozessor („SM“) 1714 auszuführen. In mindestens einer Ausführungsform ist der Pipeline-Manager 1702 konfiguriert, um Pakete, die von einer Arbeitsverteilungs-Einheit empfangen werden, in mindestens einer Ausführungsform an geeignete logische Einheiten innerhalb des GPC 1700B weiterzuleiten, und einige Pakete können an Festfunktions-Hardwareeinheiten in dem PROP 1704 und/oder der Raster-Engine 1708 weitergeleitet werden, während andere Pakete an DPCs 1706 zur Verarbeitung durch eine Primitiven-Engine 1712 oder den SM 1714 weitergeleitet werden können. In mindestens einer Ausführungsform konfiguriert der Pipeline-Manager 1702 mindestens einen der DPCs, um ein neuronales Netzwerkmodell und/oder eine Rechen-Pipeline zu implementieren.

[0288] In mindestens einer Ausführungsform ist die PROP-Einheit 1704 in mindestens einer Ausführungsform konfiguriert, um Daten, die von der Raster-Engine 1708 und den DPCs 1706 erzeugt wurden, an eine Raster-Operationen („ROP“) -Einheit in der Partitions-Einheit weiterzuleiten, die oben ausführlicher in Verbindung mit Fig. 17A beschrieben ist. In mindestens einer Ausführungsform ist die PROP-Einheit 1704 konfiguriert, um Optimierungen zur Farbenmischung durchzuführen, Pixeldaten zu organisieren, Adressenübersetzungen und mehr durchzuführen. In mindestens einer Ausführungsform umfasst die Raster-Engine 1708, 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 1708 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 Fragmente, die außerhalb eines Betrachtungstumpfes liegen, abgeschnitten werden. In mindestens einer Ausführungsform werden diejenigen Fragmente, welche die Abschneidung und Aussonderung überleben, an die Feinraster-Engine weitergeben, um Attribute für Pixelfragmente basierend auf den Ebenengleichungen zu erzeugen, die durch die Setup-Engine erzeugt werden. In mindestens einer Ausführungsform umfasst eine Ausgabe der Raster-Engine 1708 Fragmente, die durch eine geeignete Entität zu verarbeiten sind, wie beispielsweise durch einem Fragment-Shader, der innerhalb eines DPC 1706 implementiert ist.

[0289] In mindestens einer Ausführungsform umfasst jeder in dem GPC 1700B enthaltene DPC 1706, ohne darauf beschränkt zu sein, einen M-Pipe-Controller („MPC“) 1710; eine Primitiven-Engine 1712; einen oder mehrere SMs 1714; und jede geeignete Kombination davon. In mindestens einer Ausführungsform steuert der MPC 1710 den Betrieb des DPC 1706, wobei von dem Pipeline-Manager 1702 empfangene Pakete an geeignete Einheiten im DPC 1706 weitergeleitet werden. In mindestens einer Ausführungsform werden Pakete, die einer Vertex zugeordnet sind, an die Primitiven-Engine 1712 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 1714 übertragen werden.

[0290] In mindestens einer Ausführungsform umfasst der SM 1714, 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 1714 multi-threaded (umfasst mehrere Threads) und ist konfiguriert, um eine Mehrzahl von Threads (z.B. 32 Threads) von einer be-

stimmten Gruppe von Threads nebenläufig auszuführen und implementiert eine SIMD(Einzeln-Befehl, 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 der Gruppe von Threads gleiche Anweisungen aus. In mindestens einer Ausführungsform implementiert der SM **1714** eine Einzelner-Befehl, Mehrere-Thread („SIMT“)-Architektur, wobei jeder Thread in einer Gruppe von Threads konfiguriert ist, um einen unterschiedlichen Satz von Daten basierend auf dem 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 **1714** wird nachstehend ausführlicher beschrieben.

[0291] In mindestens einer Ausführungsform stellt die MMU **1718** eine Schnittstelle zwischen dem GPC **1700B** und der Partitions-Einheit (z.B. Partitions-Einheit **1722** von **Fig. 17A**) bereit und die MMU **1718** stellt eine Übersetzung von virtuellen Adressen in physische Adressen, einen Speicherschutz und eine Arbitrierung von Speicheranforderungen bereit. In mindestens einer Ausführungsform stellt die MMU **1718** einen oder mehrere Adressenübersetzungspuffer (Translation Lookaside Buffer; „TLBs“) zum Durchführen einer Übersetzung von virtuellen Adressen in physische Adressen im Speicher bereit.

[0292] Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenz- und/oder Trainingslogik **615** werden nachstehend in Verbindung mit **Fig. 6B** und/oder **6C** 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 **1700B** bereitgestellt werden. In mindestens einer Ausführungsform wird der GPC **1700B** 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 **1700B** trainiert wurde. In mindestens einer Ausführungsform kann die PPU **1700B** verwendet werden, um ein oder mehrere hier beschriebene Anwendungsfälle eines neuronalen Netzwerks durchzuführen.

[0293] **Fig. 17C** veranschaulicht eine Speicherpartitions-Einheit **1700C** eine Parallelverarbeitungseinheit („PPU“) gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform umfasst die Partitions-Einheit **1700C** eine Raster-Operationen („ROP“)-Einheit **1702**, einen Level-2 („L2“)-Cache **1704**, eine Speicherschnittstelle **1706** und jede geeignete Kombination davon. In mindestens einer Ausführungsform ist die Speicherschnittstelle **1706** mit dem Speicher gekoppelt. In mindestens einer Ausführungsform kann die Speicherschnittstelle **1706** 32-, 64-, 128-, 1024-Bit-Datenbusse oder ähnliche Implementierungen für einen Hochgeschwindigkeits-Datentransfer implementieren. In mindestens einer Ausführungsform nimmt die PPU Speicherschnittstellen **1706** auf, eine Speicherschnittstelle **1706** pro Paar von Speicherpartitions-Einheiten **1700C**, wobei jedes Paar von Speicherpartitions-Einheiten **1700C** mit einer entsprechenden Speichervorrichtung verbunden ist. In mindestens einer Ausführungsform kann die PPU mit bis zu Y Speichervorrichtungen, wie beispielsweise Speicherstapel mit hoher Bandbreite oder Graphikdoppeldatenraten, Version 5, synchronen dynamischen Direktzugriffsspeicher („GDDR5 SDRAM“) verbunden sein.

[0294] In mindestens einer Ausführungsform implementiert die Speicherschnittstelle **1706** 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 derselben physischen Packung wie die PPU lokalisiert, die wesentliche Leistungs- und Flächeneinsparungen im Vergleich mit GDDR5 SDRAM Systemen bereitstellt. In mindestens einer Ausführungsform umfasst jeder HBM2-Stapel, ohne darauf beschränkt zu sein, vier Speicher-Dies und 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 („SECCED“), um Daten zu schützen. In mindestens einer Ausführungsform stellt ECC eine höhere Zuverlässigkeit für Rechenanwendungen bereit, die gegen Datenverfälschung empfindlich sind.

[0295] In mindestens einer Ausführungsform implementiert die PPU eine Mehrebenen-Speicherhierarchie. In mindestens einer Ausführungsform unterstützt die Speicherpartitions-Einheit **1700C** 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 **1708** Adressenübersetzungsdienste, die der PPU ermöglichen, auf Seitentabellen einer CPU direkt zuzugreifen und einen vollen Zugriff auf den CPU-Speicher durch die PPU bereitzustellen.

[0296] 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 **1700C** 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 zwischen mehreren Prozessoren fixiert (d.h., 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.

[0297] Daten aus dem Speicher **1704** von **Fig. 17A** oder einem anderen Systemspeicher werden von der Speicherpartitions-Einheit **1700C** abgerufen und in dem L2-Cache-Speicher **1704** gespeichert, der On-Chip lokalisiert ist und zwischen verschiedenen GPCs gemäß mindestens einer Ausführungsform gemeinsam benutzt wird. Jede Speicherpartitions-Einheit **1700C** umfasst in mindestens einer Ausführungsform einen Bereich des L2-Cache-Speichers **1704**, der einer entsprechenden Speichervorrichtung zugeordnet ist. In mindestens einer Ausführungsform können Cache-Speicher niedrigerer Ebene dann in verschiedenen Einheiten innerhalb der GPCs implementiert sein.

[0298] In mindestens einer Ausführungsform kann jeder der SMs **1714** einen Level-1 („L1“)-Cache-Speicher implementieren, wobei der L1-Cache-Speicher ein privater Speicher ist, der einem bestimmten SM **1714** fest zugeordnet ist, und Daten von dem L2-Cache-Speicher **1704** werden abgerufen und in jedem L1-Cache-Speicher zur Verarbeitung in Funktionseinheiten der SMs **1714** gespeichert. In mindestens einer Ausführungsform ist der L2-Cache-Speicher **1704** mit der Speicherschnittstelle **1706** und dem XBar **1720** gekoppelt.

[0299] Die ROP-Einheit **1702** führt Graphik-Raster-Operationen, welche die Pixelfarbe betreffen, wie beispielsweise Farbenkomprimierung, Pixelmischung und mehr, in mindestens einer Ausführungsform durch. Die ROP-Einheit **1702** implementiert in mindestens einer Ausführungsform ebenfalls Tiefentesten in Verbindung mit der Raster-Engine **1708**, wobei eine Tiefe für einen Abtastort, der einem Pixelfragment zugeordnet ist, von einer Aussonderungs-Engine der Raster-Engine **1708** 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 das Fragment den Tiefentest für den Abtastort besteht, dann aktualisiert die ROP-Einheit **1702** den Tiefenpuffer und überträgt ein Ergebnis dieses Tiefentests an die Raster-Engine **1708**. Es wird anerkannt, dass sich die Anzahl von Speicherpartitions-Einheiten **1700C** von der Anzahl von GPCs unterscheiden kann, und daher kann in mindestens einer Ausführungsform jede ROP-Einheit **1702** mit jedem GPC gekoppelt werden. In mindestens einer Ausführungsform verfolgt die ROP-Einheit **1702** Pakete, die von unterschiedlichen GPCs empfangen werden, und bestimmt, welches ein durch die ROP-Einheit **1702** erzeugtes Ergebnis zu der Xbar **1720** durchgeleitet wird. **Fig. 17D** veranschaulicht einen Streaming-Multiprozessor („SM“) **1700D** gemäß mindestens einer Ausführungsform. In mindestens einer Ausführungsform ist der SM **1700D** der SM von **Fig. 17B**. In mindestens einer Ausführungsform umfasst der SM **1700D**, ohne darauf beschränkt zu sein, einen Anweisungs-Cache **1702**; eine oder mehrere Planer-Einheiten **1704**; eine Registerdatei **1708**, einen oder mehrere Verarbeitungskerne („cores“) **1710**, eine oder mehrere Spezialfunktionseinheiten („SFUs“) **1712**, eine oder mehrere Lade/Speicher-Einheiten („LSUs“) **1714**, ein Zwischenverbindung-Netzwerk **1716**, einen gemeinsam genutzten Speicher/Level-1 („L1“)-Cache-Speicher **1718** und jede geeignete Kombination davon.

[0300] 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 **1700D** zugeteilt. In mindestens einer Ausführungsform empfängt die Planer-Einheit **1704** Aufgaben von der Arbeitsverteilungs-Einheit und verwaltet die Anweisungsplanung für einen oder mehrere Thread-Blöcke, die dem SM **1700D** zugewiesen sind. In mindestens einer Ausführungsform plant die Planer-Einheit **1704** 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 **1704** 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 **1710**, SFUs **1712** und LSUs **1714**) während jedes Taktzyklus.

[0301] In mindestens einer Ausführungsform können sich 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 reicherem, 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 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 Cooperative Groups Programmierern, Gruppen von Threads explizit bei Sub-Block- (in mindestens einer Ausführungsform, 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. In mindestens einer Ausführungsform unterstützt das Programmiermodell 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 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.

[0302] In mindestens einer Ausführungsform ist eine Abfertigungseinheit **1706** konfiguriert, um Anweisungen an eine oder mehrere Funktionseinheiten zu übertragen, und die Planer-Einheit **1704** umfasst, ohne darauf beschränkt zu sein, zwei Abfertigungseinheiten **1706**, die ermöglichen, dass zwei unterschiedliche Anweisungen von dem gleichen Warp während jedes Taktzyklus abgefertigt werden. In mindestens einer Ausführungsform umfasst jede Planer-Einheit **1704** eine einzige Abfertigungseinheit **1706** oder zusätzliche Abfertigungseinheiten **1706**.

[0303] In mindestens einer Ausführungsform umfasst jeder SM **1700D**, ohne darauf beschränkt zu sein, eine Registerdatei **1708**, die einen Satz von Registern für die Funktionseinheiten des SM **1700D** bereitstellt. In mindestens einer Ausführungsform ist die Registerdatei **1708** jeweils zwischen Funktionseinheiten aufgeteilt, so dass jede Funktionseinheit einem zugehörigen Abschnitt der Registerdatei **1708** zugeteilt ist. In einer anderen Ausführungsform ist die Registerdatei **1708** zwischen den unterschiedlichen Warps aufgeteilt, die von dem SM **1700D** ausgeführt werden, und die Registerdatei **1708** stellt eine temporäre Speicherung für Operanden bereit, die mit Datenpfaden der Funktionseinheiten verbunden sind. In mindestens einer Ausführungsform umfasst jeder SM **1700D**, ohne darauf beschränkt zu sein, L Verarbeitungskerne **1710**. In mindestens einer Ausführungsform umfasst der SM **1700D**, ohne darauf beschränkt zu sein, eine große Anzahl (z.B., 128 oder mehr) von distinkten Verarbeitungskernen **1710**. In mindestens einer Ausführungsform umfasst jeder Verarbeitungskern **1710**, ohne darauf beschränkt zu sein, eine vollständig in einer Pipeline angeordnete (fullypipelined) 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 **1710**, ohne darauf beschränkt zu sein, 64 Einfach-Präzisions-(32-Bit)-Gleitkommakerne, 64 Ganzzahlkern, 32 Doppel-Präzisions-(64-Bit)-Gleitkommakerne und 8 Tensorkerne.

[0304] Tensorkerne sind konfiguriert, um Matrixoperationen gemäß mindestens einer Ausführungsform durchzuführen. In mindestens einer Ausführungsform sind ein oder mehrere Tensorkerne in Verarbeitungskernen **1710** enthalten. In mindestens einer Ausführungsform sind Tensorkerne konfiguriert, um Matrixarithmetik tiefen Lernens, wie beispielsweise Faltungsoperationen für neuronales Netzwerktraining und Inferenzieren, durch-

zuführen. In mindestens einer Ausführungsform arbeitet jeder Tensorkernel 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.

[0305] 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 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 ein Warp-Schnittstellenniveau an der CUDA-Ebene 16x16 große Matrizen an, die alle 32 Threads eines Warp überspannen.

[0306] In mindestens einer Ausführungsform umfasst jeder SM **1700D**, ohne darauf beschränkt zu sein, M SFUs **1712**, die Sonderfunktionen durchführen (z.B. Attributauswertung, reziproke Quadratwurzel usw.). In mindestens einer Ausführungsform können die SFUs **1712**, ohne darauf beschränkt zu sein, eine Baumtraversierungseinheit umfassen, die konfiguriert ist, um eine hierarchische Baumdatenstruktur zu durchlaufen. In mindestens einer Ausführungsform umfassen die SFUs **1712** 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 **1700D** ausgeführt werden. In mindestens einer Ausführungsform werden die Texturkarten in dem gemeinsam genutzten Speicher/L1-Cache-Speicher **1718** 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 **1700D**, ohne darauf beschränkt zu sein, zwei Textureinheiten.

[0307] Jeder SM **1700D** umfasst, ohne darauf beschränkt zu sein, N LSUs **1714**, die Lade- und Speicheroperationen zwischen dem gemeinsam genutzten Speicher/L1-Cache-Speicher **1718** und der Registerdatei **1708** in mindestens einer Ausführungsform implementieren. Jede SM **1700D** umfasst ein Zwischenverbindung-Netzwerk **1716**, das jede der Funktionseinheiten mit der Registerdatei **1708** und die LSU **1714** mit der Registerdatei **1708** und dem gemeinsam genutzten Speicher/L1-Cache-Speicher **1718** in mindestens einer Ausführungsform verbindet. In mindestens einer Ausführungsform ist das Zwischenverbindung-Netzwerk **1716** eine Kreuzschiene, die konfiguriert sein kann, um eine beliebige der Funktionseinheiten mit irgendeinem der Register in der Registerdatei **1708** zu verbinden und die LSUs **1714** mit der Registerdatei **1708** und Speicherorten in dem gemeinsam genutzten Speicher/L1-Cache-Speicher **1718** zu verbinden.

[0308] In mindestens einer Ausführungsform ist der gemeinsam benutzte Speicher/L1-Cache-Speicher **1718** eine On-Chip-Speicheranordnung, die in mindestens einer Ausführungsform Datenspeicherung und Kommunikation zwischen dem SM **1700D** und der Primitiven-Engine und zwischen Threads in dem SM **1700D** ermöglicht. In mindestens einer Ausführungsform umfasst der gemeinsam benutzte Speicher/L1-Cache-Speicher **1718** 128KB von Speicherkapazität und ist in dem Pfad von dem SM **1700D** zu der Partitions-Einheit. In mindestens einer Ausführungsform wird der gemeinsam benutzte Speicher/L1-Cache-Speicher **1718** 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 **1718**, dem L2-Cache-Speicher und dem Speicher Hintergrundspeicher.

[0309] 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 die Hälfte der Kapazität zu verwenden, und die Textur- und Lade-/Speicher-Operationen die verbleibende Kapazität verwenden können. Die Integration innerhalb des gemeinsam genutzten Speichers/L1-Caches **1718** ermöglicht dem gemeinsam genutzten Speicher/L1-Cache **1718** 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 von einer Arbeitsverteilungs-Einheit direkt DPCs in mindestens einer Ausführungsform zugewiesen und verteilt. In mindestens einer Ausführungsform führen Threads in einem Block dasselbe Programm aus, unter Verwendung einer eindeutigen Thread-ID in der Berechnung, um sicherzustellen, dass jeder Thread unter Verwendung des SM **1700D** eindeutige Ergebnisse erzeugt, um das Programm auszuführen und Berechnungen durchzuführen, eines gemeinsam genutzten Speicher/L1-Cache-Speichers **1718**, um zwischen Threads zu kommunizieren, und der LSU **1714**, um einen globalen Speicher durch den gemeinsam genutzten Speicher/L1-Cache-Speicher **1718** und die Speicherpartitions-Einheit zu lesen und zu beschreiben. In mindestens einer Ausführungsform schreibt, wenn für Allzweck-Parallelberechnung konfiguriert, der SM **1700D** Befehle, welche die Planer-Einheit **1704** verwenden kann, um neue Arbeit auf DPCs zu starten.

[0310] In mindestens einer Ausführungsform ist die PPU in einem enthalten 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 PPU's, Speicher, einem Rechner-mitreduziertem-Befehlssatz („RISC“)-CPU, einer Speicherverwaltungseinheit („MMU“), einem Digital/Analog-Wandler („DAC“) und dergleichen enthalten.

[0311] In mindestens einer Ausführungsform kann die 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 die PPU eine integrierte Graphikverarbeitungseinheit („GPU“) sein, die in einem Chipsatz der Hauptplatine enthalten ist.

[0312] Die Inferenz- und/oder Trainingslogik **615** wird verwendet, um Inferenz- und/oder Trainingsoperationen durchzuführen, die einer oder mehreren Ausführungsformen zugeordnet sind. Einzelheiten hinsichtlich der Inferenz- und/oder Trainingslogik **615** werden nachstehend in Verbindung mit **Fig. 6B** und/oder **6C** 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 **1700D** bereitgestellt werden. In mindestens einer Ausführungsform wird der SM **1700D** 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 SM **1700D** trainiert wurde. In mindestens einer Ausführungsform kann der SM **1700D** verwendet werden, um einen oder mehrere hier beschriebene Anwendungsfälle eines neuronalen Netzwerks durchzuführen.

[0313] In mindestens einer Ausführungsform kann sich eine einzige Halbleiterplattform auf eine alleinige unitäre Halbleiterbasierte integrierte Schaltung oder Chip beziehen. In mindestens einer Ausführungsform können multi-chip Module mit erhöhter Konnektivität verwendet werden, die on-chip Betrieb simulieren und erhebliche Verbesserungen gegenüber Benutzen einer Zentralverarbeitungseinheit (Central Processing Unit („CPU“) und Busimplementierung durchführen. In mindestens einer Ausführungsform können verschiedene Module ebenfalls getrennt oder in verschiedene Kombinationen von Halbleiterplattformen per Wünschen von Benutzer gelegen sein.

[0314] In mindestens einer Ausführungsform werden Computerprogramme in Form von maschinenlesbaren ausführbaren Code oder Computersteuerlogikalgorithmien im Hauptspeicher **4ee04** und/oder sekundären Speicherung gespeichert. Computerprogramme, wenn durch eine oder mehrere Prozessoren ausgeführt, ermöglichen dem System **4ee00**, verschiedene Funktionen gemäß mindestens einer Ausführungsform durchzuführen. In mindestens einer Ausführungsform sind der Speicher **4ee04**, die Speicherung und/oder jegliche andere Speicherung 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 ein Wechselspeicherlaufwerk, ein Diskettenlaufwerk, ein Magnetbandlaufwerk, ein Compact-Disk-Laufwerk, ein DVD (Digital Versatile Disk)-Laufwerk, eine Aufzeichnungsvorrichtung oder einen USB (Universal Serial Bus-Flash-Speicher usw. repräsentiert. In mindestens ei-

ner Ausführungsform werden die Architektur und/oder die Funktionalität von verschiedenen vorhergehenden Figuren im Kontext der CPU 4ee02; des Parallelverarbeitungssystems 4ee12; einer integrierten Schaltung, die mindestens einen Teil der Fähigkeiten sowohl der CPU 4ee02; des Parallelverarbeitungssystems 4ee12; eines Chipsatzes (z.B., eine Gruppe von integrierten Schaltungen, die als eine Einheit zur Durchführung verwandter Funktionen usw. entworfen und verkauft werden); und jeder geeigneten Kombination einer(von) integrierten Schaltung(en) implementiert.

[0315] In mindestens einer Ausführungsform wird die Architektur und/oder Funktionalität verschiedener vorherigen 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 4ee00 die Form eines Desktop-Computers, eines Laptop-Computers, eines Tablet-Computers, von Servern, von Supercomputern, eines Smartphones (z.B. eines drahtlosen, in der Hand haltbaren Geräts), eines persönlichen digitalen Assistenten („PDA“), einer Digitalkamera, eines Fahrzeugs, einer am Kopf angebrachten Anzeige bzw. eines Head-Mounted-Displays, eines in der Hand haltbaren elektronischen Geräts, eines Mobiltelefons, eines Fernsehers, einer Workstation, von Spielkonsolen, eines eingebetteten Systems und/oder jeder anderen geeigneten Art von Logik annehmen.

[0316] In mindestens einer Ausführungsform umfasst das Parallelverarbeitungssystem 4ee12, ohne darauf beschränkt zu sein, mehrere Parallelverarbeitungseinheiten („PPUs“) 4ee14 und zugeordnete Speicher 4ee16. In mindestens einer Ausführungsform sind die PPUs 4ee14 mit einem Host-Prozessor oder anderen peripheren Vorrichtungen via eine Zwischenverbindung 4ee18 und einen Schalter 4ee20 oder Multiplexer verbunden. In mindestens einer Ausführungsform verteilt das Parallelverarbeitungssystem 4ee12 Rechenaufgaben über PPUs 4ee14, die parallelisierbar sein können - beispielsweise als Teil der Verteilung von Rechenaufgaben über mehrere Graphikverarbeitungseinheit („GPU“) Thread-Blöcken. In mindestens einer Ausführungsform wird der Speicher gemeinsam genutzt und ist (z.B., für Lese- und/oder Schreibzugriff) über einige oder sämtliche PPUs 4ee14 zugänglich, obwohl ein derartig gemeinsam genutzter Speicher Leistungs Nachteile relativ zu der Verwendung von lokalen Speicher und Registern anfallen können, die sich in einem PPU 4ee14 befinden. In mindestens einer Ausführungsform wird der Betrieb von PPUs 4ee14 durch Verwendung eines Befehls synchronisiert, wie beispielsweise syncThreads(), wobei alle Threads in einem Block (z.B., über mehrere PPUs 4ee14 ausgeführt), um einen bestimmten Punkt der Ausführung des Code zu erreichen, bevor weitergegangen wird.

[0317] Andere Variationen liegen innerhalb des Wesens der vorliegenden Offenbarung. Somit werden, obwohl die offenbarten Techniken für verschiedene Modifikationen und alternative Konstruktionen empfänglich sind, bestimmte erläuterte Ausführungsformen davon in den Zeichnungen gezeigt und wurden vorstehend ausführlich beschrieben. Es sei jedoch zu verstehen, dass keine Absicht besteht, die Offenbarung auf die offenbarte spezielle Form oder Formen zu begrenzen, sondern die Absicht besteht im Gegenteil darin, alle Modifikationen, alternativen Konstruktionen und Äquivalente abzudecken, die in innerhalb des Wesens und des Umfangs der Offenbarung fallen, wie in den beigefügten Ansprüchen definiert.

[0318] Die Verwendung der Begriffe „einer/eine/eines“ und „der/die/das“ und ähnliche Referenzen im Kontext des Beschreibens der offenbarten Ausführungsformen (insbesondere im Kontext der folgenden Ansprüche) sind auszulegen, als sowohl den Singular als auch den Plural zu umfassen, solange nichts Anderweitiges hier angegeben oder durch den Kontext deutlich widersprochen wird, und nicht als eine Definition eines Begriffs. Die Begriffe „umfassend“, „aufweisend“, „einschließlich“ und „enthaltend“ sind als offene Begriffe auszulegen (das heißt bedeutend „umfassend, jedoch nicht beschränkt auf“), solange nicht anderweitig angemerkt. Der Begriff „verbunden“, wenn nicht modifiziert und sich auf physische Verbindungen beziehend, ist als teilweise oder vollständig innerhalb enthaltend, befestigt an oder zusammen verbunden auszulegen, sogar wenn es etwas Intervenierendes gibt. Die Nennung von Wertebereichen hier ist lediglich bestimmt, um als ein Kurzformverfahren des einzelnen Bezugnehmens auf jeden separaten Wert zu dienen, der in den Bereich fällt, es sei denn, dass hier etwas anderes angegeben ist, und jeder separate Wert wird in die Spezifikation aufgenommen, als ob er hier einzeln wiedergegeben wäre.

[0319] Die Verwendung des Begriffs „Satz“ oder „Menge“ (z. B. „eine Menge von Gegenständen“) oder „Teilmenge“ ist, sofern nicht anders angegeben oder durch Kontext widerlegt, als eine nicht leere Sammlung zu verstehen, die ein oder mehrere Elemente umfasst. Außerdem und sofern nicht anders vermerkt oder durch Kontext widerlegt, bezeichnet der Begriff „Teilmenge“ einer entsprechenden Menge nicht unbedingt eine echte Teilmenge der entsprechenden Menge, sondern Teilmenge und entsprechende Menge können gleich sein.

[0320] Verbindende Sprache, wie beispielsweise Ausdrücke der Form „mindestens eines von A, B, und C“, oder „mindestens eines von A, B und C“, es sei denn, dass es speziell anders angemerkt ist oder anderweitig eindeutig durch den Kontext widersprochen wird, ist andernfalls in dem Kontext zu verstehen, wie sie allgemein verwendet wird, um darzustellen, dass ein Objekt, ein Begriff usw. entweder A oder B oder C oder einen beliebigen nicht leeren Teilsatz des Satzes aus A und B und C sein kann. Beispielsweise beziehen sich im veranschaulichten Beispiel eines Satzes, der drei Elemente aufweist, die verbindenden Ausdrücke „mindestens eines von A, B, und C“ und „mindestens eines von A, B und C“ auf einen der folgenden Sätze: {A}, {B}, {C}, {A, B}, {A, C}, {B, C}, {A, B, C}. Somit kann eine derartige verbindende Sprache allgemein nicht bestimmt sein, zu implizieren, dass bestimmte Ausführungsformen verlangen, dass mindestens eines von A, mindestens eines von B, und mindestens eines von C jeweils vorhanden sind. Außerdem gibt, es sei denn, dass etwas anderes angemerkt ist oder durch den Kontext widersprochen wird, „eine Mehrzahl“ einen Zustand in der Mehrzahl an (z.B., „eine Mehrzahl von Elementen“ gibt mehrere Elemente an). Eine Mehrzahl ist mindestens zwei Elemente, kann jedoch mehr sein, wenn so entweder explizit oder durch den Kontext angegeben. Ferner bedeutet, es sei denn, dass etwas anderes angemerkt ist oder dies anderweitig aus dem Kontext offensichtlich ist, „basierend auf“ bedeutet „basierend mindestens teilweise auf“ und nicht „basierend allein auf“.

[0321] Vorgänge von hier beschriebenen Prozessen können in jeder geeigneten Reihenfolge durchgeführt werden, sofern hier nicht anderweitig angegeben oder der Kontext dem anderweitig eindeutig widerspricht. In mindestens einer Ausführungsform wird ein Prozess, wie beispielsweise jene Prozesse (oder Variationen und/oder Kombinationen davon), die hier beschrieben sind, unter der Steuerung von einem oder mehreren Computersystemen durchgeführt, die mit ausführbaren Anweisungen konfiguriert sind, und wird als Code (z.B., ausführbare Anweisungen, ein oder mehrere Computerprogramme oder ein oder mehrere Anwendungen) implementiert, der zusammen auf einem oder mehreren Prozessoren durch Hardware oder Kombinationen davon ausgeführt wird. In mindestens einer Ausführungsform wird ein Code auf einem computerlesbaren Speichermedium beispielsweise in Form eines Computerprogramms ausgeführt, das mehrere Anweisungen umfasst, die durch einen oder mehrere Prozessoren ausführbar sind. In mindestens einer Ausführungsform ist ein computerlesbares Speichermedium ein nicht flüchtiges computerlesbares Speichermedium, das flüchtige Signale ausschließt (z.B., eine propagierende transiente elektrische oder elektromagnetische Übertragung), jedoch nicht flüchtige Datenspeicherschaltungen (z.B., Puffer, Cache und Warteschlangen) innerhalb Transceivern von flüchtigen Signalen umfasst. In mindestens einer Ausführungsform ist der Code (z.B., ein ausführbarer Code oder Quellencode) auf einem Satz von einem oder mehreren nicht flüchtigen computerlesbaren Speichermedien gespeichert, die darauf gespeicherte ausführbare Anweisungen aufweisen (oder anderem Speicher, um ausführbare Anweisungen zu speichern) die, wenn durch einen oder mehreren Prozessoren eines Computersystems ausgeführt (d.h., als ein Ergebnis einer Ausführung) das Computersystem veranlassen, hier beschriebene Vorgänge durchzuführen. Ein Satz von nicht flüchtigen computerlesbaren Speicherungsmedien umfasst in mindestens einer Ausführungsform mehrere nicht flüchtige computerlesbare Speicherungsmedien und eines oder mehrere von einzelnen nicht flüchtigen Speicherungsmedien der mehreren nicht flüchtigen computerlesbaren Speichermedien. Während mehrere nicht flüchtige computerlesbare Speichermedien zusammen den gesamten Code speichern. In mindestens einer Ausführungsform werden ausführbare Anweisungen ausgeführt, so dass unterschiedliche Anweisungen durch unterschiedliche Prozessoren ausgeführt werden - beispielsweise speichert ein nicht flüchtiges computerlesbares Speichermedium Anweisungen und eine Haupt-Zentralverarbeitungseinheit („CPU“) führt einige Anweisungen aus, während eine Graphikverarbeitungseinheit („GPU“) andere Anweisungen ausführt. In mindestens einer Ausführungsform weisen unterschiedliche Komponenten eines Computersystems getrennte Prozessoren auf und unterschiedliche Prozessoren führen unterschiedliche Teilsätze von Anweisungen aus.

[0322] Dementsprechend sind in mindestens einer Ausführungsform Computersysteme konfiguriert, um ein oder mehrere Dienste zu implementieren, die einzeln oder gemeinsam Vorgänge von hier beschriebenen Prozessen durchführen, und derartige Computersysteme sind mit anwendbarer Hardware und/oder Software konfiguriert, welche die Durchführung von Vorgängen ermöglichen. Ferner ist ein Computersystem, das mindestens eine Ausführungsform der vorliegenden Offenbarung implementiert, eine einzelne Vorrichtung, und ist in einer anderen Ausführungsform ein verteiltes Computersystem, das mehrere Vorrichtungen umfasst, die unterschiedlich arbeiten, sodass das ein verteiltes Computersystem die hier beschriebenen Vorgänge durchführt und sodass eine einzelne Vorrichtung nicht alle Vorgänge durchführt.

[0323] Die Verwendung von einzelnen oder allen Beispielen oder einer hier beispielhaften bereitgestellten Formulierung (z.B., „wie beispielsweise“) ist bestimmt, lediglich Ausführungsformen der Offenbarung besser zu beleuchten und stellt keine Einschränkung des Umfangs der Offenbarung dar, es sei denn, dass etwas anderes beansprucht ist. Keine Sprache in der Spezifikation sollte so ausgelegt werden, dass sie ein beliebiges nicht beanspruchtes Element als wesentlich angibt, um die Offenbarung zu praktizieren.

[0324] 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.

[0325] 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.

[0326] Sofern nicht speziell anders angegeben, wird anerkannt, dass über die gesamte Spezifikation Bezugnahmen auf „Verarbeiten“, „Rechnen“, „Berechnen“, „Bestimmen“ oder dergleichen sich auf eine Aktion und/oder Verfahren eines Computers oder Rechensystems beziehen, oder einer ähnlichen elektronischen Vorrichtung, die Daten manipulieren und/oder Daten transformieren, 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.

[0327] Auf eine ähnlichen Art und Weise kann sich „ein 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. Begriffe „System“ und „Verfahren“ werden hier austauschbar verwendet, insofern als System ein oder mehrere Verfahren verkörpern und Verfahren als ein System betrachtet werden können.

[0328] 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. Erhalten, Erfassen, Empfangen oder Eingeben von analogen und digitalen Daten kann in einer Vielzahl 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, Empfangens oder Eingebens von analogen oder digitalen Daten durch Transferieren von Daten über eine serielle oder parallele Schnittstelle erreicht werden. In einer anderen Implementierung 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 Funktionsaufrufs, eines Parameters einer Anwendungsprogrammierschnittstelle oder eines Mechanismus zur Interprozesskommunikation erreicht werden.

[0329] Obwohl die obige Erläuterung beispielhafte Implementierungen von beschriebenen Techniken darlegt, 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 Zwecke der Erläuterung definiert sind, verschiedene Funktionen und Verantwortlichkeiten in unterschiedlichen Weisen abhängig von den Umständen verteilt und aufgeteilt werden.

[0330] Des Weiteren sei zu verstehen, obwohl der Gegenstand in einer für strukturelle Merkmale und/oder methodische Handlungen spezifischen Sprache beschrieben wurde, dass der in den beigefügten Ansprüchen 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.

Patentansprüche**1. Mobiles Rechenzentrumkühlsystem, umfassend:**

mindestens einen Behälter, der einen Behälterverteiler umfasst, um Kühlmittel, das einem Kühlturm zugeordnet ist, der an oder benachbart zu dem mindestens einen Behälter angebracht ist, an ein oder mehrere flüssigkeitsgekühlte Gestelle innerhalb des mindestens einen Behälters zu zirkulieren und um eine Fluidkopplung des Behälterverteilers mit einem zweiten Behälterverteiler eines zweiten Behälters zu ermöglichen.

2. Mobiles Rechenzentrumkühlsystem gemäß Anspruch 1, ferner umfassend:

mindestens einen dritten Behälter oder den mindestens einen Behälter zum Umfassen des Kühlturms, wobei der Kühlturm angepasst ist, um Kühlanforderungen zu erfüllen, die für das eine oder mehrere flüssigkeitsgekühlte Gestelle bestimmt sind, und für mindestens ein physikalisches Merkmal des mindestens einen dritten Behälters, einer Anhängerladefläche oder des mindestens einen Behälters angepasst sind.

3. Mobiles Rechenzentrumkühlsystem gemäß Anspruch 1 oder 2, ferner umfassend:

mindestens eine primäre Kühlschleife, die dem Kühlturm zugeordnet ist;
mindestens eine sekundäre Kühlschleife, die dem Behälterverteiler zugeordnet ist; und
mindestens eine Kühlverteilungseinheit (CDU), die zugeordnet ist oder innerhalb des mindestens einen Behälters zum Tauschen von Wärme zwischen der mindestens einen primären Kühlschleife und der mindestens einen sekundären Kühlschleife ist.

4. Mobiles Rechenzentrumkühlsystem gemäß einem vorangehenden Anspruch, wobei ein Merkmal des Kühlturms basierend mindestens teilweise auf einem physikalischen Merkmal bestimmt wird, das dem mindestens einem Behälter, einer Anhängerladefläche oder einem dritten Behälter zugeordnet ist, der angepasst ist, um den Kühlturm zu hosten, und teilweise auf einem zweiten Merkmal basiert, das dem einen oder mehreren flüssigkeitsgekühlten Gestellen zugeordnet ist.

5. Mobiles Rechenzentrumkühlsystem gemäß einem vorangehenden Anspruch, ferner umfassend:

Fluidkoppler, die sich von dem Behälterverteiler oder dem Behälter erstrecken und die zur Fluidkopplung zwischen dem Behälterverteiler und dem zweiten Behälterverteiler der angepasst sind.

6. Mobiles Rechenzentrumkühlsystem gemäß einem vorangehenden Anspruch, ferner umfassend:

mindestens eine Anhängerladefläche mit mindestens einer Feder, über die ein oder mehrere des mindestens einen Behälters und der Kühlturm zu tragen sind.

7. Mobiles Rechenzentrumkühlsystem gemäß einem vorangehenden Anspruch, ferner umfassend:

ein lernendes Subsystem, das mindestens einen Prozessor zum Bewerten von Temperaturanforderungen eines oder mehrerer zweiter flüssigkeitsgekühlter Gestelle, zum Bewerten von Strömungsraten oder Strömungsvolumen des Kühlmittels basierend teilweise auf ihren Zuordnungen zu den Temperaturanforderungen, zum Bewerten mindestens einer physikalischen Beschränkung des Behälters, einer Anhängerladefläche oder eines dritten Behälters, der den Kühlturm hostet, und zum Bereitstellen einer Ausgabe zum Erleichtern der Zirkulation des Kühlmittels, wobei die Ausgabe mindestens einer Kühlturmanforderung zugeordnet ist und mindestens der einen physikalischen Beschränkung des Behälters, der Anhängerladefläche oder des dritten Behälters zugeordnet ist.

8. Mobiles Rechenzentrumkühlsystem gemäß Anspruch 7, ferner umfassend:

den einen oder mehrere Strömungscontroller, um das Kühlmittel durch den Behälterverteiler und das eine oder mehrere flüssigkeitsgekühlte Gestelle zu zirkulieren; und
das lernende Subsystem, das ein Maschinenlernmodell ausführt, um:
Temperaturen, die den Temperaturanforderungen zugeordnet sind, unter Verwendung mehrerer Neuronenebenen des Maschinenlernmodells mit den Temperaturen und mit zuvor zugeordneten Strömungsraten oder Strömungsvolumen für das Kühlmittel zu verarbeiten; und
die Ausgabe, die einer Strömungsrate oder einem Strömungsvolumen für das Kühlmittel zugeordnet ist, dem einen oder mehreren Strömungscontrollern aus einer Bewertung der zuvor zugeordneten Strömungsraten oder Strömungsvolumen und der mindestens einen physikalischen Beschränkung des Behälters, der Anhängerladefläche oder des dritten Behälters bereitzustellen.

9. Mindestens ein Prozessor für ein mobiles Kühlsystem, umfassend:

mindestens eine Logikeinheit, um einen oder mehrere Strömungscontroller zu steuern, die einem Behälterverteiler zugeordnet sind, um Kühlmittel, das einem Kühlturm zugeordnet ist, der auf oder benachbart zu dem

mindestens einen Behälter angebracht ist, zu einem oder mehreren flüssigkeitsgekühlten Gestellen innerhalb mindestens eines Behälters zu zirkulieren und eine Kühlung zweiten oder mehreren flüssigkeitsgekühlten Gestellen eines zweiten Behälters zu ermöglichen, der mit dem mindestens einen Behälter gekoppelt ist.

10. Der mindestens eine Prozessor gemäß Anspruch 9, ferner umfassend:
ein lernendes Subsystem zum Bewerten von Temperaturanforderungen eines oder mehrerer zweiter flüssigkeitsgekühlter Gestelle, zum Bewerten von Strömungsraten oder Strömungsvolumen des Kühlmittels basierend teilweise auf ihren Zuordnungen zu den Temperaturanforderungen, zum Bewerten mindestens einer physikalischen Beschränkung des Behälters, einer Anhängerladefläche oder eines dritten Behälters, der den Kühlturm hostet, und zum Bereitstellen einer Ausgabe zum Erleichtern der Zirkulation des Kühlmittels, wobei die Ausgabe mindestens einer Kühlturmanforderung zugeordnet ist und mindestens der einen physikalischen Beschränkung des Behälters, der Anhängerladefläche oder des dritten Behälters zugeordnet ist.

11. Der mindestens eine Prozessor gemäß Anspruch 10, ferner umfassend:
den einen oder mehrere Strömungscontroller, um das Kühlmittel durch den Behälterverteiler und das eine oder mehrere flüssigkeitsgekühlte Gestelle zu zirkulieren; und
das lernende Subsystem, das ein Maschinenlernmodell ausführt, um:
Temperaturen, die den Temperaturanforderungen zugeordnet sind, unter Verwendung mehrerer Neuronenebenen des Maschinenlernmodells zu verarbeiten, das die Temperaturen aufweist und die zuvor zugeordneten Strömungsraten oder Strömungsvolumen für das Kühlmittel aufweist; und
die Ausgabe, die einer Strömungsrate oder einem Strömungsvolumen für das Kühlmittel zugeordnet ist, dem einen oder mehreren Strömungscontrollern aus einer Bewertung der zuvor zugeordneten Strömungsraten oder Strömungsvolumen und der mindestens einen physikalischen Beschränkung des Behälters oder des dritten Behälters bereitzustellen.

12. Der mindestens eine Prozessor gemäß Anspruch 11, ferner umfassend:
eine Anweisungsausgabe zum Kommunizieren der Ausgabe mit dem einen oder mehreren Strömungscontrollern, um die Zirkulation des Kühlmittels innerhalb des Behälterverteilers oder von dem Behälterverteiler zu dem zweiten Behälterverteiler des zweiten Behälters zu erleichtern.

13. Der mindestens eine Prozessor gemäß einem der Ansprüche 9 bis 12, ferner umfassend:
die mindestens eine Logikeinheit, die angepasst ist, um einen Temperaturwert von einem Temperatursensor innerhalb des mindestens einen Behälters zu empfangen, und angepasst ist, um die Zirkulation des Kühlmittels zu erleichtern, um das eine oder mehrere flüssigkeitsgekühlte Gestelle zu kühlen.

14. Der mindestens eine Prozessor gemäß einem der Ansprüche 9 bis 13, ferner umfassend:
eine kommunikative Kopplung mit einem Rechenzentrum-Managementsystem (DMS), die aktiviert ist innerhalb oder dem einen oder mehreren flüssigkeitsgekühlten Gestellen zugeordnet ist, wobei die kommunikative Kopplung Temperatureingaben zu empfangen und Steuerausgaben für den einen oder mehreren Strömungscontroller zu kommunizieren hat, um die Zirkulation des Kühlmittels zu erleichtern.

15. Der mindestens eine Prozessor für ein mobiles Kühlsystem, umfassend:
mindestens eine Logikeinheit, um ein oder mehrere neuronale Netzwerk mit verborgenen Schichten von Neuronen zum Bewerten von Temperaturanforderungen eines oder mehreren flüssigkeitsgekühlter Gestelle zu trainieren, die in einem Behälter zu hosten sind, zum Bewerten von Strömungsraten oder Strömungsvolumen eines Kühlmittel basierend teilweise auf ihren Zuordnungen zu den Temperaturanforderungen, zum Bewerten mindestens einer physikalischen Beschränkung des Behälter, einer Anhängerladefläche oder eines zweiten Behälters, der einen Kühlturm hostet, um das eine oder mehrere flüssigkeitsgekühlte Gestelle zu kühlen, und zum Bereitstellen einer Ausgabe zum Erleichtern der Zirkulation des Kühlmittels, wobei die Ausgabe mindestens einer Kühlturmanforderung zugeordnet ist und der mindestens einen physikalischen Beschränkung des Behälters, der Anhängerladefläche oder dem zweiten Behälter zugeordnet ist.

16. Der mindestens eine Prozessor gemäß Anspruch 15, ferner umfassend:
die mindestens eine Logikeinheit zum Bewerten der Temperaturanforderungen des einen oder mehrerer flüssigkeitsgekühlter Gestelle und der Strömungsraten oder Strömungsvolumen des Kühlmittels und zum Bereitstellen der Ausgabe, die eine Zuordnung zu mindestens einer Temperatur aufweist, die von dem einen oder mehreren flüssigkeitsgekühlten Gestellen durch die Zirkulation des Kühlmittels erlangt werden kann.

17. Der mindestens eine Prozessor gemäß Anspruch 15 oder 16, ferner umfassend:

eine Anweisungsausgabe zum Kommunizieren einer Ausgabe von der mindestens einen Logikeinheit mit dem einen oder mehreren Strömungscontrollern, um die Zirkulation des Kühlmittels innerhalb eines Behälterverteilers oder von dem Behälterverteiler zu einem zweiten Behälterverteiler eines dritten Behälters zu erleichtern, der das zweite oder mehrere flüssigkeitsgekühlte Gestelle hostet.

18. Der mindestens eine Prozessor gemäß einem der Ansprüche 15 bis 17, ferner umfassend: die mindestens eine Logikeinheit, die angepasst ist, um einen Temperaturwert von einem Temperatursensor innerhalb des Behälters zu empfangen, und angepasst ist, um die Zirkulation des Kühlmittels zu erleichtern, um das eine oder mehrere flüssigkeitsgekühlte Gestelle zu kühlen.

19. Mobiles Rechenzentrumkühlsystem, umfassend: mindestens einen Prozessor, um ein oder mehrere neuronale Netzwerke mit verborgenen Schichten von Neuronen zu trainieren, um Temperaturanforderungen eines oder mehrerer flüssigkeitsgekühlter Gestelle zu bewerten, die in einem Behälter zu hosten sind, um Strömungsraten oder Strömungsvolumen eines Kühlmittels basierend teilweise auf ihren Zuordnungen zu den Temperaturanforderungen zu bewerten, zum Bewerten mindestens einer physikalischen Beschränkung der Behälter, einer Anhängerladefläche oder eines zweiten Behälters, der einen Kühlturm hostet, um das eine oder mehrere flüssigkeitsgekühlte Gestelle zu kühlen, und zum Bereitstellen einer Ausgabe zum Erleichtern der Zirkulation des Kühlmittels, wobei die Ausgabe mindestens einer Kühlturmanforderung zugeordnet ist und mindestens der einen physikalischen Beschränkung des Behälters, der Anhängerladefläche oder des zweiten Behälters zugeordnet ist.

20. Mobiles Rechenzentrumkühlsystem gemäß Anspruch 19, ferner umfassend: den mindestens einen Prozessor zum Bewerten der Temperaturanforderungen des einen oder mehrerer flüssigkeitsgekühlter Gestelle und der Strömungsraten oder Strömungsvolumen des Kühlmittels und zum Bereitstellen der Ausgabe mit einer Zuordnung zu mindestens einer Temperatur, die von dem einen oder mehreren flüssigkeitsgekühlten Gestellen durch die Zirkulation des Kühlmittels erreichbar ist.

21. Mobiles Rechenzentrumkühlsystem gemäß Anspruch 19 oder 20, ferner umfassend: eine Anweisungsausgabe zum Kommunizieren einer Ausgabe von der mindestens einen Logikeinheit mit dem einen oder mehreren Strömungscontrollern, um die Zirkulation des Kühlmittels innerhalb eines Behälterverteilers oder von dem Behälterverteiler zu einem zweiten Behälterverteiler eines dritten Behälters zu erleichtern, der das zweite oder mehrere flüssigkeitsgekühlte Gestelle hostet.

22. Mobiles Rechenzentrumkühlsystem gemäß einem der Ansprüche 19 bis 21, ferner umfassend: mindestens einen Prozessor, der angepasst ist, um einen Temperaturwert von einem Temperatursensor innerhalb des Behälters zu empfangen, und angepasst ist, die Zirkulation des Kühlmittels zu erleichtern, um das eine oder mehrere flüssigkeitsgekühlte Gestelle zu kühlen.

23. Verfahren zur Kühlung eines mobilen Rechenzentrums, umfassend: Bereitstellen eines Behälterverteilers, um Kühlmittel zu zirkulieren, das einem Kühlturm zugeordnet ist, der angebracht ist an oder benachbart zu mindestens einem Behälter mit einem oder mehreren flüssigkeitsgekühlten Gestellen; und Ermöglichen einer Fluidkopplung des Behälterverteilers mit einem zweiten Behälterverteiler eines zweiten Behälters.

24. Verfahren gemäß Anspruch 23, ferner umfassend: Bereitstellen mindestens eines dritten Behälters, einer Anhängerladefläche oder dem Behälter, um den Kühlturm zu umfassen, wobei der Kühlturm angepasst ist, um mindestens eine Kühlturmanforderung zu erfüllen, die für das eine oder mehrere flüssigkeitsgekühlte Gestelle bestimmt und angepasst ist, um mindestens ein physikalisches Merkmal des mindestens einen dritten Behälters, der Anhängerladefläche oder des mindestens einen Behälters zu erfüllen.

25. Verfahren gemäß Anspruch 23 oder 24, ferner umfassend: Ermöglichen, dass mindestens eine primäre Kühlschleife dem Kühlturm zugeordnet ist; Ermöglichen, dass der Behälterverteiler mindestens einer sekundären Kühlschleife zugeordnet ist; und Ermöglichen mindestens einer Kühlverteilungseinheit (CDU), die dem mindestens einen Behälter zugeordnet ist, um Wärme zwischen der mindestens einen primären Kühlschleife und der mindestens einen sekundären Kühlschleife zu tauschen.

26. Verfahren gemäß einem der Ansprüche 23 bis 25, wobei ein Merkmal des Kühlturms basierend teilweise auf mindestens einem physikalischen Merkmal bestimmt wird, das dem mindestens einen Behälter, einer Anhängerladefläche oder einem dritten Behälter zugeordnet ist, der angepasst ist, um den Kühlturm zu hosten, und teilweise auf einem zweiten Merkmal basiert, das dem einen oder mehreren flüssigkeitsgekühlten Gestellen zugeordnet ist.

27. Verfahren gemäß einem der Ansprüche 23 bis 26, ferner umfassend:
Bewerten von Temperaturanforderungen eines oder mehrerer zweiter flüssigkeitsgekühlter Gestelle;
Bewerten von Strömungsraten oder Strömungsvolumen des Kühlmittels basierend teilweise auf ihren Zuordnungen mit den Temperaturanforderungen;
Bewerten mindestens einer physikalischen Beschränkung des Behälters, einer Anhängerladefläche oder eines dritten Behälters, der den Kühlturm hostet; und
Bereitstellen einer Ausgabe zum Erleichtern der Zirkulation des Kühlmittels, wobei die Ausgabe mindestens einer Kühlturmanforderung zugeordnet ist und der mindestens einen physikalischen Beschränkung des Behälters, der Anhängerladefläche oder des dritten Behälters zugeordnet ist.

28. Verfahren gemäß Anspruch 27, ferner umfassend:
Verwenden des einen oder mehrerer Strömungscontroller, um das Kühlmittel durch den Behälterverteiler und dem einen oder mehreren flüssigkeitsgekühlten Gestellen zu zirkulieren; und
Ausführen eines Maschinenlernmodells für das lernende Subsystem, wobei das Ausführen:
Temperaturen verarbeitet, die den Temperaturanforderungen zugeordnet sind, unter Verwendung mehrerer Neuronenebenen des Maschinenlernmodells mit den Temperaturen und mit zuvor zugeordnete Strömungsraten oder Strömungsvolumen für das Kühlmittel; und
die Ausgabe, die einer Strömungsrate oder einem Strömungsvolumen für das Kühlmittel zugeordnet ist, dem einen oder mehreren Strömungscontrollern aus einer Bewertung der zuvor zugeordneten Strömungsraten oder Strömungsvolumen und der mindestens einen physikalischen Beschränkung der Behälter, der Anhängerladefläche oder des dritten Behälters bereitstellt.

29. Verfahren gemäß einem der Ansprüche 23 bis 28, ferner umfassend:
Steuern, unter Verwendung mindestens eines Prozessors, eines oder mehrerer Strömungscontroller, die dem Behälterverteiler zugeordnet sind, um das dem Kühlturm zugeordnete Kühlmittel zu dem einen oder mehreren flüssigkeitsgekühlten Gestellen zu zirkulieren und dem Kühlmittel zu ermöglichen, von dem Behälterverteiler zu dem zweiten Behälterverteiler des zweiten Behälters zu strömen.

30. Verfahren gemäß einem der Ansprüche 23 bis 29, ferner umfassend:
Bereitstellen von Fluidkopplern, die sich von dem Behälterverteiler oder dem Behälter erstrecken und die für die Fluidkopplung zwischen dem Behälterverteiler und dem zweiten Behälterverteiler angepasst sind.

Es folgen 41 Seiten Zeichnungen

Anhängende Zeichnungen

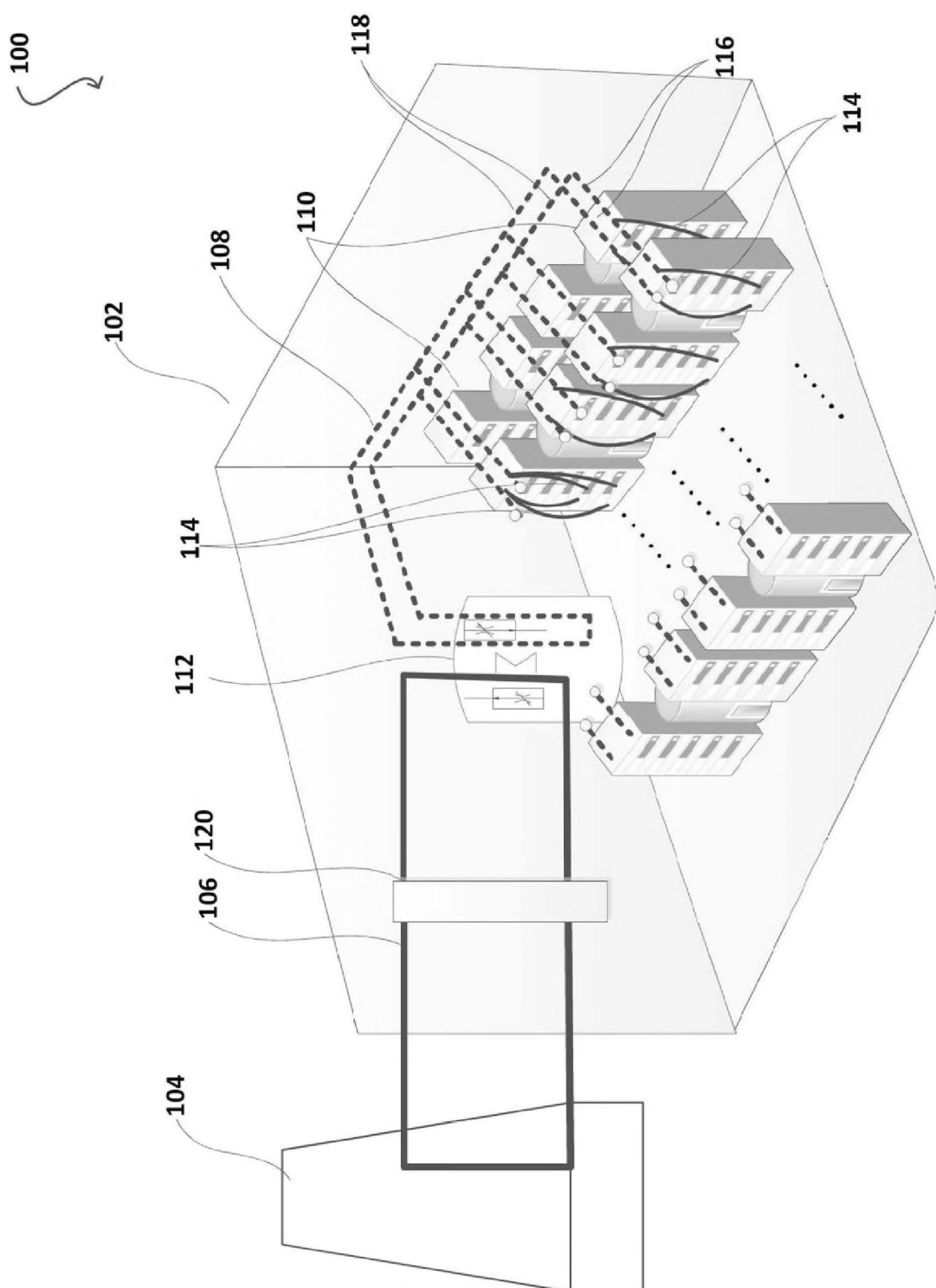


FIG. 1

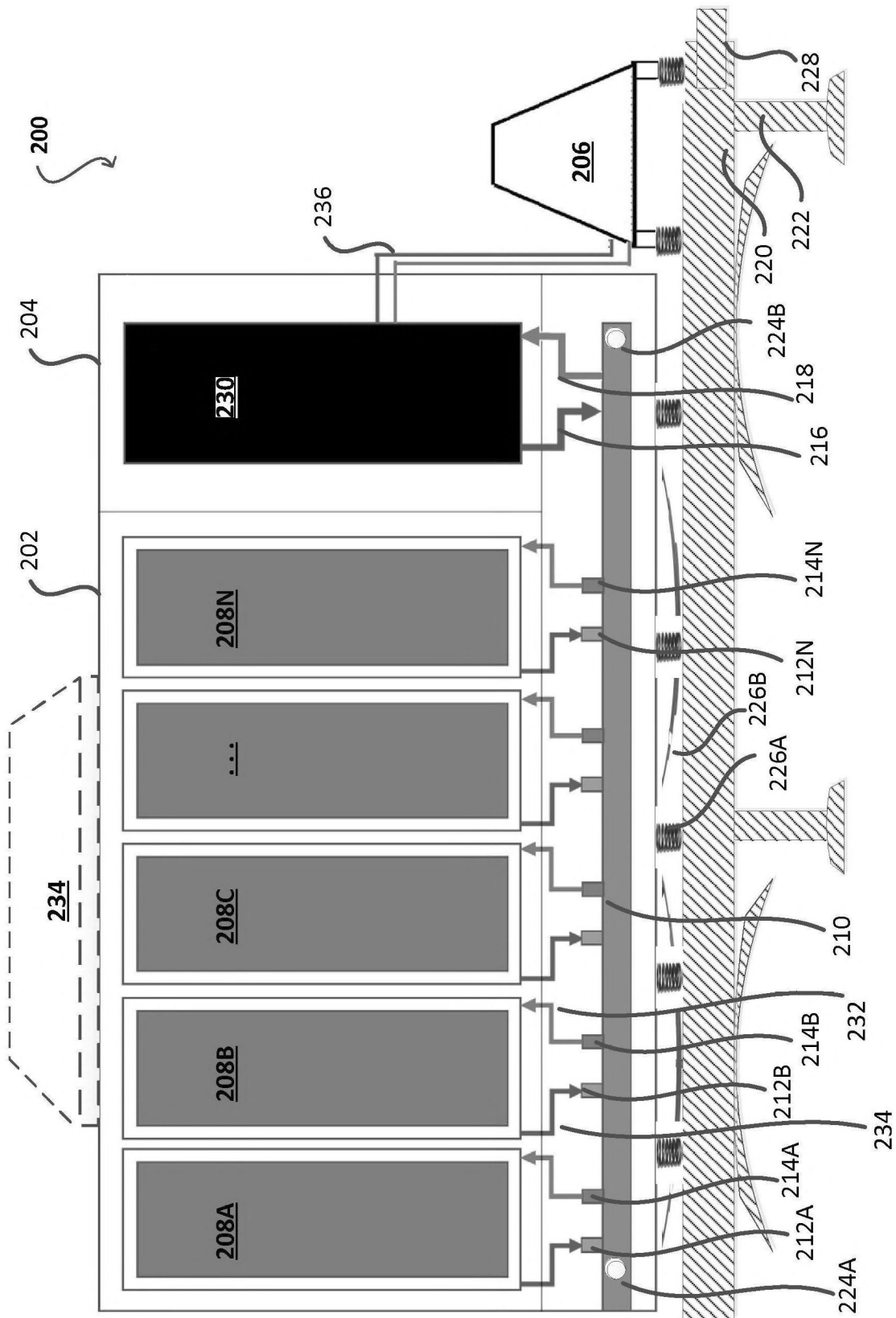


FIG. 2

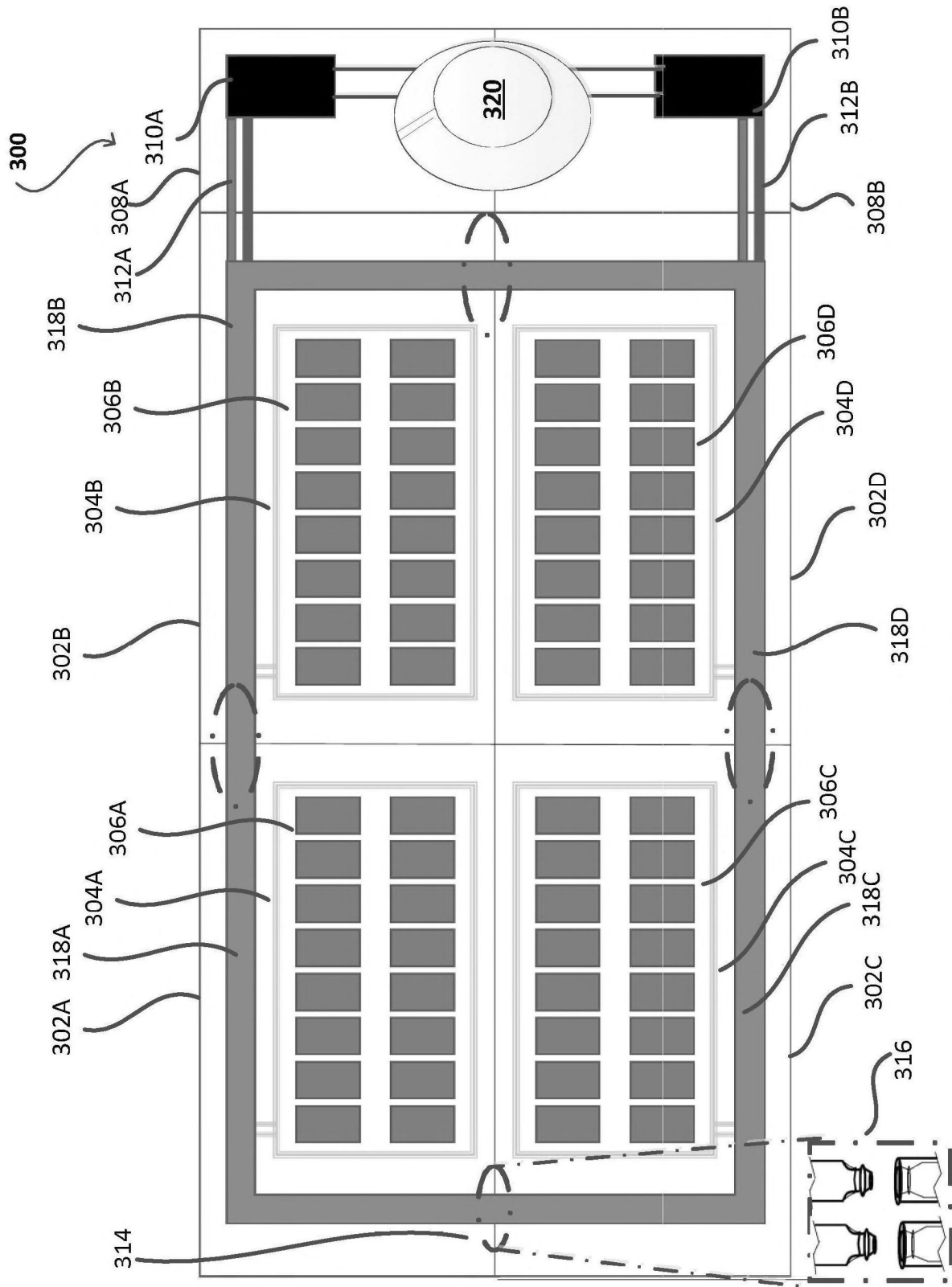


FIG. 3

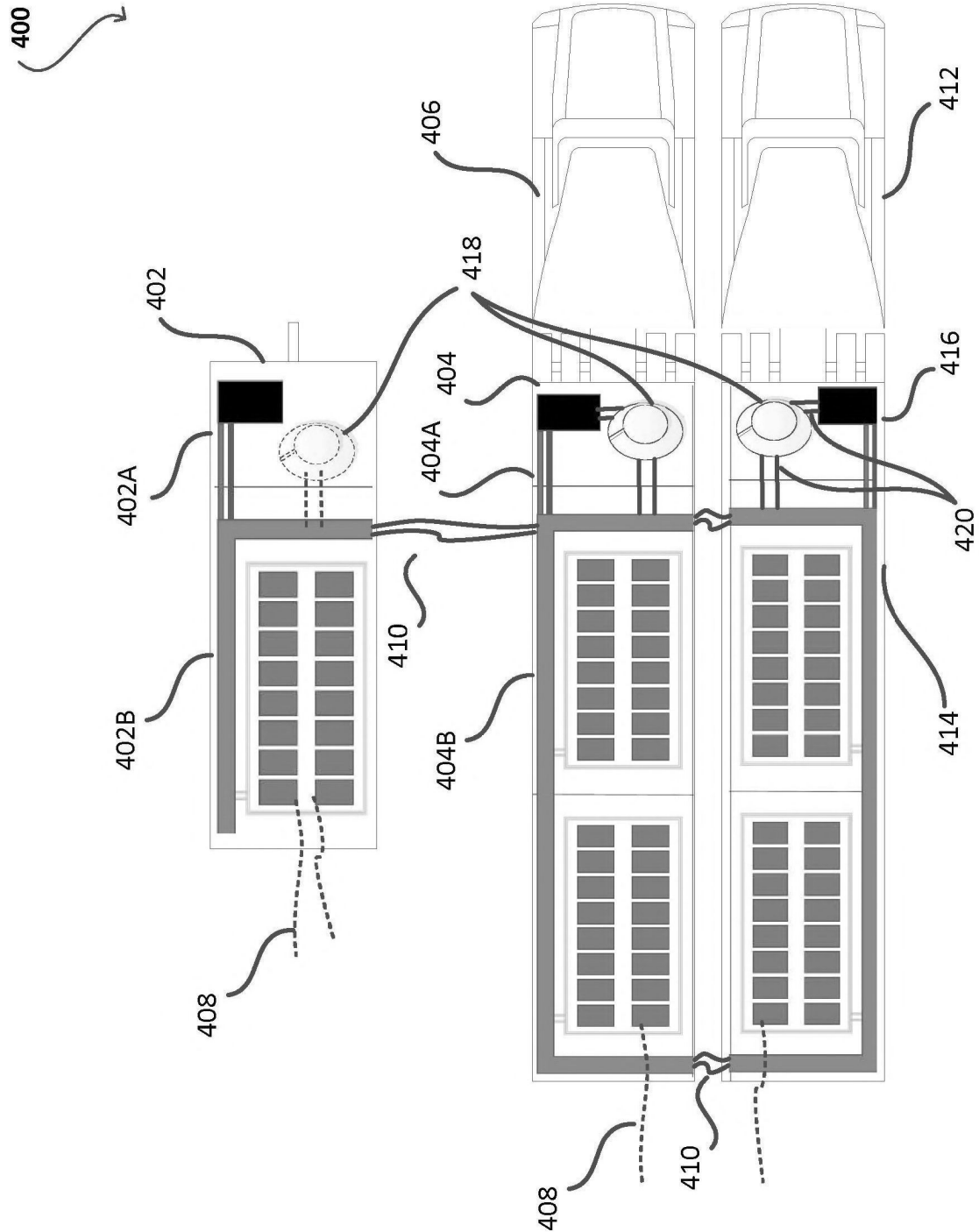


FIG. 4

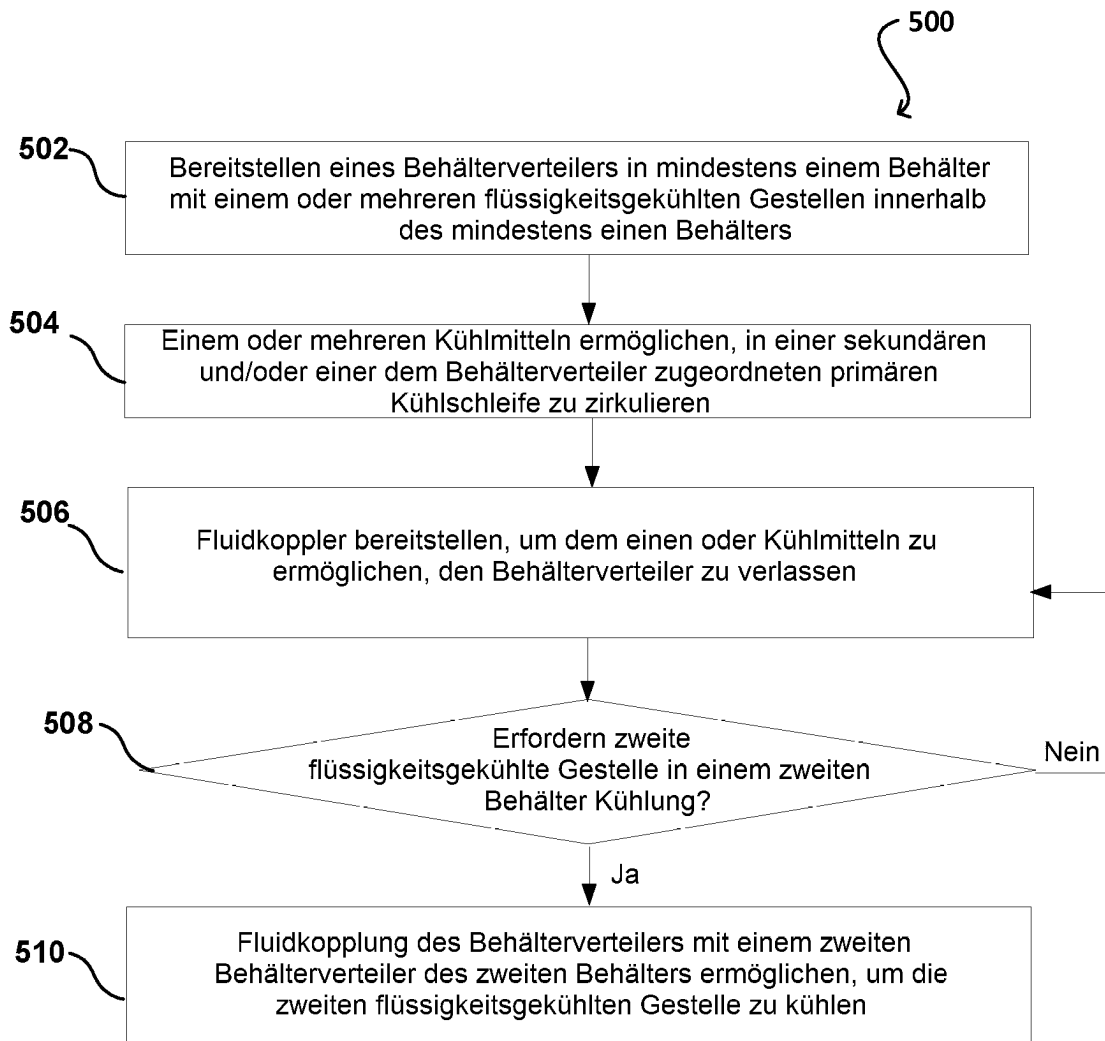


FIG. 5

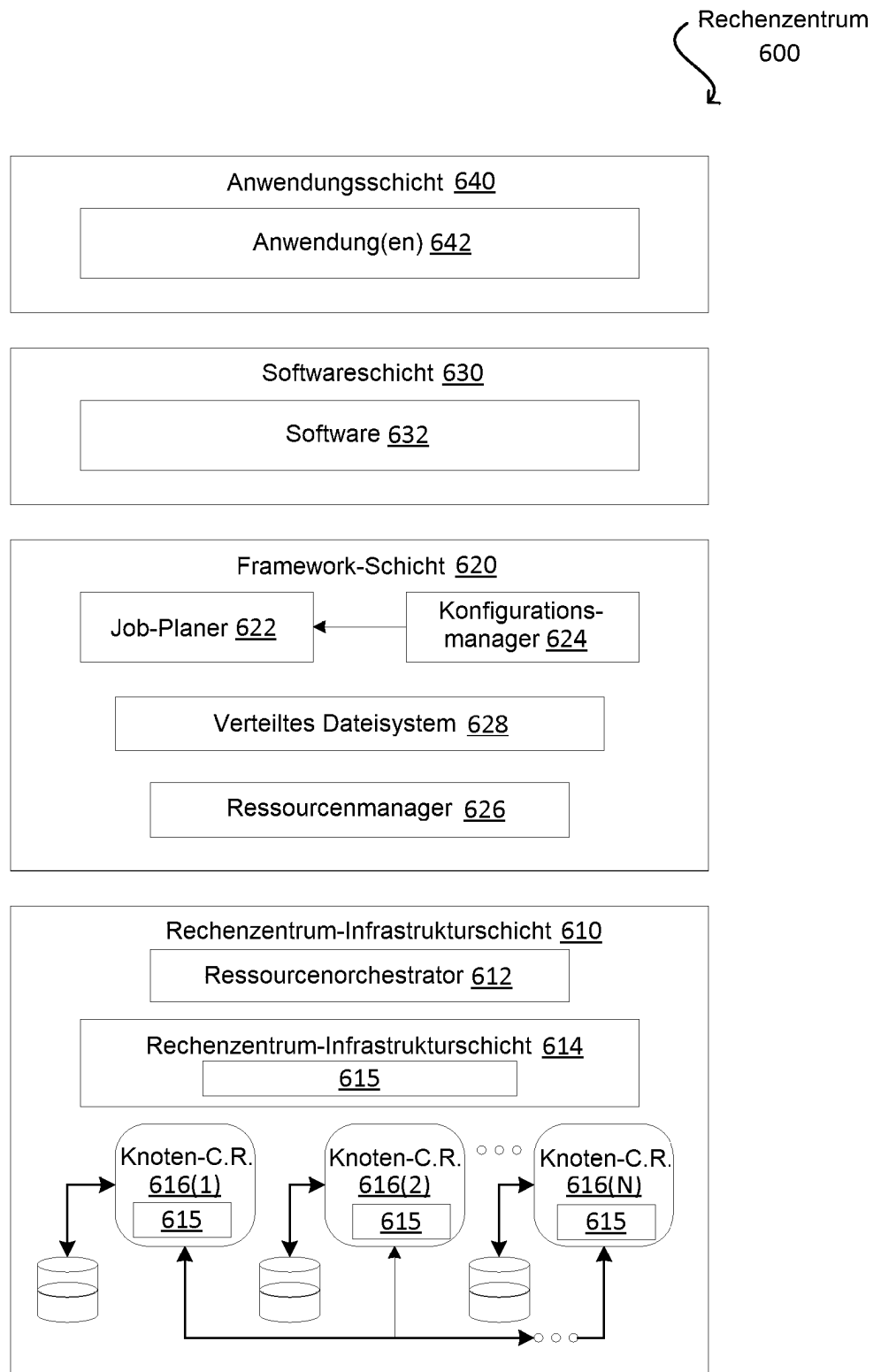


FIG. 6A

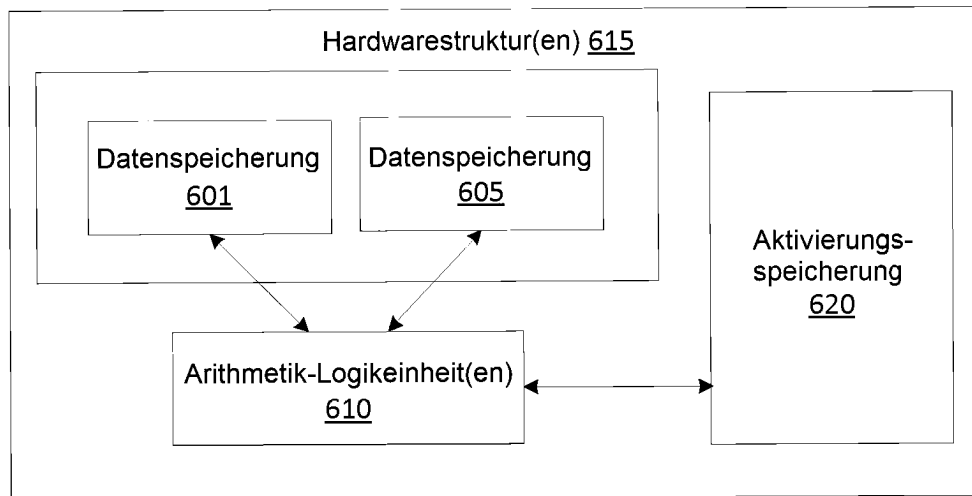


FIG. 6B

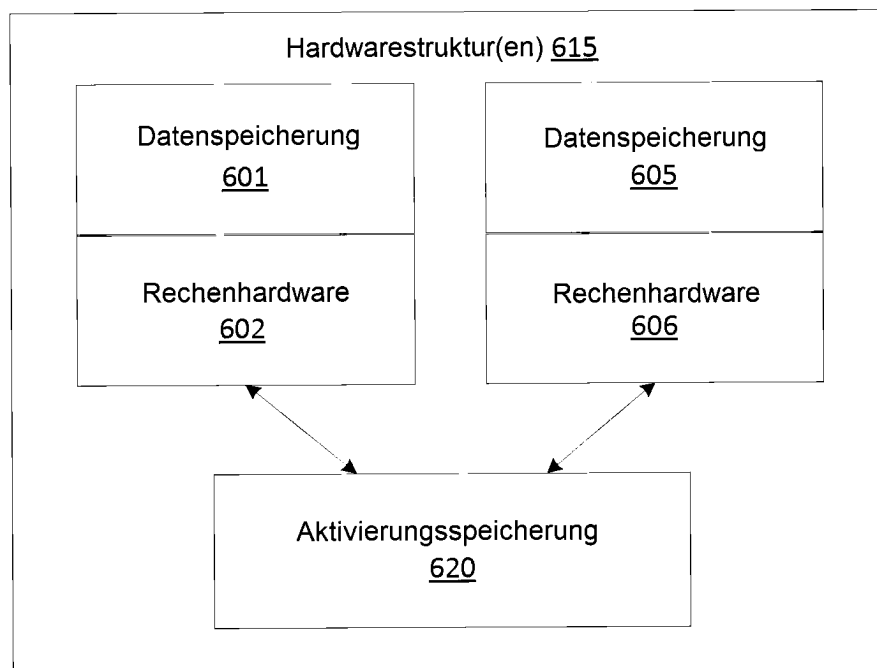
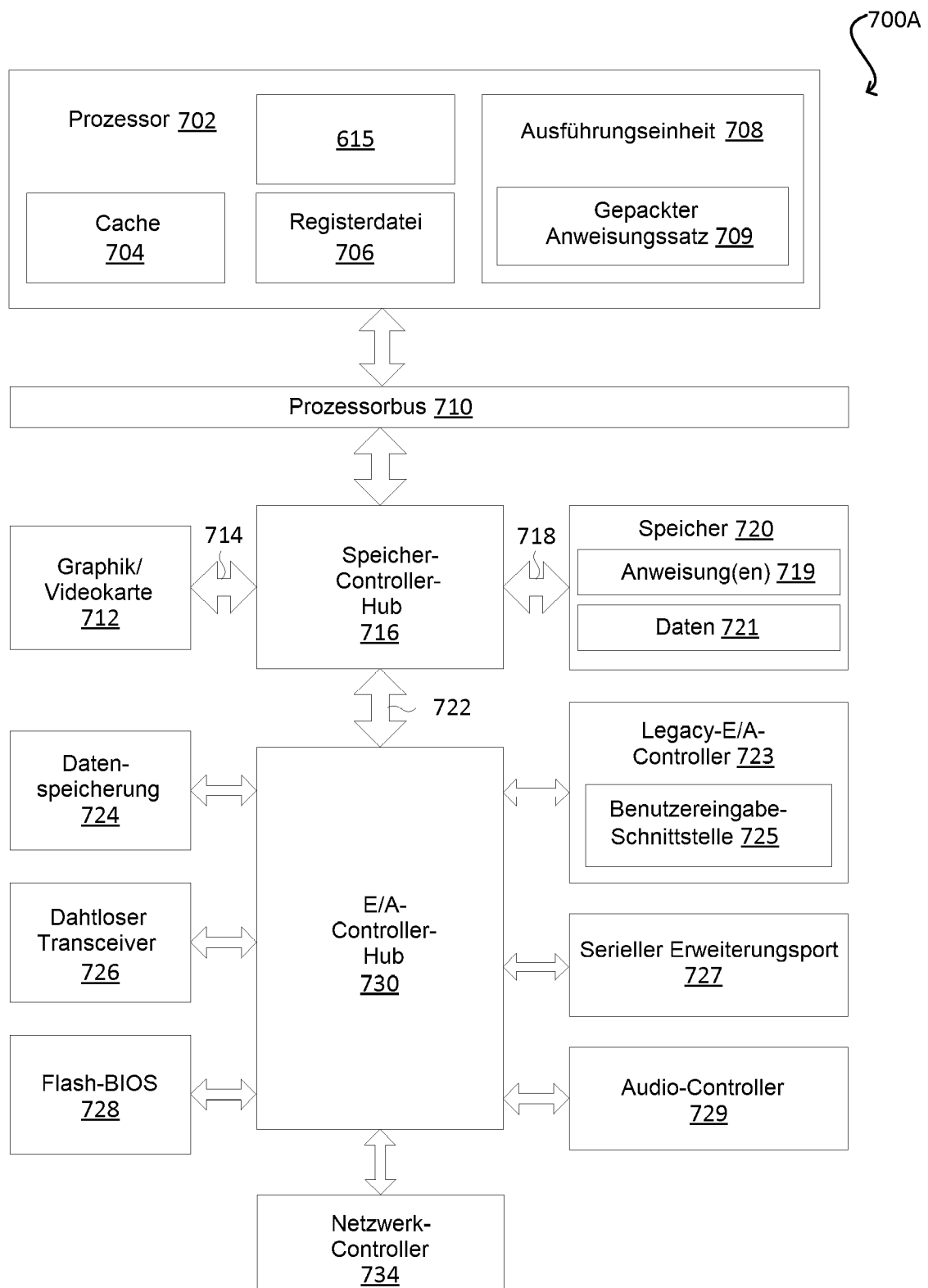


FIG. 6C

**FIG. 7A**

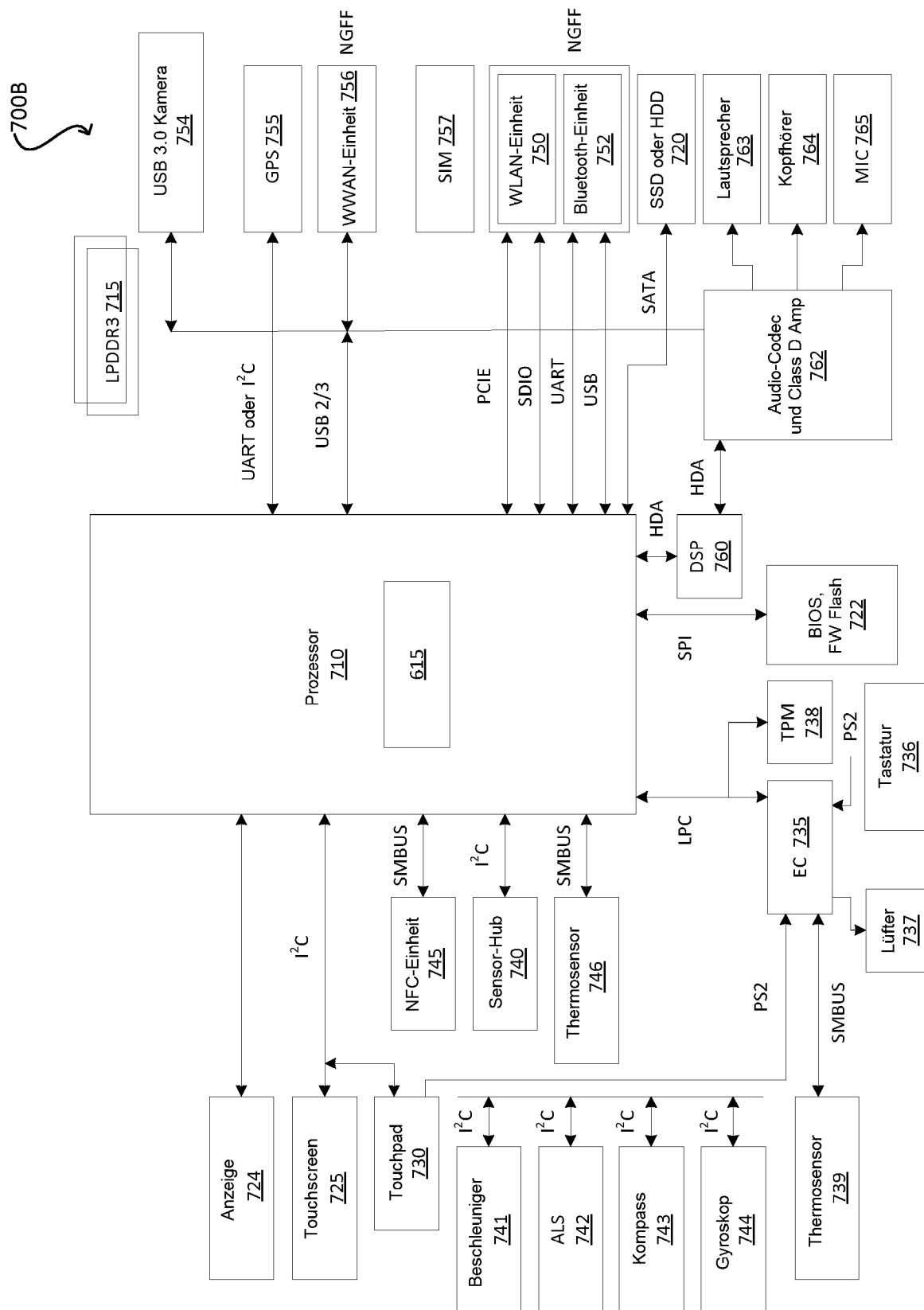


FIG. 7B

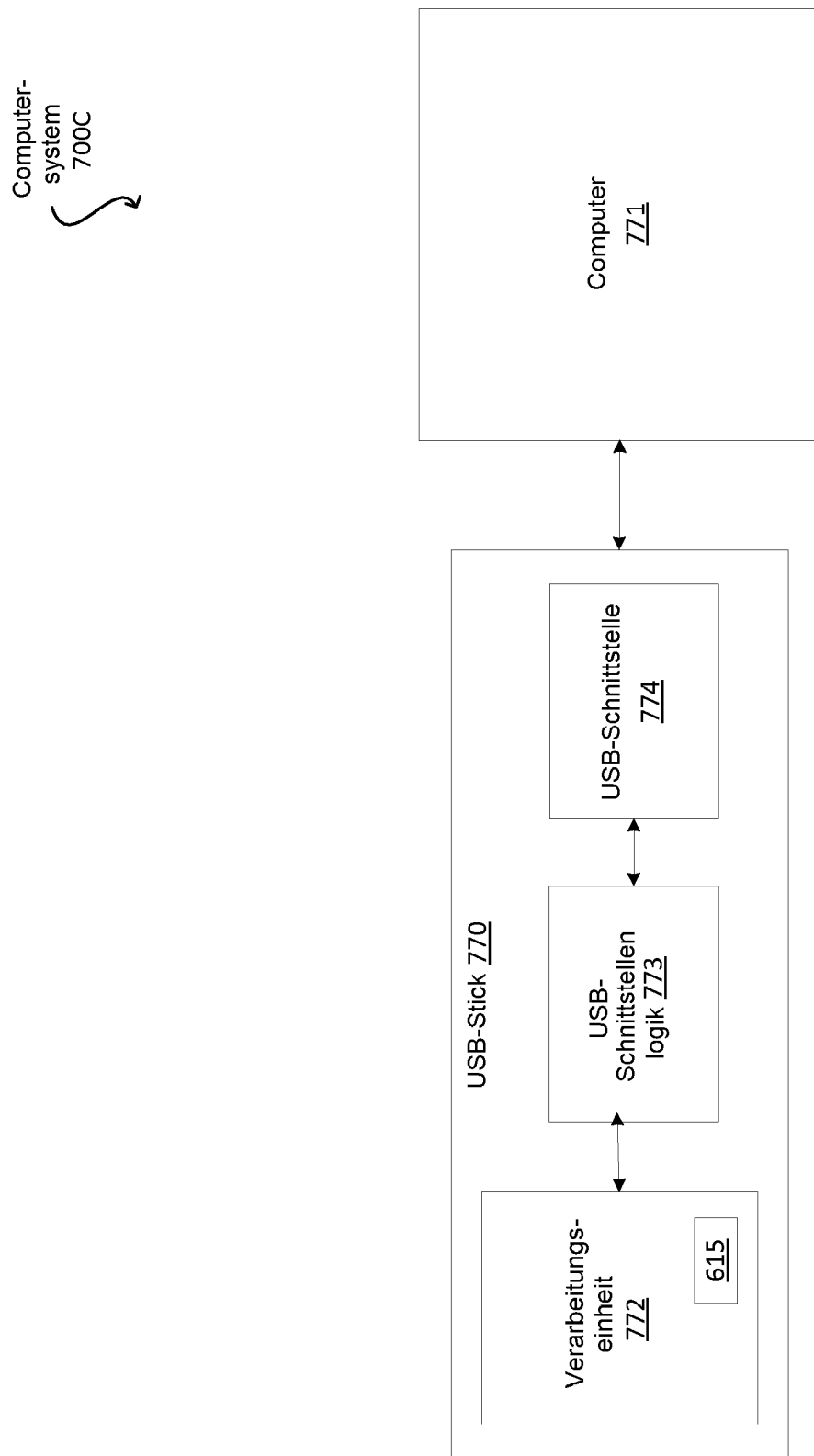


FIG. 7C

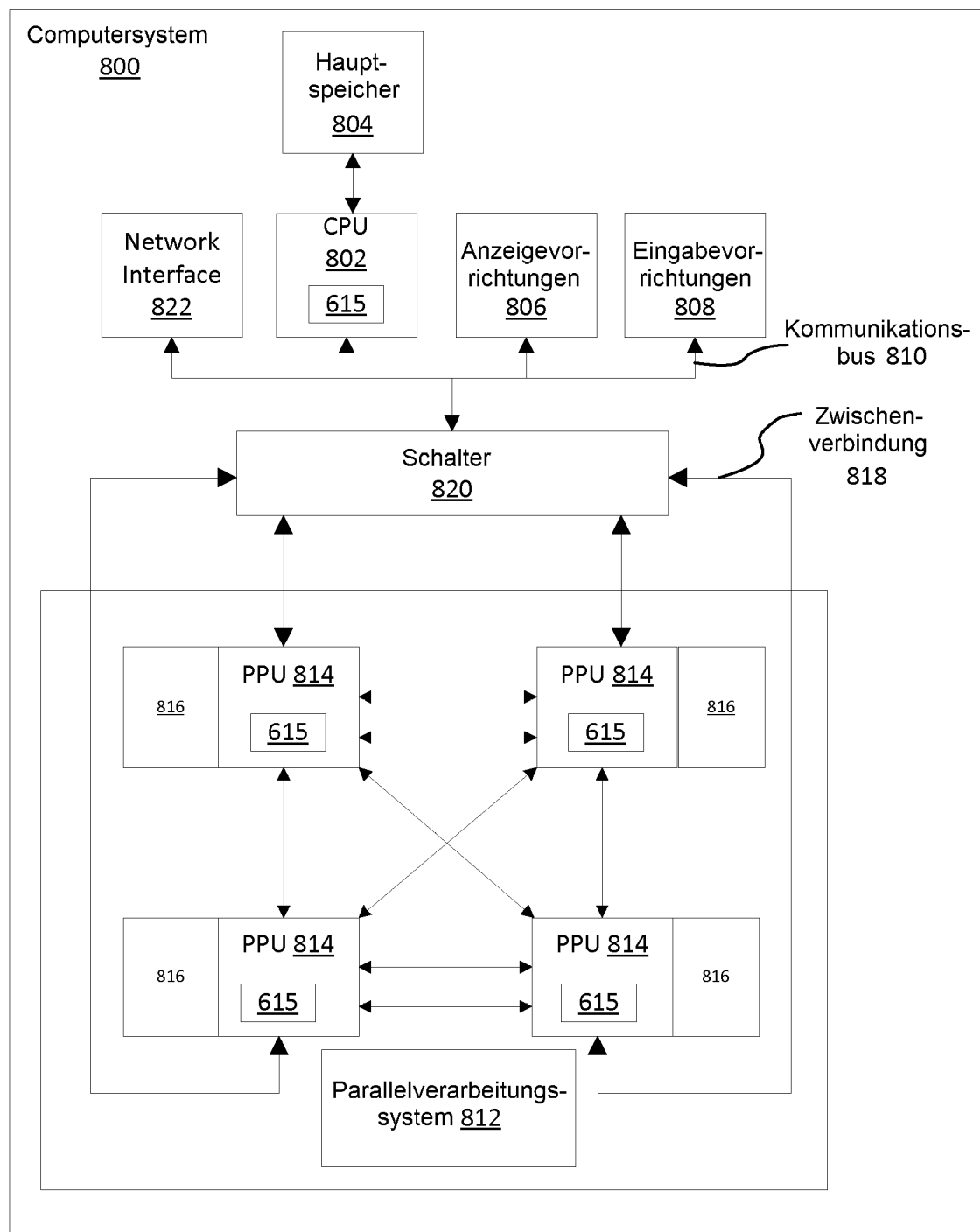


FIG. 8

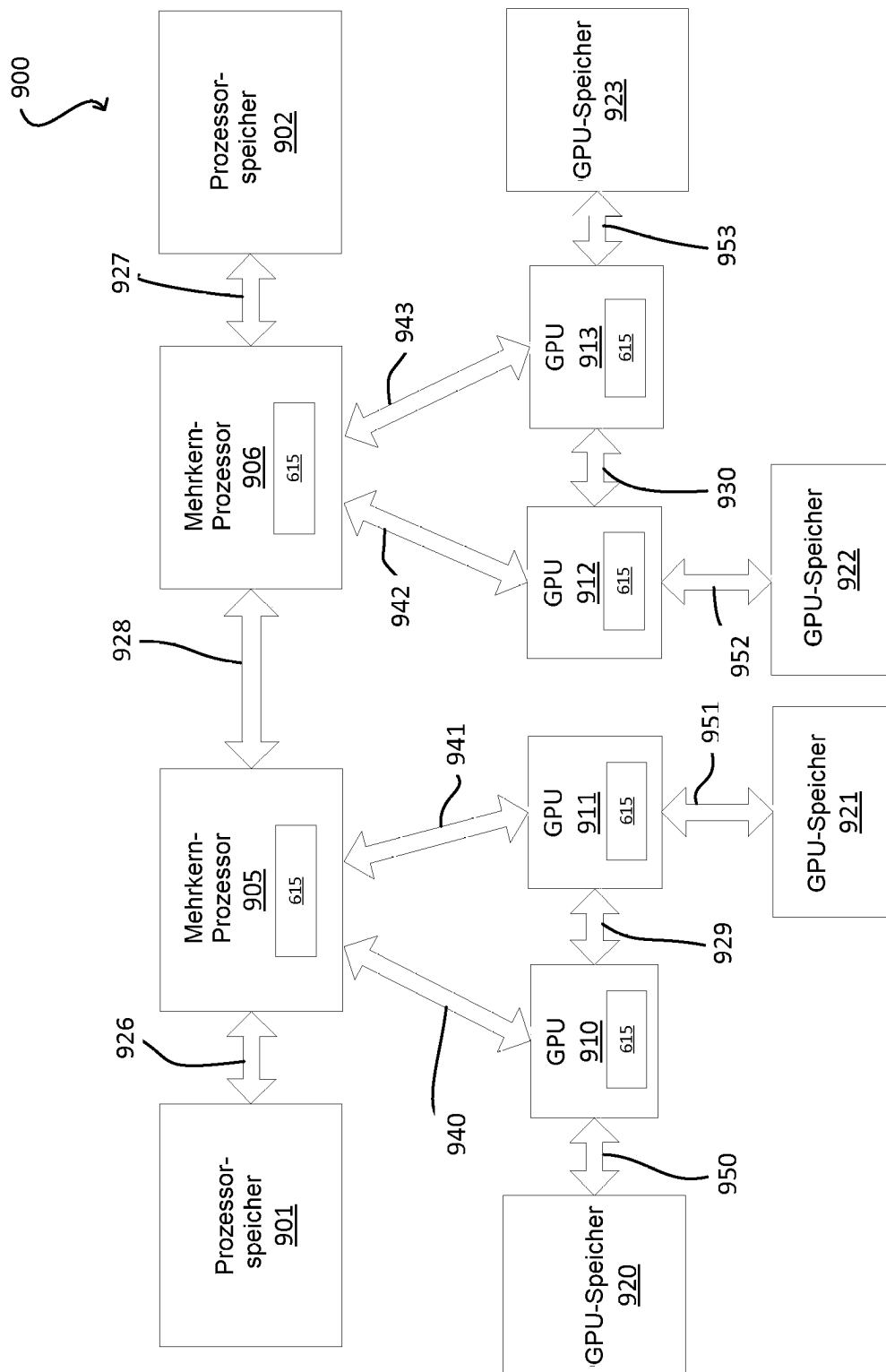
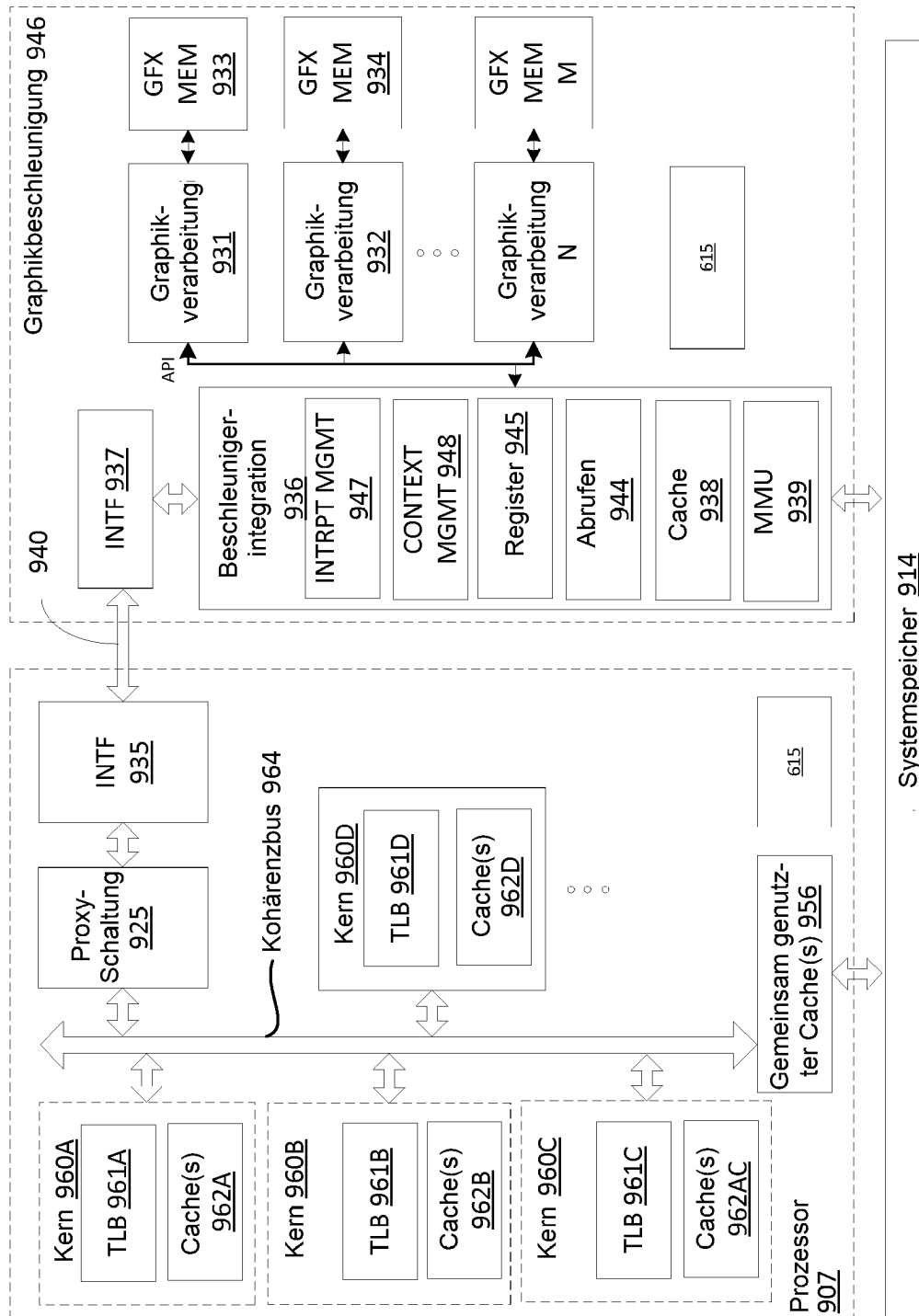


FIG. 9A

**FIG. 9B**

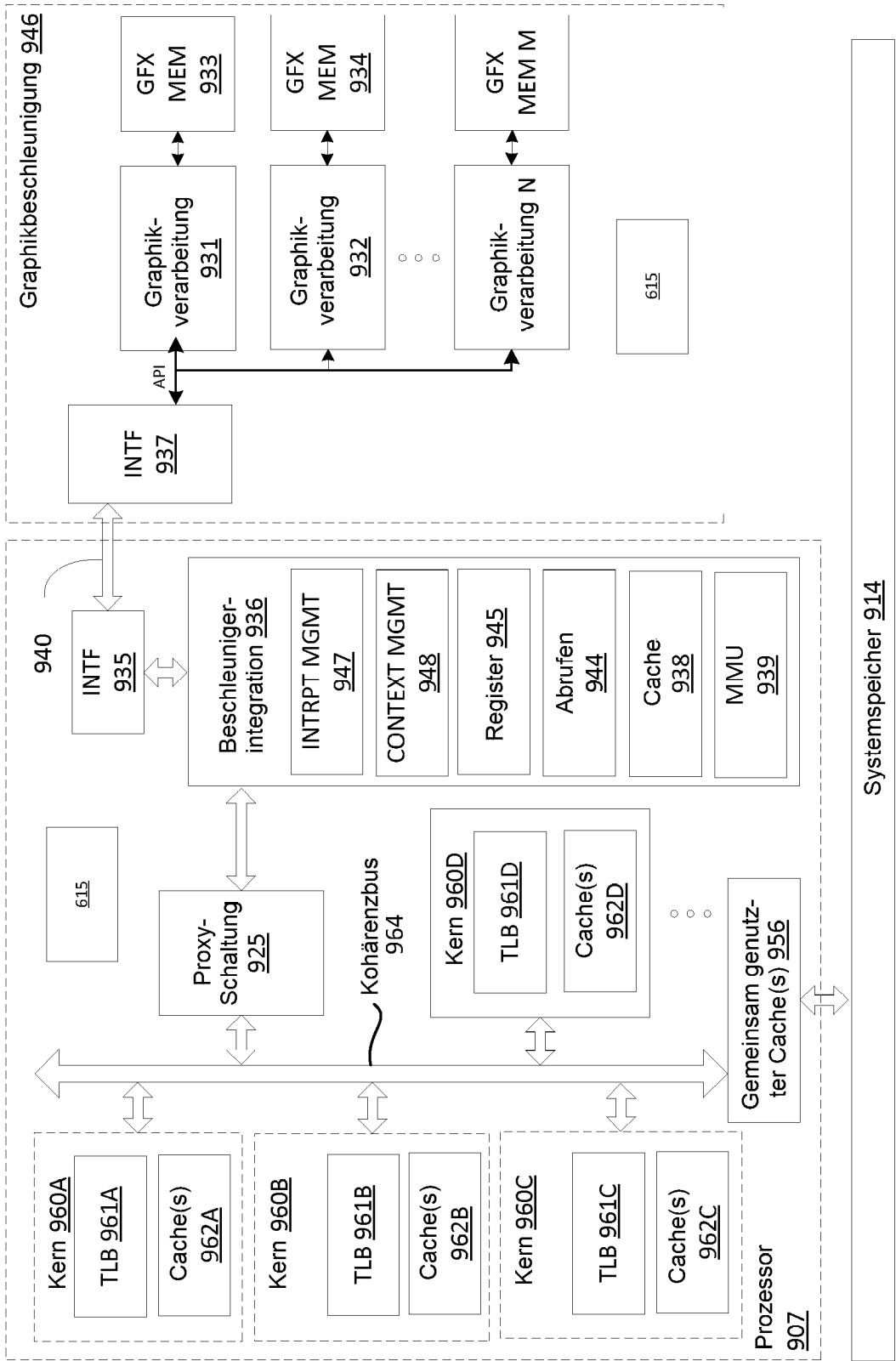
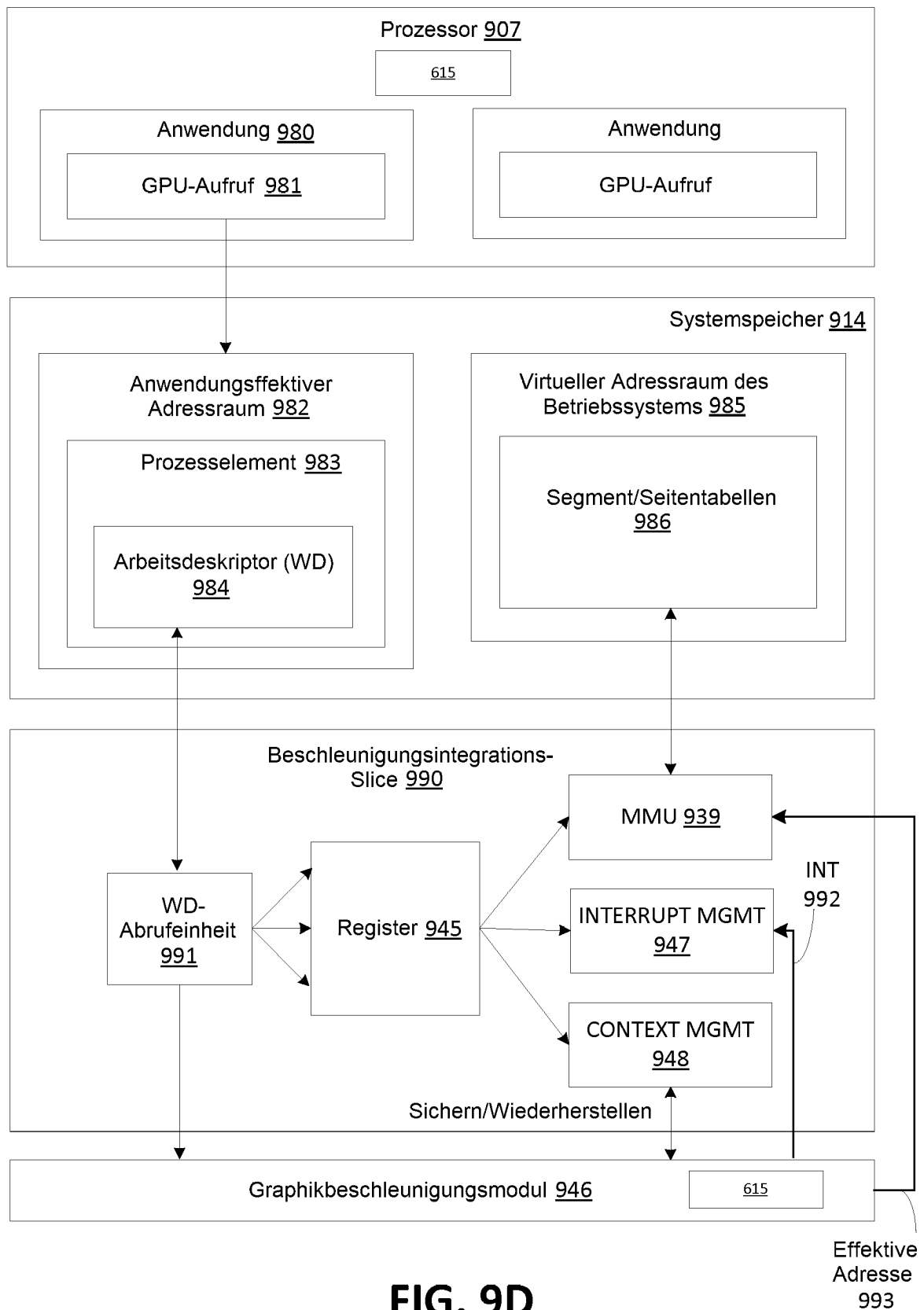


FIG. 9C



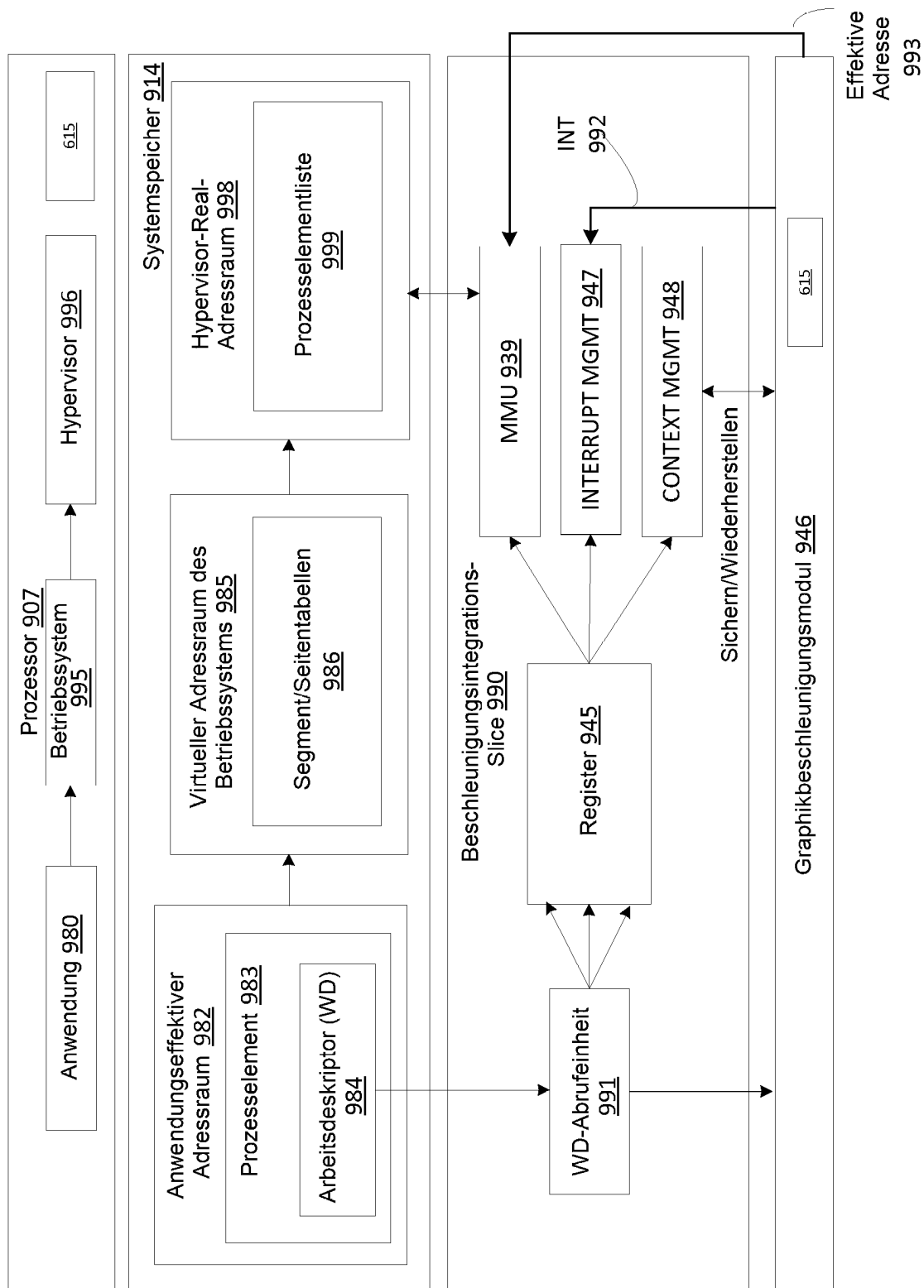


FIG. 9E

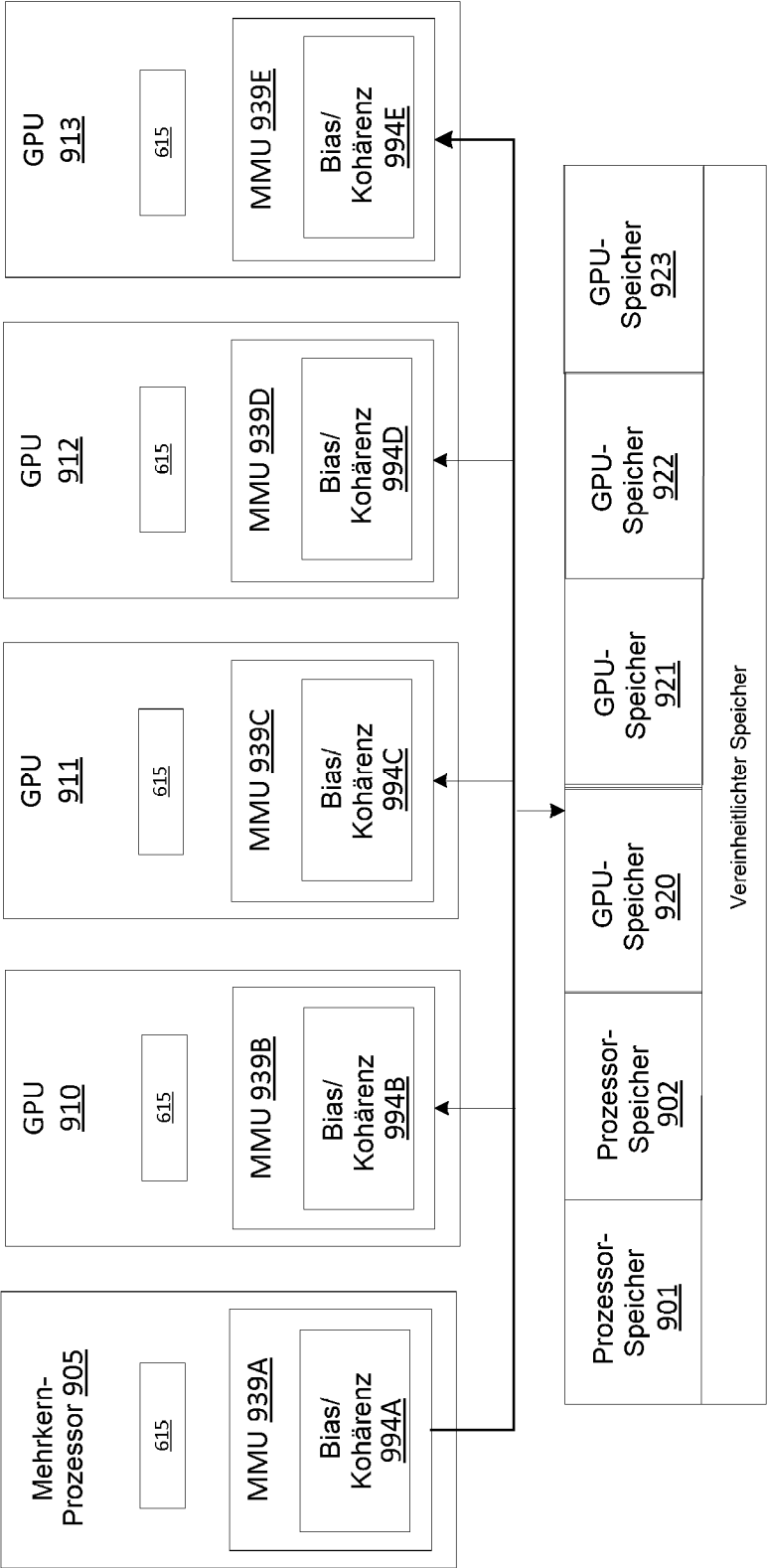


FIG. 9F

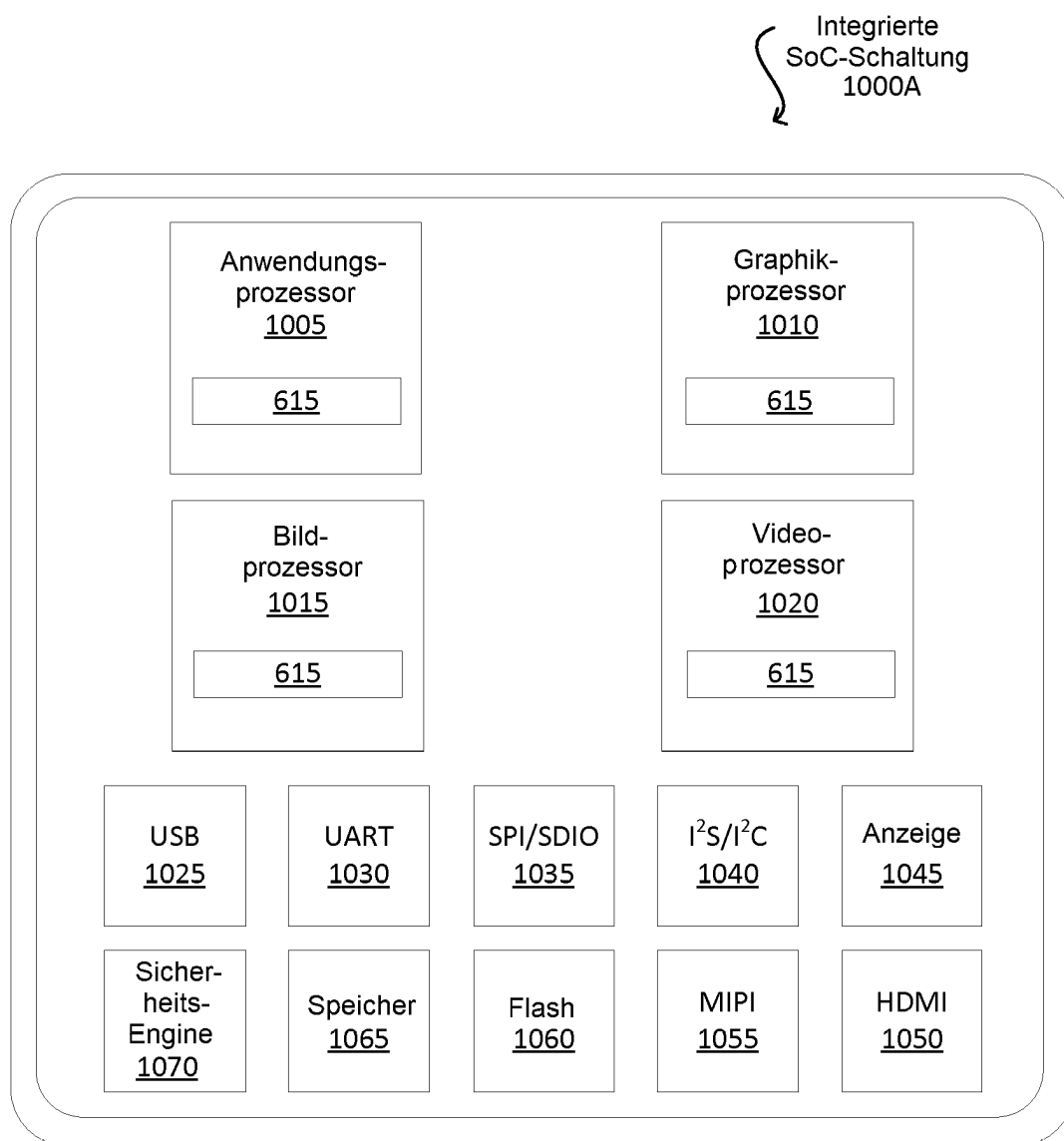


FIG. 10A

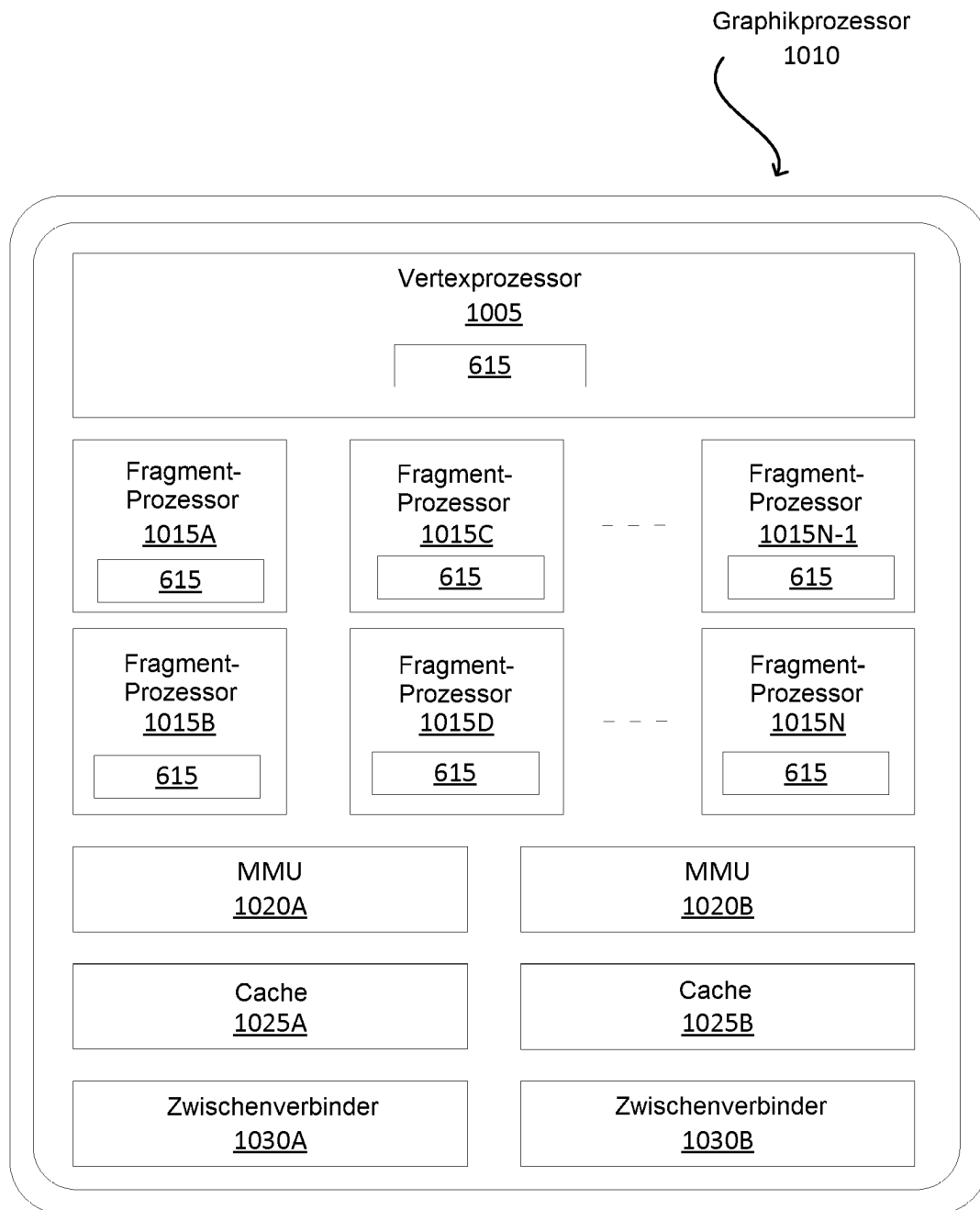


FIG. 10B

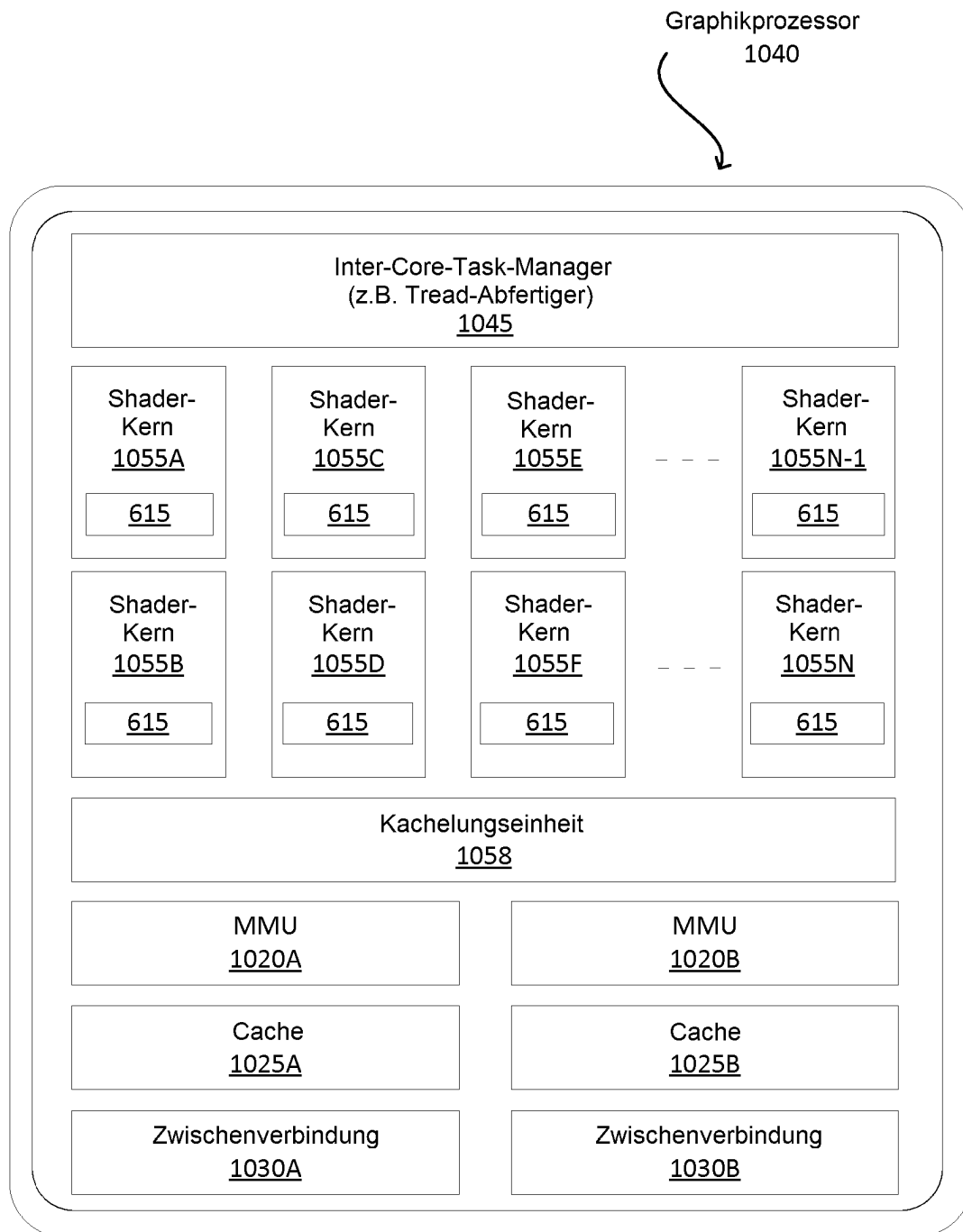
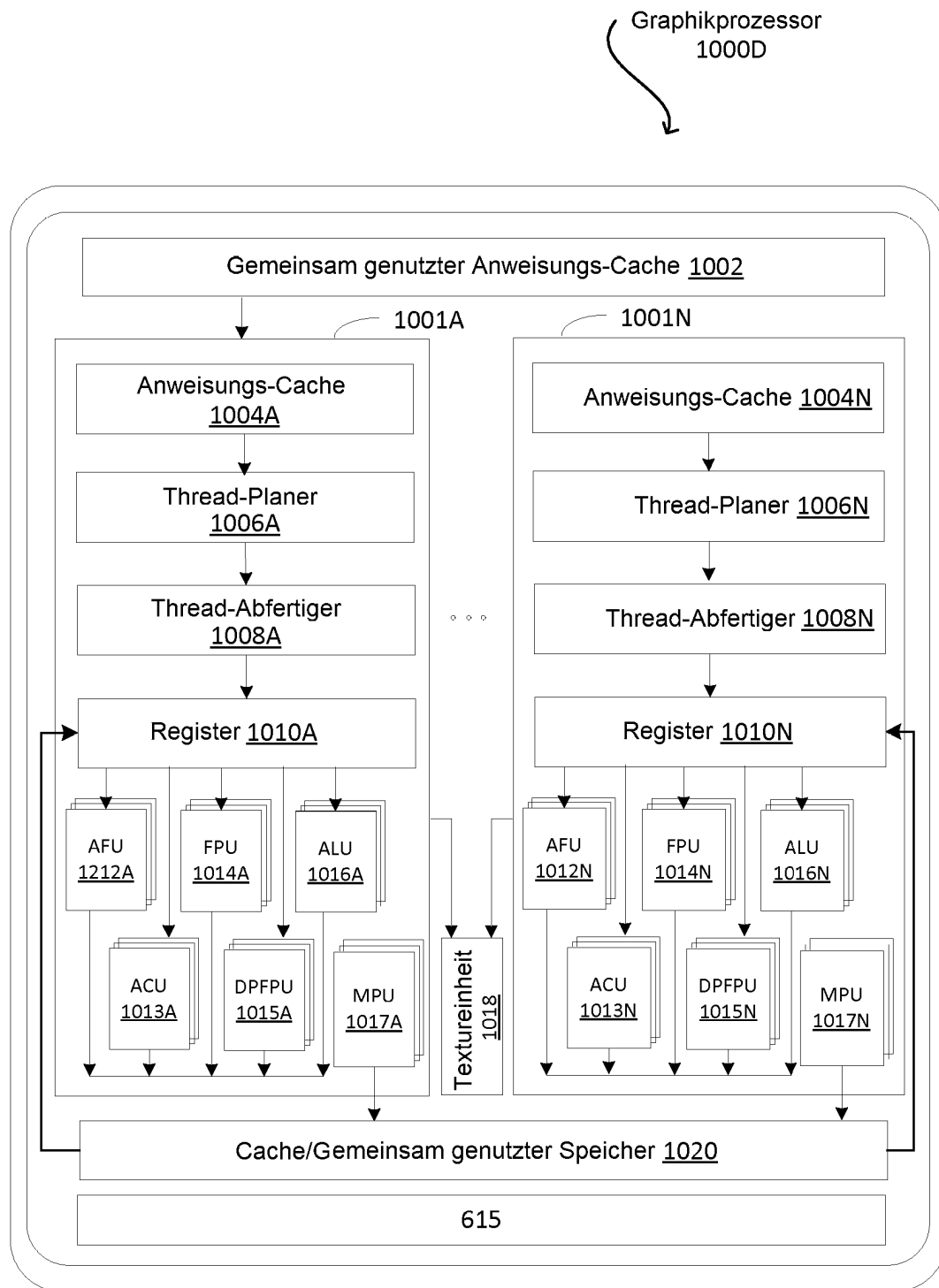


FIG. 10C

**FIG. 10D**

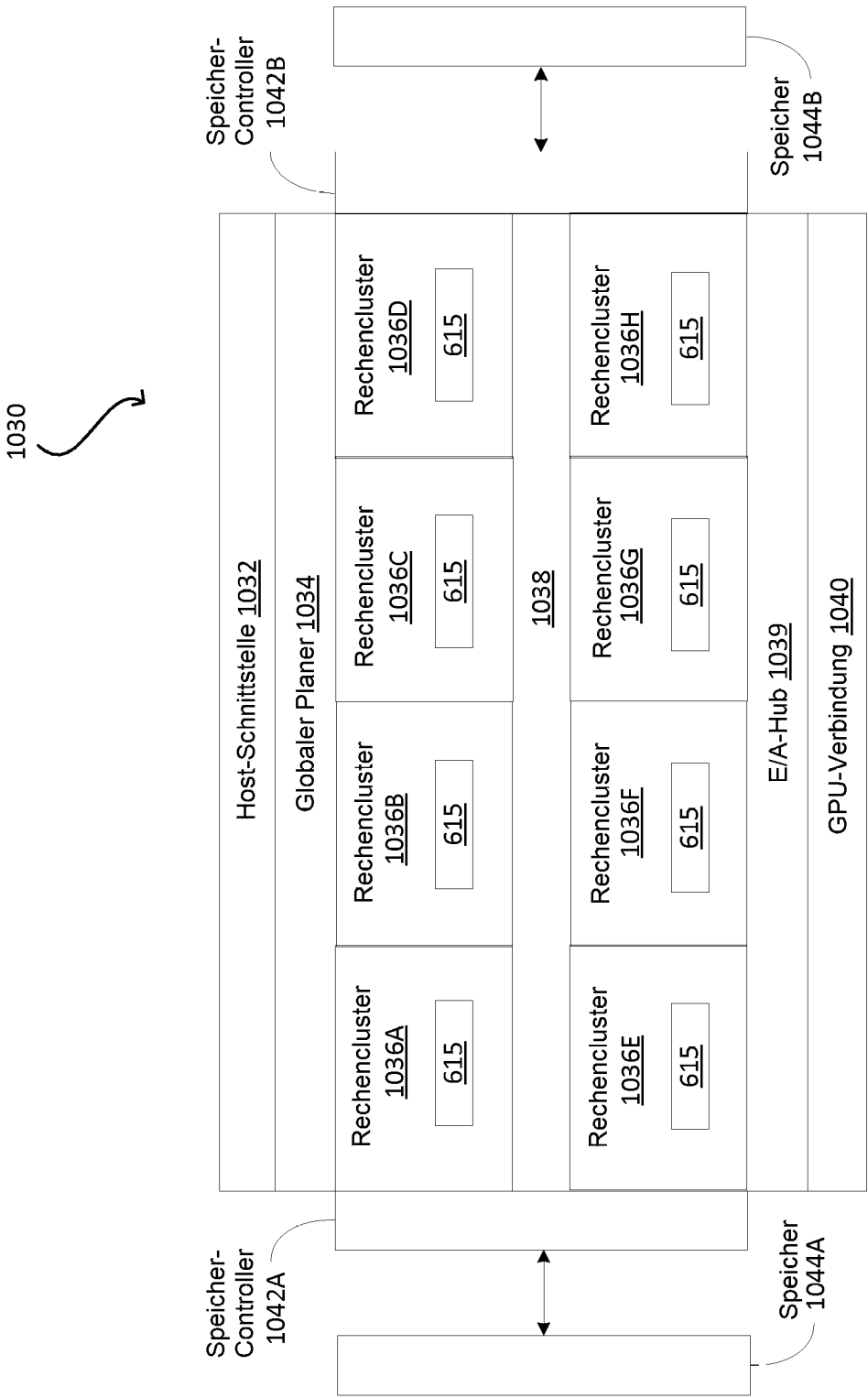


FIG. 10E

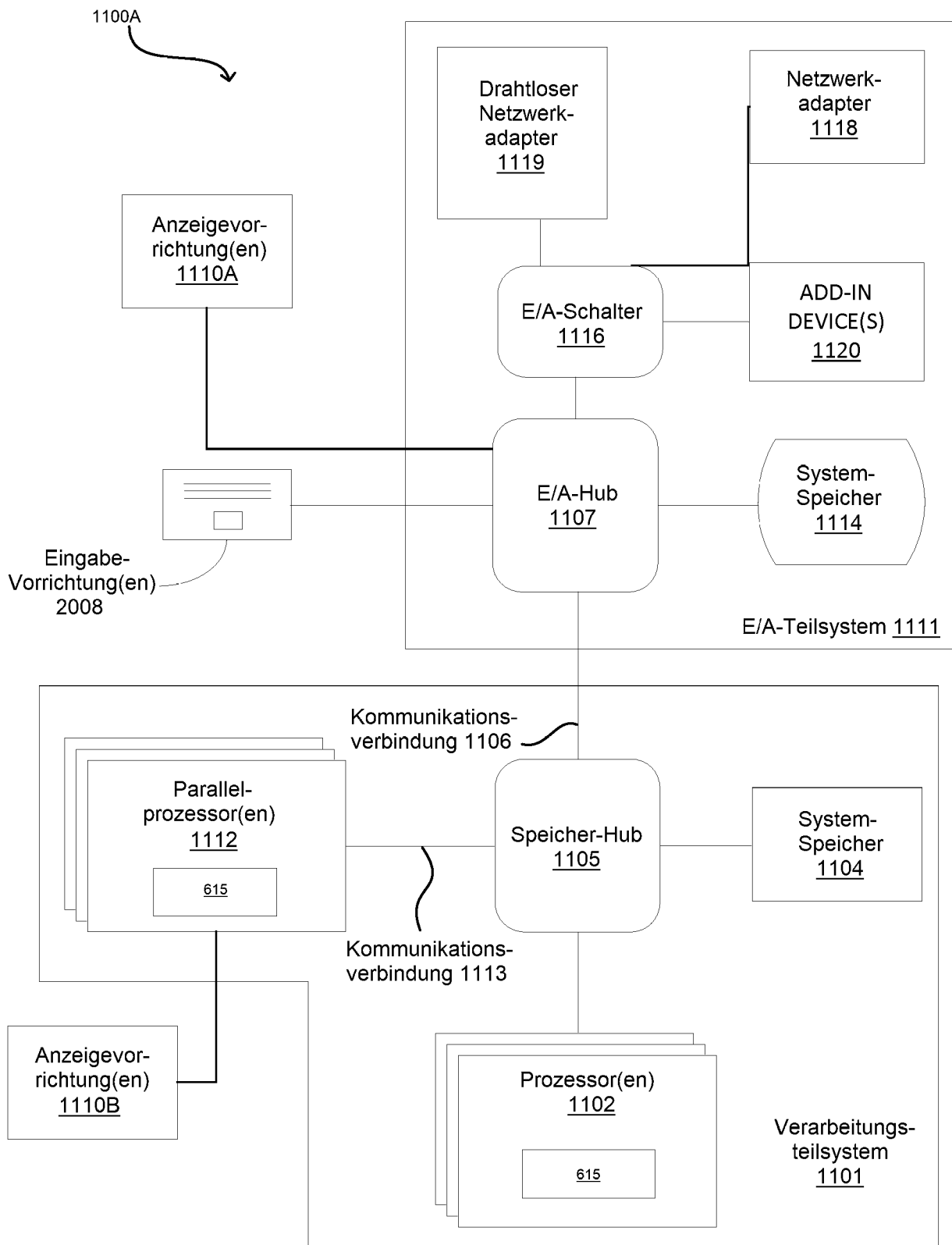


FIG. 11A

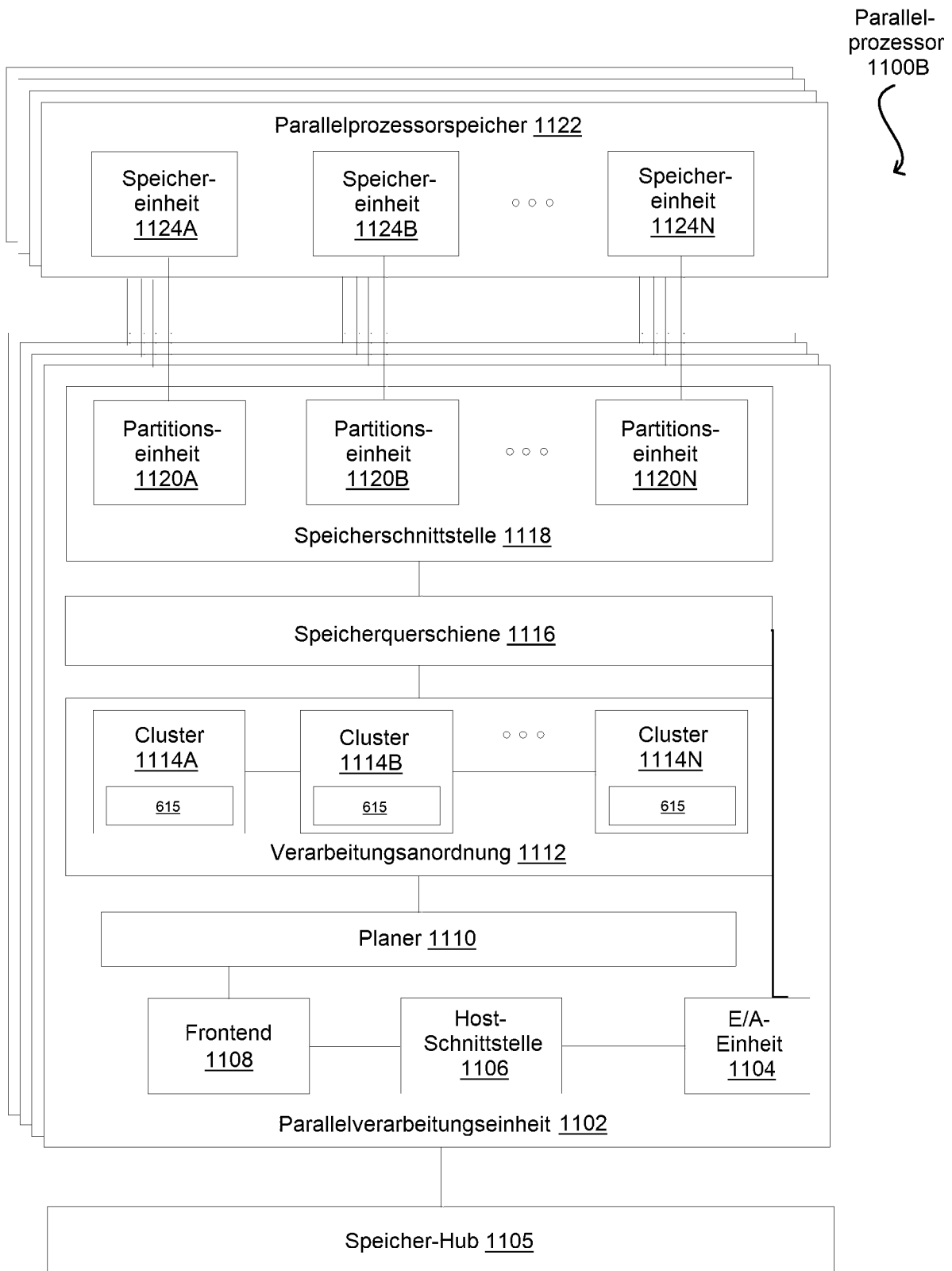


FIG. 11B

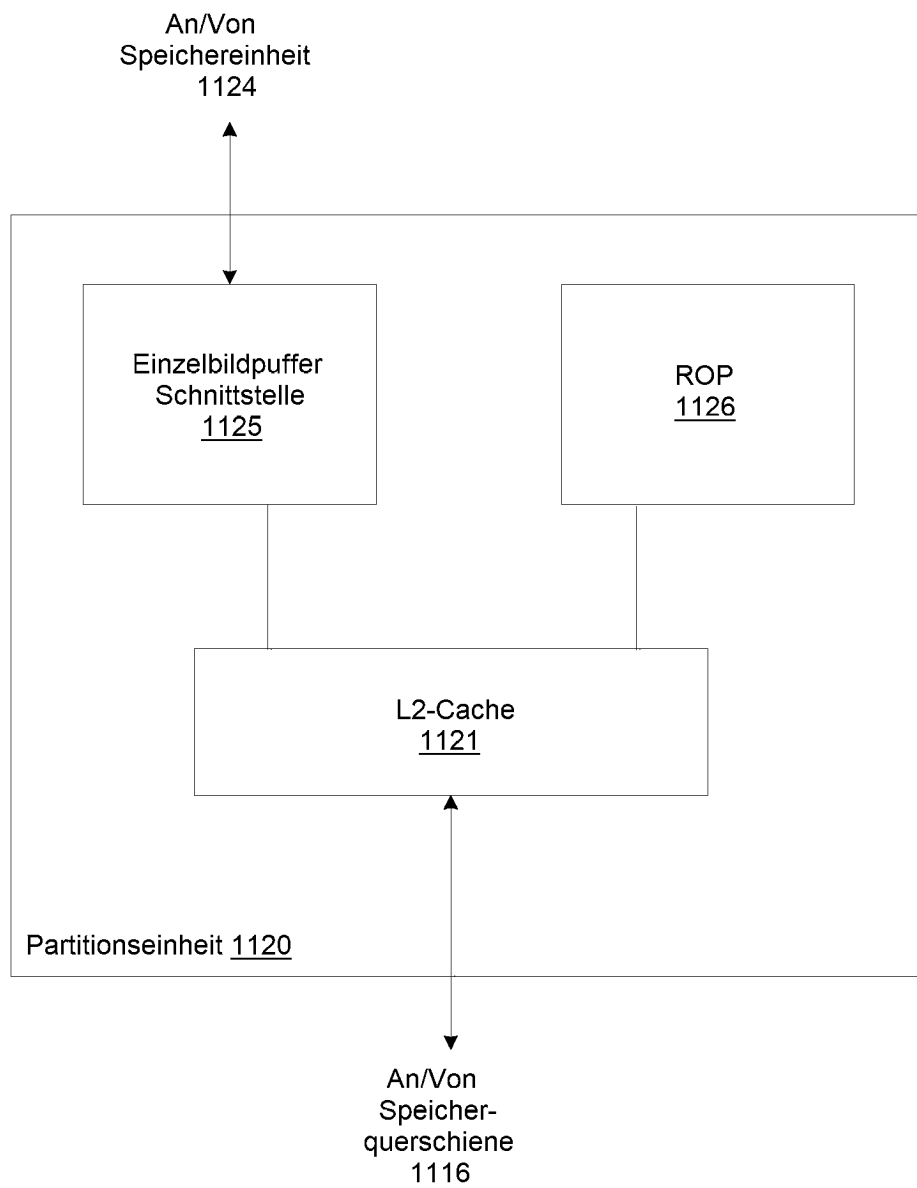


FIG. 11C

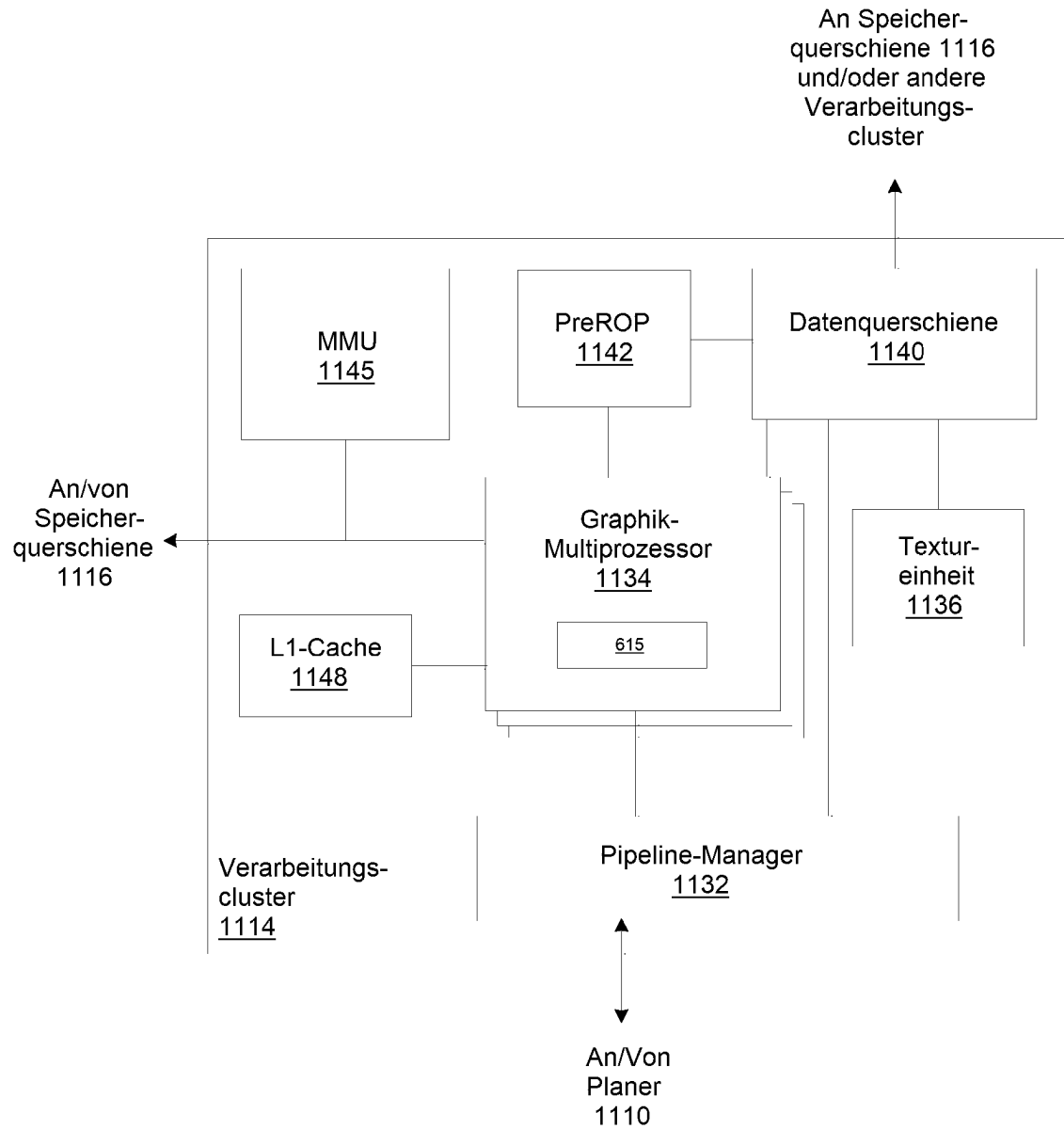


FIG. 11D

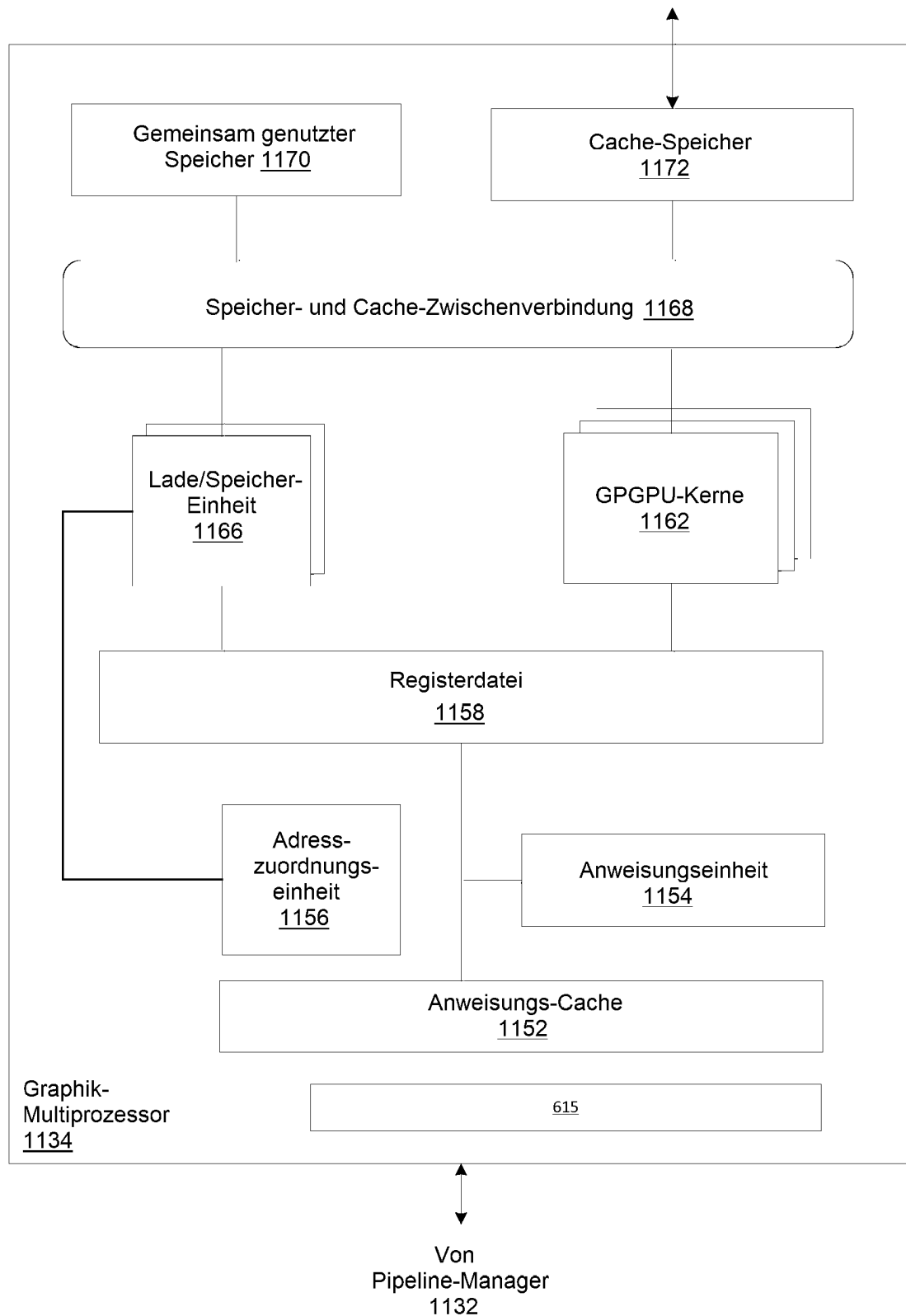
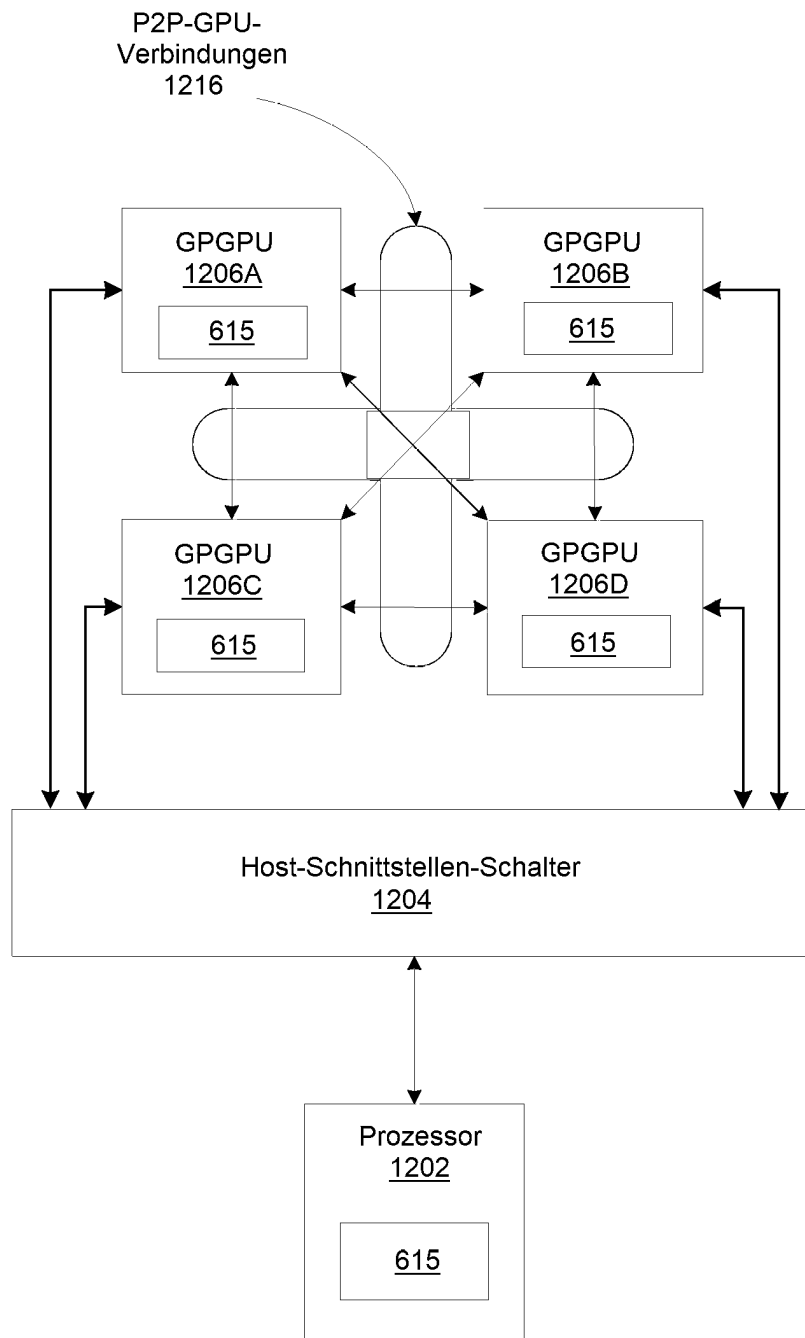


FIG. 11E

1200A**FIG. 12A**

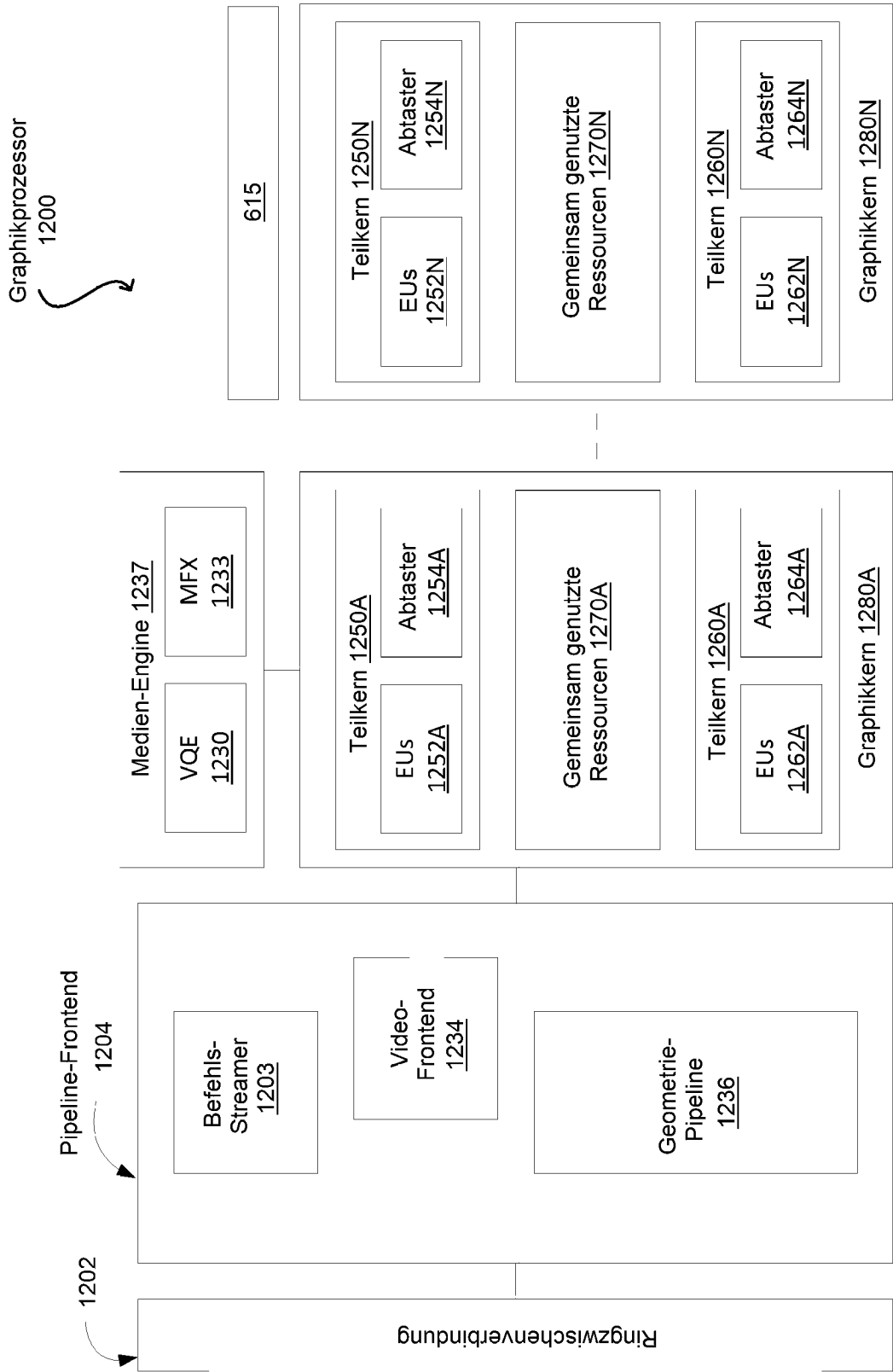


FIG. 12B

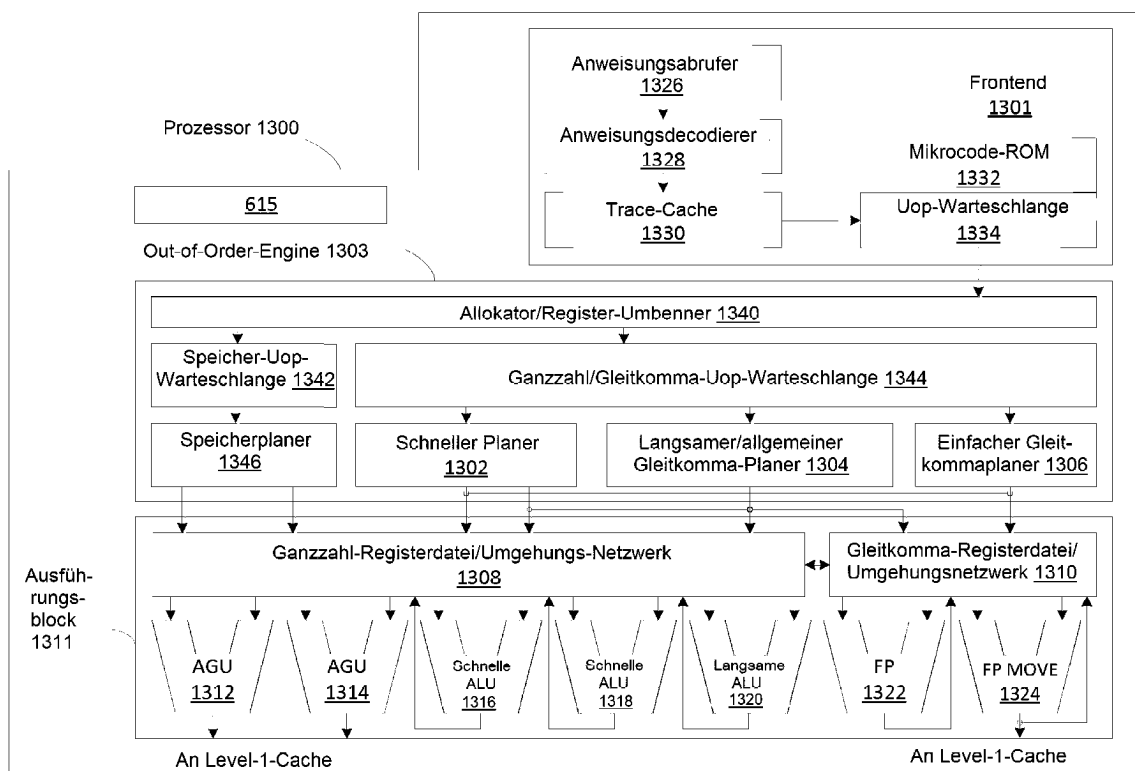


FIG. 13

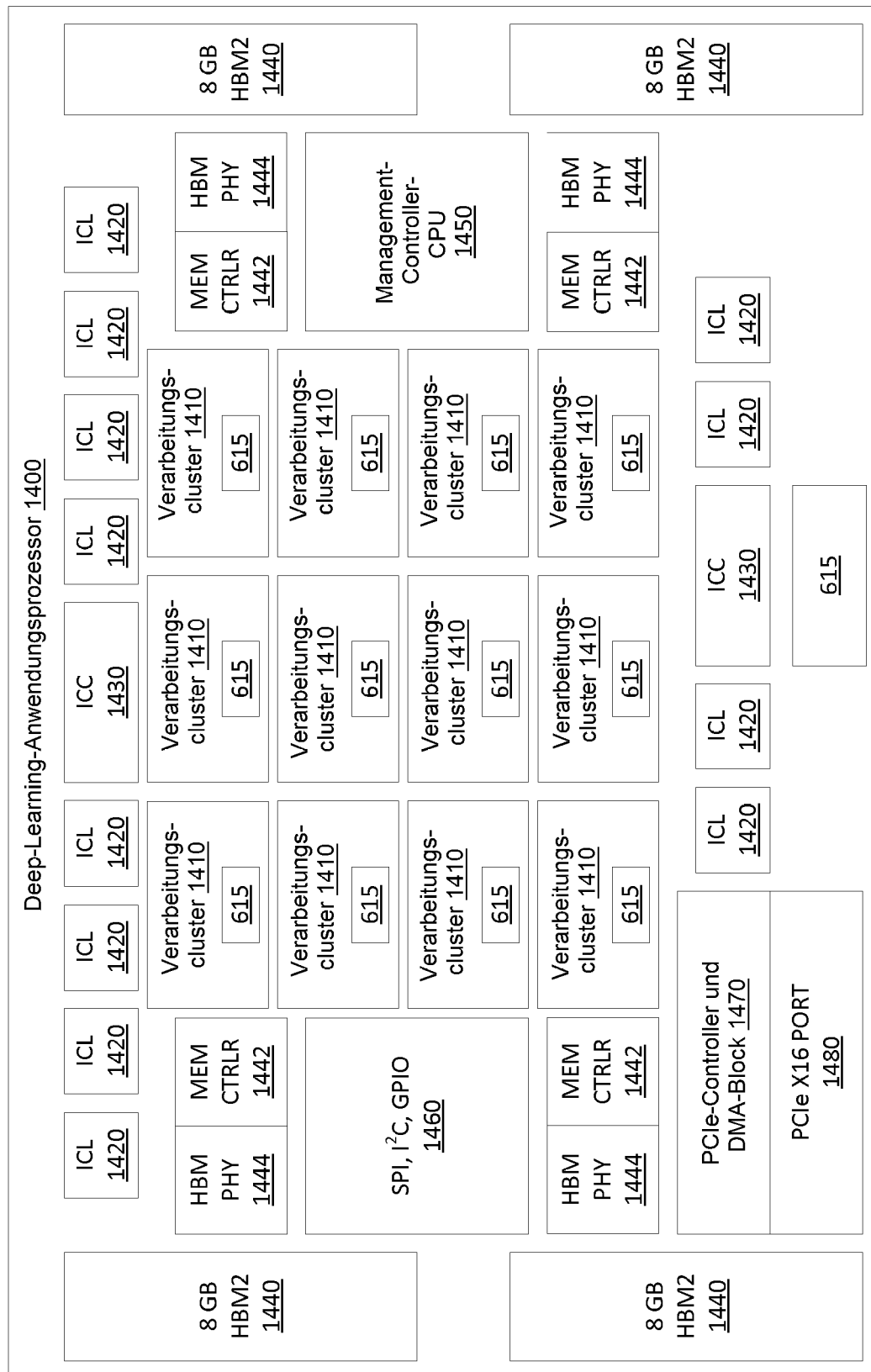
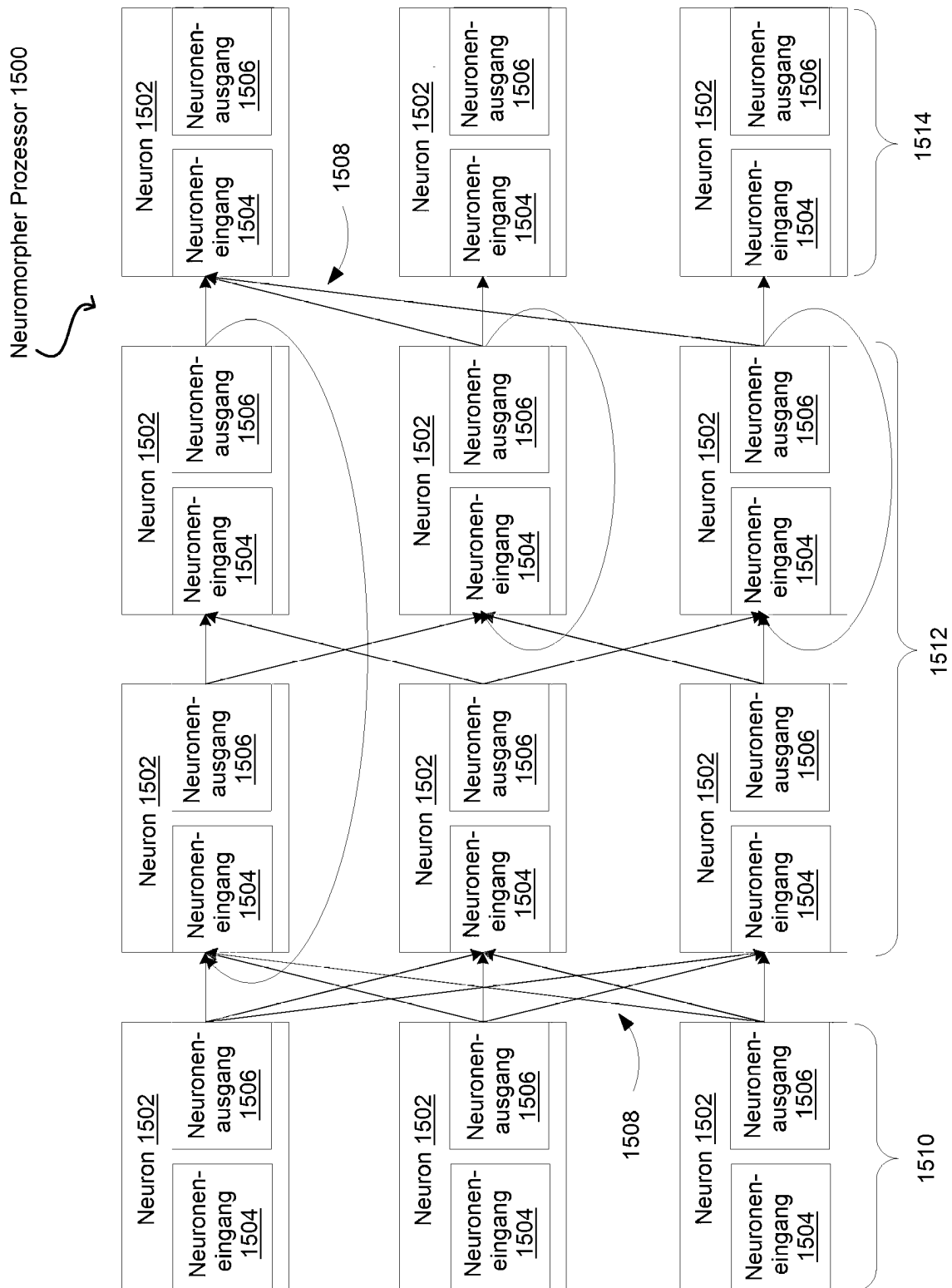


FIG. 14

**FIG. 15**

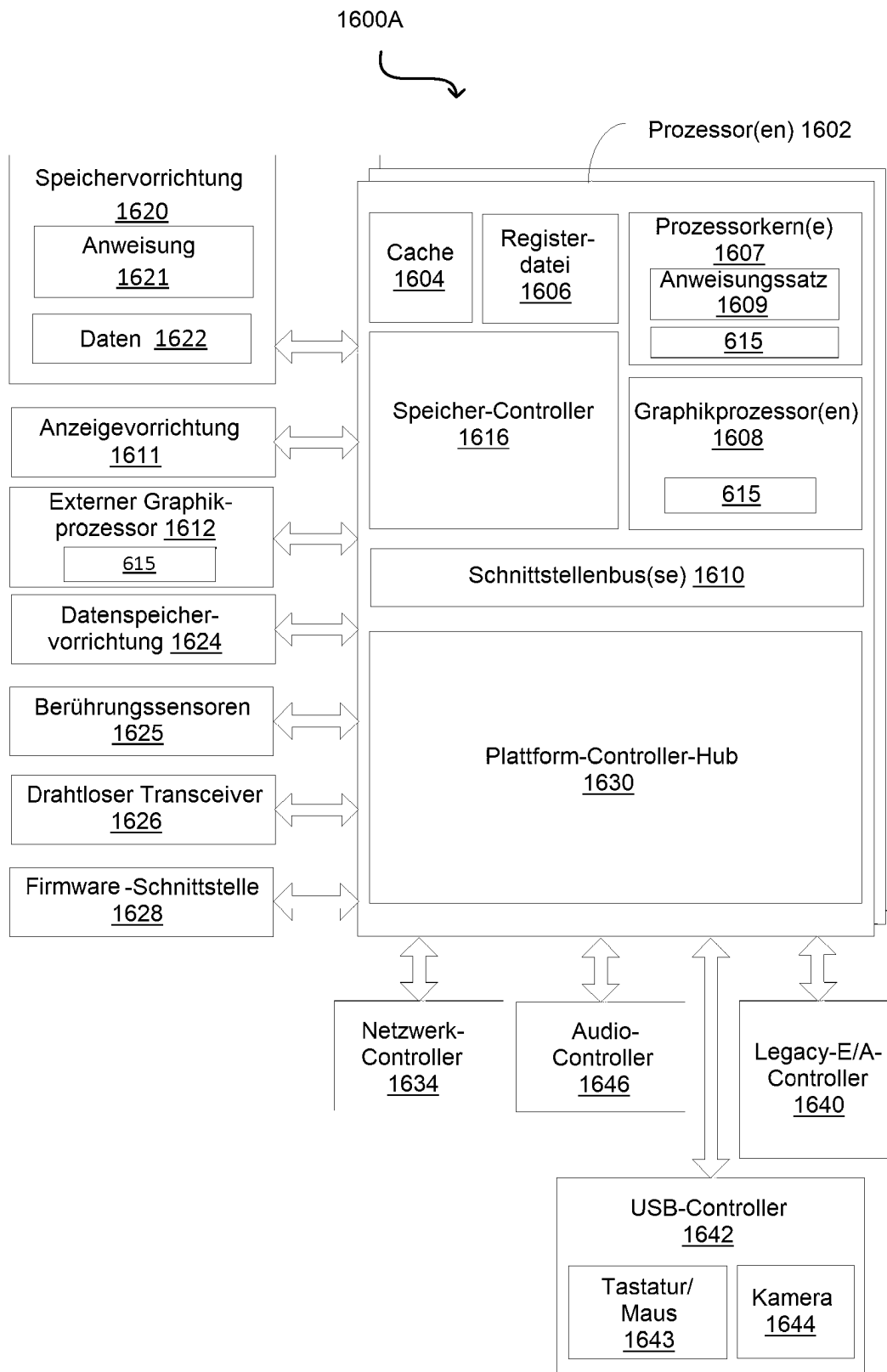


FIG. 16A

Prozessor 1600B

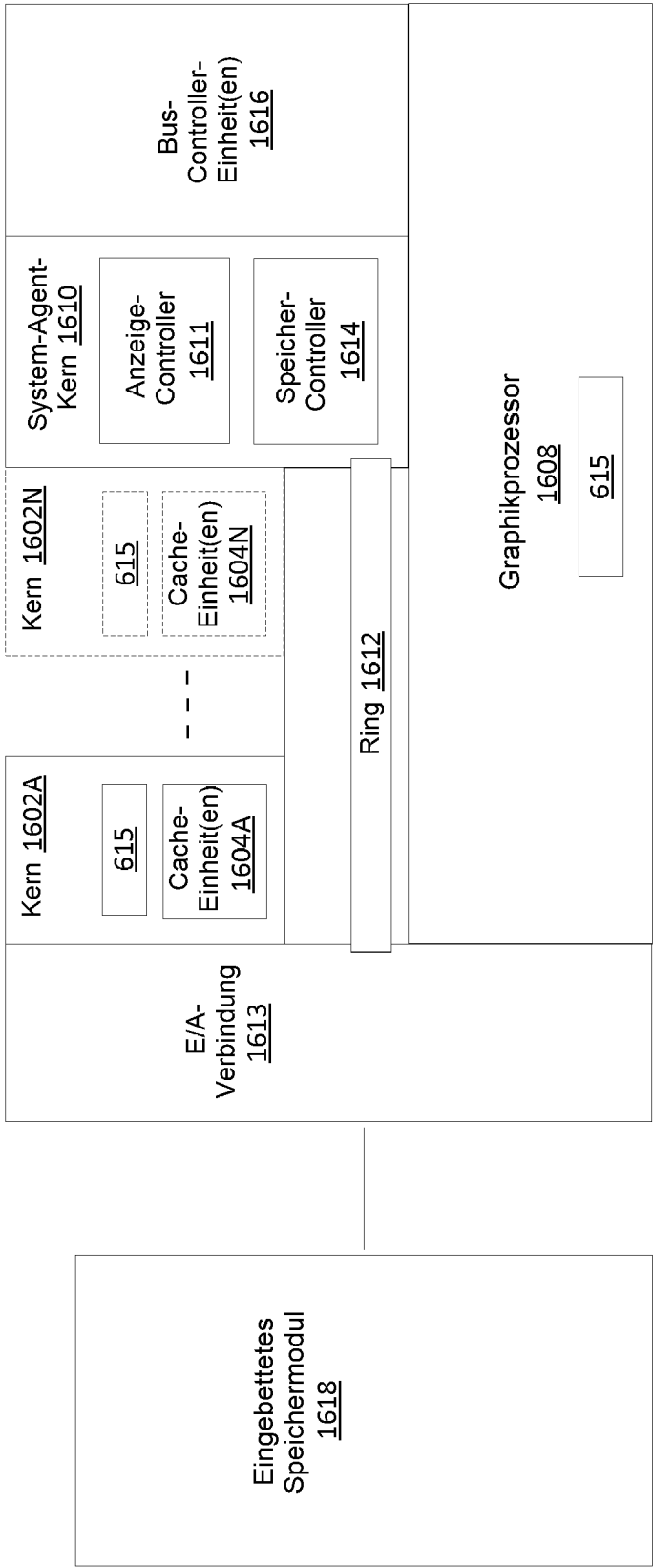


FIG. 16B

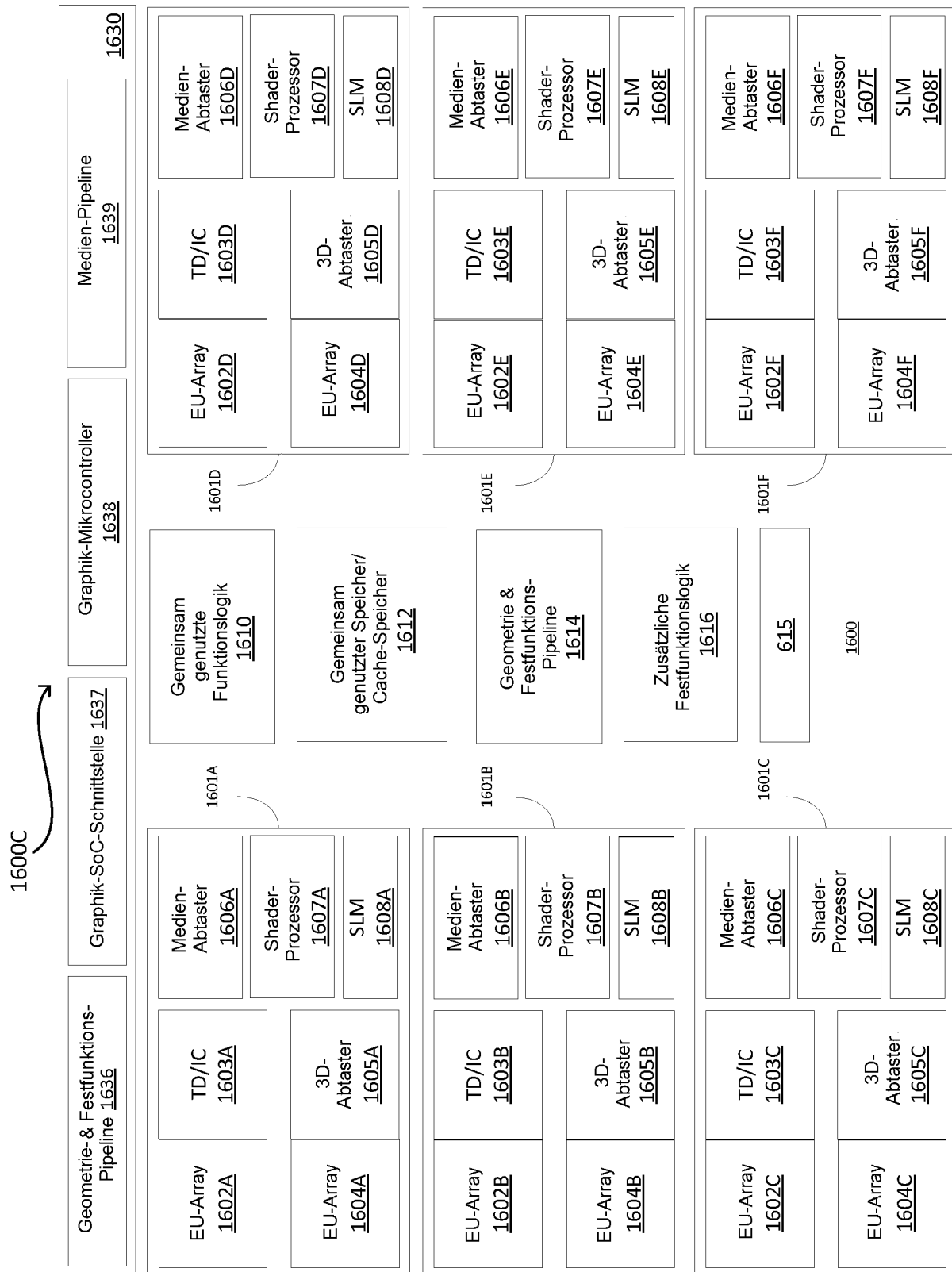


FIG. 16C

Ausführungslogik
1600D

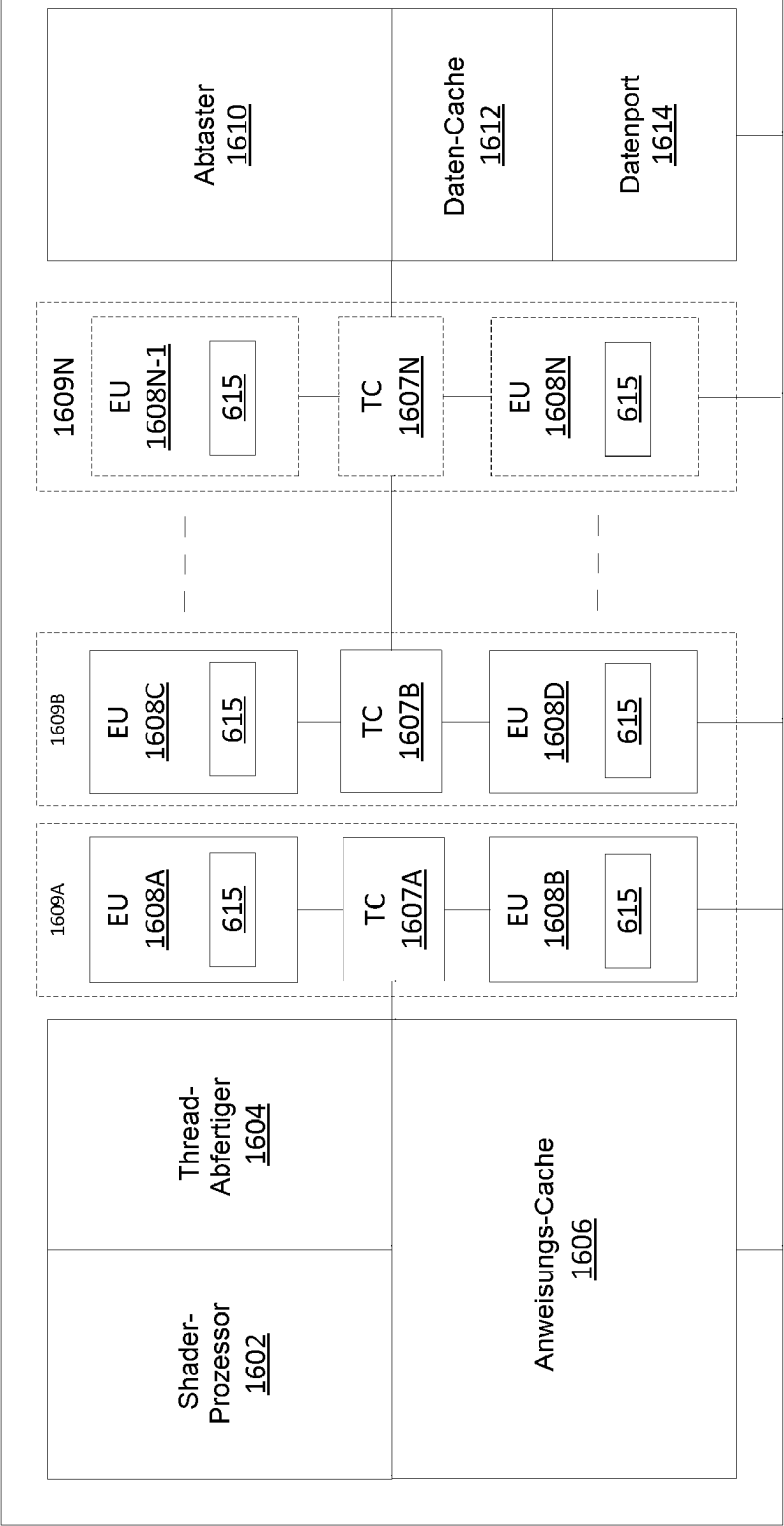


FIG. 16D

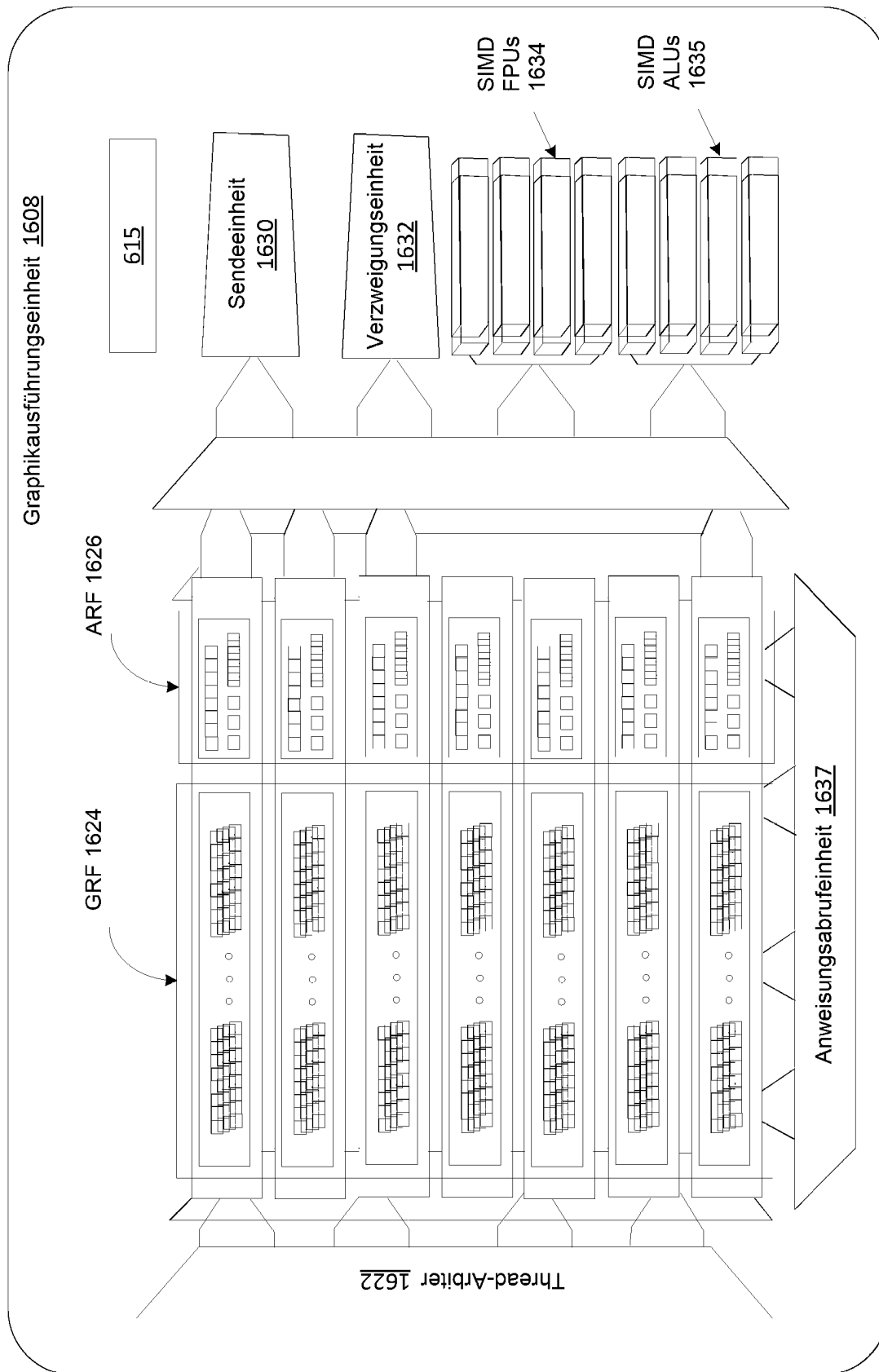
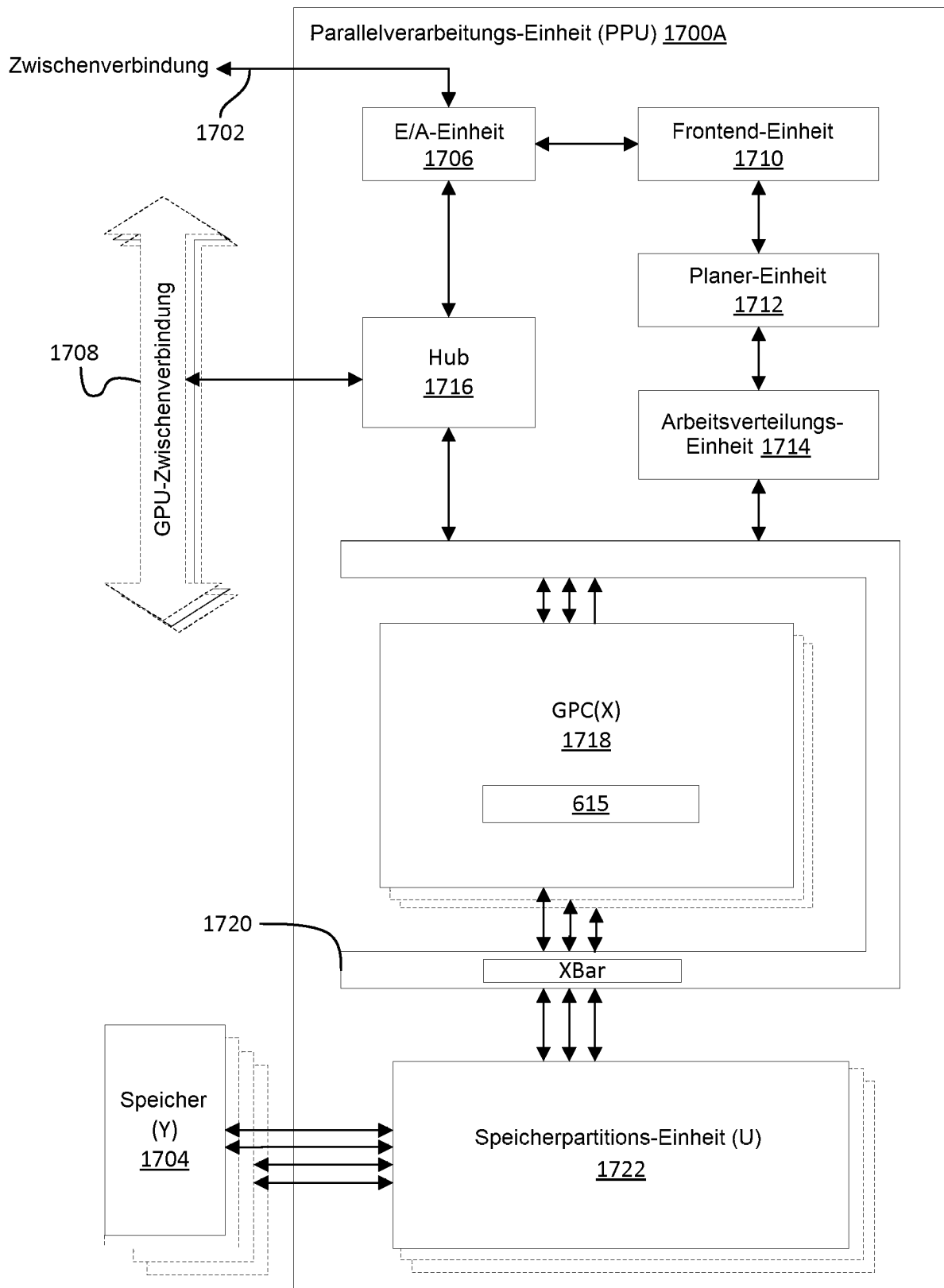


FIG. 16E

**FIG. 17A**

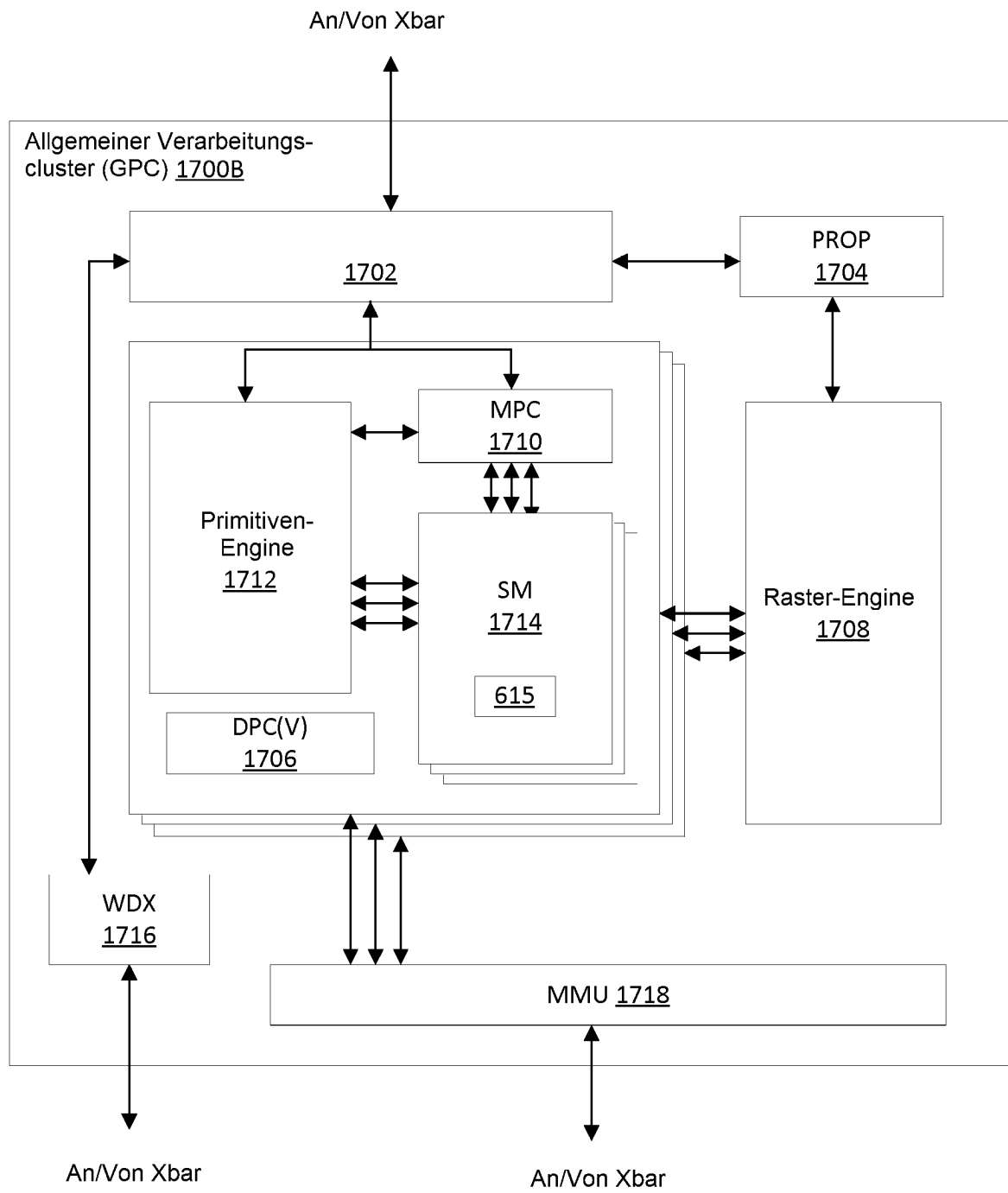


FIG. 17B

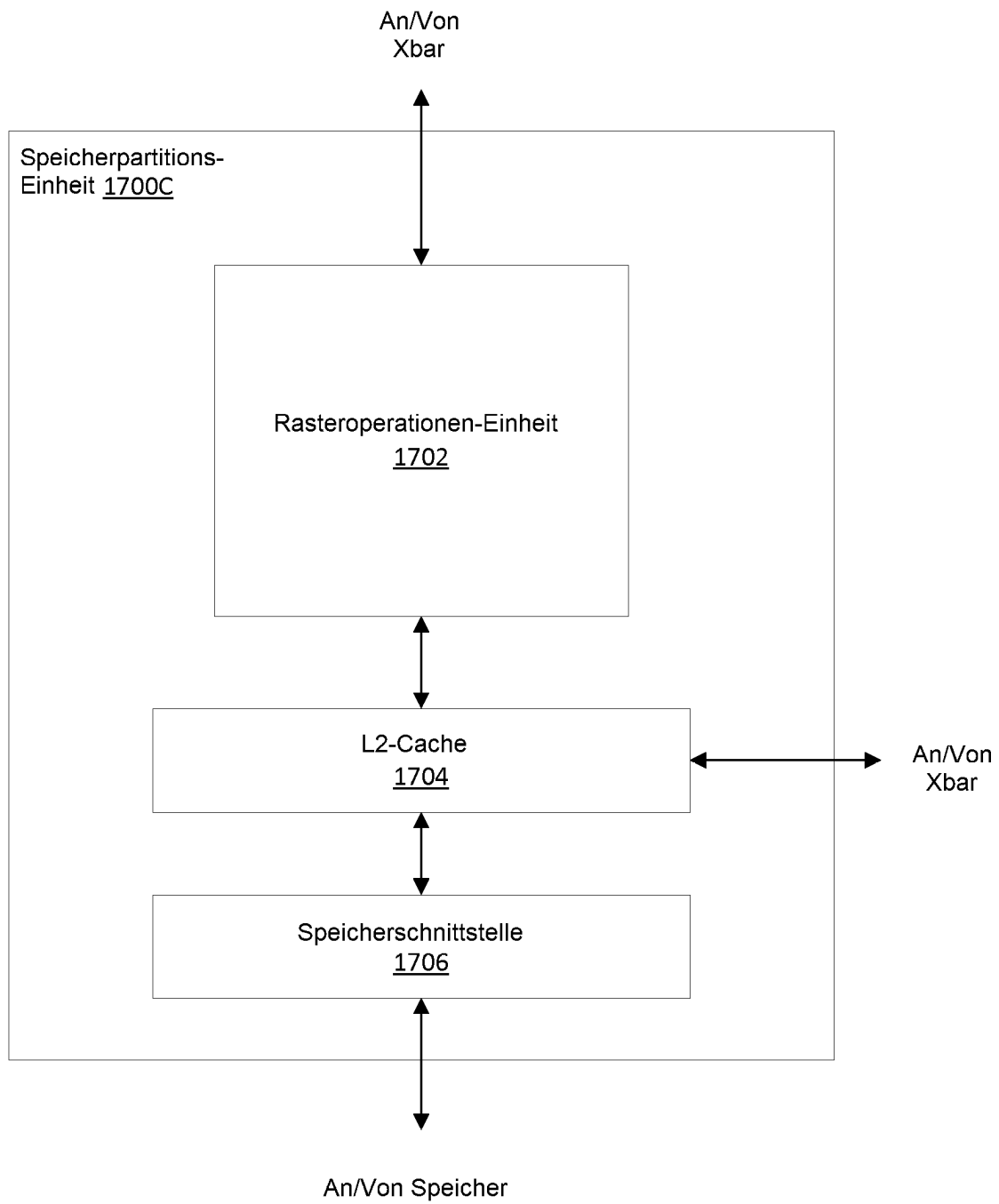


FIG. 17C

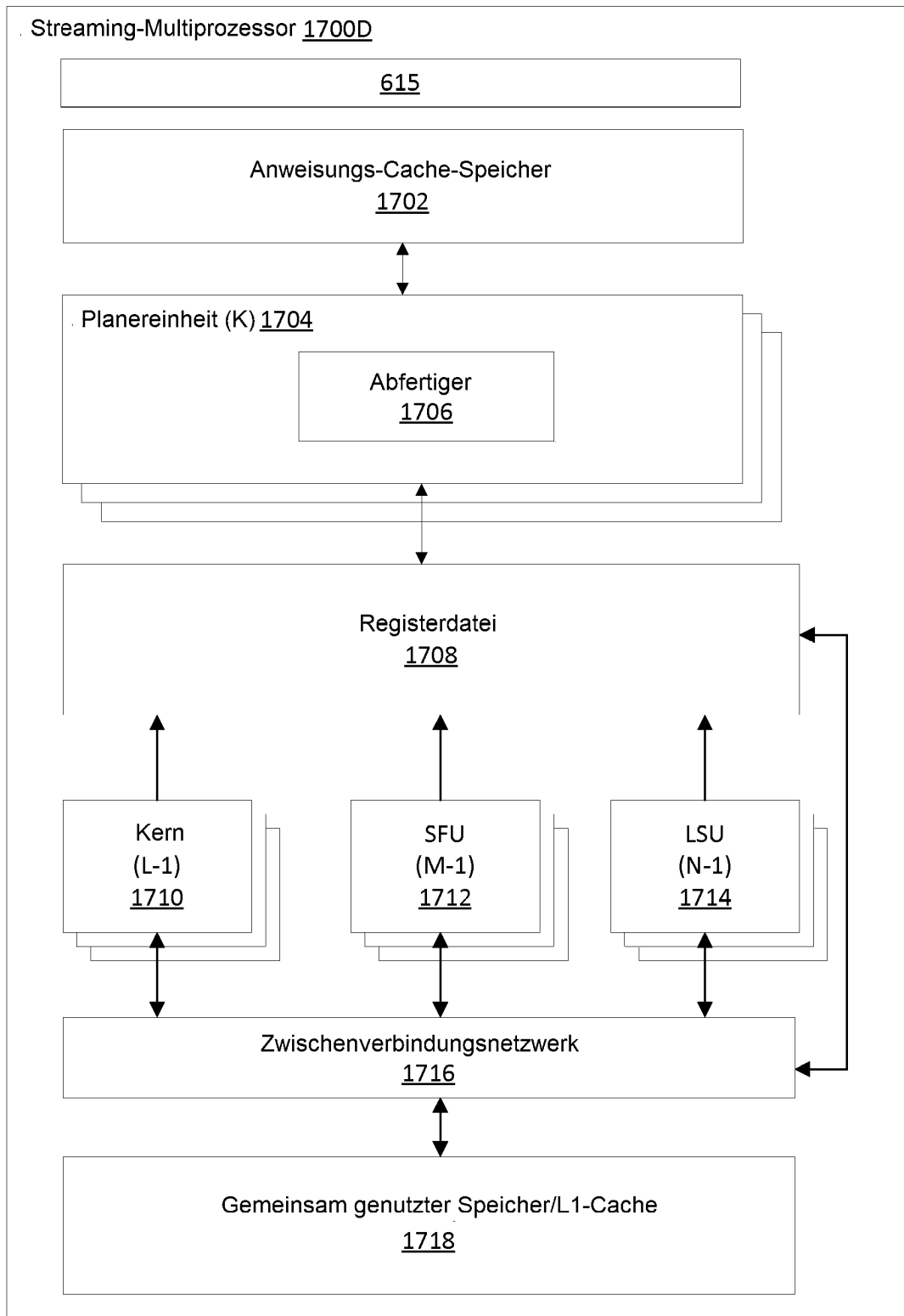


FIG. 17D