



(12) **Offenlegungsschrift**

(21) Aktenzeichen: **10 2014 119 048.3**
(22) Anmeldetag: **18.12.2014**
(43) Offenlegungstag: **18.06.2015**

(51) Int Cl.: **G06F 9/38 (2006.01)**
G06F 15/80 (2006.01)

(30) Unionspriorität:
1322388.8 **18.12.2013** **GB**

(71) Anmelder:
**Imagination Technologies Ltd., Kings Langley,
Hertfordshire, GB**

(74) Vertreter:
Olswang Germany LLP, 80331 München, DE

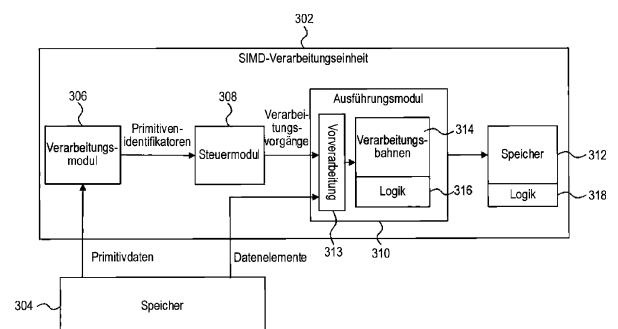
(72) Erfinder:
**Redshaw, Jonathan, c/o Imagination
Technologies L, Kings Langley, Hertfordshire,
GB; Foo, Yoong-Chert, c/o Imagination
Technologie, Kings Langley, Hertfordshire, GB;
Howson, John, c/o Imagination Technologies
Limit, Kings Langley, Hertfordshire, GB**

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Ausführung von Verarbeitungsvorgängen in einer SIMD-Verarbeitungseinheit**

(57) Zusammenfassung: Eine SIMD-Verarbeitungseinheit verarbeitet eine Mehrzahl von Verarbeitungsvorgängen, die jeweils Arbeitselemente bis zu einer vorbestimmten Höchstanzahl umfassen. Die Arbeitselemente eines Verarbeitungsvorgangs sind ausgelegt, um eine gemeinsame Sequenz von Befehlen an entsprechenden Datenelementen auszuführen. Die Datenelemente sind in Blöcken angeordnet, wobei einige der Blöcke wenigstens ein ungültiges Datenelement umfassen. Arbeitselemente, die sich auf ungültige Datenelemente beziehen, sind ungültige Arbeitselemente. Die SIMD-Verarbeitungseinheit umfasst eine Gruppe von Verarbeitungsbahnen, die ausgelegt sind, Befehle von Arbeitselementen eines bestimmten Verarbeitungsvorgangs über eine Mehrzahl von Verarbeitungszyklen auszuführen. Ein Steuermodul stellt basierend auf der Gültigkeit der Arbeitselemente Arbeitselemente zu den Verarbeitungsvorgängen zusammen, sodass ungültige Arbeitselemente des bestimmten Verarbeitungsvorgangs in den Verarbeitungsbahnen vorübergehend aufeinander ausgerichtet sind. Auf diese Weise kann die Anzahl ungenutzter Verarbeitungsschlitze aufgrund ungültiger Arbeitselemente verringert werden.



Beschreibung

Hintergrund

[0001] SIMD-(Single Instruction Multiple Data)Verarbeitungseinheiten sind ausgelegt, dieselben Vorgänge an mehreren Datenelementen gleichzeitig auszuführen. Dies ermöglicht es SIMD-Verarbeitungseinheiten, Datenelemente parallel zu verarbeiten, was effizienter sein kann als ein aufeinanderfolgendes Verarbeiten jedes Datenelements. SIMD-Verarbeitungseinheiten sind besonders nützlich, wenn derselbe Befehl an einer großen Anzahl von Datenelementen auszuführen ist, was in Multimediaanwendungen häufig der Fall ist. Zum Beispiel kann eine Grafikverarbeitungseinheit (GPU) eine SIMD-Verarbeitungseinheit verwenden, um Vorgänge an jedem einer großen Anzahl von Pixeln eines computergenerierten Bilds durchzuführen. Auf ähnliche Weise kann eine Bildverarbeitungseinheit zum Verarbeiten von Bilddateien (die z. B. durch eine Kamera aufgenommen wurden), die zum Beispiel Teil einer Kameraverarbeitungspipeline sein können, eine SIMD-Verarbeitungseinheit verwenden, um Vorgänge auf jedem einer großen Anzahl von Pixeln eines Bilds durchzuführen.

[0002] Ein Verarbeitungsvorgang kann aus einer Mehrzahl von „Arbeitselementen“ bestehen, wobei die Arbeitselemente eines Verarbeitungsvorgangs ausgeführt werden können, um dadurch eine gemeinsame Sequenz von Befehlen an entsprechenden Datenelementen auszuführen. Das heißt, ein Arbeitselement kann eine Sequenz von Befehlen umfassen, die an bzw. auf einem Datenelement auszuführen sind, wobei eine Gruppe von Arbeitselementen, die dieselbe Sequenz von auf entsprechenden Datenelementen auszuführenden Befehlen umfassen, in einem Verarbeitungsvorgang (engl. task) gruppiert sind. Jeder Verarbeitungsvorgang kann Arbeitselemente bis zu einer vorbestimmten Höchstanzahl umfassen. Die Höchstanzahl von Arbeitselementen, die in einem Verarbeitungsvorgang enthalten sein können, können in verschiedenen Systemen variieren, aber **Fig. 1** zeigt einen Verarbeitungsvorgang **100**, der bis zu zweiunddreißig Arbeitselemente **102** umfassen kann. Der Klarheit halber sind nur einige der Arbeitselemente **102** in **Fig. 1** gekennzeichnet. **Fig. 1** zeigt auch einige der zweiunddreißig verschiedenen Positionen von Elementen im Verarbeitungsvorgang (von Position 0 bis Position 31), an der ein Arbeitselement eingefügt sein kann. Die verschiedenen Arbeitselemente in einem Verarbeitungsvorgang können parallel ausgeführt werden, da sie entsprechende Fälle eines auf entsprechenden Datenelementen durchzuführenden Befehls sind. Der Verarbeitungsvorgang **100** ist nicht voll von Arbeitselementen und es heißt daher, dass er „teilweise belegt“ ist. Das bedeutet, dass der Verarbeitungsvorgang **100** weniger als zweiunddreißig Arbeitselemente umfasst, obwohl

er eine Kapazität für zweiunddreißig Arbeitselemente aufweist. Positionen im Verarbeitungsvorgang **100**, die in **Fig. 1** schattiert sind, umfassen ein Arbeitselement, während Positionen im Verarbeitungsvorgang **100**, die in **Fig. 1** nicht schattiert sind, kein Arbeitselement umfassen. Der Verarbeitungsvorgang **100** umfasst daher siebzehn Arbeitselemente zur Ausführung auf der SIMD-Verarbeitungseinheit an den Positionen 0 bis 16 und umfasst keine Arbeitselemente an den Positionen 17 bis 31. Ferner sind die Arbeitselemente **102** an den Positionen 0 bis 6, 9, 11, 15 und 16 gültige Arbeitselemente für die Ausführung durch eine SIMD-Verarbeitungseinheit. Wie weiter unten genauer beschrieben, können jedoch einige Arbeitselemente ungültig sein. In diesem Fall werden sie von der SIMD-Verarbeitungseinheit nicht ausgeführt. Die Arbeitselemente **102** an den Positionen 7, 8, 10 und 12 bis 14 sind im in **Fig. 1** gezeigten Beispiel ungültige Arbeitselemente und sind schraffiert dargestellt.

[0003] Eine SIMD-Verarbeitungseinheit kann eine Mehrzahl von Verarbeitungsbahnen umfassen, die jeweils ausgelegt sind, einen Befehl eines Arbeitselements in jedem einer Mehrzahl von Verarbeitungszyklen auszuführen. **Fig. 2** zeigt die Verarbeitung von Verarbeitungsvorgängen unter Verwendung einer SIMD-Verarbeitungseinheit, die sechzehn Verarbeitungsbahnen, in **Fig. 2** mit **200** gekennzeichnet, umfasst. Die Kombination aus einer Verarbeitungsbahn und einem Verarbeitungszyklus umfasst einen Verarbeitungs-„Schlitz“, in dem ein Befehl eines Arbeitselements verarbeitet werden kann. In diesem Fall sind die Verarbeitungszyklen Taktungszyklen, und **Fig. 2** zeigt vier Taktungszyklen, die als clk 0, clk 1, clk 2 und clk3 bezeichnet sind. Befehle der Arbeitselemente von den ersten sechzehn Positionen eines ersten Verarbeitungsvorgangs (Verarbeitungsvorgang **100**) sind zur Ausführung in den sechzehn Verarbeitungsbahnen im ersten Verarbeitungszyklus (clk 0) geplant; und Befehle der Arbeitselemente der nächsten sechzehn Positionen des Verarbeitungsvorgangs **100** sind zur Ausführung in den sechzehn Verarbeitungsbahnen im zweiten Verarbeitungszyklus (clk 1) geplant. In den nächsten Taktungszyklen (clk 2 und clk 3) sind die Verarbeitungsbahnen zur Ausführung von Arbeitselementen vom nächsten Verarbeitungsvorgang geplant. Ist ein Verarbeitungsvorgang teilweise belegt, werden einige Verarbeitungsschlitze nicht genutzt, d. h. in diesen Verarbeitungsschlitzen werden keine Arbeitselemente ausgeführt. Dies geht aus **Fig. 2** hervor, wo der Verarbeitungsvorgang **100** an den Positionen 17 bis 31 keine Arbeitselemente umfasst, weshalb im zweiten Taktungszyklus (clk 1) ein Befehl von nur einem Arbeitselement (**102**₁₆) ausgeführt wird. Es sind daher im in **Fig. 2** gezeigten Beispiel während des Taktungszyklus clk 1 15 Verarbeitungsbahnen untätig. Falls ein ungültiges Arbeitselement zur Ausführung in einem Verarbeitungsschlitz geplant ist, dann bleibt außerdem auch dieser Verarbeitungsschlitz unge-

nutzt, da ungültige Arbeitselemente nicht verarbeitet werden. Somit sind im in **Fig. 2** gezeigten Beispiel die Verarbeitungsbahnen 7, 8, 10, 12, 13 und 14 während des ersten Verarbeitungszyklus (clk 0) untätig, da die Arbeitselemente **102₇**, **102₈**, **102₁₀**, **102₁₂**, **102₁₃** und **102₁₄** im Verarbeitungsvorgang **100** ungültige Arbeitselemente sind.

[0004] Das in **Fig. 2** gezeigte System hat also aus den genannten Gründen ungenutzte Verarbeitungsschlitze zur Folge.

[0005] Moderne Grafikschnittstellen zur Anwendungsprogrammierung (API) wie OpenGL und Direct X von Microsoft definieren Befehle, die mit Pixeln in einem Pixelquad von 2x2 arbeiten. Es ist z. B. häufig erforderlich, mittels eines „Gradienten“-Vorgangs die Änderungsrate einer variierenden Menge von verschiedenen Pixeln zu bestimmen. Die Anwesenheit dieser Befehle verhindert die Entfernung von „leeren“ Pixelschlitzen (die ungültigen Arbeitselementen entsprechen), wenn Arbeitselemente in Verarbeitungsvorgängen verpackt werden.

Kurzfassung

[0006] Diese Kurzfassung wird bereitgestellt, um eine Auswahl von Konzepten in vereinfachter Form vorzustellen, die weiter unten in der detaillierten Beschreibung näher beschrieben werden. Diese Kurzfassung soll keine Schlüsselmerkmale oder wesentlichen Merkmale des beanspruchten Gegenstands ermitteln, noch den Schutzbereich des beanspruchten Gegenstands beschränken.

[0007] Bereitgestellt ist eine SIMD-(Single Instruction Multiple Data)Verarbeitungseinheit, die ausgelegt ist, eine Mehrzahl von Verarbeitungsvorgängen zu verarbeiten, die jeweils Arbeitselemente bis zu einer vorbestimmten Höchstanzahl umfassen, wobei die Arbeitselemente eines Verarbeitungsvorgangs angeordnet bzw. ausgelegt sind, um eine gemeinsame Sequenz von Befehlen an bzw. auf entsprechenden Datenelementen auszuführen, wobei die Datenelemente in Blöcken von Datenelementen angeordnet sind, wobei einige der Blöcke wenigstens ein ungültiges Datenelement umfassen und wobei Arbeitselemente, die sich auf ungültige Datenelemente beziehen, ungültige Arbeitselemente sind, wobei die SIMD-Verarbeitungseinheit umfasst: eine Gruppe von Verarbeitungsbahnen, die ausgelegt sind, Befehle von Arbeitselementen eines bestimmten Verarbeitungsvorgangs über eine Mehrzahl von Verarbeitungszyklen auszuführen; und ein Steuermodul, das ausgelegt ist, basierend auf der Gültigkeit der Arbeitselemente die Arbeitselemente zu den Verarbeitungsvorgängen zusammenzustellen, sodass ungültige Arbeitselemente des bestimmten Verarbeitungsvorgangs über die Gruppe von Verarbeitungsbah-

nen hinweg vorübergehend aufeinander ausgerichtet sind.

[0008] Bereitgestellt ist eine SIMD-(Single Instruction Multiple Data-)Verarbeitungseinheit, die ausgelegt ist, eine Mehrzahl von Verarbeitungsvorgängen zu verarbeiten, die jeweils Arbeitselemente bis zu einer vorbestimmten Höchstanzahl umfassen, wobei einige der Verarbeitungsvorgänge weniger als die vorbestimmte Höchstanzahl an Arbeitselementen umfassen, und wobei die Arbeitselemente eines Verarbeitungsvorgangs ausgelegt sind, um eine gemeinsame Sequenz von Befehlen an entsprechenden Datenelementen auszuführen, wobei die SIMD-Verarbeitungseinheit umfasst: eine Mehrzahl von parallelen Gruppen von Verarbeitungsbahnen, wobei jede Gruppe ausgelegt ist, Befehle von Arbeitselementen eines entsprechenden Verarbeitungsvorgangs über eine Mehrzahl von Verarbeitungszyklen auszuführen; und Logik, die mit den Gruppen von Verarbeitungsbahnen gekoppelt ist, die ausgelegt ist, eine bestimmte Gruppe von Verarbeitungsbahnen zu veranlassen, einen bestimmten Verarbeitungszyklus zu überspringen, unabhängig von den anderen Gruppen von Verarbeitungsbahnen, falls im bestimmten Verarbeitungszyklus in den Verarbeitungsbahnen der bestimmten Gruppe keine Arbeitselemente für die Ausführung geplant sind.

[0009] Bereitgestellt ist ein Verfahren zur Verwendung einer SIMD-(Single Instruction Multiple Data) Verarbeitungseinheit, um eine Mehrzahl von Verarbeitungsvorgängen zu verarbeiten, die jeweils Arbeitselemente bis zu einer vorbestimmten Höchstanzahl umfassen, wobei die Arbeitselemente eines Verarbeitungsvorgangs ausgelegt sind, eine gemeinsame Sequenz von Befehlen an entsprechenden Datenelementen auszuführen, wobei die Datenelemente in Blöcken von Datenelementen angeordnet sind, wobei einige der Blöcke wenigstens ein ungültiges Datenelement umfassen, wobei Arbeitselemente, die sich auf ungültige Datenelemente beziehen, ungültige Arbeitselemente sind und wobei die SIMD-Verarbeitungseinheit eine Gruppe von Verarbeitungsbahnen umfasst, die ausgelegt sind, Befehle von Arbeitselementen eines bestimmten Verarbeitungsvorgangs über eine Mehrzahl von Verarbeitungszyklen auszuführen, wobei das Verfahren umfasst: Zusammenstellen der Arbeitselemente zu den Verarbeitungsvorgängen basierend auf der Gültigkeit der Arbeitselemente, sodass ungültige Arbeitselemente des bestimmten Verarbeitungsvorgangs über die Gruppe von Verarbeitungsbahnen hinweg vorübergehend aufeinander ausgerichtet bzw. zusammengefasst sind; und Ausführen von Befehlen von Arbeitselementen des bestimmten Verarbeitungsvorgangs unter Verwendung der Gruppe von Verarbeitungsbahnen.

[0010] Bereitgestellt ist ein Verfahren zur Verwendung einer SIMD-(Single Instruction Multiple Data) Verarbeitungseinheit, um eine Mehrzahl von Verarbeitungsvorgängen zu verarbeiten, die jeweils Arbeitselemente bis zu einer vorbestimmten Höchstanzahl umfassen, wobei einige der Verarbeitungsvorgänge weniger als die vorbestimmte Höchstanzahl von Arbeitselementen umfassen, wobei die Arbeitselemente eines Verarbeitungsvorgangs ausgelegt sind, eine gemeinsame Sequenz von Befehlen an entsprechenden Datenelementen auszuführen, wobei die SIMD-Verarbeitungseinheit eine Mehrzahl von parallelen Gruppen von Verarbeitungsbahnen umfasst, wobei das Verfahren umfasst: in jeder Gruppe von Verarbeitungsbahnen Ausführen von Befehlen von Arbeitselementen eines entsprechenden Verarbeitungsvorgangs in einer Mehrzahl von Verarbeitungszyklen; und Veranlassen einer bestimmten Gruppe von Verarbeitungsbahnen, einen bestimmten Verarbeitungszyklus zu überspringen, unabhängig von den anderen Gruppen von Verarbeitungsbahnen, falls keine Arbeitselemente für die Ausführung in einer der Verarbeitungsbahnen der bestimmten Gruppe im bestimmten Verarbeitungszyklus geplant sind.

[0011] Es kann ein computerlesbares Speichermedium zum Erzeugen einer Single Instruction Multiple Data-Verarbeitungseinheit gemäß einem der hier beschriebenen Beispiele bereitgestellt sein. Ferner kann ein computerlesbares Speichermedium bereitgestellt sein, auf dem computerlesbare Codes zum Erzeugen einer Single Instruction Multiple Data-Verarbeitungseinheit gemäß einem der hier beschriebenen Beispiele codiert sind.

[0012] Die oben genannten Merkmale können nach Bedarf, wie für Fachleute offensichtlich wäre, miteinander sowie mit einem der Aspekte der hier beschriebenen Beispiele kombiniert werden.

Kurzbeschreibung der Zeichnungen

[0013] Es sollen nun in Bezug auf die beigefügten Zeichnungen Beispiele näher beschrieben werden, wobei

[0014] Fig. 1 einen Verarbeitungsvorgang zeigt, der bis zu zweiunddreißig Arbeitselemente umfassen kann;

[0015] Fig. 2 die Verarbeitung von Verarbeitungsvorgängen unter Verwendung einer SIMD-Verarbeitungseinheit nach dem Stand der Technik zeigt;

[0016] Fig. 3 eine schematische Darstellung einer SIMD-Verarbeitungseinheit ist;

[0017] Fig. 4 Primitive, die Pixel eines Bilds überlagern, zeigt;

[0018] Fig. 5 ein Flussdiagramm eines ersten Verfahrens der Verwendung einer SIMD-Verarbeitungseinheit zum Verarbeiten von Verarbeitungsvorgängen ist;

[0019] Fig. 6 die Verarbeitung von Verarbeitungsvorgängen in einer Gruppe von Verarbeitungsbahnen der SIMD-Verarbeitungseinheit in einem ersten Beispiel zeigt;

[0020] Fig. 7 ein Umordnen von Blöcken von Arbeitselementen in einem ersten Beispiel zeigt;

[0021] Fig. 8 ein Umordnen von Blöcken von Arbeitselementen in einem zweiten Beispiel zeigt;

[0022] Fig. 9 die Verarbeitung von Verarbeitungsvorgängen in einer Gruppe von Verarbeitungsbahnen der SIMD-Verarbeitungseinheit in einem zweiten Beispiel zeigt; und

[0023] Fig. 10 ein Flussdiagramm eines zweiten Verfahrens der Verwendung einer SIMD-Verarbeitungseinheit zum Verarbeiten von Verarbeitungsvorgängen ist.

[0024] Die beigefügten Zeichnungen zeigen verschiedene Beispiele. Fachleute werden erkennen, dass die gezeigten Elementabgrenzungen (z. B. Kästen, Gruppen von Kästen oder sonstige Formen) in den Zeichnungen ein Beispiel von Abgrenzungen zeigen. Es ist möglich, dass in einigen Beispielen ein Element als mehrere Elemente dargestellt ist oder dass mehrere Elemente als ein Element gekennzeichnet sind. In den Figuren werden gemeinsame Bezugszeichen verwendet, um, wenn angebracht, ähnliche Merkmale anzuzeigen.

Detaillierte Beschreibung

[0025] Es sollen nun auf rein beispielhafte Weise Ausführungsformen beschrieben werden. Die hier beschriebenen Beispiele stellen eine SIMD-(Single Instruction Multiple Data)Verarbeitungseinheit bereit, die ausgelegt ist, die Anzahl ungenutzter Verarbeitungsschlitze aufgrund ungültiger Arbeitselemente in Verarbeitungsvorgängen und/oder aufgrund von teilweise belegten Verarbeitungsvorgängen zu verringern. Das kann erreicht werden, indem ungültige Arbeitselemente in einer Gruppe von Verarbeitungsbahnen, die ausgelegt sind, Arbeitselemente von einem Verarbeitungsvorgang auszuführen, aufeinander ausgerichtet werden. Falls in einem bestimmten Verarbeitungszyklus keine gültigen Arbeitselemente für die Ausführung in den Verarbeitungsbahnen der Gruppe geplant sind, kann der bestimmte Verarbeitungszyklus übersprungen werden, sodass die Verarbeitungsschlitze im bestimmten Verarbeitungszyklus nicht ungenutzt bleiben.

[0026] In einem weiteren Beispiel kann ferner eine Mehrzahl von Gruppen von Verarbeitungsbahnen vorliegen, wobei jede Gruppe von Verarbeitungsbahnen Arbeitselemente eines entsprechenden Verarbeitungsvorgangs in einer Mehrzahl von Verarbeitungszyklen ausführt. In diesem Fall kann eine bestimmte Gruppe von Verarbeitungsbahnen einen bestimmten Verarbeitungszyklus überspringen, unabhängig von den anderen Gruppen von Verarbeitungsbahnen, falls keine Arbeitselemente für die Ausführung in den Verarbeitungsbahnen der bestimmten Gruppe im bestimmten Verarbeitungszyklus geplant sind, ungeachtet dessen, ob im bestimmten Verarbeitungszyklus in einer anderen Gruppe (d. h. Arbeitselemente von einem anderen Verarbeitungsvorgang) Arbeitselemente zur Ausführung geplant sind.

[0027] Fig. 3 zeigt ein beispielhaftes System, das eine SIMD-Verarbeitungseinheit **302** und einen Speicher **304** umfasst. In einem hier detailliert beschriebenen Beispiel ist die SIMD-Verarbeitungseinheit **302** in einer GPU implementiert, die Bilddaten verarbeitet, um eine Szene wiederzugeben. Es können zum Beispiel Primitive verwendet werden, um Gegenstände in einer Szene zu definieren, wodurch die Primitive durch Primitivdaten definiert werden, die zum Beispiel die Stelle und Textur von den Primitiven in der Szene bestimmen. Der Speicher **304** kann ein Systemspeicher oder ein auf dem Chip befindlicher Speicher sein. Die SIMD-Verarbeitungseinheit **302** umfasst ein Verarbeitungsmodul **306**, ein Steuermodul **308**, ein Ausführungsmodul **310** und einen Speicher **312**. Das Ausführungsmodul **310** umfasst ein Vorverarbeitungsmodul **313**, einen Satz von Verarbeitungsbahnen **314** und Logik **316**, die an die Verarbeitungsbahnen **314** gekoppelt ist. Der Speicher **312** ist an eine Speicherlogik **318** gekoppelt, die das Speichern von Daten im Speicher **312** steuern kann. Die SIMD-Verarbeitungseinheit **302**, insbesondere das Verarbeitungsmodul **306**, ist ausgelegt, um Primitivdaten vom Speicher **304** zu empfangen, wobei die Primitivdaten Informationen umfassen, die die Positionen von Primitiven definieren, die in einem Bild wiederzugeben sind (z. B. Standortdaten für Scheitelpunkte der Primitive). Ein Ausgang des Verarbeitungsmoduls **306** ist an einen Eingang des Steuermoduls **308** gekoppelt. Das Steuermodul **308** ist ausgelegt, um Verarbeitungsvorgänge von Arbeitselementen zu bilden, die ausgelegt sind, um eine gemeinsame Sequenz von Befehlen auf entsprechenden Datenelementen auszuführen. Ein Ausgang des Steuermoduls **308** ist mit einer Eingabe des Ausführungsmoduls **310** gekoppelt, um Verarbeitungsvorgänge aus dem Steuermodul **308** an das Verarbeitungsmodul **310** auszugeben. Das Ausführungsmodul **310** ist auch ausgelegt, um Datenelemente aus dem Speicher **304** zu empfangen, um die Arbeitselemente auf den entsprechenden Datenelementen auszuführen. Das Vorverarbeitungsmodul **313** ist ausgelegt, um die Arbeitselemente teilweise vorzuverarbeiten, be-

vor sie an die Verarbeitungsbahnen **314** weitergegeben werden. Die Verarbeitungsbahnen **314** des Verarbeitungsmoduls **310** sind ausgelegt, um die Arbeitselemente auszuführen. Das Ausführungsmodul **310** ist an den Speicher **312** gekoppelt und ausgelegt, um verarbeitete Datenelemente, die aus der Ausführung der Verarbeitungsvorgänge resultieren, zur Speicherung im Speicher **312** bereitzustellen. In diesem Beispiel sind die Datenelemente Pixelwerte. In anderen Beispielen könnten die Datenelemente andere Arten von Datenelementen sein, die von einer SIMD-Verarbeitungseinheit zu verarbeiten sind. Zum Beispiel kann die SIMD-Verarbeitungseinheit **302** ausgelegt sein, um Befehle auf Fragmentwerten oder Scheitelwerten auszuführen, wobei die Datenelemente jeweils Fragmentwerte oder Scheitelwerte sein können.

[0028] Fig. 4 zeigt einen Renderingraum (Rasterungsraum) **402**, der von einer GPU verwendet wird, um ein Bild wiederzugeben. Der Renderingraum **402** umfasst eine Mehrzahl von Pixeln, von denen einige in Fig. 4 nummeriert sind (0 bis 35). Wie in Fig. 4 gezeigt, sind die Pixel in Pixelblöcken ausgelegt. Insbesondere sind in diesem Beispiel die Pixel in Pixelquads angeordnet, sodass jeder Block vier der Pixel umfasst, die in einem Block von 2×2 angeordnet sind. Zwei der Pixelquads sind in Fig. 4 als **404₁** und **404₂** bezeichnet. Pixel eines Pixelquads werden von der SIMD-Verarbeitungseinheit **302** insofern zusammen verarbeitet, als dass, falls einer oder mehrere der Pixel in einem Pixelquad zu verarbeiten ist/sind, alle Pixel im Pixelquad zur Verarbeitung von der SIMD-Verarbeitungseinheit **302** eingeplant werden. Fig. 4 zeigt auch zwei Primitive **406** und **408**, die im Renderingraum **402** wiederzugeben sind. Wie in Fig. 4 zu sehen ist, überlagern die Primitive **406** und **408** einige, aber nicht alle der Pixel im Renderingraum **402**. Ein Primitiv kann als ein Pixel überlagernd angesehen werden, falls das Primitiv den Abtastpunkt des Pixels bedeckt, der als in der Mitte des dem Pixel zugewiesenen Bereichs liegend vorausgesetzt wird. Pixel, die ein Primitiv **406** überlagern, können von der SIMD-Verarbeitungseinheit **302** verarbeitet werden, z. B. um eine Texturierung für das Primitiv **406** auf die entsprechenden Pixel anzuwenden. Es ist zu erkennen, dass der Gegenstand **406** einige der Pixel (z. B. Pixel 0, 2, 5 oder 7) nicht überlagert, wie in Fig. 4 gezeigt, aber da der Gegenstand andere Pixel der Pixelquads **404₁** und **404₂** überlagert, werden alle Pixelwerte in den Pixelquads **404₁** und **404₂** an die SIMD-Verarbeitungseinheit **302** zur Verarbeitung gesendet, obwohl für einige der Pixelwerte möglicherweise keine Verarbeitung durchzuführen ist. Die Pixelwerte, für die keine Verarbeitung durchzuführen ist, die aber an die SIMD-Verarbeitungseinheit **302** zur Verarbeitung gesendet werden (z. B. Pixel 0, 2, 5 und 7) können insofern als „ungültig“ betrachtet werden, als ein Befehl (z. B. Anwendung der Texturierung des Primitivs **406**) für diese Pixelwerte nicht durchzuführen ist. Es wird

jedoch angemerkt, dass diese ungültigen Pixel nicht verworfen werden könnten, da einige von Standard-API wie OpenGL definierte Befehle möglicherweise erfordern, dass Daten von mehreren Pixeln miteinander zu kombinieren sind, z. B. aus den vier Pixeln in einem Pixelquad. Zum Beispiel kann das Vorverarbeitungsmodul **313** die Arbeitselemente im Verarbeitungsvorgang teilweise verarbeiten, z. B. um durch einen „Gradienten“-Vorgang die Änderungsrate einer variierenden Menge von verschiedenen Pixeln in einem Pixelquad zu bestimmen, bevor die Verarbeitungsvorgänge an die Verarbeitungsbahnen **314** weitergegeben werden. In diesem Fall werden die ungültigen Arbeitselemente, die in den Verarbeitungsvorgängen umfasst sind, verwendet, um die Vorverarbeitung im Modul **313** durchzuführen, sodass die Verarbeitungsvorgänge Befehle für die gültigen Arbeitselemente korrekt in den Verarbeitungsbahnen **314** umsetzen können. Mit anderen Worten sind ungültige Arbeitselemente für ungültige Pixel eines Pixelquads, das einen oder mehrere gültige Pixel umfasst, in einem Verarbeitungsvorgang enthalten, sodass der/die gültige(n) Pixel im Pixelquad im Ausführungsmodul **310** korrekt verarbeitet werden kann/können. In den hier beschriebenen Beispielen ist es jedoch so, dass, falls es in einem Pixelquad gar keine gültigen Pixel gibt, keine Arbeitselemente für irgendein Pixel im Pixelquad in einem vom Ausführungsmodul **310** auszuführenden Verarbeitungsvorgang umfasst sind.

[0029] Es wird angemerkt, dass das in **Fig. 4** gezeigte Beispiel ausschließlich der Veranschaulichung dient, und die in **Fig. 4** gezeigte Anordnung der Primitive **406** und **408** über den Pixeln nicht der Anordnung von ungültigen Arbeitselementen entspricht, wie sie in **Fig. 1**, **Fig. 2**, **Fig. 6**, **Fig. 7** oder **Fig. 8** gezeigt ist.

[0030] Der Betrieb der SIMD-Verarbeitungseinheit **302** in einem Beispiel ist in Bezug auf ein in **Fig. 5** gezeigtes Flussdiagramm beschrieben. Wie oben beschrieben, werden Primitivdaten im Verarbeitungsmodul **306** vom Speicher **304** empfangen. In diesem Beispiel ist das Verarbeitungsmodul **306** ausgelegt, um eine Entfernung verdeckter Oberflächen zu implementieren, wie sie nach Stand der Technik bekannt ist, um Primitive in Pixelpositionen zu entfernen, falls diese Primitive von anderen Primitiven in der Szene verdeckt sind und daher nicht zum endgültigen wiedergegebenen Bild beitragen. Die Details des Betriebs des Verarbeitungsmoduls **306** liegen außerhalb des Schutzbereichs dieser Patentanmeldung, außer dass das Verarbeitungsmodul **306** für jede einer Mehrzahl von Pixelpositionen einen Primitivenidentifikator an das Steuermodul **308** weitergibt, der ein zu verarbeitendes sichtbares Primitiv identifiziert.

[0031] In Schritt S502 gruppiert das Steuermodul **308** Arbeitselemente in Verarbeitungsvorgängen, um vom Ausführungsmodul **310** verarbeitet zu werden,

d. h. das Steuermodul **308** stellt die Arbeitselemente zu den Verarbeitungsvorgängen zusammen. Die Arbeitselemente eines Verarbeitungsvorgangs sind ausgelegt, um einen gemeinsamen Befehlsstrom an entsprechenden Pixelwerten auszuführen. Zum Beispiel können die Arbeitselemente eines Verarbeitungsvorgangs sich auf ein Anwenden der Textur des Primitivs **406** auf entsprechende Pixel im Renderingraum **402** beziehen. Weitere Details zur Zusammenstellung der Arbeitselemente in Verarbeitungsvorgängen werden weiter unten in verschiedenen Beispielen beschrieben. Die Verarbeitungsvorgänge werden an das Ausführungsmodul **310** weitergegeben.

[0032] Das Ausführungsmodul **310** empfängt auch die Datenelemente (z. B. die Pixelwerte), um die Arbeitselemente von den Verarbeitungsvorgängen auf den Datenelementen auszuführen. Im in **Fig. 3** gezeigten Beispiel werden die Datenelemente im Ausführungsmodul **310** vom Speicher **304** empfangen. Es wird angemerkt, dass in anderen Beispielen die Datenelemente vom Verarbeitungsmodul **306** an das Steuermodul **308** und dann an das Ausführungsmodul **310** weitergegeben werden können, aber der Effizienz halber werden im in **Fig. 3** gezeigten Beispiel die Datenelemente nur dann aus dem Speicher **304** abgerufen, wenn sie gebraucht werden, d. h. vom Ausführungsmodul **310** und nicht vom Verarbeitungsmodul **306** oder dem Steuermodul **308**.

[0033] In Schritt S504 werden die Arbeitselemente der Verarbeitungsvorgänge im Ausführungsmodul **310** ausgeführt. Wie oben beschrieben, kann ein Teil der Vorverarbeitung an den Arbeitselementen vom Verarbeitungsmodul **313** durchgeführt werden, z. B. um durch einen „Gradienten“-Vorgang die Änderungsrate einer variierenden Menge von verschiedenen Pixeln in einem Pixelquad zu bestimmen, bevor die Arbeitselemente der Verarbeitungsvorgänge an die Verarbeitungsbahnen **314** weitergegeben werden. In manchen Beispielen ist das Vorverarbeitungsmodul **313** nicht in der SIMD-Verarbeitungseinheit implementiert und/oder es wird vor der Weitergabe der Arbeitselemente an die Verarbeitungsbahnen **314** keine Vorverarbeitung durchgeführt. Die Arbeitselemente werden dann von den Verarbeitungsbahnen **314** ausgeführt. **Fig. 6** zeigt ein Ausführungsschema **600**, das definiert, wie in einem Beispiel Befehle der Arbeitselemente eines Verarbeitungsvorgangs Verarbeitungsschritten in den Verarbeitungsbahnen **314** zugewiesen werden. In diesem in **Fig. 6** gezeigten Beispiel werden entsprechende Befehle der zweiunddreißig möglichen Arbeitselemente eines Verarbeitungsvorgangs (z. B. Verarbeitungsvorgang **100**) zur Ausführung über eine Gruppe von vier Verarbeitungsbahnen und acht aufeinanderfolgenden Verarbeitungszyklen eingeplant. Es kann im Ausführungsmodul **310** mehr als vier Verarbeitungsbahnen **314** geben, von denen einige verwendet wer-

den können, um simultan zur Verarbeitung des Verarbeitungsvorgangs **100** Arbeitselemente von anderen Verarbeitungsvorgängen zu verarbeiten. Wie oben beschrieben, kann es zum Beispiel sechzehn parallele Verarbeitungsbahnen geben, die in vier Gruppe von jeweils vier Verarbeitungsbahnen unterteilt werden können, die ausgelegt sind, um Befehle von Arbeitselementen von entsprechenden Verarbeitungsvorgängen über acht Verarbeitungszyklen zu verarbeiten. Falls alle Verarbeitungsvorgänge voll von Arbeitselementen sind (z. B. falls sie jeweils zweiunddreißig Arbeitselemente umfassen), könnten daher Befehle von allen Arbeitselementen in den vier Verarbeitungsvorgängen in acht Verarbeitungszyklen verarbeitet werden. Dies entspricht der Anzahl von Verarbeitungsvorgängen über acht Verarbeitungszyklen des in **Fig. 2** gezeigten Beispiels, in dem Befehle von allen Arbeitselementen in einem Verarbeitungsvorgang über alle Verarbeitungsbahnen über zwei Verarbeitungszyklen ausgeführt werden können.

[0034] **Fig. 6** zeigt jedoch die Abbildung des Verarbeitungsvorgangs **100** (der in **Fig. 1** gezeigt ist) auf die Verarbeitungsschlitze der ersten Gruppe von vier Verarbeitungsbahnen. Wie oben beschrieben, umfasst der Verarbeitungsvorgang **100** nur siebzehn Arbeitselemente, und die Arbeitselemente 7, 8, 10, 12, 13 und 14 sind ungültig. In diesem Fall umfassen, wie in **Fig. 6** gezeigt, drei Verarbeitungszyklen (clk 5, clk 6, clk 7) keine eingeplanten Arbeitselemente des Verarbeitungsvorgangs **100**. Falls in einem bestimmten Verarbeitungszyklus für eine Gruppe von Verarbeitungsbahnen, die dem Verarbeitungsvorgang **100** zugeordnet sind, keine gültigen Arbeitselemente enthalten sind, bewirkt die Logik **316**, dass die Gruppe von Verarbeitungsbahnen den bestimmten Verarbeitungszyklus überspringt. Das Überspringen der Verarbeitungszyklen kann für verschiedene Gruppen von Verarbeitungsbahnen unabhängig durchgeführt werden. Daher können im in **Fig. 6** gezeigten Beispiel die Verarbeitungszyklen „clk 5“, „clk 6“, und „clk 7“, als **602** gekennzeichnet, übersprungen werden, da vom Verarbeitungsvorgang **100** keine Arbeitselemente zur Ausführung in diesen Verarbeitungszyklen eingeplant sind. Das Überspringen erfolgt unabhängig von den anderen Gruppen von Verarbeitungszyklen, daher können einige andere Verarbeitungsvorgänge Arbeitselemente umfassen, die zur Ausführung in anderen Gruppen von Verarbeitungsbahnen in den Verarbeitungszyklen „clk 5“, „clk 6“, und „clk 7“ eingeplant sind, doch verhindert dies nicht, dass diese Verarbeitungszyklen für die Gruppe von Verarbeitungsbahnen, die den Verarbeitungsvorgang **100** verarbeiten, übersprungen werden. Arbeitselemente, die zur Ausführung von der in **Fig. 6** gezeigten Gruppe von Verarbeitungsbahnen im nächsten Verarbeitungszyklus (clk 8, der in **Fig. 6** nicht gezeigt ist) eingeplant sind, können dann vorwärts bewegt werden, um im Verarbeitungszyklus (clk 5) ausgeführt zu werden. Obwohl der Verarbeitungsvorgang **100**

nur siebzehn Arbeitselemente (von möglichen zweiunddreißig Arbeitselementen) umfasst, kann daher erkannt werden, dass nur drei Verarbeitungsschlitze aufgrund der teilweisen Belegung des Verarbeitungsvorgangs **100** ungenutzt bleiben. Dies steht im Gegensatz zum in **Fig. 2** gezeigten Ausführungsschema, das denselben Verarbeitungsvorgang **100** verarbeitet, in dem jedoch 15 Verarbeitungsschlitze aufgrund der teilweisen Belegung des Verarbeitungsvorgangs **100** ungenutzt bleiben. Es liegt daher auf der Hand, dass durch Teilen der Verarbeitungsbahnen **314** in Gruppen und Verwenden einer Gruppe zum Verarbeiten der Arbeitselemente eines bestimmten Verarbeitungsvorgangs weniger Verarbeitungsschlitze aufgrund der teilweisen Belegung von Verarbeitungsvorgängen ungenutzt bleiben. Es wird angemerkt, dass dies erreicht wird, indem den Gruppen von Verarbeitungsbahnen ermöglicht wird, Verarbeitungszyklen unabhängig von anderen Gruppen von Verarbeitungsbahnen zu überspringen. Es wird auch angemerkt, dass bezugnehmend auf das in **Fig. 2** gezeigte Beispiel dies erreicht wird, ohne die Gesamtzahl von Verarbeitungsbahnen oder die Anzahl von Arbeitselementen in einem Verarbeitungsvorgang zu verändern.

[0035] Im in **Fig. 6** gezeigten Beispiel ist ein Block von Arbeitselementen (der einem Pixelquad, d. h. einem Block von Datenelementen, entspricht) ausgelegt, um in einer Verarbeitungsbahn über aufeinanderfolgenden Verarbeitungszyklen auszuführen. Zum Beispiel führt die erste Verarbeitungsbahn die Arbeitselemente 0 bis 3 über die ersten vier Verarbeitungszyklen aus, und die zweite Verarbeitungsbahn führt die Arbeitselemente 4 bis 7 über die ersten vier Verarbeitungszyklen aus usw. Das bedeutet, dass die Blöcke von Arbeitselementen in einer Spalte-zuerst-Ordnung (engl. column-first order) angeordnet sind, wobei die Höhe der Spalten der Anzahl an Arbeitselementen in einem Block (z. B. vier Arbeitselemente für Pixelquads) entspricht. In weiteren Beispielen kann die Höhe der Spalten unterschiedlich sein, z. B. können die Spalten acht Arbeitselemente umfassen, sodass die erste Verarbeitungsbahn die Arbeitselemente 0 bis 7 über die ersten acht Verarbeitungszyklen ausführt. In einem Extrembeispiel kann die Höhe der Spalten der Anzahl an Arbeitselementen in einem Verarbeitungsvorgang (z. B. zweiunddreißig) entsprechen, sodass die Gruppe von Verarbeitungsbahnen nur eine Bahn breit ist und die Arbeitselemente eines Verarbeitungsvorgangs der Reihe nach von einer einzigen Verarbeitungsbahn ausgeführt werden. Dies kann helfen, ein Überspringen mehrerer Verarbeitungszyklen zu ermöglichen, doch es würden die Vorteile der Durchführung desselben Befehls auf mehreren Datenelementen auf SIMD-Weise eingebüßt werden, sodass die Verarbeitungseinheit komplexer und/oder weniger effizient sein kann. Wie nachstehend detaillierter beschrieben wird, ermöglicht ein Anordnen der Blö-

cke von Arbeitselementen in einer Spalte-zuerst-Ordnung eine teilweise Umordnung der Arbeitselemente innerhalb eines Blocks von zu implementierenden Arbeitselementen, um ungültige Arbeitselemente über die Verarbeitungsbahnen hinweg aufeinander auszurichten bzw. in eine Linie zu bringen (engl. align).

[0036] Es kann viele Gründe dafür geben, warum ein Verarbeitungsvorgang unter Umständen nicht die Höchstanzahl von Arbeitselementen umfasst (z. B. zweiunddreißig Arbeitselemente in den hier beschriebenen Beispielen). Zum Beispiel kann eine Verzögerung in der Verarbeitungspipeline vorliegen, z. B. aufgrund einer Verzögerung bei der Verarbeitung von Primitivdaten im Verarbeitungsmodul **306**, sodass nicht genug Arbeitselemente zur Ausführung bereit sind, um einen Verarbeitungsvorgang zu füllen, wenn ein Arbeitsvorgang bzw. Verarbeitungsvorgang an das Verarbeitungsmodul **310** gesendet werden soll. In diesem Fall kann es als besser angesehen werden, ohne weitere Verzögerungen dem Ausführungsmodul **310** einen Verarbeitungsvorgang zu senden, der nicht vollständig gefüllt ist, anstatt den Verarbeitungsvorgang zu verzögern, um abzuwarten, dass ausreichend Arbeitselemente zur Ausführung bereit sind, um den Verarbeitungsvorgang zu füllen.

[0037] Wie oben in einem Beispiel, in dem die Datenelemente Pixel sind, beschrieben, können die Datenelemente (z. B. Pixel) in Blöcken von Datenelementen (z. B. Pixelquads) angeordnet sein. Im in **Fig. 6** gezeigten Beispiel ist jede der Verarbeitungsbahnen der Gruppe ausgelegt, um Befehle von Arbeitselementen, die sich auf einen entsprechenden Block von Datenelementen beziehen, über eine Mehrzahl von aufeinanderfolgenden Verarbeitungszyklen auszuführen. Das heißt, dass Befehle eines ersten Blocks von Arbeitselementen (Arbeitselemente 0 bis 3), die sich auf einen ersten Pixelquad beziehen, eingeplant sind, in einer ersten Verarbeitungsbahn in den ersten vier Verarbeitungszyklen ausgeführt zu werden; Befehle eines zweiten Blocks von Arbeitselementen (Arbeitselemente 4 bis 7), die sich auf einen zweiten Pixelquad beziehen, eingeplant sind, in einer zweiten Verarbeitungsbahn in den ersten vier Verarbeitungszyklen ausgeführt zu werden; Befehle eines dritten Blocks von Arbeitselementen (Arbeitselemente 8 bis 11), die sich auf einen dritten Pixelquad beziehen, eingeplant sind, in einer dritten Verarbeitungsbahn in den ersten vier Verarbeitungszyklen ausgeführt zu werden; und Befehle eines vierten Blocks von Arbeitselementen (Arbeitselemente 12 bis 15), die sich auf einen vierten Pixelquad beziehen, eingeplant sind, in einer vierten Verarbeitungsbahn in den ersten vier Verarbeitungszyklen ausgeführt zu werden. Aus **Fig. 6** ist ersichtlich, dass der erste Block von Arbeitselementen (Arbeitselemente 0 bis 3) alle gültig sind; der zweite Block von Arbeitselementen (Arbeitselemente 4 bis 7) drei gültige Arbeitselemente und ein ungültiges Arbeitselement

umfasst; der dritte Block von Arbeitselementen (Arbeitselemente 8 bis 11) zwei gültige Arbeitselemente und zwei ungültige Arbeitselemente umfasst; und der vierte Block von Arbeitselementen (Arbeitselemente 12 bis 15) ein gültiges Arbeitselement und drei ungültige Arbeitselemente umfasst.

[0038] Falls alle Arbeitselemente, die zur Ausführung in der Gruppe von Verarbeitungsbahnen in einem Verarbeitungszyklus eingeplant sind, ungültige Arbeitselemente sind, dann kann die Logik **316** die Gruppe von Verarbeitungsbahnen veranlassen, die Ausführung der ungültigen Arbeitselemente zu überspringen. Dies erfolgt auf dieselbe Weise wie das Überspringen der Verarbeitungszyklen, für die keine Arbeitselemente eingeplant wurden. Im in **Fig. 6** gezeigten Beispiel gibt es keine Verarbeitungszyklen, für die nur ungültige Arbeitselemente eingeplant sind, weshalb keiner der ersten fünf Verarbeitungszyklen (clk 0 bis clk 4) übersprungen werden kann. Da jedoch in der Gruppe nur vier Verarbeitungsbahnen existieren, ist es wahrscheinlicher, dass alle Arbeitselemente in einem Verarbeitungszyklus in der Gruppe von Verarbeitungszyklen ungültig sind, verglichen mit dem in **Fig. 2** gezeigten System, in dem sechzehn Arbeitselemente ungültig sein müssten, damit ein Verarbeitungszyklus von diesem Verfahren übersprungen wird.

[0039] Außerdem kann die Wahrscheinlichkeit, dass alle Arbeitselemente, die in einem bestimmten Verarbeitungszyklus in einer Gruppe von Verarbeitungszyklen eingeplant sind, ungültige Arbeitselemente sind, auf die Weise, wie das Steuermodul **308** in Schritt S502 die Arbeitselemente in den Verarbeitungsvorgängen zusammenstellt, erhöht werden. Insbesondere stellt das Steuermodul **308** die Arbeitselemente zu den Verarbeitungsvorgängen basierend auf der Gültigkeit der Arbeitselemente zusammen, sodass ungültige Arbeitselemente eines bestimmten Verarbeitungsvorgangs vorübergehend über einer Gruppe von Verarbeitungsbahnen hinweg aufeinander ausgerichtet sind. Eine diesbezügliche Einschränkung ist, dass das Steuermodul **308** die Arbeitselemente zu den Verarbeitungsvorgängen so zusammenstellt, dass Arbeitselemente eines Blocks von Arbeitselementen, die sich auf einen Block von Datenelementen beziehen, im selben Verarbeitungsvorgang gruppiert sind. Das heißt, dass die Blöcke von Arbeitselementen (z. B., in Verbindung mit Pixelquads) nicht getrennt werden, sodass sie in verschiedenen Verarbeitungsvorgängen enthalten sind. Das „Zusammenstellen“ der Arbeitselemente in den Verarbeitungsvorgängen kann umfassen: i) Gruppieren von Arbeitselementen zu bestimmten Verarbeitungsvorgängen und/oder ii) Anordnen der Arbeitselemente in einem bestimmten Verarbeitungsvorgang.

[0040] Indem die Anzahl von überspringbaren Verarbeitungszyklen erhöht wird, erhöht sich die Effizienz

enz der Ausführung der Verarbeitungsvorgänge, da weniger Verarbeitungsschlitze ungenutzt bleiben. Es wird angemerkt, dass viele Blöcke von Arbeitselementen (z. B. mehr als in einen einzigen Verarbeitungsvorgang passen) zur Ausführung bereit sein können, und dass das Steuermodul **308** bestimmen kann, wie die Arbeitselemente in Verarbeitungsvorgängen gruppiert werden. Eine Art und Weise, wie das Steuermodul **308** die Arbeitselemente basierend auf der Gültigkeit der Arbeitselemente in den Verarbeitungsvorgängen zusammenstellen kann, ist, die Blöcke basierend auf der Anzahl ungültiger Arbeitselemente in den entsprechenden Blöcken von Arbeitselementen in Verarbeitungsvorgängen zu gruppieren. Vollständig gültige Blöcke von Arbeitselementen (d. h. Blöcke, die nur gültige Arbeitselemente umfassen) können zum Beispiel zusammen zu bestimmten Verarbeitungsvorgängen verpackt werden, sodass diese Arbeitsvorgänge keine ungültigen Arbeitselemente umfassen. Im Gegensatz dazu können Blöcke von Arbeitselementen, die wenigstens ein ungültiges Arbeitselement umfassen, zusammen zu anderen Verarbeitungsvorgängen verpackt werden. Dies kann hilfreich sein, um die Wahrscheinlichkeit, dass diese anderen Verarbeitungsvorgänge ungültige Arbeitselemente aufweisen, die zur Ausführung in einer ganzen Gruppe von Verarbeitungsbahnen in einem bestimmten Verarbeitungszyklus eingeplant sind, zu erhöhen, sodass der bestimmte Verarbeitungszyklus übersprungen werden kann. Im in **Fig. 6** gezeigten Beispiel sind alle Arbeitselemente im ersten Block (Arbeitselemente 0 bis 3) gültig, und daher kann keiner der ersten vier Verarbeitungszyklen übersprungen werden, ungeachtet der Gültigkeit der Arbeitselemente in den anderen Blöcken von Arbeitselementen, die zur Ausführung in diesen Verarbeitungszyklen eingeplant sind. Durch Aufteilen vollständig gültiger Blöcke von Arbeitselementen in verschiedene Verarbeitungsvorgänge, um teilweise gültige Blöcke von Arbeitselementen zu bilden, steigt die Wahrscheinlichkeit, dass die Verarbeitungsvorgänge für teilweise gültige Blöcke von Arbeitselementen ungültige Arbeitselemente aufweisen, die zur Ausführung in einer ganzen Gruppe von Verarbeitungsbahnen in einem bestimmten Verarbeitungszyklus eingeplant sind.

[0041] Diese Idee kann weiterentwickelt werden, sodass Blöcke von Arbeitselementen, die drei gültige Arbeitselemente umfassen, in bestimmte Verarbeitungsvorgänge gruppiert werden; Blöcke von Arbeitselementen, die zwei gültige Arbeitselemente umfassen, in andere Verarbeitungsvorgänge gruppiert werden; und Blöcke von Arbeitselementen, die ein gültiges Arbeitselement umfassen, in weitere Verarbeitungsvorgänge gruppiert werden. Daher kann jeder der Blöcke von Arbeitselementen in einem bestimmten Verarbeitungsvorgang dieselbe Anzahl von gültigen Arbeitselementen umfassen.

[0042] Ferner kann das Steuermodul **308** ausgelegt sein, um Arbeitselemente in einem Block von Arbeitselementen umzuordnen, um die ungültigen Arbeitselemente aus verschiedenen Blöcken von Arbeitselementen in einem Verarbeitungsvorgang aufeinander auszurichten. **Fig. 7** zeigt ein Beispiel, in dem Blöcke von Arbeitselementen von einer ersten Konfiguration **702** ausgehend umgeordnet werden, um eine zweite Konfiguration **704** zu ergeben, in der ungültige Arbeitselemente von den Blöcken von Arbeitselementen vorübergehend in der Gruppe von Verarbeitungsbahnen aufeinander ausgerichtet sind. Das in **Fig. 7** gezeigte Beispiel betrifft einen Fall, in dem ein Verarbeitungsvorgang Blöcke von Arbeitselementen umfasst, die jeweils ein ungültiges Arbeitselement umfassen. Wie in **Fig. 7** gezeigt, gibt es vier Blöcke von Arbeitselementen, die als **706₁**, **706₂**, **706₃** und **706₄** gekennzeichnet sind, die in den vier Verarbeitungsbahnen einer Gruppe in vier Verarbeitungszyklen auszuführen sind. Die Arbeitselemente 0, 6, 9 und 15 sind ungültig, während die anderen Arbeitselemente gültig sind. Vor dem Umordnen der Arbeitselemente kann gemäß der Konfiguration **702** keiner der Verarbeitungszyklen übersprungen werden, weil sie alle wenigstens ein gültiges Arbeitselement aus einem der Blöcke **706** umfassen.

[0043] Das Steuermodul **308** ordnet die Arbeitselemente in einigen der Blöcke **706** um, z. B. durch Rotation der Arbeitselemente in den Blöcken. Zum Beispiel werden die Arbeitselemente im Block **706₁** dreimal rotiert. Jeder Rotationsvorgang beinhaltet ein Verschieben der Arbeitselemente eine Position weiter nach unten, wobei das unterste Arbeitselement ganz oben platziert wird. Das Ergebnis der drei Rotationsvorgänge ist im Rotationsblock **708₁** gezeigt, sodass das ungültige Arbeitselement sich an der untersten Position des Blocks **708₁** befindet. Es wird angemerkt, dass die Arbeitselemente in Gegenrichtung um eine Position weiterrotiert werden könnten, um dasselbe Ergebnis zu erzielen. Allgemein ist, falls n die Anzahl von Rotationsvorgängen in eine erste Richtung und W die Anzahl von Arbeitselementen in einem Block ist, die Anzahl von Rotationsvorgängen m in die Gegenrichtung, die zu den n Rotationen in die erste Richtung äquivalent ist, gegeben durch: $m = W - n$.

[0044] Das Steuermodul **308** führt an den Arbeitselementen des zweiten Blocks **706₂** außerdem einen Rotationsvorgang durch, um zum umgeordneten Block **708₂** zu gelangen. Daher befindet sich das ungültige Arbeitselement an der untersten Position des Blocks **708₂**. Gleichermaßen führt das Steuermodul **308** außerdem zwei Rotationsvorgänge an den Arbeitselementen des dritten Blocks **706₃** aus, um zum umgeordneten Block **708₃** zu gelangen. Daher befindet sich das ungültige Arbeitselement 9 an der untersten Position des Blocks **708₃**. Der vierte Block **706₄** muss nicht umgeordnet werden, da sich das un-

gültige Arbeitselement **15** bereits an der untersten Position des Blocks **706₄** befindet. Daher ist der Block **708₄** derselbe wie der Block **706₄**. Es ist ersichtlich, dass die ungültigen Arbeitselemente der Blöcke **708** in der umgeordneten Konfiguration **704** vorübergehend aufeinander ausgerichtet sind. Wenn Verarbeitungsvorgänge zur Ausführung im Ausführungsmodul **314** eingeplant sind, wird daher die Logik **316** bestimmen, dass der Verarbeitungszyklus **clk3** übersprungen werden kann, weil alle in diesem Verarbeitungszyklus zur Ausführung auf den Verarbeitungsbahnen der Gruppe eingeplanten Arbeitselemente ungültig sind und daher nicht verarbeitet werden müssen. Es ist offensichtlich, dass die Umordnung der Arbeitselemente in den Blöcken von Arbeitselementen das Überspringen der Verarbeitung von ungültigen Arbeitselementen ermöglicht, wodurch die Anzahl ungenutzter Verarbeitungsschlitze verringert wird.

[0045] In dem oben beschriebenen Beispiel sind alle ungültigen Arbeitselemente an den untersten Positionen der Blöcke **708** ausgerichtet. In anderen Beispielen könnten die ungültigen Arbeitselemente an anderen Positionen in den Blöcken **708** ausgerichtet sein. Die Anzahl von Rotationsvorgängen, die an den Arbeitselementen jedes der Blöcke **706** durchgeführt werden, wird vom Steuermodul **308** bestimmt, um die ungültigen Arbeitselemente vorübergehend aufeinander auszurichten.

[0046] Das Steuermodul **308** setzt Indikatoren (z. B. einige Bits pro Arbeitselement oder pro Block von Arbeitselementen), um anzuzeigen, wie die Arbeitselemente zu den Verarbeitungsvorgängen zusammengestellt wurden. Die Indikatoren können in den Arbeitselementen eingebaut sein, da sie zum Ausführungselement **310** gesendet werden oder können in einem Speicher gespeichert werden (der ein anderer Speicher als Speicher **304** sein kann und vorzugsweise ein Speicher auf dem Chip ist, sodass das Beschreiben und Auslesen des Speichers eine kurze Verzögerung umfasst), um anschließend abgerufen zu werden, sodass die verarbeiteten Datenelemente am Ausgang der Verarbeitungsbahnen **314** umgeordnet werden können, entsprechend einer etwaigen Umordnung der Arbeitselemente, die beim Verpacken der Arbeitselemente in die Verarbeitungsvorgänge erfolgt ist. Im in **Fig. 7** gezeigten Beispiel wird jeder Block von Arbeitselementen um eine Anzahl von Rotationsvorgängen zwischen null Rotationsvorgängen und drei Rotationsvorgängen rotiert. In einem Beispiel kann daher ein Indikator mit zwei Bits für jeden Block von Arbeitselementen gesetzt werden, um die Anzahl von Rotationsvorgängen anzuzeigen, die an den Arbeitselementen des Blocks **706** vorgenommen wurden. In einem weiteren Beispiel braucht einer der Blöcke von Arbeitselementen (z. B. der erste Block von Arbeitselementen im Verarbeitungsvorgang) keinen Indikator, und die Arbeitselemente in

den anderen Blöcken können umgeordnet werden, um der Ausrichtung der ungültigen Arbeitselemente im besagten einen Block von Arbeitselementen zu entsprechen. Dies kann die Anzahl von Bits, die für die Indikatoren erforderlich sind, reduzieren. Die Indikatoren für die Blöcke von Arbeitselementen zeigen die Ordnung der Arbeitselemente in den Blöcken von Arbeitselementen an.

[0047] **Fig. 8** zeigt ein weiteres Beispiel, in dem Blöcke von Arbeitselementen von einer ersten Konfiguration **802** ausgehend umgeordnet werden, um eine zweite Konfiguration **804** zu ergeben, in der ungültige Arbeitselemente von den Blöcken von Arbeitselementen vorübergehend in der Gruppe von Verarbeitungsbahnen aufeinander ausgerichtet sind. Das in **Fig. 8** gezeigte Beispiel betrifft einen Fall, in dem ein Verarbeitungsvorgang Blöcke von Arbeitselementen umfasst, die jeweils zwei ungültige Arbeitselemente umfassen. Wie in **Fig. 8** gezeigt, gibt es vier Blöcke von Arbeitselementen, die als **806₁**, **806₂**, **806₃** und **806₄** gekennzeichnet sind, die in den vier Verarbeitungsbahnen einer Gruppe in vier Verarbeitungszyklen auszuführen sind. Die Arbeitselemente 0, 2, 5, 7, 9, 10, 14, und 15 sind ungültig, während die anderen Arbeitselemente gültig sind. Vor dem Umordnen der Arbeitselemente kann gemäß der Konfiguration **802** keiner der Verarbeitungszyklen übersprungen werden, weil sie alle wenigstens ein gültiges Arbeitselement aus einem der Blöcke **806** umfassen.

[0048] Das Steuermodul **308** ordnet die Arbeitselemente in einigen der Blöcke **806** um, z. B. durch selektives Umtauschen der ersten zwei Arbeitselemente im Block und Rotieren der Arbeitselemente in den Blöcken. Zum Beispiel wird ein Umtauschvorgang durchgeführt, um die ersten zwei Arbeitselemente im Block **806₁** umzutauschen und dann werden die Arbeitselemente im Block **806₁** einmal rotiert. Wie oben beschrieben, beinhaltet der Rotationsvorgang ein Verschieben der Arbeitselemente eine Position weiter nach unten, wobei das unterste Arbeitselement dann ganz oben platziert wird. Das Ergebnis des Umtauschvorgangs und des Rotationsvorgangs ist im rotierten Block **808₁** gezeigt, sodass die ungültigen Arbeitselemente (Arbeitselemente 0 und 2) sich an den untersten zwei Positionen des Blocks **808₁** befinden. Wie oben beschrieben, wird angemerkt, dass der Rotationsvorgang in die andere Richtung durchgeführt werden könnte.

[0049] Das Steuermodul **308** führt auch an den ersten zwei Arbeitselementen im Block **806₂** einen Umtauschvorgang und an den Arbeitselementen des zweiten Blocks **806₂** drei Rotationsvorgänge durch, um zum umgeordneten Block **808₂** zu gelangen. Daher befinden sich die ungültigen Arbeitselemente 5 und 7 an den untersten zwei Positionen des Blocks **808₂**. Das Steuermodul **308** führt keinen Umtauschvorgang an den Arbeitselementen im Block **806₃**

durch, sondern einen Rotationsvorgang an den Arbeitselementen des dritten Blocks **806₃**, um zum umgeordneten Block **808₃** zu gelangen. Daher befinden sich die ungültigen Arbeitselemente 9 und 10 an den untersten zwei Positionen des Blocks **808₃**. Der vierte Block **806₄** muss nicht umgeordnet werden, da sich die ungültigen Arbeitselemente 14 und 15 bereits an den untersten zwei Positionen des Blocks **806₄** befinden. Daher ist der Block **808₄** derselbe wie der Block **806₄**. Es ist ersichtlich, dass die ungültigen Arbeitselemente der Blöcke **808** in der umgeordneten Konfiguration **804** vorübergehend aufeinander ausgerichtet sind. Wenn Verarbeitungsvorgänge zur Ausführung im Ausführungsmodul **314** eingeplant sind, wird daher die Logik **316** bestimmen, dass die Verarbeitungszyklen **clk2** und **clk3** übersprungen werden können, weil alle in diesen Verarbeitungszyklen zur Ausführung auf den Verarbeitungsbahnen der Gruppe eingeplanten Arbeitselemente ungültig sind und daher nicht verarbeitet werden müssen. Es ist offensichtlich, dass die Umordnung der Arbeitselemente in den Blöcken von Arbeitselementen das Überspringen der Verarbeitung von ungültigen Arbeitselementen ermöglicht, wodurch die Anzahl ungenutzter Verarbeitungsschlitze verringert wird.

[0050] In dem oben beschriebenen Beispiel sind alle ungültigen Arbeitselemente an den untersten zwei Positionen der Blöcke **808** ausgerichtet. In anderen Beispielen könnten die ungültigen Arbeitselemente an anderen Positionen in den Blöcken **808** ausgerichtet sein. Das selektive Umtauschen der ersten zwei Arbeitselemente in den Blöcken und die Anzahl von Rotationsvorgängen, die an den Arbeitselementen jedes der Blöcke **806** durchgeführt werden, werden vom Steuermodul **308** bestimmt, um die ungültigen Arbeitselemente vorübergehend aufeinander auszurichten.

[0051] Ferner können in anderen Beispielen verschiedene der Arbeitselemente von den Umtauschvorgängen selektiv umgetauscht werden. Die Positionen der Arbeitselemente, die umgetauscht werden können, sind jedoch in einem bestimmten Beispiel vorbestimmt, sodass keine Bits erforderlich sind, um anzuzeigen, welche Arbeitselemente umgetauscht worden sind. Jeder Indikator zum Anzeigen, wie die Arbeitselemente für eine Mehrzahl der Blöcke **808** umgeordnet wurden, kann drei Bits umfassen: zwei Bits, um eine Rotation der Arbeitselemente im Block **808** zu definieren und ein Bit, um zu definieren, ob ein Umtauschvorgang an zwei der Arbeitselemente an den vorbestimmten Positionen im Block von Arbeitselementen durchgeführt wurde.

[0052] Für Verarbeitungsvorgänge, die Blöcke von Arbeitselementen umfassen, die jeweils drei ungültige Arbeitselemente umfassen, wird die Umordnung der Blöcke entsprechend der in **Fig. 7** gezeigten und oben in Bezug auf Verarbeitungsvorgänge, die Blö-

cke von Arbeitselementen mit einem ungültigen Arbeitselement umfassen, beschriebene Weise durchgeführt. Das heißt, es werden keine Umtauschvorgänge durchgeführt, aber es können einige Rotationsvorgänge durchgeführt werden, um die ungültigen Arbeitselemente aus verschiedenen Blöcken von Arbeitselementen in einem Verarbeitungsvorgang vorübergehend aufeinander auszurichten.

[0053] Wenn die Arbeitselemente von den Verarbeitungsbahnen **314** verarbeitet wurden, werden in Schritt **S506** die aus den Verarbeitungsbahnen **314** ausgegebenen Datenelemente im Speicher **312** gespeichert. Die Speicherlogik **318** verwendet den Indikator, der anzeigt, wie die Arbeitselemente in den Verarbeitungsvorgängen zusammengestellt wurden, um Adressen zum Speichern der verarbeiteten Datenelemente im Speicher **312** zu bestimmen. Obwohl die Arbeitselemente zur Ausführung von den Verarbeitungsbahnen **314** umgeordnet werden können, wird die Speicheradresse, wenn die Ergebnisse im Speicher gespeichert werden, bestimmt, sodass die verarbeiteten Datenelemente an den korrekten Adressen im Speicher **312** gespeichert werden, z. B. als wären die Arbeitselemente nicht umgeordnet worden.

[0054] In den oben beschriebenen Beispielen können die Arbeitselemente eine von zwei Gültigkeitsebenen aufweisen: ein Arbeitselement ist entweder gültig oder ungültig. In anderen Beispielen können jedoch mehr als zwei Gültigkeitsebenen, die Arbeitselemente aufweisen könnten, existieren. Zum Beispiel kann an einem Pixel ein Texturierungsvorgang vorgenommen werden, der ein Verwenden der Gradienten der benachbarten Pixel (z. B. unter Ausschluss diagonalen Nachbarn) beinhalten kann. In diesem Fall, falls der Texturierungsvorgang an einem bestimmten Pixel eines Pixelquads durchzuführen ist, kann das bestimmte Pixel als gültiges Datenelement angesehen werden, auf dem eine große Anzahl von Befehlen von den Verarbeitungsbahnen **314** durchzuführen ist, um den Texturierungsvorgang zu implementieren; während die benachbarten Pixel im Pixelquad, mit Ausnahme des diagonalen Nachbarn, als teilweise ungültige Datenelemente zu betrachten sein können, auf denen eine geringere Anzahl von Befehlen von den Verarbeitungsbahnen **314** durchzuführen ist, um den Texturierungsvorgang zu implementieren; während der diagonale Nachbar im Pixelquad als vollständig ungültiges Datenelement zu betrachten sein kann, auf dem keine oder eine sehr geringe Anzahl von Befehlen von den Verarbeitungsbahnen **314** durchzuführen ist, um den Texturierungsvorgang zu implementieren. Es kann hilfreich sein, Arbeitselemente, die dieselbe Gültigkeitsebene aufweisen, aufeinander auszurichten. Auf diese Weise werden, wenn Befehle an gültigen und teilweise gültigen Datenelementen durchzuführen sind, aber keine Befehle an ungültigen Datenelementen

durchzuführen sind, die ungültigen Arbeitselemente, die den ungültigen Datenelementen entsprechen, aufeinander ausgerichtet und können übersprungen werden. Wenn alle Befehle für die teilweise ungültigen Datenelemente durchgeführt worden sind, aber weiterhin Befehle an den gültigen Datenelementen durchzuführen sind, werden dann die teilweise ungültigen Arbeitselemente aufeinander ausgerichtet und können übersprungen werden.

[0055] Wie bereits festgestellt, verwenden einige Befehle Informationen von anderen Pixeln in einem 2×2-Pixelblock, zum Beispiel werden beim Texturieren Gradienten in Pixeln berechnet, um auf nach Stand der Technik bekannte Weise eine Detailebene zu bestimmen. Als solche werden Befehle, die Ergebnisse in diese Art von Befehl (z. B. Texturierungsbefehle) einspeisen, auf einigen ungültigen Pixeln in einem 2×2-Pixelblock ausgeführt, z. B. von dem Vorverarbeitungsmodul **313**. Da einige Befehle nicht dazu beitragen, 2×2-Block-Befehle zu kreuzen, ist es wünschenswert, zwischen den zwei Arten von Befehlen zu unterscheiden. Dazu wird vorgeschlagen, ein Bit in den Befehl einzubauen, der codiert, um anzuzeigen, ob ein Befehl in einem ungültigen Fall auszuführen ist.

[0056] Ein weiteres Beispiel dafür, wie Arbeitselemente aus Verarbeitungsvorgängen in einer Mehrzahl von Verarbeitungsbahnen ausgeführt werden können, wird in Bezug auf **Fig. 9** und **Fig. 10** beschrieben. **Fig. 9** zeigt ein Ausführungsschema **900**, das definiert, wie die Arbeitselemente eines Verarbeitungsvorgangs in einem Beispiel Verarbeitungsschlitzen in den Verarbeitungsbahnen **314** zugeordnet werden. In diesem Beispiel gibt es sechzehn Verarbeitungsbahnen (gleich wie in den oben beschriebenen Beispielen), und die Verarbeitungsbahnen sind in vier parallele Gruppen **902₁**, **902₂**, **902₃**, **902₄** unterteilt, die jeweils vier Verarbeitungsbahnen umfassen. Ähnlich dem oben in Bezug auf **Fig. 6** beschriebenen Beispiel sind Befehle der zweiunddreißig möglichen Arbeitselemente eines Verarbeitungsvorgangs zur Ausführung über eine Gruppe von vier Verarbeitungsbahnen und acht aufeinanderfolgenden Verarbeitungszyklen eingeplant. Das heißt, dass jede Gruppe von Verarbeitungsbahnen ausgelegt ist, Arbeitselemente eines entsprechenden Verarbeitungsvorgangs über eine Mehrzahl von Verarbeitungszyklen auszuführen. **Fig. 9** zeigt, wie Befehle der Arbeitselemente eines Verarbeitungsvorgangs, der nur siebzehn Arbeitselemente umfasst, in diesem Beispiel in den Verarbeitungsbahnen der Gruppe **902₁** eingeplant sind.

[0057] Im in **Fig. 9** gezeigten Beispiel ist die Anzahl von Arbeitselementen in einem Block von Arbeitselementen (z. B. vier im Fall von Pixelquads, wie oben beschrieben) gleich der Anzahl von Verarbeitungsbahnen in jeder der Gruppen von Verarbeitungsbah-

nen (z. B. vier). Auf diese Weise sind die Verarbeitungsbahnen der Gruppe für jede der Gruppen, in jedem der Verarbeitungszyklen ausgelegt, Befehle von Arbeitselementen in Bezug auf entsprechende Datenelemente aus einem Block von Datenelementen auszuführen. Mit anderen Worten, die Blöcke von Arbeitselementen sind in einer Spalte-zuerst-Ordnung angeordnet. Zum Beispiel werden, wie in **Fig. 9** gezeigt, Befehle der Arbeitselemente 0 bis 3 des ersten Blocks von Arbeitselementen eines Verarbeitungsvorgangs in einem ersten Verarbeitungszyklus (clk 0) in den Verarbeitungsbahnen der Gruppe **902₁** ausgeführt; Befehle der Arbeitselemente 4 bis 7 des zweiten Blocks von Arbeitselementen eines Verarbeitungsvorgangs in einem zweiten Verarbeitungszyklus (clk 1) in den Verarbeitungsbahnen der Gruppe **902₁** ausgeführt; Befehle der Arbeitselemente 8 bis 11 des dritten Blocks von Arbeitselementen eines Verarbeitungsvorgangs in einem dritten Verarbeitungszyklus (clk 2) in den Verarbeitungsbahnen der Gruppe **902₁** ausgeführt; Befehle der Arbeitselemente 12 bis 15 des vierten Blocks von Arbeitselementen eines Verarbeitungsvorgangs in einem vierten Verarbeitungszyklus (clk 3) in den Verarbeitungsbahnen der Gruppe **902₁** ausgeführt; und ein Befehl des Arbeitselements 16 des fünften Blocks von Arbeitselementen eines Verarbeitungsvorgangs in einem fünften Verarbeitungszyklus (clk 4) in einer der Verarbeitungsbahnen der Gruppe **902₁** ausgeführt.

[0058] **Fig. 10** zeigt die Grundschriffe des Verfahrens des Verwendens der SIMD-Verarbeitungseinheit **302** zum Verarbeiten von Verarbeitungsvorgängen unter Verwendung des in **Fig. 9** gezeigten Ausführungsschemas **900**. In Schritt S1002 werden die Arbeitselemente der Verarbeitungsvorgänge in den entsprechenden Gruppen von Verarbeitungsbahnen **902** gemäß dem in **Fig. 9** gezeigten Ausführungsschema **900** ausgeführt. In Schritt S1004, ähnlich wie oben in Bezug auf das in **Fig. 6** gezeigte Beispiel beschrieben, überspringt eine bestimmte Gruppe von Verarbeitungsbahnen einen bestimmten Verarbeitungszyklus unabhängig von den anderen Gruppen der Verarbeitungsbahnen, falls keine Arbeitselemente zur Ausführung in den Verarbeitungsbahnen der bestimmten Gruppe im bestimmten Verarbeitungszyklus eingeplant sind.

[0059] In Bezug auf **Fig. 9** zum Beispiel umfasst der der Gruppe von Verarbeitungsbahnen **902₁** zugewiesene Verarbeitungsvorgang keine Arbeitselemente, die in den Verarbeitungszyklen clk5 bis clk7 auszuführen sind, wie in **Fig. 9** durch **904** gekennzeichnet. Daher kann die Gruppe von Verarbeitungsbahnen **902₁** diese Verarbeitungszyklen **904** überspringen. Andere Arbeitselemente aus darauffolgenden Verarbeitungsvorgängen (in **Fig. 9** nicht gezeigt) können in den Verarbeitungsbahnen der Gruppe **902₁** in den Verarbeitungszyklen **904** ausgeführt werden. Auf diese Weise bleiben die Verarbeitungsschlitze in der

ersten Gruppe von Verarbeitungsbahnen **902**₁ in den Verarbeitungszyklen **904** nicht ungenutzt. Dies entspricht dem, was weiter oben in Bezug auf **Fig. 6** beschrieben wurde.

[0060] In einigen Fällen können im in **Fig. 9** gezeigten Beispiel jedoch weniger Verarbeitungsschritte ungenutzt bleiben als im in **Fig. 6** gezeigten Beispiel. Falls zum Beispiel der der Gruppe **902**₁ zugewiesene Verarbeitungsvorgang 20 Arbeitselemente umfasst, würden gemäß dem Ausführungsschema **900** die Arbeitselemente alle Verarbeitungsschritte in der Gruppe von Verarbeitungsbahnen **902**₁ in den ersten fünf Verarbeitungszyklen beanspruchen, doch wären die späteren Verarbeitungszyklen **904** weiterhin leer und könnten somit übersprungen werden. Im Gegensatz dazu würde im in **Fig. 6** gezeigten Beispiel ein Verarbeitungsvorgang, der 20 Arbeitselemente umfasst, ein auszuführendes Arbeitselement in jedem der acht Verarbeitungszyklen, clk 0 bis clk7, einplanen und somit zwölf Verarbeitungsschritte ungenutzt lassen.

[0061] Darüber hinaus kann das in **Fig. 9** gezeigte Ausführungsschema **900** besonders für die Verwendung mit der Steuerung von divergentem Fluss geeignet sein. Wie oben beschrieben ist die SIMD-Verarbeitungseinheit **302** ausgelegt, Verarbeitungsvorgänge auszuführen, die mehrere Fälle desselben, auf entsprechenden Datenelementen durchzuführenden Befehls umfassen. Das heißt, dass derselbe Befehl auf mehreren Datenelementen in einem bestimmten Verarbeitungszyklus für einen Verarbeitungsvorgang durchgeführt wird. Einige Befehle können jedoch bewirken, dass basierend auf den speziellen Datenelementen, die gerade verarbeitet werden, verschiedene aufeinanderfolgende Befehle durchgeführt werden. Zum Beispiel können WENN-DANN-SONST-Konstruktionen zur Folge haben, dass ein erster Befehl („Befehl A“) zur Ausführung auf einer ersten Gruppe von Datenelementen eingeplant wird und ein zweiter Befehl („Befehl B“) zur Ausführung auf einer zweiten Gruppe von Datenelementen eingeplant wird. In diesem Fall kann der erste Befehl auf allen Datenelementen ausgeführt werden, aber eine erste Maske wird verwendet, sodass nur die Ergebnisse der Ausführung des Befehls auf der ersten Gruppe von Datenwerten (d. h. nicht der zweiten Gruppe von Datenwerten) verwendet wird, um die Werte im Speicher **312** zu aktualisieren. Dann kann der zweite Befehl auf allen Datenelementen ausgeführt werden, es wird jedoch eine zweite Maske verwendet, sodass nur die Ergebnisse der Ausführung des Befehls auf der zweiten Gruppe von Datenwerten (d. h. nicht der ersten Gruppe von Datenwerten) verwendet werden, um die Werte im Speicher **312** zu aktualisieren. Das bedeutet, dass der Befehl zweimal durchgeführt wird und daher für mehr als zweimal die Anzahl von Verarbeitungszyklen eingeplant wird. Falls alle Arbeitselemente, die für ei-

nen Verarbeitungszyklus eingeplant sind, ausgeblendet werden, kann dieser Verarbeitungszyklus allerdings übersprungen werden. Da die Gruppe von Verarbeitungsbahnen für einen Verarbeitungsvorgang nur vier Verarbeitungsbahnen umfasst, ist es wahrscheinlicher, dass im in **Fig. 2** gezeigten Beispiel alle Verarbeitungsbahnen für einen Verarbeitungsvorgang ein ausgeblendetes Arbeitselement in einem bestimmten Verarbeitungszyklus umfassen. Da Verarbeitungszyklen für Gruppen von Verarbeitungsbahnen unabhängig von anderen Verarbeitungsbahnen übersprungen werden können, können mehr Verarbeitungsschritte durch das Unterteilen der Gesamtanzahl von Verarbeitungsbahnen in Gruppen übersprungen werden. Das bedeutet, dass das Unterteilen der Gesamtanzahl von Verarbeitungsbahnen in Gruppen, wie hier beschrieben, eine feinere Körnung bereitstellt, wenn Verarbeitungszyklen mit Steuerung von divergentem Fluss übersprungen werden.

[0062] Ferner kann das in **Fig. 9** gezeigte Ausführungsschema **900** für den Gebrauch mit der Steuerung von divergentem Fluss noch geeigneter sein als das in **Fig. 6** gezeigte Ausführungsschema **600**. Grund dafür ist, dass Arbeitselemente aus demselben Block von Arbeitselementen häufig auf ähnliche Weise divergieren, und gemäß dem Ausführungsschema **900** jede Gruppe von Verarbeitungsbahnen **902** Arbeitselemente aus nur einem Block in jedem Verarbeitungszyklus verarbeitet. Falls ein Arbeitselement aus einem Block von Arbeitselementen für einen bestimmten Verarbeitungszyklus ausgeblendet wird, dann ist es daher wahrscheinlich, dass auch die anderen Arbeitselemente in diesem Block für den bestimmten Verarbeitungszyklus ausgeblendet werden, sodass der bestimmte Verarbeitungszyklus für die geeignete Gruppe von Verarbeitungsbahnen übersprungen werden kann.

[0063] Wie oben erwähnt, umfassen Verarbeitungsvorgänge eine Mehrzahl von Arbeitselementen, die einen gemeinsamen Befehlsstrom auf einer entsprechenden Mehrzahl von Datenelementen ausführen. Anstatt nacheinander alle Befehle eines Verarbeitungsvorgangs auszuführen und dann zum nächsten Verarbeitungsvorgang überzugehen, kann es eine Mehrzahl von Verarbeitungsvorgängen geben, für die Befehle der Reihe nach durchgeführt werden. Das heißt, dass es eine Gruppe von $N_{\text{Verarbeitungsvorgang}}$ Verarbeitungsvorgängen geben kann (z. B. kann $N_{\text{Verarbeitungsvorgang}}$ gleich 20 sein), für die Befehle der Reihe nach durchgeführt werden, sodass ein Befehl von einem ersten Verarbeitungsvorgang durchgeführt wird, dann ein Befehl vom nächsten Verarbeitungsvorgang und so weiter für alle $N_{\text{Verarbeitungsvorgang}}$ Verarbeitungsvorgänge der Gruppe. Dann läuft die Reihenfolge rückwärts, sodass der nächste Befehl vom ersten Verarbeitungsvorgang durchgeführt wird, dann der nächste Befehl vom nächsten Verarbeitungsvorgang und so weiter. Auf diese Weise be-

steht ein Zeitraum zwischen der Ausführung von Befehlen von einem bestimmten Verarbeitungsvorgang. Dies kann hilfreich sein, da es eine Latenzzeit des Prozesses (z. B. zum Ablesen oder Schreiben von Daten usw.) ermöglicht, die mit den Befehlen verbunden sein kann, die auf den Datenelementen durchgeführt werden, bevor der nächste Befehl des Verarbeitungsvorgangs ausgeführt wird.

[0064] Die Anzahl von Verarbeitungsvorgängen, $N_{\text{Verarbeitungsvorgang}}$, die in der Gruppe von Verarbeitungsvorgängen enthalten sind, für die vor dem Wiederholen von Verarbeitungsvorgängen Arbeitselemente ausgeführt werden, können in verschiedenen Beispielen unterschiedlich sein und in verschiedenen Beispielen fest oder variabel sein. Zum Beispiel kann $N_{\text{Verarbeitungsvorgang}}$ abhängig von der Anzahl von z. B. aufgrund der hier beschriebenen Verfahren übersprungenen Verarbeitungszyklen variieren. Je mehr Verarbeitungszyklen übersprungen werden, desto eher wird nach Abschluss des vorhergehenden Befehls des Verarbeitungsvorgangs der nächste Befehl eines Verarbeitungsvorgangs durchgeführt, und desto weniger Zeit ist daher für die Latenzzeit des Prozesses vorhanden. Falls daher die Anzahl von in einem gegebenen Zeitraum übersprungenen Verarbeitungszyklen zunimmt, dann kann die Anzahl $N_{\text{Verarbeitungsvorgang}}$ erhöht werden, z. B. mit dem Ziel, einen annähernd konstanten Zeitraum zwischen der Ausführung aufeinanderfolgender Befehle von einem bestimmten Verarbeitungsvorgang bereitzustellen, um so eine bestimmte Latenzzeit des Prozesses für die Befehle des Verarbeitungsvorgangs zu ermöglichen.

[0065] In den oben beschriebenen Beispielen umfasst ein Verarbeitungsvorgang eine Höchstanzahl von zweiunddreißig Arbeitselementen. Allgemeiner gesagt, kann in anderen Beispielen die Höchstanzahl von Arbeitselementen in einem Verarbeitungsvorgang, N_{Arbeit} , irgendeine Zahl sein, aber der Einfachheit halber wäre N_{Arbeit} normalerweise eine Zweierpotenz.

[0066] In den oben beschriebenen Beispielen gibt es sechzehn Verarbeitungsbahnen **314** im Ausführungsmodul **310**. Allgemeiner gesagt, kann in anderen Beispielen irgendeine Anzahl von Verarbeitungsbahnen im Ausführungsmodul vorhanden sein, doch der Einfachheit halber wäre die Anzahl von Verarbeitungsbahnen im Ausführungsmodul, N_{Verarb} , normalerweise eine Zweierpotenz.

[0067] In den oben beschriebenen Beispielen sind ferner in einem Block von Datenelementen vier Datenelemente vorhanden. In den hier beschriebenen Hauptbeispielen bedeutet das, dass die Blöcke von Datenelementen Pixelquads sind. Allgemeiner gesagt, kann irgendeine Anzahl von Datenelementen in jedem Block von Datenelementen vorhanden sein, doch der Einfachheit halber wäre die Anzahl von

Datenelementen in jedem Block von Datenelementen, N_{Elemente} , normalerweise eine Zweierpotenz. Darüber hinaus kann der Einfachheit halber N_{Elemente} ausgewählt sein, ein Faktor von N_{Verarb} (d. h. $N_{\text{Verarb}} \bmod N_{\text{Elemente}} = 0$) sein, sodass eine ganze Zahl von Blöcken von Arbeitselementen in allen Verarbeitungsbahnen in einem Verarbeitungszyklus verarbeitet werden kann.

[0068] Wie oben beschrieben sind außerdem die Datenelemente im Fall der SIMD-Verarbeitungseinheit **302**, die Grafikdaten als Teil einer GPU verarbeitet, Pixelwerte. In anderen Beispielen könnten die Datenelemente allerdings nicht Pixelwerte sein, sondern andere Arten von Datenelementen, wie Fragmentwerte oder Scheitelwerte, oder Nicht-Grafikdatenelemente, zum Beispiel wenn die Verarbeitungseinheit **302** auf einem OpenCL-Programm läuft, um eine Parallelberechnung für allgemeine Zwecke durchzuführen. Die Datenelemente sind insofern zum Verarbeiten durch eine SIMD-Verarbeitungseinheit geeignet, als mehrere Fälle eines Befehls auf verschiedenen Datenelementen parallel durchgeführt werden.

[0069] In den hier beschriebenen Beispielen sind die Verarbeitungszyklen Taktungszyklen. Allgemeiner gesagt, kann in anderen Beispielen jeder Verarbeitungszyklus einen oder mehrere Taktungszyklen umfassen.

[0070] Im Allgemeinen können alle oben beschriebenen Funktionen, Verfahren, Techniken oder Bauteile in Modulen implementiert sein, die Software, Firmware, Hardware (z. B. Festlogikschaltungen) oder eine beliebige Kombination aus diesen Implementierungen verwenden. Die Begriffe „Modul“, „Funktionalität“, „Bauteil“, „Einheit“ und „Logik“ werden hier so verwendet, dass sie allgemein für eine Software, Firmware Hardware oder eine beliebige Kombination daraus stehen.

[0071] Im Fall einer Softwareimplementierung stellt das Modul, die Einheit, die Funktionalität, das Bauteil oder die Logik einen Programmcode dar, der spezielle Verarbeitungsvorgänge durchführt, wenn er auf einem Prozessor (z. B. einer oder mehrere CPU) ausgeführt wird. In einem Beispiel können die beschriebenen Verfahren von einem Computer durchgeführt werden, der mit einer auf einem computerlesbaren Medium gespeicherten Software in maschinenlesbarer Form konfiguriert ist. Eine solche Konfiguration eines computerlesbaren Mediums ist ein Signalübermittlungsmedium, das daher ausgelegt ist, die Befehle (z. B. als Trägerwelle) an die Berechnungsvorrichtung zu übertragen, etwa über ein Netz. Das computerlesbare Medium kann auch als ein nicht vorübergehendes computerlesbares Speichermedium ausgelegt sein und daher kein Signalübermittlungsmedium sein. Beispiele eines computerlesbaren Speicherelements umfassen einen Direktzugriffsspeicher (RAM),

einen Festwertspeicher (ROM), eine optische Platte, einen Flash-Speicher, einen Festplattenspeicher und sonstige Speichervorrichtungen, die Magnet-, Optik- und sonstige Techniken verwenden, um Befehle oder sonstige Daten zu speichern, und auf die eine Maschine zugreifen kann.

[0072] Die Software kann in Form eines Computerprogramms vorliegen, das einen Computerprogrammcode zum Konfigurieren eines Computers zur Durchführung der wesentlichen Abschnitte von beschriebenen Verfahren umfasst, oder in Form eines Computerprogramms, das Computerprogrammcode-mittel umfasst, die geeignet sind, alle Schritte irgendeines der hier beschriebenen Verfahren durchzuführen, wenn das Programm auf einem Computer läuft, und wo das Computerprogramm auf einem computerlesbaren Medium vorliegen kann. Der Programmcode kann auf einem oder mehreren computerlesbaren Medien gespeichert sein. Die Merkmale der hier beschriebenen Verfahren sind plattformunabhängig, was bedeutet, dass die Techniken auf einer Mehrzahl von Berechnungsplattformen mit einer Mehrzahl von Prozessoren implementiert sein können.

[0073] Fachleute werden ebenfalls erkennen, dass die Gesamtheit oder ein Teil der Funktionalität, der Techniken oder der Verfahren von einer Festschaltung, einer anwendungsspeziellen integrierten Schaltung, einer programmierbaren Logikanordnung, einer feldprogrammierbaren Gatter-Anordnung oder dergleichen durchgeführt werden kann. Beispielsweise kann das Modul, die Einheit, die Funktionalität, das Bauteil oder die Logik Hardware in Form von Schaltungen umfassen. Solche Schaltungen können Transistoren und/oder andere Hardware-Elemente, die in einem Herstellungsprozess verfügbar sind, umfassen. Solche Transistoren und/oder anderen Elemente können verwendet werden, um Schaltungen oder Strukturen zu bilden, die z. B. einen Speicher wie Register, Flip-Flop-Schaltungen oder Verriegelungen, logische Operatoren wie Boolesche Operatoren, mathematische Operatoren wie Addierer, Multiplikatoren oder Verschieber und Zwischenschaltungen implementieren und/oder enthalten. Solche Elemente können als spezielle Schaltungen oder Standard-Zellenbibliotheken, Makros oder auf anderen Abstraktionsebenen bereitgestellt sein. Solche Elemente können in einer speziellen Anordnung miteinander verbunden sein. Das Modul, die Einheit, die Funktionalität, das Bauteil oder die Logik kann Schaltungen mit einer festen Funktion und Schaltungen, die programmiert sein können, eine Funktion oder Funktionen auszuführen, umfassen; eine solche Programmierung kann von einer Firmware- oder Softwareaktualisierung oder einem Steuermechanismus bereitgestellt werden. In einem Beispiel weist eine Hardwarelogik Schaltungen auf, die einen Festfunktionsbetrieb, eine Zustandsmaschine oder einen Vorgang ausführen.

[0074] Ebenfalls soll Software beinhaltet sein, die die Konfiguration von Hardware „beschreibt“ oder definiert, die ein Modul, eine Einheit, eine Funktionalität, ein Bauteil oder eine Logik, die oben beschrieben wurden, ausführt, wie HDL(Hardware Description Language)-Software, wie sie für das Erstellen von integrierten Schaltungen oder für das Konfigurieren von programmierbaren Chips verwendet wird, um gewünschte Funktionen auszuführen. Das heißt, es kann ein computerlesbares Speichermedium mit einem darauf kodierten computerlesbaren Programmcode zum Erstellen einer Prozessoreinheit bereitgestellt sein, um jedes der hier beschriebenen Verfahren durchzuführen oder um eine Prozessoreinheit zu erstellen, die eine der hier beschriebenen Vorrichtungen umfasst.

[0075] Die Begriffe „Prozessor“ und „Rechner“ werden hier so verwendet, dass sie sich auf jede Vorrichtung oder einen Teil davon mit einer Prozessorfähigkeit beziehen, sodass sie Befehle ausführen kann, oder dass sie sich auf eine Festschaltung beziehen, die in der Lage ist, die Gesamtheit oder einen Teil der Funktionalität oder der Verfahren oder eine beliebige Kombination daraus durchzuführen.

[0076] Obwohl der Gegenstand in einer Sprache beschrieben wurde, die speziell für strukturelle Merkmale und/oder methodologische Vorgänge ist, gilt zu verstehen, dass der Gegenstand, der in den beigefügten Patentansprüchen definiert ist, nicht notwendigerweise auf die oben beschriebenen speziellen Merkmale oder Vorgänge beschränkt ist. Vielmehr sind die speziellen oben beschriebenen Merkmale und Vorgänge als beispielhafte Formen zum Implementieren der Patentansprüche offenbart. Es gilt zu verstehen, dass der oben beschriebene Nutzen und die Vorteile ein Beispiel oder mehrere Beispiele betreffen können.

[0077] Jeder beliebige Bereich oder Wert, der hier angegeben ist, kann erweitert oder verändert werden, ohne dabei den gewünschten Effekt zu verlieren, wie Fachleuten bewusst sein dürfte. Die Schritte der hier beschriebenen Verfahren können in jeder geeigneten Reihenfolge oder, falls angemessen, gleichzeitig durchgeführt werden. Aspekte jedes der oben beschriebenen Beispiele können mit Aspekten von jedem der anderen beschriebenen Beispiele kombiniert werden, um weitere Beispiele zu bilden, ohne den gewünschten Effekt zu verlieren.

Patentansprüche

1. SIMD(Single Instruction Multiple Data)-Verarbeitungseinheit, die ausgelegt ist, eine Mehrzahl von Verarbeitungsvorgängen zu verarbeiten, die jeweils Arbeitselemente bis zu einer vorbestimmten Höchstanzahl umfassen, wobei die Arbeitselemente eines Verarbeitungsvorgangs ausgelegt sind, eine

gemeinsame Sequenz von Befehlen an entsprechenden Datenelementen auszuführen, wobei die Datenelemente in Blöcken von Datenelementen angeordnet sind, wobei einige der Blöcke wenigstens ein ungültiges Datenelement umfassen und wobei Arbeitselemente, die sich auf ungültige Datenelemente beziehen, ungültige Arbeitselemente sind, wobei die SIMD-Verarbeitungseinheit umfasst:

eine Gruppe von Verarbeitungsbahnen, die ausgelegt sind, Befehle von Arbeitselementen eines bestimmten Verarbeitungsvorgangs über eine Mehrzahl von Verarbeitungszyklen auszuführen; und ein Steuermodul, das ausgelegt ist, basierend auf der Gültigkeit der Arbeitselemente die Arbeitselemente zu den Verarbeitungsvorgängen zusammenzustellen, sodass ungültige Arbeitselemente des bestimmten Verarbeitungsvorgangs über die Gruppe von Verarbeitungsbahnen hinweg vorübergehend aufeinander ausgerichtet sind.

2. SIMD-Verarbeitungseinheit nach Anspruch 1, wobei jede der Verarbeitungsbahnen der Gruppe ausgelegt ist, Befehle von Arbeitselementen in Bezug auf einen entsprechenden Block von Datenelementen über eine Mehrzahl von aufeinanderfolgenden Verarbeitungszyklen auszuführen.

3. SIMD-Verarbeitungseinheit nach Anspruch 1 oder 2, ferner umfassend Logik, die mit der Gruppe von Verarbeitungsbahnen gekoppelt ist, die ausgelegt ist, die Gruppe von Verarbeitungsbahnen zu veranlassen, die Ausführung einer Gruppe von ungültigen Arbeitselementen zu überspringen, falls die Gruppe von ungültigen Arbeitselementen die einzigen Arbeitselemente sind, die für die Ausführung in der Gruppe von Verarbeitungsbahnen in einem Verarbeitungszyklus geplant sind.

4. SIMD-Verarbeitungseinheit nach einem der vorhergehenden Ansprüche, wobei das Steuermodul ausgelegt ist, Indikatoren zu setzen, um anzuzeigen, wie die Arbeitselemente zu den Verarbeitungsvorgängen zusammengestellt wurden.

5. SIMD-Verarbeitungseinheit nach Anspruch 4, ferner umfassend:
einen Speicher, der ausgelegt ist, die verarbeiteten Datenelemente, die von der Gruppe von Verarbeitungsbahnen ausgegeben wurden, zu speichern; und Speicherlogik, die ausgelegt ist, Adressen zum Speichern der verarbeiteten Datenelemente im Speicher basierend auf den Indikatoren zu speichern.

6. SIMD-Verarbeitungseinheit nach einem der vorhergehenden Ansprüche, wobei das Steuermodul ausgelegt ist, die Arbeitselemente zu den Verarbeitungsvorgängen zusammenzustellen, sodass Arbeitselemente eines Blocks von Arbeitselementen, die sich auf einen Block von Datenelementen be-

ziehen, zum selben Verarbeitungsvorgang gruppiert sind.

7. SIMD-Verarbeitungseinheit nach Anspruch 6, wobei das Steuermodul ausgelegt ist, die Arbeitselemente zu den Verarbeitungsvorgängen zusammenzustellen, sodass Blöcke von Arbeitselementen basierend auf der Anzahl von ungültigen Arbeitselementen in den entsprechenden Blöcken von Arbeitselementen zu Verarbeitungsvorgängen gruppiert sind.

8. SIMD-Verarbeitungseinheit nach Anspruch 6 oder 7, wobei das Steuermodul ausgelegt ist, die Arbeitselemente zu den Verarbeitungsvorgängen zusammenzustellen, sodass Arbeitselemente in einem Block von Arbeitselementen umgeordnet werden, um dadurch die ungültigen Arbeitselemente aus verschiedenen Blöcken von Arbeitselementen innerhalb eines Verarbeitungsvorgangs aufeinander auszurichten.

9. SIMD-Verarbeitungseinheit nach Anspruch 8, wobei das Steuermodul ausgelegt ist, Arbeitselemente in einem Block von Arbeitselementen umzuordnen, indem wenigstens eines von einem Rotationsvorgang und einem Tauschvorgang der Arbeitselemente im Block von Arbeitselementen durchgeführt wird.

10. SIMD-Verarbeitungseinheit nach Anspruch 9, wenn abhängig von Anspruch 4, wobei das Steuermodul ausgelegt ist, einen entsprechenden Indikator für eine Mehrzahl der Blöcke von Arbeitselementen zu setzen, um die Reihenfolge der Arbeitselemente innerhalb der Mehrzahl von Blöcken von Arbeitselementen anzuzeigen.

11. SIMD-Verarbeitungseinheit nach Anspruch 10, wobei jeder Block von Arbeitselementen vier Arbeitselemente umfasst und wobei jeder Indikator drei Bits umfasst: zwei Bits zum Definieren einer Rotation der Arbeitselemente innerhalb des Blocks von Arbeitselementen und ein Bit, um zu definieren, ob ein Tauschvorgang an zwei der Arbeitselemente an vorbestimmten Stellen innerhalb des Blocks von Arbeitselementen durchgeführt wurde.

12. SIMD-Verarbeitungseinheit nach einem der vorhergehenden Ansprüche, wobei mehr als zwei Ebenen Gültigkeit für die Arbeitselemente existieren und wobei das Steuermodul ausgelegt ist, die Arbeitselemente basierend auf der Gültigkeit der Arbeitselemente zu den Verarbeitungsvorgängen zusammenzustellen, sodass Arbeitselemente des bestimmten Verarbeitungsvorgangs, die dieselbe Gültigkeitsebene aufweisen, vorübergehend über die Gruppe von Verarbeitungsbahnen hinweg aufeinander ausgerichtet sind.

13. SIMD-Verarbeitungseinheit nach einem der vorhergehenden Ansprüche, wobei die Datenelemente Pixelwerte sind.

14. SIMD-Verarbeitungseinheit nach Anspruch 13, wobei der Block von Datenelementen ein Pixelquad ist.

15. SIMD-Verarbeitungseinheit nach einem der vorhergehenden Ansprüche, wobei einige der Verarbeitungsvorgänge weniger als die vorbestimmte Höchstanzahl von Arbeitselementen umfassen und wobei die SIMD-Verarbeitungseinheit umfasst: eine Mehrzahl von parallelen Gruppen von Verarbeitungsbahnen, wobei jede Gruppe ausgelegt ist, Befehle von Arbeitselementen eines entsprechenden Verarbeitungsvorgangs über eine Mehrzahl von Verarbeitungszyklen auszuführen; und Logik, die mit den Gruppen von Verarbeitungsbahnen gekoppelt ist, die ausgelegt ist, eine bestimmte Gruppe von Verarbeitungsbahnen zu veranlassen, einen bestimmten Verarbeitungszyklus zu überspringen, unabhängig von den anderen Gruppen von Verarbeitungsbahnen, falls keine Arbeitselemente zur Ausführung in einer der Verarbeitungsbahnen der bestimmten Gruppe im bestimmten Verarbeitungszyklus geplant sind.

16. Verfahren zur Verwendung einer SIMD-(Single Instruction Multiple Data)Verarbeitungseinheit, um eine Mehrzahl von Verarbeitungsvorgängen zu verarbeiten, die jeweils Arbeitselemente bis zu einer vorbestimmten Höchstanzahl umfassen, wobei die Arbeitselemente eines Verarbeitungsvorgangs ausgelegt sind, um eine gemeinsame Sequenz von Befehlen an entsprechenden Datenelementen auszuführen, wobei die Datenelemente in Blöcken von Datenelementen angeordnet sind, wobei einige der Blöcke wenigstens ein ungültiges Datenelement umfassen, wobei Arbeitselemente, die sich auf ungültige Datenelemente beziehen, ungültige Arbeitselemente sind und wobei die SIMD-Verarbeitungseinheit eine Gruppe von Verarbeitungsbahnen umfasst, die ausgelegt sind, Befehle von Arbeitselementen eines bestimmten Verarbeitungsvorgangs über eine Mehrzahl von Verarbeitungszyklen auszuführen, wobei das Verfahren umfasst:
Zusammenstellen der Arbeitselemente zu den Verarbeitungsvorgängen basierend auf der Gültigkeit der Arbeitselemente, sodass ungültige Arbeitselemente des bestimmten Verarbeitungsvorgangs über die Gruppe von Verarbeitungsbahnen hinweg vorübergehend aufeinander ausgerichtet sind; und
Ausführen von Befehlen von Arbeitselementen des bestimmten Verarbeitungsvorgangs unter Verwendung der Gruppe von Verarbeitungsbahnen.

17. Verfahren nach Anspruch 16, ferner umfassend ein Veranlassen der Gruppe von Verarbeitungsbahnen, die Ausführung eines Satzes von ungültigen

Arbeitselementen zu überspringen, falls der Satz von ungültigen Arbeitselementen die einzigen Arbeitselemente sind, die für die Ausführung in der Gruppe von Verarbeitungsbahnen in einem Verarbeitungszyklus geplant sind.

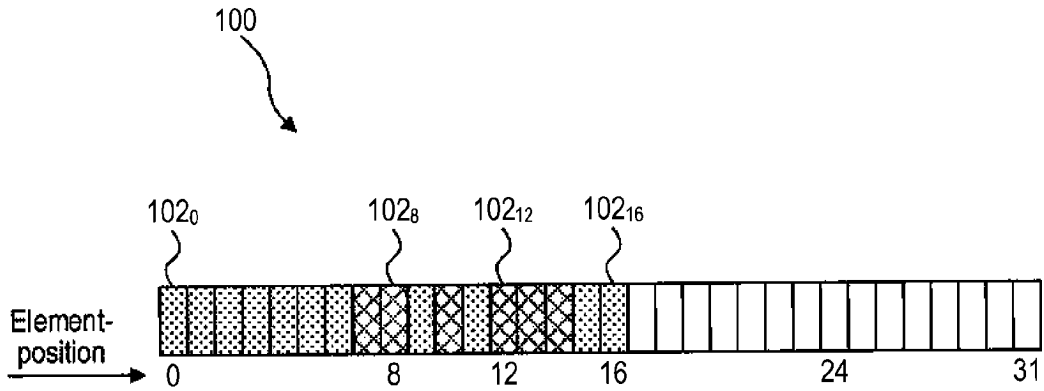
18. Verfahren nach Anspruch 16 oder 17, ferner umfassend ein Setzen von Indikatoren, um anzuzeigen, wie die Arbeitselemente zu den Verarbeitungsvorgängen zusammengestellt wurden.

19. Verfahren nach einem der Ansprüche 16 bis 18, wobei einige der Verarbeitungsvorgänge weniger als die vorbestimmte Höchstanzahl von Arbeitselementen umfassen und wobei die SIMD-Verarbeitungseinheit eine Mehrzahl von parallelen Gruppen von Verarbeitungsbahnen umfasst, wobei das Verfahren umfasst:
in jeder Gruppe von Verarbeitungsbahnen Ausführen von Befehlen von Arbeitselementen eines entsprechenden Verarbeitungsvorgangs in einer Mehrzahl von Verarbeitungszyklen; und
Veranlassen einer bestimmten Gruppe von Verarbeitungsbahnen, einen bestimmten Verarbeitungszyklus zu überspringen, unabhängig von den anderen Gruppen von Verarbeitungsbahnen, falls keine Arbeitselemente zur Ausführung in einer der Verarbeitungsbahnen der bestimmten Gruppe im bestimmten Verarbeitungszyklus geplant sind.

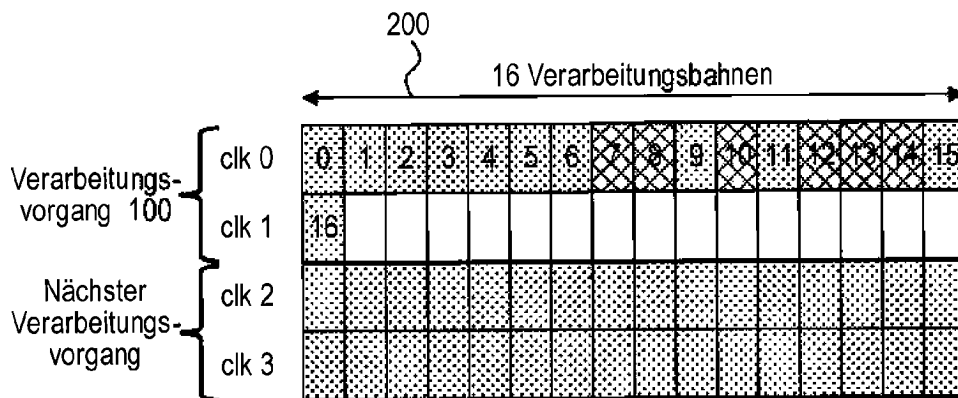
20. Computerlesbares Speichermedium, auf dem computerlesbarer Code zum Erzeugen einer SIMD-(Single Instruction Multiple Data)Verarbeitungseinheit gemäß einem der Ansprüche 1 bis 15 codiert ist.

Es folgen 9 Seiten Zeichnungen

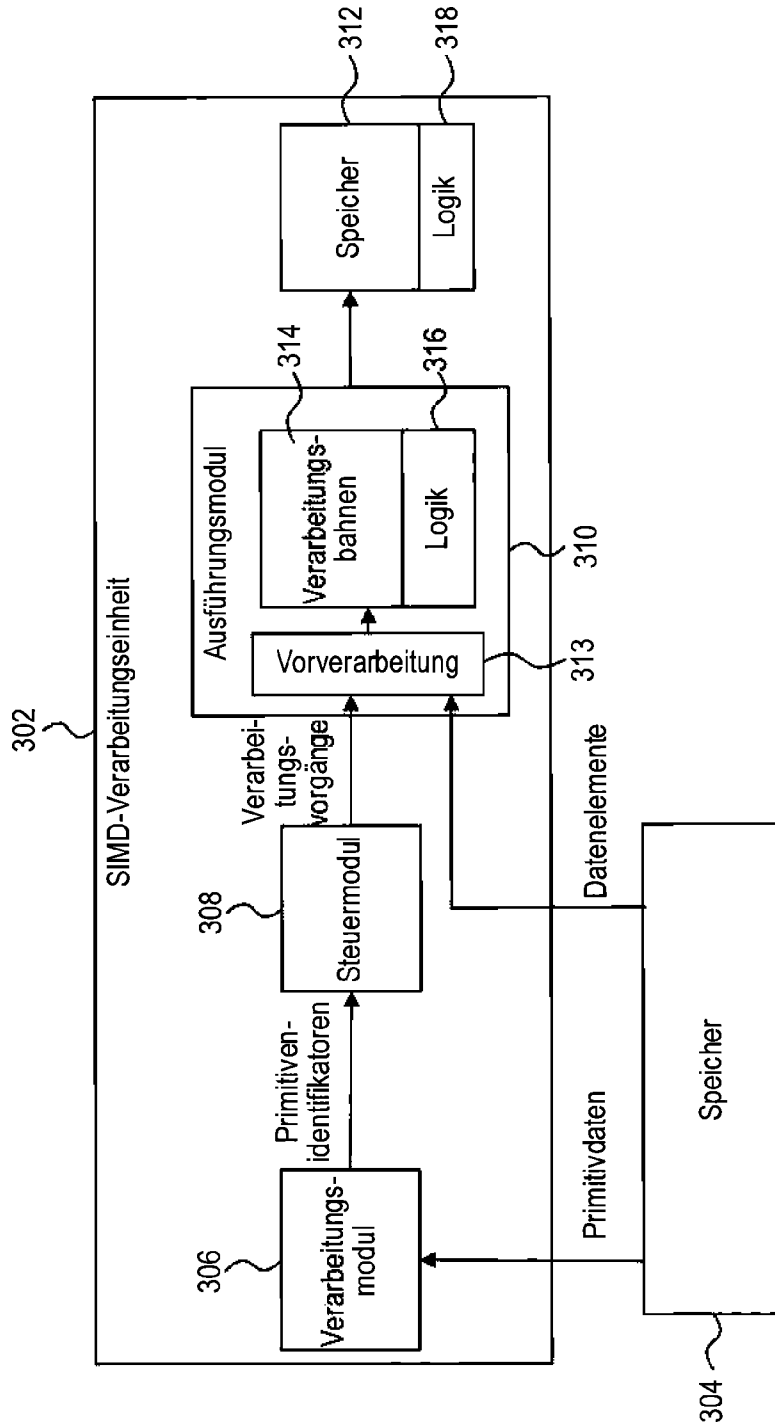
Anhängende Zeichnungen



FIGUR 1



FIGUR 2



FIGUR 3

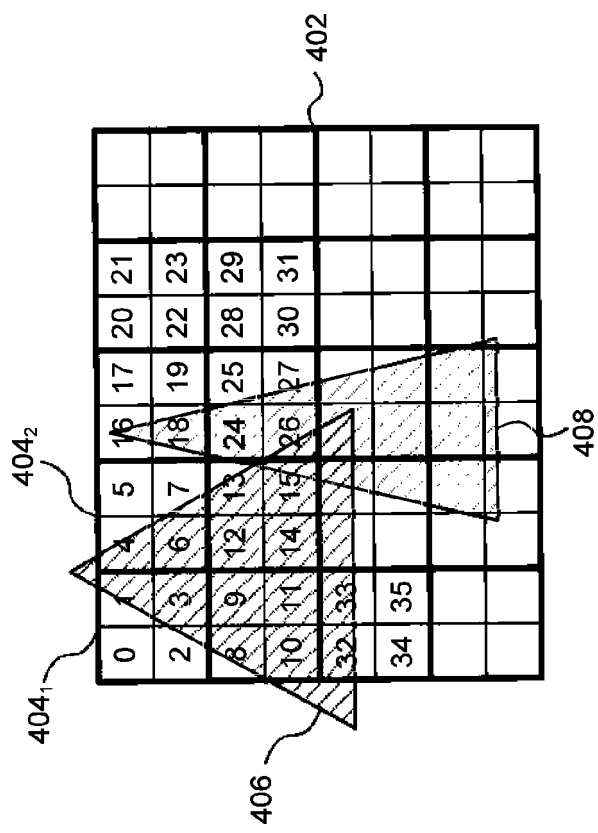
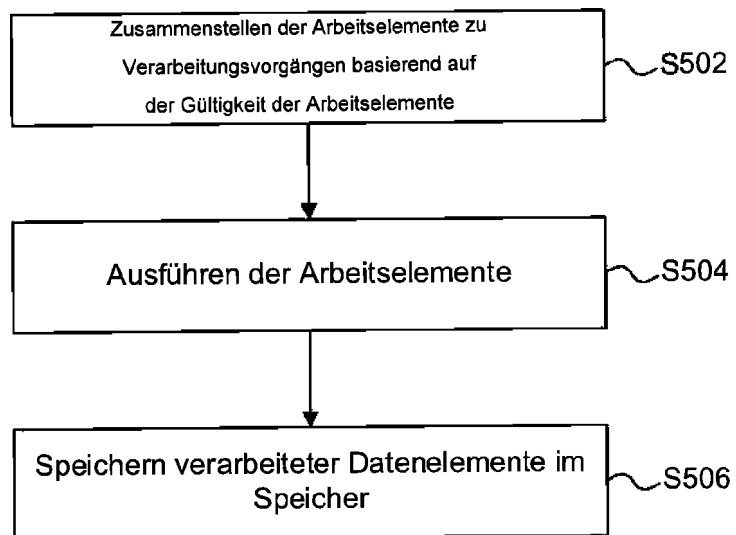
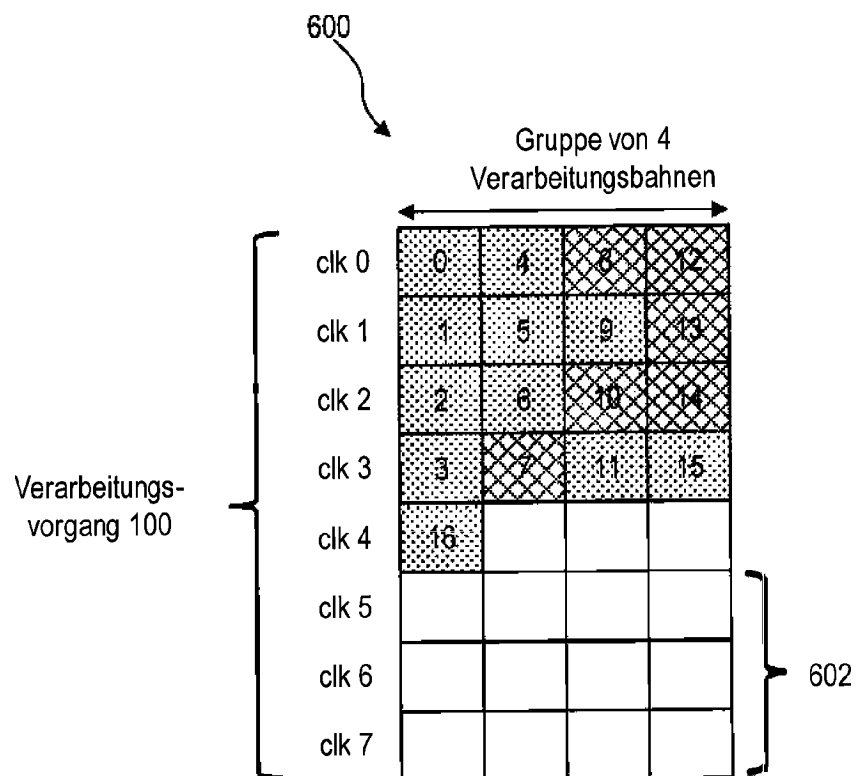


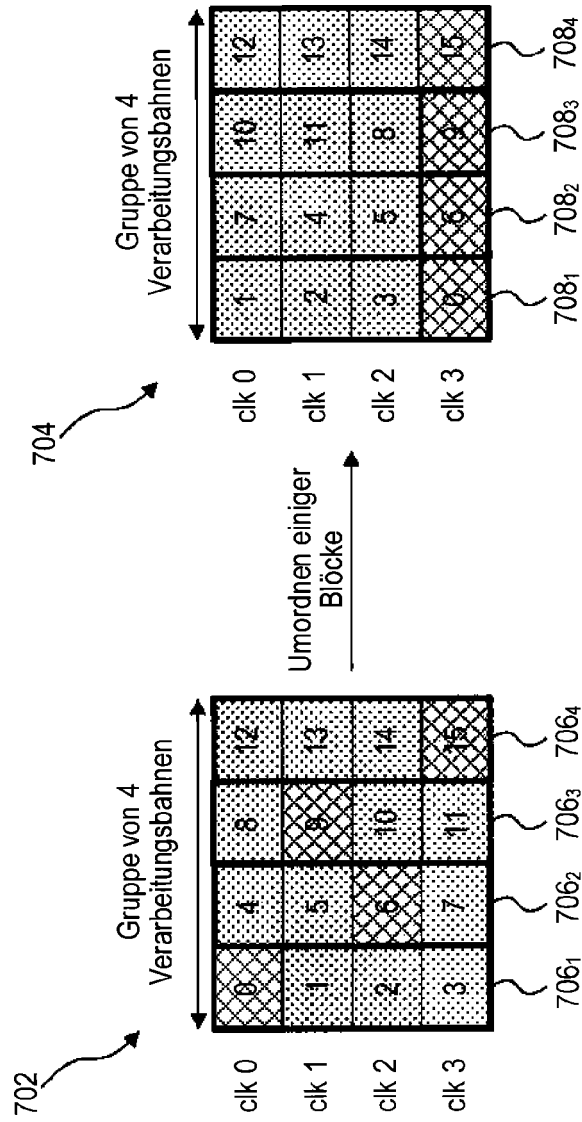
FIGURE 4



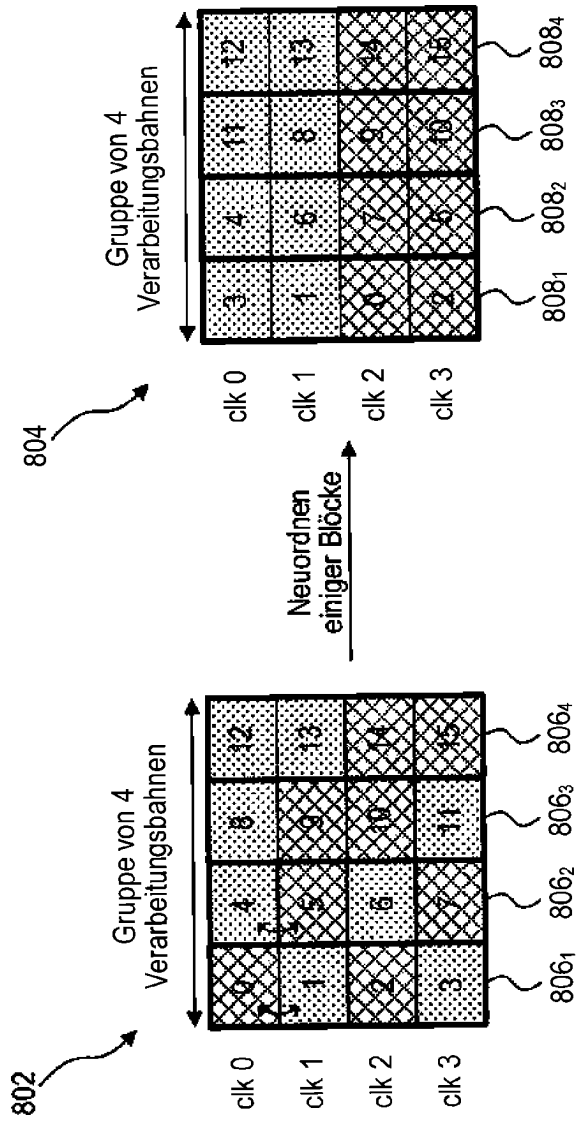
FIGUR 5



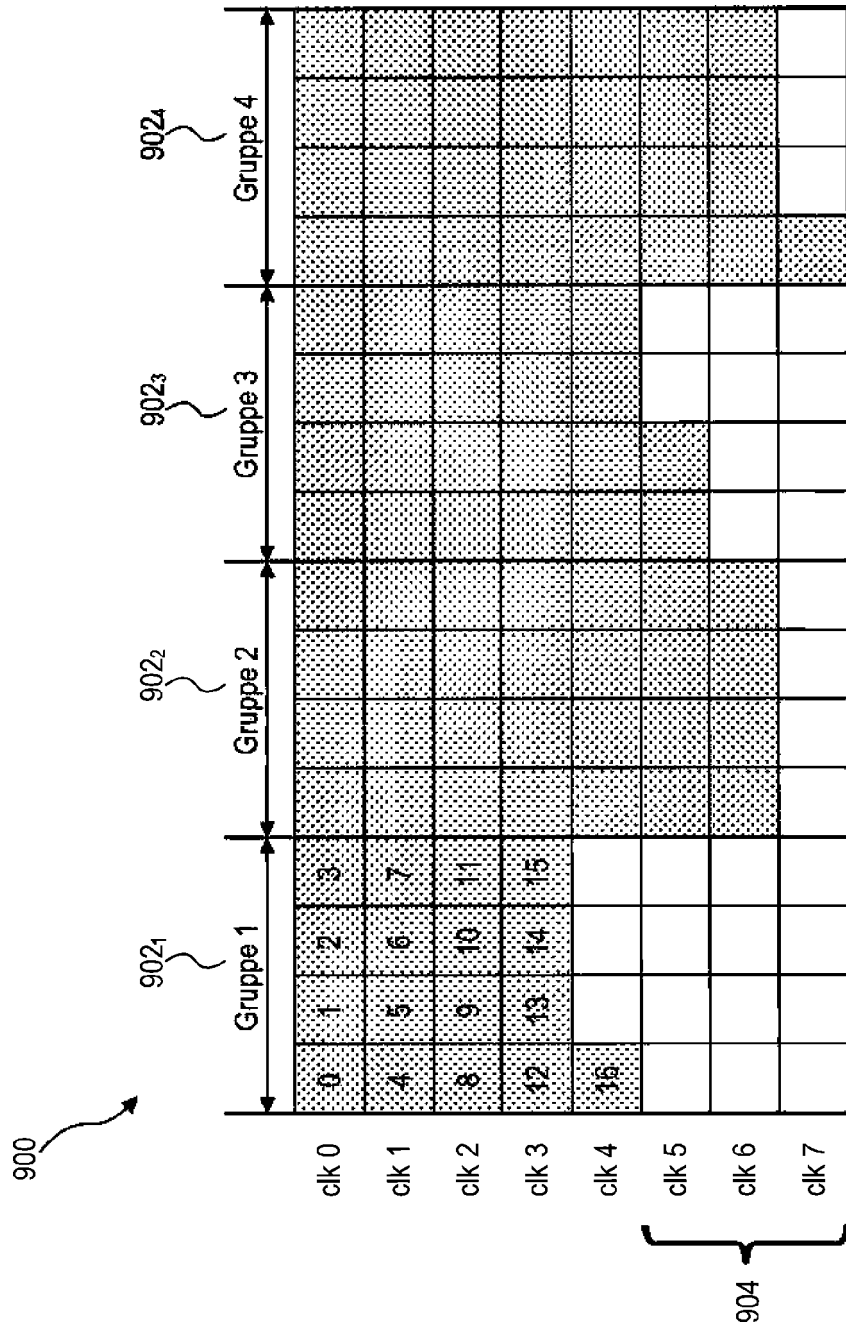
FIGUR 6



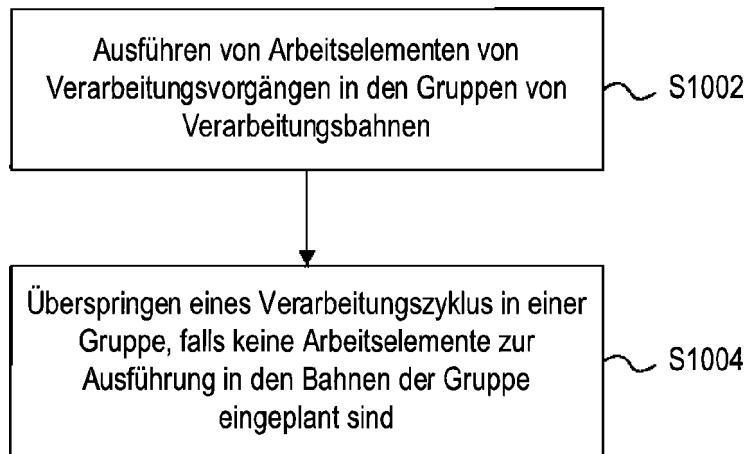
FIGUR 7



FIGUR 8



FIGUR 9



FIGUR 10