



(10) **DE 11 2011 104 770 B4** 2022.08.25

(12) **Patentschrift**

(21) Deutsches Aktenzeichen: **11 2011 104 770.3**
(86) PCT-Aktenzeichen: **PCT/CA2011/000080**
(87) PCT-Veröffentlichungs-Nr.: **WO 2012/100316**
(86) PCT-Anmeldetag: **25.01.2011**
(87) PCT-Veröffentlichungstag: **02.08.2012**
(43) Veröffentlichungstag der PCT Anmeldung
in deutscher Übersetzung: **31.10.2013**
(45) Veröffentlichungstag
der Patenterteilung: **25.08.2022**

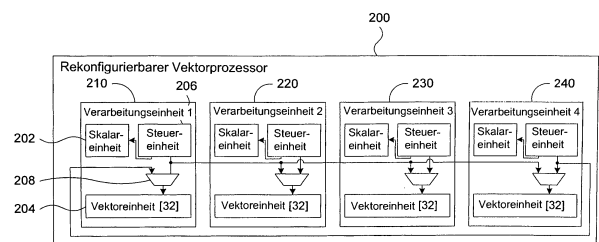
(51) Int Cl.: **G06F 15/76 (2006.01)**
G06F 15/16 (2006.01)
G06F 9/38 (2006.01)

Innerhalb von neun Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

<p>(73) Patentinhaber: NXP Canada Inc., Toronto, CA</p> <p>(74) Vertreter: Dilg, Haeusler, Schindelmann Patentanwalts-gesellschaft mbH, 80636 München, DE</p> <p>(72) Erfinder: Stewart, Malcom, Ottawa, Ontario, CA; Ors, Ali Osman, Ottawa, Ontario, CA; Laroche, Daniel, Kemptville, Ontario, CA</p> <p>(56) Ermittelte Stand der Technik: Nutt, G. J.: Microprocessor implementation of a parallel processor, Report CU-CS-091-76,</p>	<p>Department of Computer Science, University of Colorado, 1976. URL: http://www.cs.colorado.edu/depart-ment/publications/reports/docs/CU-CS-091-76.pdf [abgerufen am 04.08.2016] Schwederski, T.; Meyer, D. G.; Siegel, H. J.: Computer Architecture: Concepts and Systems. New York, NY : Elsevier Science Ltd, 1987. S. 178- 184, 191-193, 196-198, 204-207, 214-224. Kapitel 5: Parallel Processing. - ISBN 044401019X. http://www.engr.colostate.edu/~hj/book_chapters/11.pdf [abgerufen am 04.08.2016] Siegel, H. J. [u.a.]: An Overview of the PASM Parallel Processing System. In: Computer Architecture, 1987, S. 387-407. http://www.engr.colostate.edu/~hj/book_chapters/9.pdf [abgerufen am 04.08.2016]</p>
--	---

(54) Bezeichnung: **Apparat und Verfahren zum gemeinsamen Vektoreinheit Benutzen**

(57) Zusammenfassung: Ein rekonfigurierbarer Vektorprozessor wird beschrieben, welcher erlaubt, die Größe seiner Vektoreinheiten zu ändern, um Vektoren von unterschiedlichen Größen zu verarbeiten. Der rekonfigurierbare Vektorprozessor weist eine Mehrzahl von Prozessoreinheiten auf. Jede der Prozessoreinheiten weist eine Steuereinheit zum Dekodieren von Instruktionen und Generieren von Steuersignalen, eine Skalareinheit zum Verarbeiten von Instruktionen für Skalar-daten und eine Vektoreinheit zum Verarbeiten von Instruktionen unter Steuerung der generierten Steuersignalen auf. Die rekonfigurierbare Vektorprozessorarchitektur weist ebenfalls einen Vektorsteuerselektor zum selektiven Bereitstellen der Steuersignale, welche mittels einer Prozessoreinheit der Mehrzahl von Prozessoreinheiten generiert ist, an die Vektoreinheit einer unterschiedlichen Prozessoreinheit der Mehrzahl von Prozessoreinheiten auf.



Beschreibung

Technisches Gebiet

[0001] Die vorliegende Offenbarung betrifft ein Bilder- oder Daten-Verarbeiten und insbesondere auf einen Prozessor, welcher rekonfigurierbare Vektordaten Verarbeitungseinheiten bereitstellt.

Hintergrund

[0002] Das Verarbeiten von Bildern mag rechenkomplex sein und in einigen Anwendungen von dedizierten Prozessor- oder Verarbeitungstechniken profitieren. Um Bildinformation in Echtzeit oder fast in Echtzeit zu verarbeiten, mögen verschiedene Verarbeitungstechniken, bei dem Design und in der Implementation spezifischen Prozessoren für Bildanwendungen, verwendet werden. Zum Beispiel mögen bestimmte Prozessoren sowohl eine Skalar Verarbeitungseinheit als auch eine oder mehr Vektor Verarbeitungseinheiten beinhalten, welche eine Instruktion auf mehreren Stücken von Information auf einmal ausführen können. Die Verwendung einer Vektor Verarbeitungseinheit kann verbesserte Verarbeitungseffizienzen bereitstellen, insbesondere wenn die Vektorverarbeitung-Leistungsfähigkeit (vector processing capability) auf die Verarbeitungsaufgabe, welche bereitgestellt ist, abgestimmt ist.

[0003] Obwohl die Verwendung einer Vektoreinheit verbesserte Effizienzen bereitstellen mag, haben frühere Vektoreinheiten eine unveränderbare Verarbeitungsbreite im Sinne von Berechnungseinheiten (computational units) (CUs) gehabt, welche mit jeder Vektoreinheit assoziiert sind. Sofern die Aufgabe, welche mit der Vektoreinheit assoziiert ist, nicht von gleicher Breite war, oder umgekehrt, kann eine ineffiziente Prozessorbenutzung oder eine schwache Ausführungsperformanz resultieren.

[0004] Es gibt daher ein Bedürfnis für ein verbessertes gemeinsames Vektoreinheit Benutzen in Prozessoren.

Zusammenfassung

[0005] In Übereinstimmung mit einem Aspekt der vorliegenden Offenbarung, ist ein rekonfigurierbarer Vektorprozessor bereitgestellt, welcher aufweist: eine Mehrzahl von Prozessoreinheiten, wobei jede aufweist: eine Steuereinheit zum Dekodieren von Instruktionen und Generieren von Steuersignalen, eine Skalareinheit zum Verarbeiten von Instruktionen auf Skalarwerten, und eine Vektoreinheit zum Verarbeiten von Instruktionen auf Vektordaten basierend auf den generierten Steuersignalen, und einen Vektorsteuerselektor zum selektiven Bereitstellen von Steuersignalen, welche mittels einer der Mehrzahl von Prozessoreinheiten generiert ist, an eine Vektor-

einheit, welche mit einer unterschiedlichen Prozessoreinheit der Mehrzahl von Prozessoreinheiten assoziiert ist.

[0006] In Übereinstimmung mit einem Aspekt der vorliegenden Offenbarung ist ein Verfahren zum Verarbeiten von Daten bereitgestellt, welches einen rekonfigurierbaren Vektorprozessor verwendet, welcher zwei oder mehr Verarbeitungseinheiten aufweist, wobei jede eine Vektoreinheit hat, wobei das Verfahren aufweist: Konfigurieren des rekonfigurierbaren Vektorprozessors, um eine Vektoreinheit einer ersten Größe zum Verarbeiten von Vektordaten der ersten Größe bereitzustellen, Ausführen einer oder mehr Instruktionen, was die Vektoreinheit der ersten Größe verwendet, Rekonfigurieren des rekonfigurierbaren Vektorprozessors, um die Größe der Vektoreinheit zu einer zweiten Größe zu ändern, und Ausführen einer oder mehr Instruktionen, was die Vektoreinheit der zweiten Größe verwendet, um Vektordaten der zweiten Größe zu verarbeiten.

Figurenliste

[0007] Weitere Merkmale und Vorteile werden aus der folgenden detaillierten Beschreibung offensichtlich werden, wenn sie zusammen mit den angefügten Zeichnungen genommen wird, in welchen:

Fig. 1A einen Skalar in einem Blockdiagramm darstellt;

Fig. 1B einen Vektor in einem Blockdiagramm darstellt;

Fig. 2 einen rekonfigurierbaren Vektorprozessor in einem Blockdiagramm darstellt;

Fig. 3A eine Anordnung von verfügbaren Skalar- und Vektoreinheiten in einem Blockdiagramm darstellt;

Fig. 3B eine weitere Anordnung von verfügbaren Skalar- und Vektoreinheiten in einem Blockdiagramm darstellt;

Fig. 4 einen rekonfigurierbaren Vektorprozessor in einem Blockdiagramm darstellt;

Fig. 5A eine Anordnung von verfügbaren Skalar- und Vektoreinheiten von acht Verarbeitungseinheiten in einem Blockdiagramm darstellt;

Fig. 5B eine weitere Anordnung von verfügbaren Skalar- und Vektoreinheiten von acht Verarbeitungseinheiten in einem Blockdiagramm darstellt;

Fig. 6 ein weiteres Ausführungsbeispiel eines rekonfigurierbaren Vektorprozessors in einem Blockdiagramm darstellt;

Fig. 7 ein weiteres Ausführungsbeispiel eines rekonfigurierbaren Vektorprozessors in einem Blockdiagramm darstellt;

Fig. 8 ein System in einem Blockdiagramm darstellt, in welchem ein rekonfigurierbarer Vektorprozessor verwendet werden mag;

Fig. 9 eine APU mit Verbindungen zu verschiedenen Komponenten eines Systems in einem Blockdiagramm darstellt, welches einen rekonfigurierbaren Vektorprozessor verwendet;

Fig. 10 Komponenten eines ACP in einem Blockdiagramm darstellt;

Fig. 11 Komponenten einer CU in einem Blockdiagramm darstellt;

Fig. 12A mögliche Vektorgrößen darstellt, welche Verarbeitungseinheiten steuern mögen, falls jede Verarbeitungseinheit zum Steuern der Vektoreinheiten der anderen Verarbeitungseinheiten geeignet ist;

Fig. 12B mögliche Vektorgrößen darstellt, welche Verarbeitungseinheiten steuern mögen, falls nur die Verarbeitungseinheiten 0 und 2 zum Steuern der Vektoreinheiten der anderen Verarbeitungseinheiten geeignet sind;

Fig. 13 einen rekonfigurierbaren Vektorprozessor mit einem Crossbar-Schalter in einem Blockdiagramm darstellt;

Fig. 14 ein Teilsatz der 24 möglichen Vektoreinheit Steuerkonfigurationen für den rekonfigurierbaren Vektorprozessor von **Fig. 13** darstellt,

Fig. 15 ein beispielhaftes Bild darstellt, welches unter Verwendung der rekonfigurierbaren Vektorprozessor und Systeme verarbeitet werden mag;

Fig. 16 ein Verfahren zum Verarbeiten von Daten mit einem rekonfigurierbaren Vektorprozessor in einem Flowchart darstellt, und

Fig. 17 ein Verfahren zum Konfigurieren einer Größe einer Vektoreinheit in einem Flowchart darstellt.

Detaillierte Beschreibung

[0008] Ausführungsbeispiele eines Apparats, Systems und Verfahrens zum gemeinsamen Vektoreinheit Benutzen werden lediglich auf dem Wege eines Beispiels mit Bezug auf die **Fig. 1** bis **Fig. 17** beschrieben.

[0009] Ein rekonfigurierbarer Vektorprozessor wird beschrieben, welcher der Größe seiner Vektoreinheit erlaubt, geändert zu werden, um Vektoren von unterschiedlichen Größen zu verarbeiten. Die rekonfigurierbaren Vektorprozessoren weisen eine Mehrzahl

von Prozessoreinheiten auf. Jede der Prozessoreinheiten weist eine Steuereinheit zum Dekodieren von Instruktionen und zum Generieren von Steuersignalen, eine Skalareinheit zum Verarbeiten von Instruktionen auf Skalardaten, und eine Vektoreinheit zum Verarbeiten von Instruktionen auf Vektordaten unter Steuerung der Steuersignale auf. Der rekonfigurierbare Vektorprozessor weist auch einen Vektorsteuerselektor zum selektiven Bereitstellen von Steuersignalen, welche mittels einer Prozessoreinheit der Mehrzahl von Prozessoreinheiten generiert ist, an eine Vektoreinheit einer unterschiedlichen Prozessoreinheit der Mehrzahl von Prozessoreinheiten auf. Obwohl der Begriff Prozessor in der vorliegenden Offenbarung verwendet wird, mag ein Prozessor in einem Prozessor integriert sein, welcher einen oder mehr Kerne (cores) aufweist, oder als ein unabhängiger Stand-Alone Prozessor bereitgestellt sein oder funktionell zwischen mehreren physikalischen Prozessoren aufgeteilt sein, um die Funktionen des Vektorprozessors zu implementieren. Alternativ mag der Vektorprozessor eine Prozessorarchitektur definieren, welche benutzt wird, um die beschriebenen funktionalen Charakteristiken in einer Anzahl von physikalischen Implementierungen zu implementieren.

[0010] **Fig. 1A** stellt einen Skalar 100 in einem Blockdiagramm dar. Der Skalar 100 mag verwendet werden, um verschiedene Daten zu repräsentieren, welche verarbeitet werden können. Zum Beispiel mag der Skalar 100 eine Zahl sein, welche verwendet wird, um die Intensität einer Farbe in einem Abschnitt eines Bildes zu repräsentieren. Der Skalar 100 weist eine Mehrzahl von Bits auf. Der Skalar 100 ist dargestellt, eine Breite von 8 Bits zu haben, es wird jedoch gewürdigt werden, dass Skalare von anderen Breiten abhängig von der Architektur des Prozessors, welcher verwendet wird, möglich sind. Zum Beispiel mag ein Skalar 4 Bits, 8 Bits (typischerweise als ein Byte bezeichnet), 16 Bits, 32 Bits, 64 Bits, etc. sein.

[0011] **Fig. 1B** stellt einen Vektor 120 in einem Blockdiagramm dar. Der Vektor 120 weist eine Mehrzahl von skalaren Werten (datum) 122, 124, 126, 128 auf. Typischerweise hat jeder Skalar 122, 124, 126, 128 des Vektors 120 die gleiche Breite, zum Beispiel 8 Bits. Der Vektor 120, welche die Skalare 122, 124, 126, 128 aufweist, ist dargestellt, eine Vektorgröße von vier zu haben,. Es wird gewürdigt werden, dass Vektoren unterschiedlicher Größe sein mögen. Zum Beispiel mag ein Vektor aus einem oder mehr Skalaren bestehen, welche von 0 bis „Größe (-)-1“ („size (-)-1“) beziffert sind. Die Breite der Skalare, welche der Vektor aufweist, mag ebenfalls variieren. Zum Beispiel mag ein erster Vektor aus 4 Skalaren bestehen, wobei jeder 8 Bits breit ist; ein zweiter Vektor mag aus 2 Skalaren bestehen, wobei jeder 16 Bits

breit ist, und ein dritter Vektor mag aus 1 Skalar bestehen, welcher 32 Bits breit ist.

[0012] Die Gesamtbreite des Vektors wird basierend auf der Summe der Breite der Skalare des Vektors bestimmt. Eine Vektoreinheit, welche zum Verarbeiten des Vektors geeignet ist, muss ausgebildet sein, eine Anzahl von Skalaren der speziellen Breiten zu verarbeiten. Zum Beispiel würde eine Vektoreinheit, welche zum Verarbeiten des Vektors 120 geeignet sein würde, im Stande sein müssen, 4 Skalare, wobei jeder 8 Bits breit ist, zu verarbeiten. Frühere Vektoreinheiten sind von einer unveränderbaren Gesamtbreite, zum Beispiel einer unveränderbaren Gesamtbreite von 64 Bits, gewesen. Als solche würde sie zum Verarbeiten eines Vektors aus einem einzelnen 64 Bit Skalar oder eines Vektors aus zwei 32 Bit Skalaren, etc. geeignet sein. Jedoch war, mit der früheren Vektor Verarbeitungseinheit, falls nicht alle der Bits der Gesamtbreite benötigt waren, beispielsweise wenn es nur nötig ist, einen Vektor aus zwei 8 Bit Skalaren zu verarbeiten, die zusätzliche Verarbeitungsleistungsfähigkeiten der Vektor Verarbeitungseinheit nicht nutzbar.

[0013] Fig. 2 stellt einen rekonfigurierbaren Vektorprozessor 200 in einem Blockdiagramm dar. Der Vektorprozessor 200 weist eine Mehrzahl von Verarbeitungseinheiten 210, 220, 230, 240 auf. Jede Verarbeitungseinheit weist eine Skalareinheit 202, eine Vektoreinheit 204, eine Steuereinheit 206 und einen Vektor-Steuer-Multiplexer 208 auf.

[0014] Die Skalareinheit 202 ist zum Verarbeiten von Instruktionen auf Skalardaten geeignet. Zum Beispiel mag die Skalareinheit 202 zwei skalare Zahlen zusammen addieren. Die Skalareinheit 202 hat eine unveränderbare Breite, welche die maximale Breite eines Skalars ist, welche sie verarbeiten kann. Zum Beispiel mag eine Skalareinheit 8 Bits breit, 16 Bits breit, 32 Bits breit, 64 Bits breit, sein, obwohl andere Breiten möglich sind. Die Breite der Skalareinheit 202 bezieht sich auf die maximale Breite eines Skalars, welchen sie verarbeiten kann. Um einen oder mehr Skalare zu verarbeiten, dekodiert die Verarbeitungseinheit eine Instruktion, welche auszuführen ist, wie zum Beispiel Addiere A zu B, und steuert die Skalareinheit durch angemessene Steuersignale, um die gewünschte Instruktion aufzuführen.

[0015] Die Vektoreinheit 204 ist zum Verarbeiten von Instruktionen auf Vektordaten geeignet. Die Vektoreinheit 204, welche in der Fig. 2 dargestellt ist, ist gezeigt, eine Größe von 32 CUs zu haben. Für die Klarheit der Beschreibung wird jede Vektoreinheit beschrieben, im Stande zu sein, einen Vektor aus 32 CUs zu verarbeiten, wobei jede zum Verarbeiten von 16 Bit Skalaren geeignet ist, obwohl es gewürdigt werden wird, dass jede Vektoreinheit im Stande

ist, unterschiedliche Anordnungen von Vektoren, wie zum Beispiel acht 64 Bit Skalare, zu verarbeiten. Jede Vektoreinheit 204 ist zum Verarbeiten eines Vektors mit einer maximalen Gesamtvektorbreite von 512 Bit geeignet. Um einen oder mehr Vektoren zu verarbeiten, dekodiert eine der Verarbeitungseinheiten eine Instruktion, welche auszuführen ist, wie zum Beispiel addiere die Skalare des Vektors A zu den Skalaren des Vektors B, und steuert die Vektoreinheit durch geeignete Steuersignale, um die gewünschte Instruktion aufzuführen.

[0016] Die Steuereinheit 206 empfängt einen oder mehr Eingänge, wie zum Beispiel Instruktionen zum Ausführen, dekodiert sie und stellt sie an der Skalareinheit 202 und an der Vektoreinheit 204 als angemessen bereit.

[0017] Die Verarbeitungseinheiten 210, 220, 230, 240 sind dargestellt, jeweils einen Vektor-Steuer-Multiplexer 208 aufzuweisen. Für die Klarheit der Beschreibung wird der Vektor-Steuer-Multiplexer 208 jeder Verarbeitungseinheit 210, 220, 230, 240 dargestellt, zwei Steuereingänge an einen einzelnen Steuereingang zu multiplexen, welcher von der jeweiligen Vektoreinheiten 204 erwartet wird. Jeder Vektor-Steuer-Multiplexer empfängt Steuersignale von der Steuereinheit der jeweiligen Verarbeitungseinheit, welche für die Klarheit als die interne Steuereinheit bezeichnet werden mag. Der Vektor-Steuer-Multiplexer 208 mag ebenfalls Steuersignale von einer oder mehr Steuereinheiten von anderen Verarbeitungseinheiten empfangen, welche für die Klarheit als die externe Steuereinheit bezeichnet werden mag. Jeder Vektor-Steuer-Multiplexer 208, welcher in der Fig. 2 dargestellt ist, empfängt zwei Steuersignale, eines aus der internen Steuereinheit und eines aus einer externen Steuereinheit. In der Fig. 2 empfängt jeder der Vektor-Steuer-Multiplexer 208 der Verarbeitungseinheiten 220, 230 und 240 Steuersignale aus der externen Steuereinheit der Verarbeitungseinheit 210. Der Vektor-Steuer-Multiplexer der Verarbeitungseinheiten 210 empfängt Steuersignale aus der externen Steuereinheit der Verarbeitungseinheit 240.

[0018] Mittels geeigneten Steuerns des Vektor-Steuer-Multiplexers 208 von jeder Verarbeitungseinheit kann die Vektorgröße vergrößert werden, welche mittels einer Vektoreinheit verarbeitet werden kann. Dies erlaubt vorteilhafterweise den Verarbeitungsleistungsfähigkeiten wie benötigt anpassbar zu sein. Zum Beispiel mag der Vektor-Steuer-Multiplexer der Verarbeitungseinheiten 220, 230 und 240 so gesteuert werden, dass sie jeweils die Steuersignale aus der externen Steuereinheit der Verarbeitungseinheit 210 an die jeweiligen Vektoreinheiten bereitstellen. Dies stellt eine Vektoreinheit bereit, welche einen größer ausgelegten Vektor, nämlich 128 CUs, verarbeiten kann. Falls die maximale Vektorgröße

nicht benötigt wird, mögen individuelle Vektoreinheiten an die Steuerung der jeweiligen internen Steuereinheiten zurückgegeben werden. Als solches mag die Verarbeitungseffizienz vergrößert werden. Zusätzlich mag die Verarbeitungseinheit, während eine Vektoreinheit unter der Steuerung einer externen Steuereinheit ist, weiterhin als eine Skalareinheit betrieben werden. Alternativ kann die Steuereinheit sowohl die Skalar als auch die Vektoreinheit(en) gleichzeitig betreiben.

[0019] Fig. 3A stellt eine Anordnung von verfügbaren Skalar- und Vektoreinheiten in einem Blockdiagramm dar. **Fig. 3A** stellt eine mögliche Anordnung von vier Verarbeitungseinheiten dar. Wie dargestellt ist, ist die Vektoreinheit von jeder Verarbeitungseinheit unter Steuerung der jeweiligen internen Steuereinheiten. Jede Verarbeitungseinheit stellt eine Skalareinheit und eine Vektoreinheit zum Verarbeiten eines Vektors einer Größe 32 bereit, so dass vier separate Skalare oder vier separate Vektoren, jeder mit der Größe 32, oder eine Kombination davon simultan verarbeitet werden mag. Eine Verarbeitungseinheit betreibt typischerweise die Skalareinheit und die Vektoreinheit individuell. Jedoch könnten abhängig von dem spezifischen Design der Steuereinheit sowohl die Vektoreinheit als auch die Skalareinheit gleichzeitig laufen gelassen werden. Des Weiteren ist es möglich, falls die Vektoreinheit unter der Steuerung einer Steuereinheit einer unterschiedlichen Verarbeitungseinheit betrieben wird, die Skalareinheit unter Verwendung der Steuereinheit zu steuern. Des Weiteren mag eine Steuereinheit sowohl die Skalareinheit betreiben als auch eine oder mehr andere Vektoreinheiten zur gleichen Zeit steuern.

[0020] Fig. 3B stellt eine weitere Anordnung von verfügbaren Skalar- und Vektoreinheiten in einem Blockdiagramm dar. **Fig. 3B** stellt ebenfalls eine mögliche Anordnung von vier Verarbeitungseinheiten dar. In der Anordnung von **Fig. 3B** sind die Vektoreinheiten der Verarbeitungseinheiten 220 und 230 übernommen worden und sind unter der externen Steuerung der Steuereinheit der Verarbeitungseinheit 210. Als ein Resultat sind vier Skalareinheiten zum Verarbeiten verfügbar. Nur zwei Vektoreinheiten sind zum Verarbeiten verfügbar, jedoch hat eine der Verarbeitungseinheiten eine maximale Vektorgröße von 96 mittels Kombinierens der CU der assoziierten Vektoreinheit, während die zweite Vektoreinheit eine maximale Vektorgröße von 32 hat.

[0021] Wie aus den **Fig. 3A** und **Fig. 3B** offensichtlich ist, mag die maximale Vektorgröße geändert werden. Diese Änderung mag auf verschiedenen Faktoren basieren, wie zum Beispiel dem gewünschten Energiegebrauch, wie zeitkritisch das Verarbeiten ist, die Komplexität der Verarbeitung, die Größe der zu verarbeitenden Daten, zusätzliches Verarbeiten,

welches benötigt wird, durchgeführt zu werden, etc. Ohne Rücksicht auf die Gründe für das Anpassen der Größe der verfügbaren Vektoreinheiten, wird es gewürdigt werden, dass die Vektoreinheitgröße angepasst werden kann, während es den Verarbeitungseinheiten, deren jeweilige Vektoreinheiten übernommen worden sind, erlaubt wird, weiterhin die Verarbeitungsleistungsfähigkeiten der Skalareinheit bereitzustellen.

[0022] Der rekonfigurierbare Vektorprozessor 200, in der **Fig. 2**, welcher oben beschrieben ist, war beschrieben, vier im Wesentlichen identische Verarbeitungseinheiten 210, 220, 230, 240 zu haben. Wie weiter unten beschrieben wird, ist es möglich, einen rekonfigurierbaren Vektorprozessor mit unterschiedlichen Konfigurationen der Verarbeitungseinheiten bereitzustellen.

[0023] Fig. 4 stellt einen rekonfigurierbaren Vektorprozessor 400 in einem Blockdiagramm dar. Der rekonfigurierbare Vektorprozessor 400 ist darin ähnlich zu dem rekonfigurierbaren Vektorprozessor 200, dass die Vektorgröße der verfügbaren Vektoreinheiten geändert werden mag. Jedoch benutzt, im Gegensatz zu dem rekonfigurierbaren Vektorprozessor 200, der rekonfigurierbare Vektorprozessor 400 unterschiedliche Verarbeitungseinheiten. Wie dargestellt ist, beinhaltet der rekonfigurierbare Vektorprozessor 400 zwei Master Verarbeitungseinheiten 402, 404 und sechs Slave Verarbeitungseinheiten 406, 408, 410, 412, 414, 416. Die zwei Master Verarbeitungseinheiten 402, 404 sind ähnlich zu den Verarbeitungseinheiten 210, 220, 230, 240, jedoch haben die zwei Master Verarbeitungseinheiten keinen Vektor-Steuer-Multiplexer. Als solches wird die Vektoreinheit von jeder Master Verarbeitungseinheit 402, 404 immer unter der Steuerung der jeweiligen internen Steuereinheiten sein.

[0024] Jede der Slave Verarbeitungseinheiten 406, 408, 410, 412, 414, 416 sind darin ähnlich zu den Verarbeitungseinheiten 210, 220, 230, 240, dass jede Verarbeitungseinheit einen Vektor-Steuer-Multiplexer beinhaltet, welcher den Vektoreinheiten der Slave Verarbeitungseinheiten erlaubt, mittels einer der Steuereinheiten der Master Verarbeitungseinheiten gesteuert zu werden. Wie dargestellt ist, beinhaltet jeder Vektor-Steuer-Multiplexer drei Eingänge von Steuersignalen. Die Steuersignale sind aus der internen Steuereinheit von jeder Slave Verarbeitungseinheit und der externen Steuereinheit von jeder Master Verarbeitungseinheit 402, 404. Unter der angemessenen Steuerung stellt der Steuervektor Multiplexer die gewünschten Steuersignale an den jeweiligen Vektoreinheiten bereit, welche es der Größe der Vektoren erlauben, welche verarbeitet werden können, geändert zu werden.

[0025] Die Vektoreinheiten sowohl der Master als auch der Slave Verarbeitungseinheiten sind in der **Fig. 4** dargestellt, dass sie von der Größe 16 CUs sind.

[0026] Es sollte beachtet werden, dass der rekonfigurierbare Vektorprozessor 200, **Fig. 2**, so betrieben werden kann, dass nur eine einzelne Vektoreinheit der Größe 128 (vier mal 32) verfügbar ist (zusätzlich zu den Skalareinheiten). Jedoch wird der rekonfigurierbare Vektorprozessor 400, **Figur 400**, immer ein Minimum von zwei Vektoreinheiten haben, da die Vektoreinheiten jedes Master Prozessors 402, 404 immer unter der Steuerung der jeweiligen internen Steuereinheit sind.

[0027] **Fig. 5A** stellt eine Anordnung von verfügbaren Skalar- und Vektoreinheiten von acht Verarbeitungseinheiten in einem Blockdiagramm dar, welche zwei Master Verarbeitungseinheiten und sechs Slave Verarbeitungseinheiten aufweisen. Wie in der **Fig. 5A** dargestellt ist, hat die Master Verarbeitungseinheit 402 die Steuerung der Vektoreinheiten von zwei Slave Verarbeitungseinheiten 406 und 408 übernommen. Als solches kann die Master Verarbeitungseinheit 402 eine Vektoreinheit mit einer Größe von 48 bereitstellen. Die Master Verarbeitungseinheit 404 hat die Steuerung der Vektoreinheit von einer Slave Verarbeitungseinheit 412 übernommen. Als solches kann die Master Verarbeitungseinheit 404 eine Vektoreinheit mit einer Größe von 32 bereitstellen. Die verbleibenden Slave Verarbeitungseinheiten 408, 410, 414, 416 mögen entweder als eine Vektoreinheit der Größe 16 oder eine Skalareinheit oder beides betrieben werden abhängig von den jeweiligen Steuereinheiten. Die internen Steuereinheiten jeder Slave Verarbeitungseinheit 406, 408, 412, welche ihre Vektoreinheiten unter externer Steuerung haben, sind zum Betreiben der Verarbeitungseinheiten als Skalareinheiten geeignet.

[0028] **Fig. 5B** stellt eine weitere Anordnung von verfügbaren Skalar- und Vektoreinheiten von acht Verarbeitungseinheiten in einem Blockdiagramm dar, welche zwei Master Verarbeitungseinheiten und sechs Slave Verarbeitungseinheiten aufweisen. Wie in der **Fig. 5B** dargestellt ist, hat die Master Verarbeitungseinheit 402 die Steuerung der Vektoreinheiten von allen der Slave Verarbeitungseinheiten 406, 408, 410, 412, 414, 416 übernommen. Als solche kann die Master Verarbeitungseinheit 402 eine Vektoreinheit mit einer Größe von 112 bereitstellen. Die Master Verarbeitungseinheit 404 hat nicht die Steuerung der Vektoreinheiten von irgendeiner der Slave Verarbeitungseinheiten übernommen. Als solche kann die Master Verarbeitungseinheit 404 eine Vektoreinheit mit einer Größe von 16 bereitstellen. Die internen Steuereinheiten jeder Slave Verarbeitungseinheit 406, 408, 410, 412, 414, 416, welche ihre Vektoreinheiten unter externer Steuerung

haben, sind zum Betreiben der Verarbeitungseinheiten als Skalareinheiten geeignet.

[0029] Das Obere hat verschiedene rekonfigurierbare Vektorprozessoren beschrieben, welche eine Mehrzahl von Verarbeitungseinheiten jede mit einem Vektor-Steuer-Multiplexer aufweisen. Abhängig von den Instruktionen, welche mittels der Vektoreinheiten implementiert sind, mag es wünschenswert oder notwendig sein, zusätzliche Multiplexer aufzunehmen.

[0030] **Fig. 6** stellt ein weiteres Ausführungsbeispiel eines rekonfigurierbaren Vektorprozessors in einem Blockdiagramm dar. Der rekonfigurierbare Prozessor 600 mag abhängig von dem Typ der Instruktionen verwendet werden, welche die Vektoreinheiten durchführen können. Zum Beispiel mag es brauchbar sein, wenn die Vektoreinheiten zum Durchführen einer Verschiebe-nach-rechts-Instruktion geeignet sind, welche die individuellen Elemente oder Skalare des Vektors veranlasst, sich eine Position nach rechts zu verschieben. Das letzte Element mag auf das erste Element verschoben werden. Die Skalareinheiten der Verarbeitungseinheiten sind, für die Klarheit der Beschreibung, in der **Fig. 6** weggelassen worden. Der rekonfigurierbare Vektorprozessor 600 weist eine Master Verarbeitungseinheit 602 und zwei Slave Verarbeitungseinheiten 606, 608 auf. Jede Slave Verarbeitungseinheit 606, 608 beinhaltet einen Vektor-Steuer-Multiplexer zum Auswählen der angemessenen Steuereinheit, welche die Vektoreinheit steuert, wie vorhergehend beschrieben ist.

[0031] Die Verarbeitungseinheiten 602, 606, 608 beinhalten jeweils einen jeweiligen ersten Datenmultiplexer 604, 610, 612. Die Master Verarbeitungseinheit 602 beinhaltet den ersten Eingangsdatenmultiplexer 604. Die Eingänge des ersten Eingangsdatenmultiplexers sind die Ausgänge von jeder der Vektoreinheiten, welche die Master Verarbeitungseinheit 602 steuern kann. In dem Ausführungsbeispiel der **Fig. 6** beinhaltet dies sowohl die Ausgänge aus den beiden Slave Verarbeitungseinheiten 606, 608 als auch die Vektoreinheit der Master Verarbeitungseinheit 602 selber.

[0032] Die Slave Verarbeitungseinheiten 606, 608 beinhalten auch erste Eingangsdatenmultiplexer 610, 612. Jedoch sind die Eingänge der ersten Datenmultiplexer 610, 612 entweder der Ausgang aus der Vektoreinheit der jeweiligen Slave Verarbeitungseinheiten oder der Ausgang aus der Vektoreinheit einer früheren Verarbeitungseinheit in einer Kette von kombinierten Vektoreinheiten. Es sollte beachtet werden, dass die Eingänge der ersten Datenmultiplexer der Slave Verarbeitungseinheiten bestimmen, welche Slave Verarbeitungseinheiten zusammen kombiniert werden können und das volle Instruktionssset bereitstellen können, zum Beispiel

einschließlich der Verschiebe-nach-rechts-Instruktion, welche den ersten Datenmultiplexer benötigen mag.

[0033] Mit Bezug auf die **Fig. 6** kann, falls eine Verschiebe-nach-rechts-Funktion die ersten Datenmultiplexer benötigt, die Master Verarbeitungseinheit nicht die Vektoreinheit nur der Slave Verarbeitungseinheit 608 steuern, da dies als Eingang den Ausgang aus der Vektoreinheit der Master Verarbeitungseinheit benötigen würde, welcher für den ersten Datenmultiplexer nicht als Eingang verfügbar ist. Die Master Verarbeitungseinheit 602 könnte die Vektoreinheit der Slave Verarbeitungseinheit 606 steuern, welche einen Eingang an den ersten Daten Multiplexer 610 aus der Vektoreinheit der Master Verarbeitungseinheit beinhaltet. Die Master Verarbeitungseinheit 602 könnte auch die Vektoreinheiten der beiden Slave Verarbeitungseinheiten 606, 608 steuern, da ein Eingang des ersten Daten Multiplexers der Slave Verarbeitungseinheit 606 der Ausgang der Master Verarbeitungseinheiten 602 Vektoreinheit ist und ein Eingang des ersten Daten Multiplexers der Slave Verarbeitungseinheit 608 der Ausgang der Vektoreinheit der früheren Verarbeitungseinheit in dem kombinierten Vektor, nämlich die Slave Verarbeitungseinheit 606, ist. Als ein Resultat der Kette von Vektoreinheiten, welche mit den ersten Daten Multiplexern 604, 610, 612 erzeugt sind, ist es möglich, Instruktionen zu implementieren, welche ein Verschieben der Elemente eines Vektors benötigen, während die Flexibilität eines Geeignets-eins, die Vektoreinheitsgröße zu ändern, beibehalten bleibt. Es wird gewürdigt werden, dass die Flexibilität, mittels derer Verarbeitungseinheiten unabhängig gesteuert werden können, basierend auf den Eingängen an die ersten Daten Multiplexer von jeder der Verarbeitungseinheiten bestimmt wird. So falls zum Beispiel der ersten Daten Multiplexer 612 einen Eingang aus dem Ausgang der Vektoreinheit der Master Verarbeitungseinheit beinhaltet, dann könnte der Master Prozessor die Vektoreinheit und die assoziierten Berechnungseinheiten der Slave Verarbeitungseinheit 608 unabhängig davon steuern, ob oder ob nicht die Slave Verarbeitungseinheit 606 auch unter Steuerung der Master Verarbeitungseinheit 602 ist.

[0034] **Fig. 7** stellt ein weiteres Ausführungsbeispiel eines rekonfigurierbaren Vektorprozessors in einem Blockdiagramm dar. Der rekonfigurierbare Prozessor 700 mag abhängig von dem Typ der Instruktionen verwendet werden, welche die Vektoreinheit durchführen kann. Zum Beispiel mag es brauchbar sein, wenn die Vektoreinheiten zum Durchführen von sowohl einer Verschiebe-nach-links und einer Verschiebe-nach-rechts Instruktion geeignet sind. Um sowohl ein Verschieben-nach-links und ein Verschieben-nach-rechts durchzuführen, muss eine Vektor-

einheit im Stande sein, das letzte Element in einem Vektor mit dem ersten Element in einem Vektor, und umgekehrt, zu ersetzen. Dies ist schematisch in der **Fig. 7** durch die Verwendung eines Paares von Multiplexern - einen für jede Richtung - repräsentiert.

[0035] Der rekonfigurierbare Vektorprozessor 700 weist vier Verarbeitungseinheiten 702, 704, 706, 708 auf. Die Verarbeitungseinheit 702 fungiert als eine Master Verarbeitungseinheit und mag die Steuerung der Vektoreinheiten von einer oder mehr der Verarbeitungseinheiten 704, 706, 708 übernehmen, welche als Slave Verarbeitungseinheiten fungieren. Jede der Verarbeitungseinheiten 702, 704, 706, 708 weist einen ersten Daten Multiplexer 710 a, 710 b, 710 c, 710 d (insgesamt als 710 bezeichnet) und einen zweiten Daten Multiplexer 712 a, 712 b, 712 c, 712 d (insgesamt als 712 bezeichnet) auf. Sowohl die ersten als auch die zweiten Daten Multiplexer empfangen als Eingang den Ausgang von den Vektoreinheiten der Verarbeitungseinheiten. Die ersten und die zweiten Daten Multiplexer 710, 712 erlauben es, eine Kette von Vektoreinheiten zu erzeugen, welche Elemente zwischen den Vektoreinheiten nach links und nach rechts verschieben kann. Wie bei den ersten Daten Multiplexer 604, 610, 612 der rekonfigurierbaren Vektoreinheit 600, **Fig. 6**, bestimmen die Eingänge zu den verschiedenen ersten und zweiten Daten Multiplexern 710, 712, welche Vektoreinheiten mittels der Master Verarbeitungseinheit 702 gesteuert werden können, unabhängig davon, welche anderen Vektoreinheiten gesteuert werden. Zum Beispiel, wenn die Vektoreinheit der Slave Verarbeitungseinheit 708 mittels der Steuereinheit der Master Verarbeitungseinheit 702 zu steuern ist, muss sie auch die Vektoreinheiten der anderen Slave Verarbeitungseinheiten 704, 706 steuern, so dass die Vektoreinheiten zusammen in einer Kette verbunden sein können, welche die ersten und zweiten Datenmultiplexer 710, 712 verwendet.

[0036] **Fig. 8** stellt ein System in einem Blockdiagramm dar, in welchem ein rekonfigurierbarer Vektorprozessor verwendet werden mag. Das System 800 weist einen Mikroprozessor 802 auf, welcher Instruktionen ausführen kann, welche in einem Speicher (nicht gezeigt) gespeichert sind. Der Prozessor 802 mag zum Beispiel ein ARM™ Prozessor sein, obwohl andere Prozessoren oder Mikroprozessoren verwendet sein mögen. Das System beinhaltet ferner einen Sequenzer (sequencer) 804, welcher Instruktionen für den rekonfigurierbaren Vektorprozessor zum Ausführen empfängt. Der Sequenzer 804 stellt die angemessenen Steuersignale an den verschiedenen Array Verarbeitungseinheiten (APU) 806, 808, 810, 812 bereit, um die Instruktionen in der richtigen Sequenz/Ordnung auszuführen. Jeder der APUs 806, 808, 810, 812 ist ähnlich in einer Funktionalität zu den Verarbeitungseinheiten, welche oben mit Bezug auf die rekonfigurierbaren Vektorpro-

zessoren 200, 400, 600, 700 beschrieben sind. Jede APU weist sowohl einen Array-Steuerprozessor (ACP) 814 Prozessor als auch die Steuereinheit auf, welche die Instruktionen dekodiert und geeignete Steuersignale an die Berechnungseinheiten (CUs) 816 bereitstellt. Jede Gruppe von CUs 816 der APUs 806, 808, 810, 812 fungiert als eine Vektoreinheit, wie oben beschrieben ist. Jede APU ist dargestellt, 32 CUs zu haben, was eine Vektoreinheit bereitstellt, welche eine Größe von 32 hat. Jede Gruppe von CUs kann zusammen kombiniert werden und mittels einer einzelnen Steuereinheit gesteuert werden, welche einen ACP bereitstellt, und so ein Verarbeiten von Vektoren von unterschiedlichen Größen, zum Beispiel 32, 64, 96 und 128 bereitstellen.

[0037] Der Prozessor 802 kann Instruktionen an den Sequenzer 804 bereitstellen, welche mittels der APUs 806, 808, 810, 812 auszuführen sind. Zum Beispiel mag der Prozessor 802 Instruktionen an den Sequenzer 804 bereitstellen, vier unterschiedliche Instruktionen auf vier unterschiedlichen jeweiligen Vektoren auszuführen, wobei jeder eine Größe von 32 hat. Alternativ mag der Prozessor 802 Instruktionen bereitstellen, eine Instruktion auf einem einzelnen Vektor der Größe 128 und drei unterschiedliche Instruktionen auf drei unterschiedlichen Skalaren auszuführen.

[0038] Wie oben beschrieben ist, kann das System 800 die Vektorgröße rekonfigurieren, welche verarbeitet werden kann, um eine effiziente Verteilung der Verarbeitungsressourcen bereitzustellen. Wenn die APUs rekonfiguriert werden, um ihren jeweiligen CUs zu erlauben, mittels einer unterschiedlichen APU gesteuert zu werden, mag die Skalareinheit, welche mittels der ACPs bereitgestellt ist, weiterhin verwendet werden, um ein Verarbeiten eines oder mehr Skalaren bereitzustellen.

[0039] Fig. 9 stellte eine APU mit Verbindungen zu verschiedenen Komponenten eines Systems in einem Blockdiagramm dar, welches einen rekonfigurierbaren Vektorprozessor verwendet. Die APU 902 mag als eine der APUs 806, 808, 810, 812 des Systems 800 verwendet werden. Die APU 902 ist mit einem fortgeschrittenen erweiterbaren Interface (advanced extensible interface) (AXI) Bus 904 verbunden, welcher der APU erlaubt, Daten zu und von anderen Komponenten eines Systems zu transferieren, welche mit dem AXI Bus 904 verbunden sind. Obwohl er als ein AXI Bus beschrieben ist, mag irgendein anderer Typ von Bus oder Verbindung verwendet werden, welcher oder welche der APU erlaubt, Daten zu und von anderen Komponenten zu transferieren. Die APU mag ebenfalls mit einem DMA Interface 906 zum Bereitstellen von direktem Speicherzugriff zum Laden von Daten in den Vektordaten Speicher (CMEM) verbunden sein. Sowohl das AXI 904 als auch das DMA 906 mögen Daten

in den Speicherraum der APU (1004, 1006 in der Fig. 10, 1118 in der Fig. 11) laden. Die APU 902 oder insbesondere der ACP 908 der APU 902, mag ebenfalls mit einem Sequenzer 910 gekoppelt sein, wie zum Beispiel Sequenzer 804 des Systems 800, welcher Interrupt und/oder Steuersignale an die APU 902 bereitstellt.

[0040] Zusätzlich zu den AXI 904 und den DMA 906 Verbindungen, welche oben beschrieben sind, mag die APU 902 ebenfalls mit den AXI und CMEM_DMA Verbindungen durch ein Interface (CMEM_IF) 912 hindurch zum Laden von Daten in und aus der APU 902 verbunden sein. Das CMEM_IF 912 kann Daten vektorisieren und die vektorisierten Daten an die CUs 914 der APU 902 bereitstellen. Das CMEM_IF 912 kann ebenfalls Skalarwerten an die APU bereitstellen. Das CMEM_IF 912 kann ebenfalls Vektordaten und Skalarwerten aus der APU 902 an das AXI 904 oder das CMEM_DMA 906 bereitstellen. Das CMEM_IF 912 mag verwendet werden, um neue Daten in die APU oder den Speicher (CMEM), welcher mit der APU assoziiert ist, zu laden und um alte oder verarbeitete Daten herauszuziehen, während die APU aktuelle Daten verarbeitet.

[0041] Fig. 10 stellt Komponenten eines ACP in einem Blockdiagramm dar. Der ACP 1002 mag als die ACPs 806, 808, 810, 812 der Fig. 8 oder 908 der Fig. 9 verwendet werden. Der ACP 1002 beinhaltet Speicher für Instruktionen (IMEM) 1004 und Speicher für Daten (DMEM) 1006. Der IMEM 1004 und der DMEM 1006 können Daten, entweder Instruktionen oder Daten, welche zu verarbeiten sind, von einem Host-Interface (Host_If) 1008 empfangen. Das Host-Interface 1006 mag mit einem AXI Bus und einem oder mehr DMA Kanälen verbunden sein. Der ACP 1002 mag ebenfalls eine Skalarverarbeitungseinheit 1010 beinhalten. Die Skalarverarbeitungseinheit 1010 empfängt eine oder mehr Instruktionen oder Steuersignale, wie zum Beispiel einen Opcode, eine Speicheradresse, Signale aus dem Sequenzer oder ein globales-und/globales-oder Signal. Die Skalarverarbeitungseinheit mag ebenfalls mit dem CMem verbunden sein, um Daten zu empfangen und/oder bereitzustellen. Die Skalarverarbeitungseinheit 1010 kann die Instruktionen und Steuersignale dekodieren und Steuersignale an andere Komponenten bereitstellen, zum Beispiel an eine oder mehr Vektoreinheiten.

[0042] Zusätzlich zu dem Dekodieren von Instruktionen und Bereitstellen von Steuersignalen an eine oder mehr Vektoreinheiten beinhaltet die Skalarverarbeitungseinheit 1010 ebenfalls Komponenten zum Verarbeiten von Skalaren. Die Skalarverarbeitungskomponenten mögen zum Beispiel eine arithmetische Logikeinheit (ALU) 1012 zum Durchführen von arithmetischen Funktionen wie zum Beispiel Addition

auf einem oder mehr Skalaren beinhalten. Die Skalarverarbeitungseinheit 1010 mag ebenfalls eine Verschiebekomponente 1014 zum Verschieben von Bits eines Skalars nach links oder nach rechts, eine Multiplizierkomponente 1016 zum zusammen Multiplizieren von Skalaren und eine Adress-Generiereinheit (AGU) Komponente 1018 zum Generieren der Adresse der nächsten Instruktion, welche auszuführen ist, beinhalten.

[0043] Fig. 11 stellt Komponenten einer CU in einem Blockdiagramm dar. Die CU 1102 mag als die CUs 816, 914 verwendet werden. Die CU 1102 beinhaltet ein Register 1104 und zwei Bypass Multiplexer 1106, 1108. Wie verstanden wird, erlauben die Bypass Multiplexer Daten aus unterschiedlichen Stufen in der Verarbeitungspipeline als Eingänge für die Prozessor Funktionaleinheiten 1110, 1112, 1114, und 1116 zu verwenden.

[0044] Die CU beinhaltet verschiedene Komponenten zum Ausführen von Instruktionen, einschließlich einer ALU Einheit 1110 zum Durchführen von arithmetischen Funktionen auf Elementen eines oder mehr Vektoren, einer Multiplikationskomponente 1112 zum Multiplizieren von Elementen von Vektoren, einer Laden/Speichern Einheit 1114, welche mit einem Speicher 1118 gekoppelt ist, zum Laden oder Speichern von Daten. Die CU 1102 beinhaltet ebenfalls eine Bewegen/Verschieben Komponente 1116. Die Bewegen/Verschieben Komponente 1116 bewegt oder verschiebt Daten zwischen CUs. Die Bewegen/Verschieben Komponente 1116 ist mit der Bewegen/Verschieben Komponente von anderen CUs verbunden. Falls Vektoreinheiten von unterschiedlichen Verarbeitungseinheiten zusammen gekoppelt sind, um eine einzelne Verarbeitungseinheit zu bilden, mögen die Bewegen/Verschieben Einheiten der letzten CUs der Vektoreinheiten mit den Bewegen/Verschieben Einheiten der ersten CUs der Vektoreinheiten durch einen oder mehr Datenmultiplexer hindurch gekoppelt sein.

[0045] Obwohl nicht in der **Fig. 11** dargestellt, ist es möglich rekonfigurierbare Verbindungen zwischen CUs bereitzustellen. Zum Beispiel mag eine Vektoreinheit 128 x 16 Bits sein. Die CUs Verbindungen zwischen den CUs könnten rekonfiguriert sein, um, zum Beispiel 64 x 32 Bits oder 256 x 8 Bits bereitzustellen. Sowohl rekonfigurierbare Vektoreinheiten als auch Systeme, welche eine rekonfigurierbare Vektoreinheit benutzen, sind beschrieben worden. Die rekonfigurierbaren Vektoreinheiten und Systeme, welche oben beschrieben sind, mögen in verschiedenen Anwendungen verwendet werden, welche ein Verarbeiten von Daten benötigen. Zum Beispiel mögen die rekonfigurierbaren Vektoreinheiten und Systeme verwendet werden, um Audiodaten, Videodaten oder sowohl Audio als auch Videodaten zu verarbeiten.

[0046] Die obige Beschreibung hat verschiedene Ausführungsbeispiele von rekonfigurierbaren Vektoreinheiten beschrieben, andere Variationen sind möglich. Zum Beispiel mag eine rekonfigurierbare Vektoreinheit eine Mehrzahl von Verarbeitungseinheiten aufweisen, wobei jede zum Steuern der Vektoreinheiten der anderen Verarbeitungseinheiten geeignet ist.

[0047] Fig. 12A stellt mögliche Vektorgrößen dar, welche jede der Verarbeitungseinheiten steuern mag, falls jede Verarbeitungseinheit zum Steuern der Vektoreinheiten der anderen Verarbeitungseinheiten, wie oben beschrieben ist, geeignet ist. Wie dargestellt ist, mag jede der vier Verarbeitungseinheiten eine Vektoreinheit der Größe 32, 64, 96 oder 128 steuern. Die Schattierung jedes Blockes, welcher eine Vektoreinheit repräsentiert, stellt dar, welche der Verarbeitungseinheiten die jeweilige Vektoreinheit steuert.

[0048] Die Programmierflexibilität mag reduziert sein, indem nur einem Teilsatz der Verarbeitungseinheiten erlaubt wird, Vektoreinheiten von anderen Verarbeitungseinheiten zu steuern.

[0049] Fig. 12B stellt mögliche Vektorgrößen dar, welche jede der Verarbeitungseinheiten steuern mag, falls nur Verarbeitungseinheiten 0 und 2 zum Steuern der Vektoreinheiten der anderen Verarbeitungseinheiten befähigt sind, wie oben beschrieben ist. Die Schattierung jedes Blockes, welcher eine Vektoreinheit repräsentiert, stellt dar, welche der Verarbeitungseinheiten die jeweilige Vektoreinheit steuert.

[0050] Fig. 13 stellt einen rekonfigurierbaren Vektorprozessor 1300 in einem Blockdiagramm dar. Der Vektorprozessor 1300 weist eine Mehrzahl von Verarbeitungseinheiten 1310, 1320, 1330, 1340 auf. Jede Verarbeitungseinheit ist ähnlich zu den Verarbeitungseinheiten 210, 220, 230, 240, welche oben beschrieben sind, und jede weist eine Skalareinheit 202, eine Vektoreinheit 204 und eine Steuereinheit 206 auf. Jedoch weist der rekonfigurierbare Vektor Prozessor anstelle eines Vektor-Steuer-Multiplexers 208 einen Crossbar-Schalter 1302 auf.

[0051] Der Crossbar-Schalter 1302 erlaubt irgendeiner Steuereinheit einer Verarbeitungseinheit irgendeine Vektoreinheit einer anderen Verarbeitungseinheit zu steuern. Ein möglicher Vorteil an dieser Anordnung ist, dass es möglich ist, eine Steuerung einer Vektoreinheit Allokation durch unterschiedliche Verarbeitungseinheiten hindurchzuleiten, ohne irgendwelche Daten bewegen zu müssen.

[0052] Fig. 14 stellt ein Teilsatz der 24 möglichen Vektoreinheit Steuerkonfigurationen für Vektoreinheiten der Größe 32 dar. Obwohl **Fig. 14** nur eine

Steuerung von Vektoreinheiten der Größe 32 darstellt, wird es eingesehen werden, dass der Crossbar-Schalter ebenfalls verwendet werden mag, wenn unterschiedliche Vektoreinheiten von variierenden Größen gesteuert werden, wie oben beschrieben ist. Verarbeitungseinheiten mögen Vektoreinheiten unterschiedlicher Größe, wie zum Beispiel 64, 96 oder 128, steuern. Um Routing Platz (routing space) in dem Datenpfad des Crossbar-Schalters 1302 zu reduzieren, mögen die Vektoreinheiten, welche zusammen gruppiert sein können und mittels einer einzelnen Verarbeitungseinheit gesteuert sein können, angrenzend zu einander angeordnet sein.

[0053] Fig. 15 stellt ein beispielhaftes Bild dar, welches unter Verwendung des rekonfigurierbaren Vektorprozessors und Systemen, welche oben beschrieben sind, verarbeitet werden mag. Die Anwendung, welche mit Bezug auf Fig. 15 beschrieben ist, mag ein Echtzeit Verarbeiten von Frames eines Videos bereitstellen, welches mittels einer Videokamera auf einem Automobil aufgenommen wurde. Die Frames des Videos mögen verarbeitet werden, um Elemente in dem Framebild zu suchen und zu identifizieren. Zum Beispiel mögen die Framebilder verarbeitet werden, um sowohl Autos, Straßenschilder zu identifizieren als auch zusätzliche Funktionalität durchzuführen, wie zum Beispiel Objekte in einem blinden Punkt eines Autos detektieren oder das Framebild zu entzerren (dewarping).

[0054] Das gesamte Bild mag zuerst verarbeitet werden, um Kandidatenelemente in dem Bild zu suchen. Die Kandidatenelemente mögen zum Beispiel Automobile in dem Bild sein. Der Prozess identifiziert ein oder mehr Kandidatenelemente wie mittels der Boxen 1502, 1504, 1506, 1508, 1510, 1512, 1514 der Fig. 15 dargestellt ist. Die Kandidatenelemente mögen Objekte wie zum Beispiel Straßenschilder 1516, 1518 beinhalten, obwohl Straßenschilder durch einen separaten Prozess detektiert werden mögen. Da ein Verarbeiten des gesamten Bildes ein Verarbeiten der größten Menge von Daten benötigt, mag der rekonfigurierbare Vektorprozessor, welche zum Verarbeiten der Bilddaten verwendet wird, konfiguriert sein, um die größte mögliche Vektorgröße bereitzustellen, wenn die Kandidatensuche- oder Verarbeitungsfunktion auf den Bilddaten durchgeführt wird. Wenn die rekonfigurierbare Vektoreinheit konfiguriert ist, um die größte Vektorgröße bereitzustellen, sind drei Skalareinheiten verfügbar, um verwendet zu werden. Diese Skalareinheiten mögen verwendet werden, wenn die Kandidatensuche durchgeführt wird, um zusätzliche Funktionalität bereitzustellen. Diese zusätzliche Funktionalität kann zum Beispiel ein Entzerren von Bildern, um Fischaugenlinsen zu korrigieren, welche verwendet worden sein mögen, um die Bilder aufzunehmen, oder ein Detektieren von

Objekten in einem blinden Punkt eines Autos beinhalten.

[0055] Sobald ein Bild verarbeitet worden ist, um Kandidatenelemente zu identifizieren, mögen die individuellen Kandidatenelemente, zum Beispiel die individuellen Blöcke 1502, 1504, 1506, 1508, 1510, 1512, 1514, 1516, 1518, welche in der Fig. 15 dargestellt sind, verarbeitet werden, um die individuellen Kandidatenelemente zu verifizieren und zu klassifizieren. Da die Größe von jedem individuellen Kandidatenelement kleiner als das gesamte Bild sein wird, ist es nicht notwendig, die maximal verfügbare Vektorgröße zu verwenden. Als solches kann der rekonfigurierbare Vektorprozessor konfiguriert sein, zwei Vektoreinheiten von gleicher Größe bereitzustellen. Jede Vektoreinheit kann für unterschiedliche Funktionen verwendet werden. Zum Beispiel mag eine der Vektoreinheiten verwendet werden, um die individuellen Elemente zu verifizieren und zu klassifizieren, während die zweite Vektoreinheit verwendet werden mag, um einen Abschnitt des Framebilds auf Straßenschilder abzusuchen oder um die identifizierten Schilder weiter zu verarbeiten.

[0056] Wie oben beschrieben ist, erlaubt ein System, welches einen rekonfigurierbaren Vektorprozessor hat, dass mehrere Algorithmen mittels des Systems ausgeführt werden. Vorteilhafterweise erlaubt der rekonfigurierbare Vektorprozessor dem System, dass es konfiguriert wird, basierend auf den Voraussetzungen des Algorithmus, welcher verarbeitet wird, ein effizientes Verarbeiten bereitzustellen.

[0057] Fig. 16 stellt ein Verfahren zum Verarbeiten von Daten mit einem rekonfigurierbaren Vektorprozessor in einem Flowchart dar. Das Verfahren 1600 konfiguriert zuerst eine Größe einer Vektoreinheit eines Vektorprozessors (1602) mittels Verteilens von CUs, welche mit einer oder mehr einer Mehrzahl von jeweiligen anderen (Slave oder externen) Verarbeitungseinheiten assoziiert sind. Die Größe der Vektoreinheit mag mittels Steuern der Mehrzahl der Vektoreinheiten mit einer einzelnen Steuer-(Master oder interne) Verarbeitungseinheit konfiguriert werden. Eine oder mehr Instruktionen werden unter Verwenden der Vektoreinheit der konfigurierten Größe (1604) ausgeführt. Die Slave Verarbeitungseinheiten können mittels der Master Einheit gesteuert werden oder benutzt werden, um Instruktionen zu verarbeiten, welche kleinere Vektoreinheiten benötigen, oder Skalar Verarbeitungsfunktionalität nur bereitzustellen, wenn dies benötigt wird. Die Größe der Vektoreinheit mag dann rekonfiguriert werden, um eine Vektoreinheit (1606) unterschiedlicher Größe bereitzustellen. Die CUs, welche zu der Vektoreinheit der Master Verarbeitungseinheit allokiert sind, mögen in der Größe vergrößert oder verkleinert werden, wie es von der speziellen Verarbeitungsaufgabe angefordert ist. Sobald die Größe der

Vektoreinheit rekonfiguriert ist, werden eine oder mehr Instruktionen unter Verwenden der Vektoreinheit der rekonfigurierten Größe ausgeführt. Unter Verwenden des oben beschriebenen Verfahrens ist es möglich, die Vektorgröße zu modifizieren, welche mittels einer Vektoreinheit verarbeitet werden kann.

[0058] Fig. 17 stellt ein Verfahren zum Konfigurieren einer Größe einer Vektoreinheit eines rekonfigurierbaren Vektorprozessors in einem Flowchart dar. Eine benötigte Vektorgröße wird für eine benötigte Verarbeitungsfunktion bestimmt, welche mittels des rekonfigurierbaren Vektorprozessors (1702) durchzuführen ist. Falls mehrere Verarbeitungsfunktionen gleichzeitig benötigt werden, wird die Vektorgröße für jede Funktion bestimmt und auf die Leistungsfähigkeit des Prozessors und der Anzahl der verfügbaren Verarbeitungseinheiten optimiert. Dies mag dynamisch mittels eines Steuer-Mikroprozessors während des Betriebs oder Startens des rekonfigurierbaren Vektorprozessors oder basierend auf vordefinierten Verarbeitungsvoraussetzungen, welche während einer Programmier- oder einer Kompilierzeit des für den Betrieb des rekonfigurierbaren Vektorprozessors benötigten Codes durchgeführt werden. Die Anzahl der Berechnungseinheiten (CUs), welche benötigt werden, um die Vektorgröße (1704) zu erreichen, wird basierend auf den verfügbaren Konfigurationen der Verarbeitungseinheiten und den assoziierten Vektoreinheiten bestimmt. Ein oder mehr Prozessorkonfigurationen werden bestimmt, was benötigt wird, um die bestimmte CU Verteilung (1706) basierend auf einer oder mehr der Verarbeitungsfunktionen bereitzustellen. Die Konfiguration bestimmt, welcher bzw. welche Prozessor(en) als Master oder Steuerprozessor oder Slave Prozessor agieren werden, und welche Vektoreinheiten neu verteilt werden. Die Steuersignale werden zum Steuern der Vektoreinheit einer ersten Verarbeitungseinheit (1708) generiert. Die Steuersignale mögen mittels eines Steuerprozessors dynamisch bestimmt werden oder mögen basierend auf Konfigurationsinformation (1707) generiert werden, welche während einer Kodier- oder der Kompilierzeit basierend auf vorgegebenen Allokationsmetriken generiert werden, welche Verarbeitungsvoraussetzungen definieren, welche mit Instruktionen oder Algorithmen assoziiert sind, welche mittels des Vektorprozessors durchzuführen sind. Die generierten Steuersignale werden an eine Vektoreinheit sowohl der ersten Verarbeitungseinheit als auch einer Vektoreinheit einer zweiten Verarbeitungseinheit (1710) bereitgestellt. Mittels Bereitstellens der generierten Steuersignale an beide Vektoreinheiten, wird die Größe der Vektoreinheit, welche mittels der ersten Verarbeitungseinheit gesteuert wird, vergrößert. Die Größe der Vektoreinheit mag mittels Bereitstellens der geeigneten generierten Steuersignale an unterschiedliche Komponenten des rekonfigurierbaren Vektorprozessors wie zum Beispiel Vektor-Steuer-

Multiplexer, Datenmultiplexer oder Schalter, konfiguriert oder rekonfiguriert werden. Die Steuersignale an die zweite Verarbeitungseinheit mögen die Skalar Verarbeitungsressourcen der zweiten Verarbeitungseinheit (1712) allozieren, um die Verarbeitungsfunktionen mittels Verwendens von Skalarfunktionalität der zweiten Verarbeitungseinheit ohne eine Vektoreinheit auszuführen oder die zweite Verarbeitungseinheit als eine Slave Verarbeitungsfunktion für die Master Verarbeitungseinheit zu benutzen. Zusätzlich zu dem Ausführen von Instruktionen unter Verwenden der Vektoreinheit der ersten oder zweiten Größe, mag das Verfahren ebenfalls unter Verwenden eines oder mehr Skalarprozessoren oder einer oder mehr zusätzlicher Vektoreinheiten Instruktionen ausführen.

[0059] Bestimmte Adaptionen und Modifikationen der beschriebenen Ausführungsbeispiele können vorgenommen werden. Daher werden die oben diskutierten Ausführungsbeispiele als illustrativ und nicht als einschränkend erachtet werden.

Patentansprüche

1. Ein rekonfigurierbarer Vektorprozessor (200), welcher aufweist:
 - eine Mehrzahl von Prozessoreinheiten (210, 220, 230, 240), wobei jede Prozessoreinheit folgendes aufweist:
 - eine Skalareinheit (202) zum Verarbeiten von Instruktionen auf Skalarwerten;
 - eine Vektoreinheit (204) zum Verarbeiten von Instruktionen auf Vektordaten;
 - eine Steuereinheit (206) zum Dekodieren von Instruktionen, Generieren von Steuersignalen und Steuern der Skalareinheit (202) und der Vektoreinheit (204) basierend auf den generierten Steuersignalen; und
 - einen Vektorsteuerselektor (208) zum selektiven Bereitstellen von Steuersignalen, welche mittels einer der Steuereinheit von einer Prozessoreinheit (240) von der Mehrzahl von Prozessoreinheiten (210, 220, 230, 240) generiert sind, an eine Vektoreinheit (204), welche mit einer unterschiedlichen Steuereinheit (202) einer anderen Prozessoreinheit (210) von der Mehrzahl von Prozessoreinheiten (210, 220, 230, 240) assoziiert ist.
2. Der rekonfigurierbare Vektorprozessor (200) gemäß Anspruch 1, wobei der Vektorsteuerselektor (208) einen Vektorsteuermultiplexer aufweist, welcher mit einer ersten Prozessoreinheit (210) von der Mehrzahl der Prozessoreinheiten (210, 220, 230, 240) zum selektiven Koppeln der Vektoreinheit (204) der ersten Prozessoreinheit (210) an die Steuereinheit (206) der ersten Prozessoreinheit (210) oder an eine Steuereinheit einer zweiten Prozessoreinheit (220, 230, 240) von der Mehrzahl der Prozessoreinheiten (210, 220, 230, 240) aufweist,

um das eine oder mehr Steuersignale, welche mittels der ersten Prozessoreinheit (210) oder der zweiten Prozessoreinheit (220, 230, 240) generiert sind, an die Vektoreinheit (204) der ersten Prozessoreinheit (210) bereitzustellen.

3. Der rekonfigurierbare Vektorprozessor (1300) gemäß Anspruch 1, wobei der Vektorsteuerselektor (208) einen Crossbar-Schalter (1302) zum Empfangen einer Mehrzahl von jeweiligen Steuersignalen aus einer oder mehr der Mehrzahl der Prozessoreinheiten (1310, 1320, 1330, 1340) und zum selektiven Bereitstellen einer oder mehr der empfangenen Mehrzahl von jeweiligen Steuersignalen an die jeweiligen Vektoreinheiten (204) von einer oder mehr Prozessoreinheiten der Mehrzahl der Prozessoreinheiten (1310, 1320, 1330, 1340) aufweist.

4. Die rekonfigurierbare Vektorprozessorarchitektur (600; 700) gemäß Anspruch 1, welche ferner einen Vektordatenverbinder (610, 612; 710a-710d, 712a-712d) zum selektiven Koppeln der Vektoreinheit von einer Verarbeitungseinheit (608; 704) an die Vektoreinheit von einer anderen Verarbeitungseinheit (606; 702) aufweist, welche die Steuersignale bereitstellt.

5. Der rekonfigurierbare Vektorprozessor (400) gemäß irgendeinem der Ansprüche 1 bis 4, welcher ferner eine Mehrzahl von Vektorsteuerselektoren aufweist, wobei jeder Vektorsteuerselektor einen Vektorsteuermultiplexer aufweist, welcher mit einer jeweiligen Prozessoreinheit (402, 404, 410, 412, 414, 416) der Mehrzahl von Prozessoreinheiten assoziiert ist.

6. Der rekonfigurierbare Vektorprozessor (400) gemäß irgendeinem der Ansprüche 1 bis 5, wobei die jeweiligen Steuereinheiten einer oder mehr Prozessoreinheiten (402, 404), welche hierin später als Mastereinheiten (402, 404) bezeichnet werden, mit dem Vektorsteuermultiplexer gekoppelt sind, welcher mit einer unterschiedlichen Prozessoreinheit (406, 410, 412, 414, 416) assoziiert ist.

7. Der rekonfigurierbare Vektorprozessor (600) gemäß Anspruch 6, wobei eine oder mehr der Mastereinheiten (602) einen Vektorsteuermultiplexer zum selektiven Koppeln der Vektoreinheit mit der Steuereinheit einer anderen Mastereinheit aufweisen.

8. Der rekonfigurierbare Vektorprozessor gemäß irgendeinem der Ansprüche 1 bis 7, wobei die Skalareinheit (202, 1010) von jeder der Mehrzahl von Prozessoreinheiten (210, 220, 230, 240; 1002) arithmetische, logische und Verschieboperationen durchführen kann.

9. Der rekonfigurierbare Vektorprozessor gemäß irgendeinem der Ansprüche 1 bis 8, wobei jede der Mehrzahl von Prozessoreinheiten (210, 220, 230, 240; 1002) ferner eine Adressen-Generiereinheit-Komponente (1018) zum Generieren der Adresse der nächsten Instruktion aufweist, welche mittels der Prozessoreinheit (1002) auszuführen ist.

10. Der rekonfigurierbare Vektorprozessor (200) gemäß irgendeinem der Ansprüche 1 bis 9, wobei die Skalareinheit (202) von jeder der Mehrzahl von Prozessoreinheiten (210, 220, 230, 240) gleichzeitig mit ihren jeweiligen Vektoreinheiten (204) betrieben werden können.

11. Der rekonfigurierbare Vektorprozessor (200) gemäß irgendeinem der Ansprüche 1 bis 9, wobei die Skalareinheit (202) von jeder der Prozessoreinheiten (210, 220, 230, 240) autonom von ihren jeweiligen Vektoreinheiten (204) betrieben werden können.

12. Der rekonfigurierbare Vektorprozessor (600) gemäß Anspruch 1, wobei eine oder mehr der Mehrzahl von Prozessoreinheiten (602, 606, 608) jeweils ferner einen oder mehr Datenmultiplexer (604, 610, 612) zum selektiven zusammen Koppeln der Vektoreinheiten der einen oder mehr Prozessoreinheiten (602, 606, 608) aufweist.

13. Der rekonfigurierbare Vektorprozessor (800; 900) gemäß irgendeinem der Ansprüche 1 bis 12, wobei jede Vektoreinheit (816; 914) eine Mehrzahl von Berechnungseinheiten (CUs 0..31) aufweist, wobei jede zum Verarbeiten von Daten einer definierten Bitlänge ist.

14. Der rekonfigurierbare Vektorprozessor gemäß Anspruch 13, wobei jede CU konfiguriert ist, Addier- und Verschiebe-Operationen auf empfangenen Daten durchzuführen.

15. Der rekonfigurierbare Vektorprozessor gemäß Anspruch 13, wobei jede CU aufweist: ein Datenregister (1104); eine Mehrzahl von Bypass-Multiplexern (1106, 1108), welche mit dem Datenregister (1104) gekoppelt ist; eine arithmetische Logikeinheit (1110), welche mit den Ausgängen der Mehrzahl der Bypass-Multiplexern (1106, 1108) gekoppelt ist; eine Multiplikationseinheit (1112), welche mit den Ausgängen der Mehrzahl der Bypass-Multiplexern (1106, 1108) gekoppelt ist; eine Laden/Speicherneinheit (1114), welche mit den Ausgängen der Mehrzahl der Bypass-Multiplexern (1106, 1108) und einem Speicher (1118) gekoppelt ist; eine Bewege/Verschiebeeinheit (1116), welche mit den Ausgängen der Mehrzahl der Bypass-Multiplexern

xern (1106, 1108) und einer oder mehreren anderen Berechnungseinheiten (816; 914) gekoppelt ist.

16. Ein Verfahren (1600) für ein Verarbeiten von Daten unter Verwenden eines rekonfigurierbaren Vektorprozessors (200), welcher zwei oder mehr Verarbeitungseinheiten (210, 220, 230, 240) aufweist, jede mit einer Vektoreinheit (204), wobei das Verfahren (1600) aufweist:

Konfigurieren (1602) des rekonfigurierbaren Vektorprozessors (200), um eine Vektoreinheit (204) einer ersten Größe zum Verarbeiten von Vektordaten (120) einer ersten Größe bereitzustellen;

Ausführen (1604) von einer oder mehr Instruktionen unter Verwenden der Vektoreinheit (204) der ersten Größe, um Vektordaten (120) der ersten Größe zu verarbeiten;

Rekonfigurieren (1606) des rekonfigurierbaren Vektorprozessors (200), um die Größe der Vektoreinheit (204) auf eine zweite Größe zu ändern; und

Ausführen (1608) von einer oder mehr Instruktionen unter Verwenden der Vektoreinheit (204) der zweiten Größe, um Vektordaten der zweiten Größe zu verarbeiten,

wobei das Konfigurieren (1602) und das Rekonfigurieren (1606) der Größe der Vektoreinheit (204) jeweils aufweist:

Generieren (1708) von Steuersignalen zum Steuern der Vektoreinheit (204) einer ersten Verarbeitungseinheit (210), und

Bereitstellen (1710) der generierten Steuersignale an die Vektoreinheit (204) der ersten Verarbeitungseinheit (210) und an die Vektoreinheit einer zweiten Verarbeitungseinheit (220, 230, 240), um eine Vektoreinheit mit einer Gesamtgröße der Summe der individuellen Vektoreinheiten der ersten (210) und zweiten (220, 230, 240) Verarbeitungseinheit bereitzustellen.

17. Das Verfahren gemäß Anspruch 16, wobei das Konfigurieren (1710) der Größe der Vektoreinheit (204) aufweist:

Bereitstellen (1710) von geeigneten Steuersignalen an eine oder mehrere Komponenten (602-608; 702-708) des rekonfigurierbaren Vektorprozessors (600; 700), welcher einen Vektorsteuermultiplexer oder Datenmultiplexer (604, 610, 612; 710, 712) aufweist.

18. Das Verfahren gemäß irgendeinem der Ansprüche 16 bis 17, welches ferner aufweist:

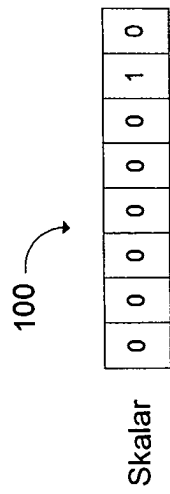
Ausführen von Instruktionen unter Verwenden von einem oder mehreren Skalarprozessoren (202) des rekonfigurierbaren Vektorprozessors (200), wenn die Instruktionen ausgeführt werden, welche die Vektoreinheit (204) verwenden.

19. Das Verfahren gemäß irgendeinem der Ansprüche 16 bis 18, welches ferner aufweist:

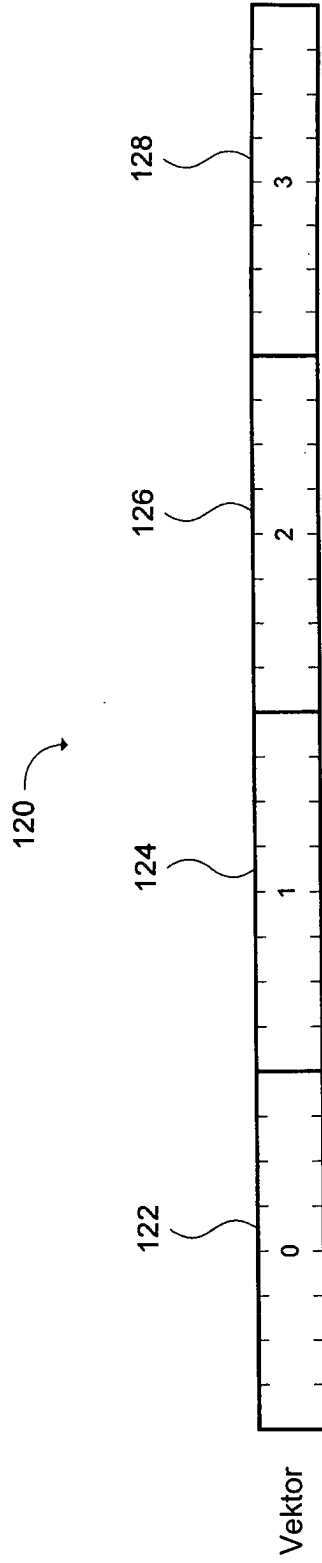
Konfigurieren des rekonfigurierbaren Vektorprozessors (200), um eine oder mehr zusätzliche Vektoreinheiten zum Verarbeiten von Vektordaten bereitzustellen.

Es folgen 19 Seiten Zeichnungen

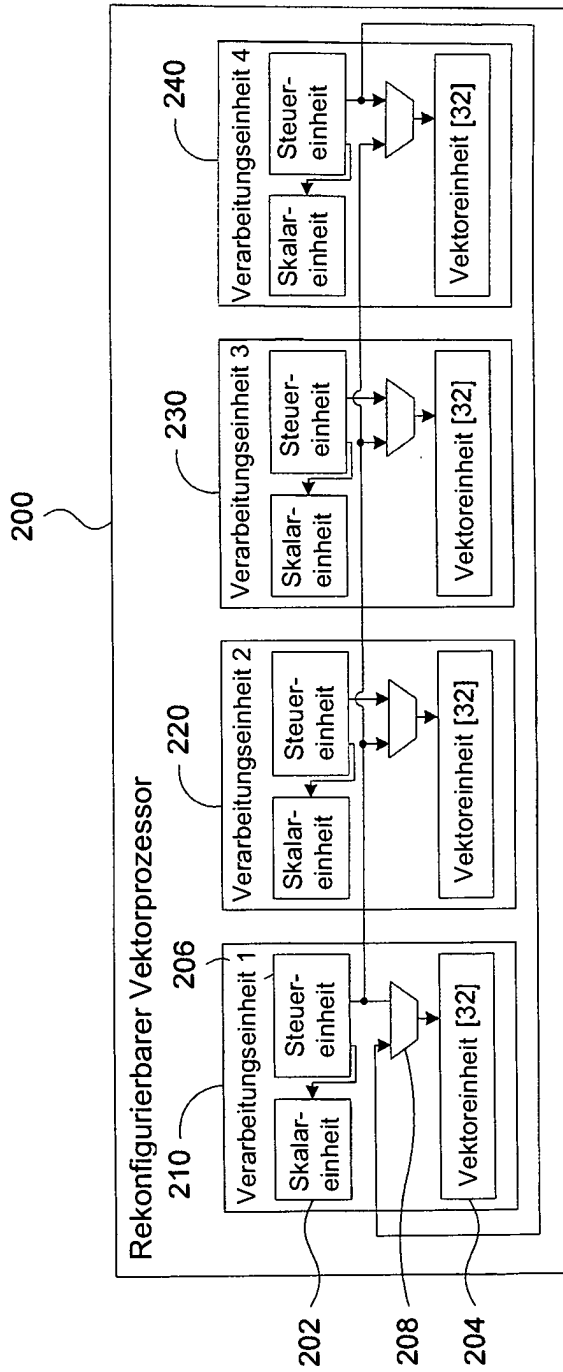
Anhängende Zeichnungen



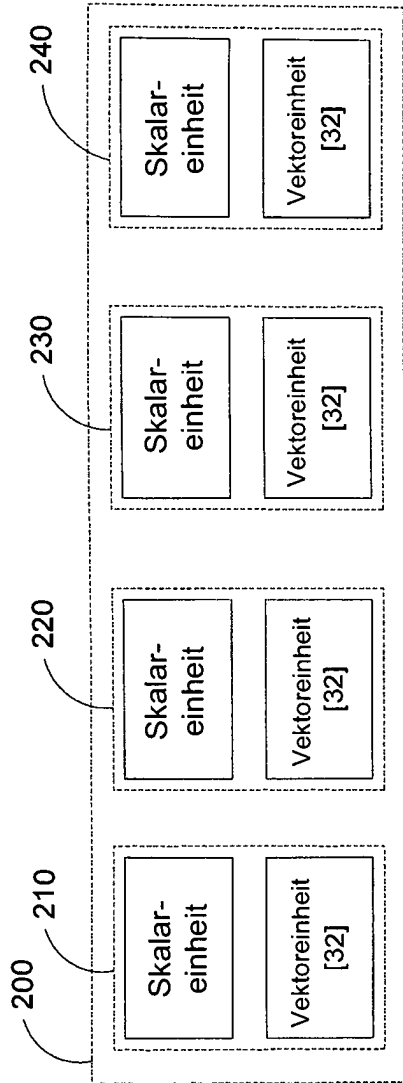
Figur 1A



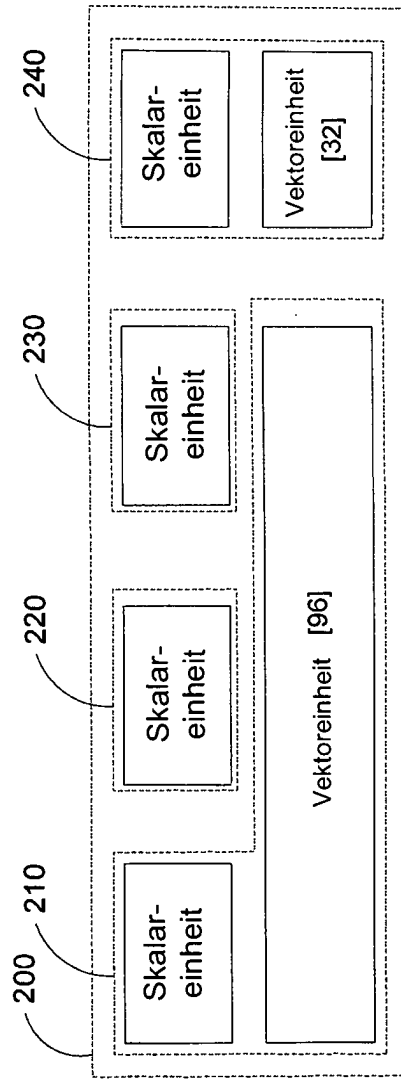
Figur 1B



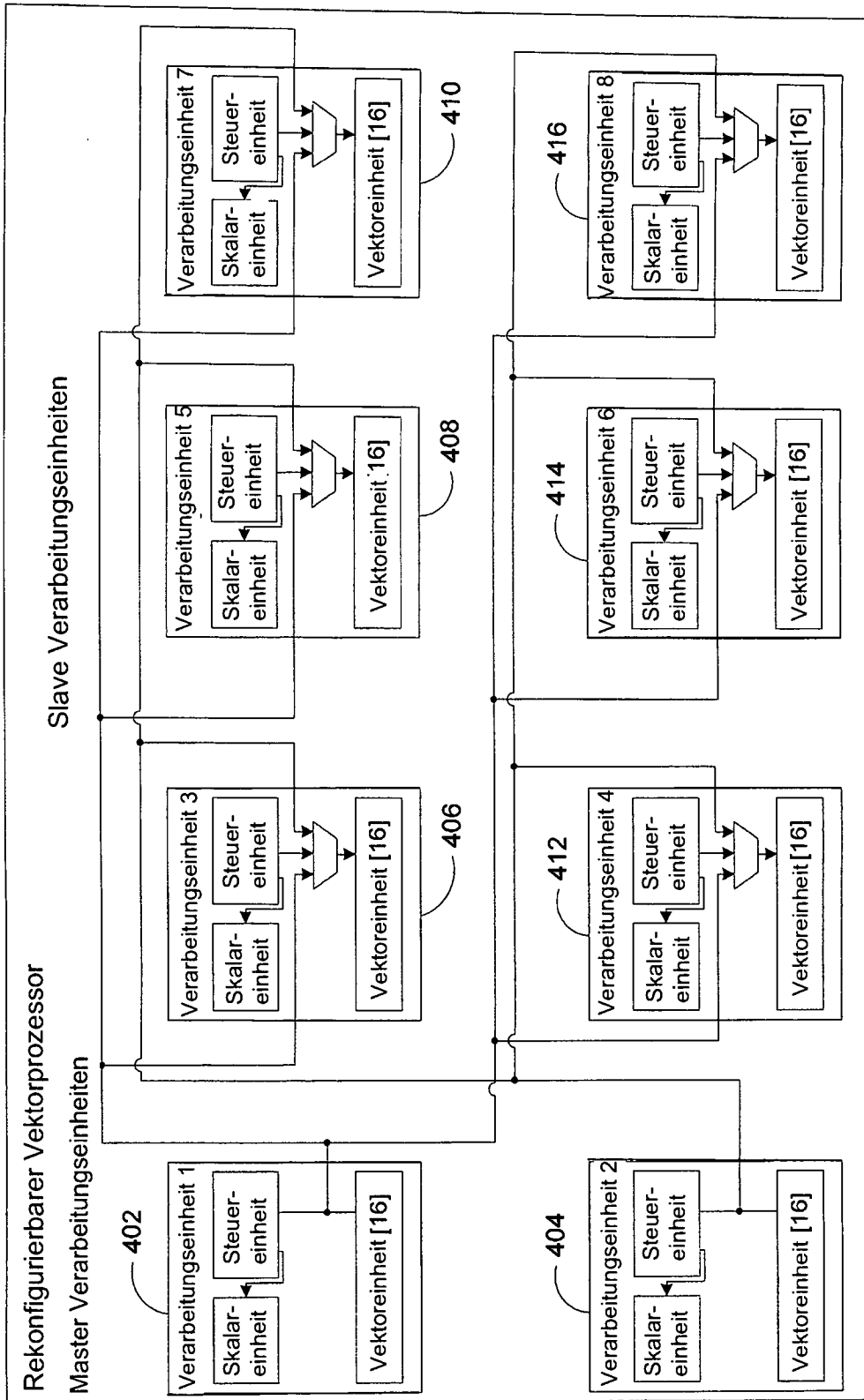
Figur 2



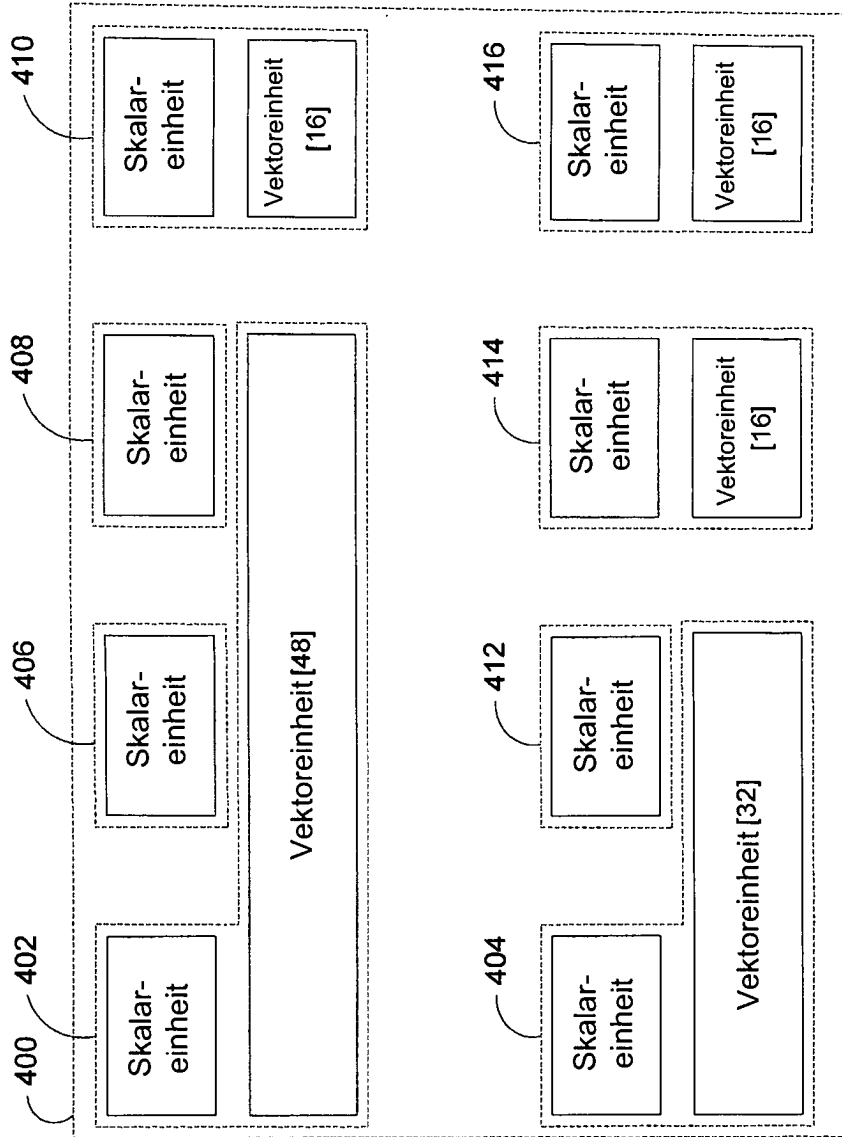
Figur 3A



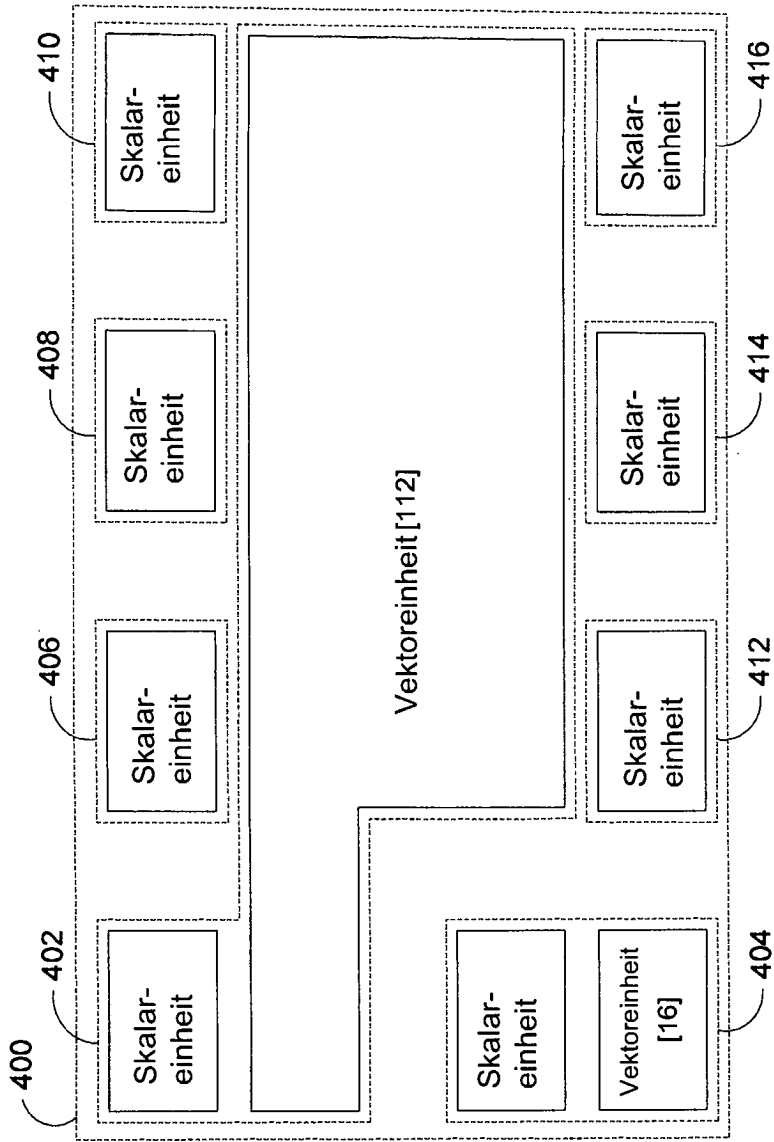
Figur 3B



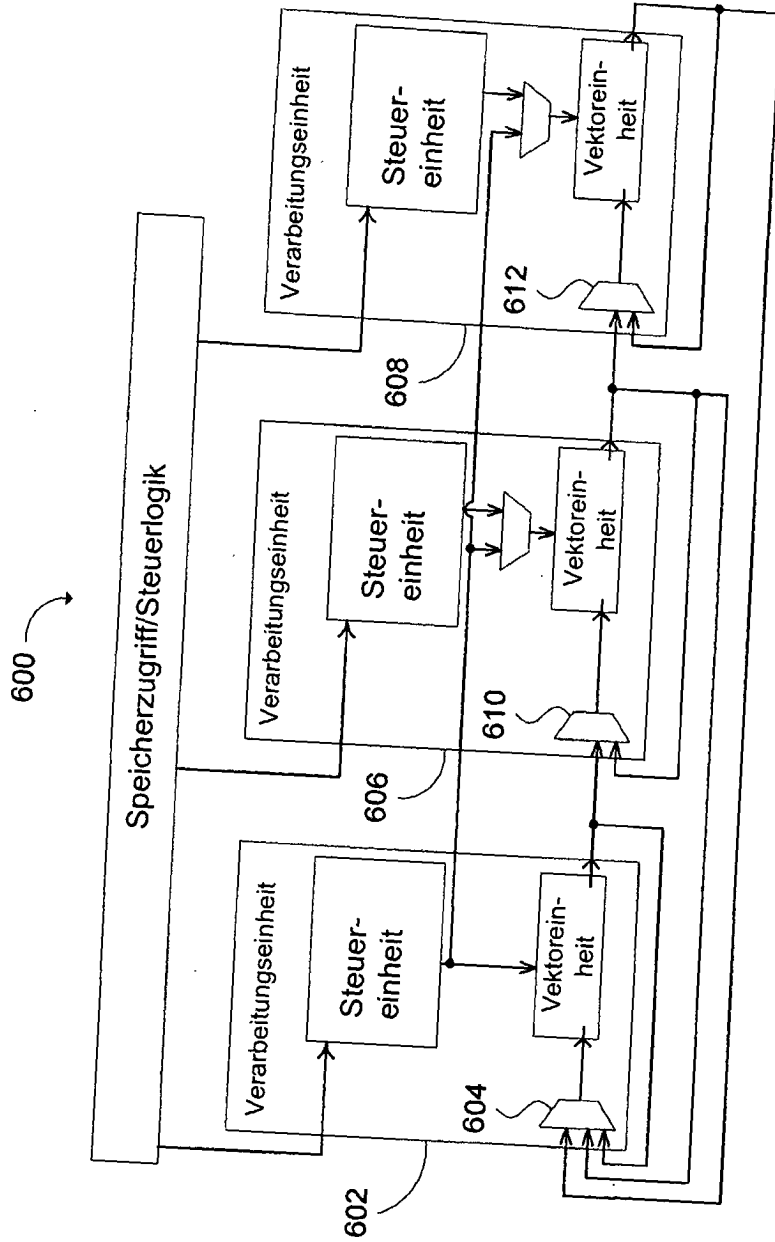
Figur 4



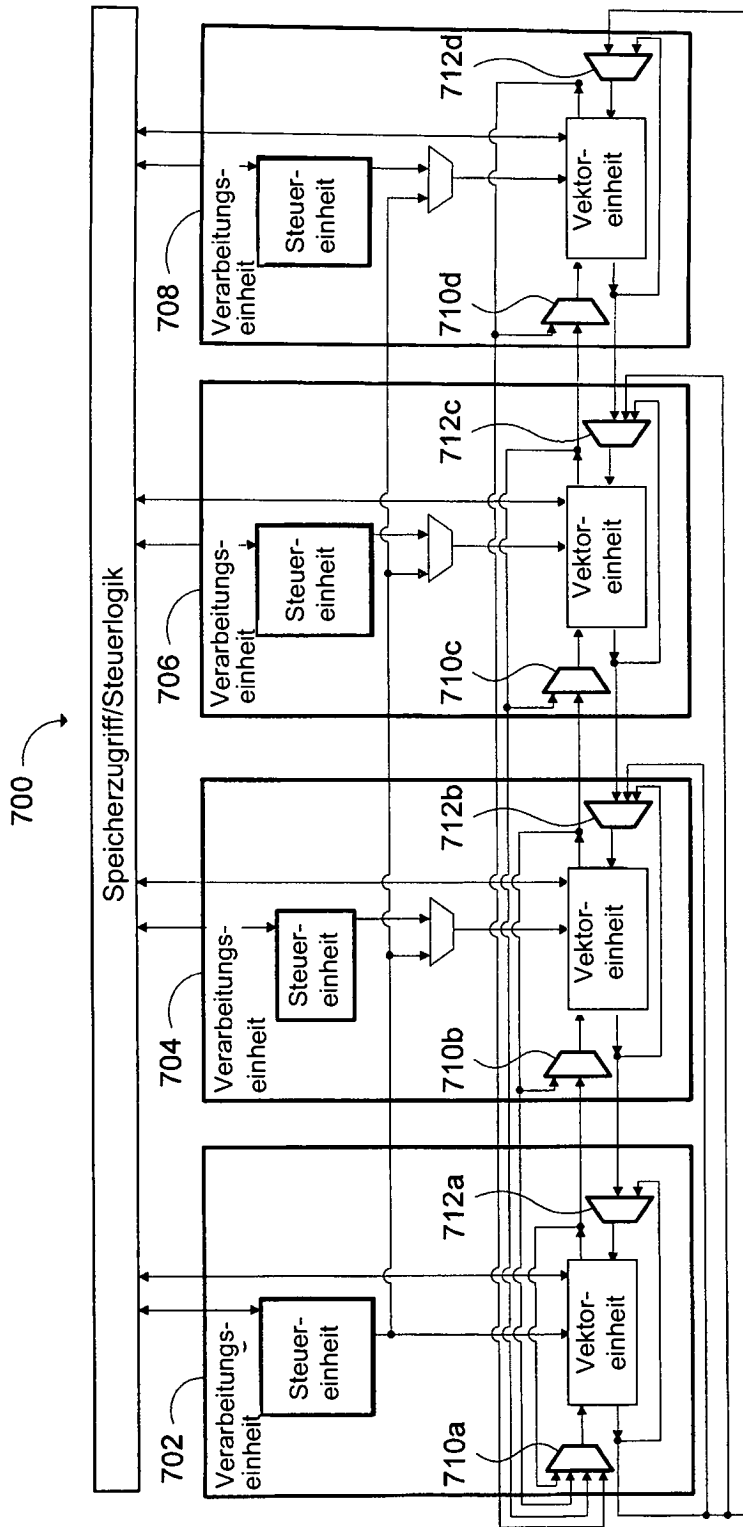
Figur 5A



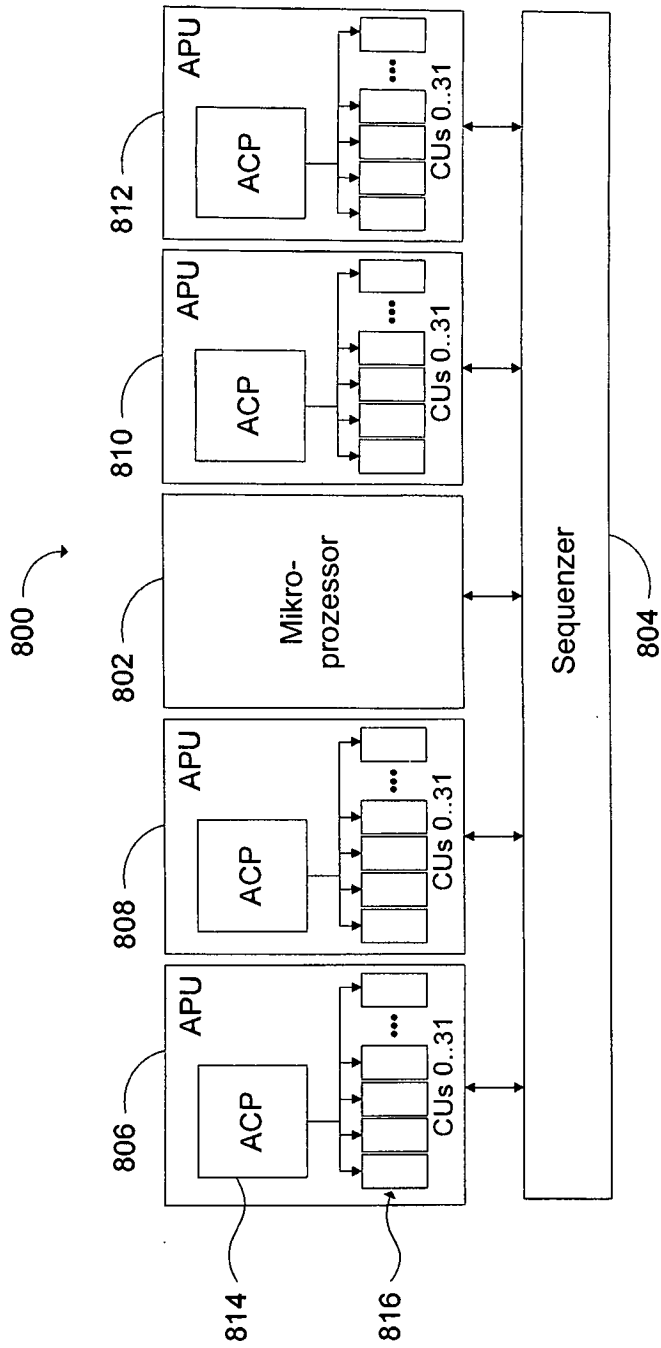
Figur 5B



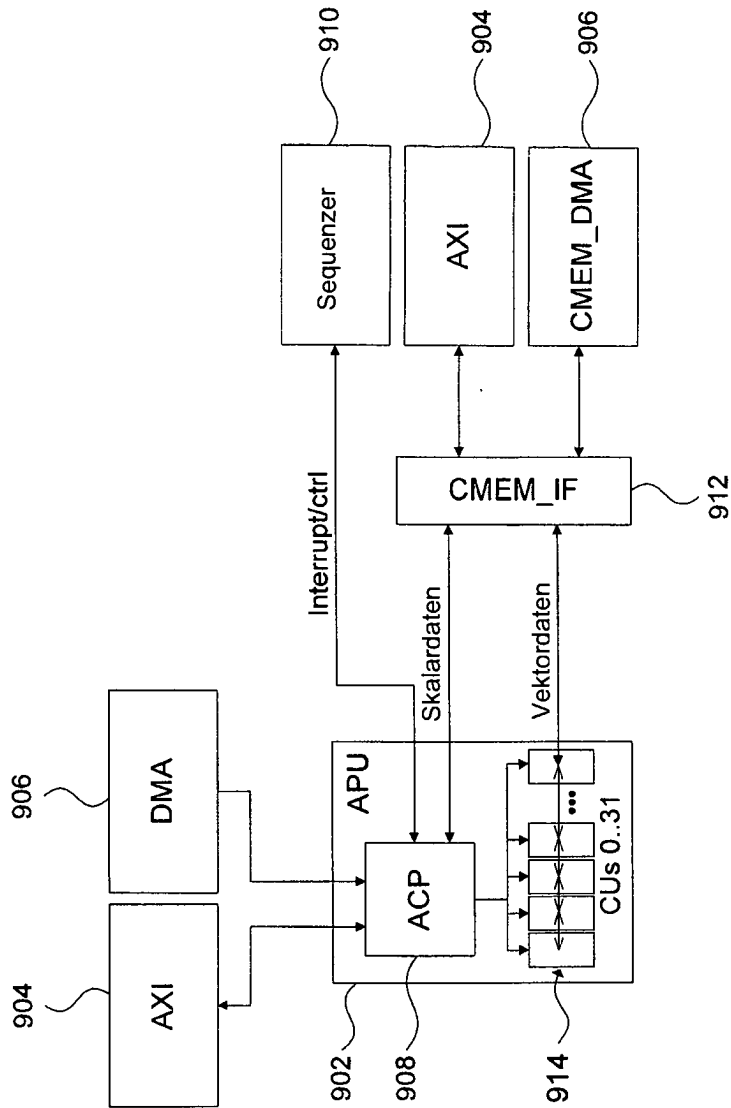
Figur 6



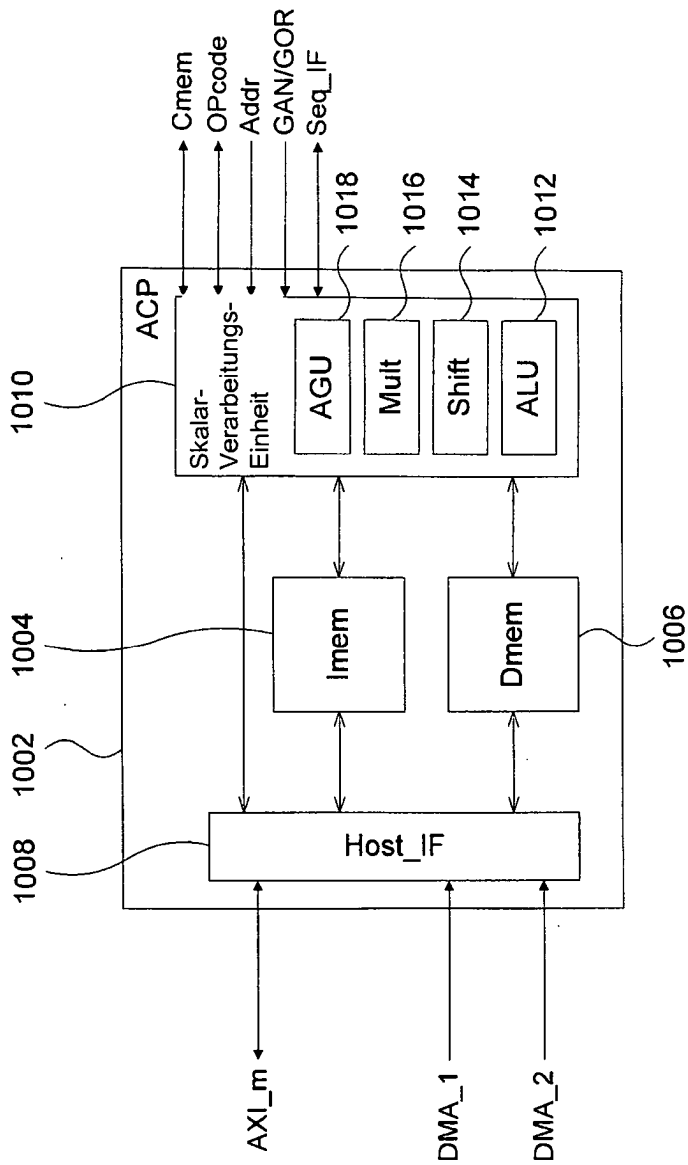
Figur 7



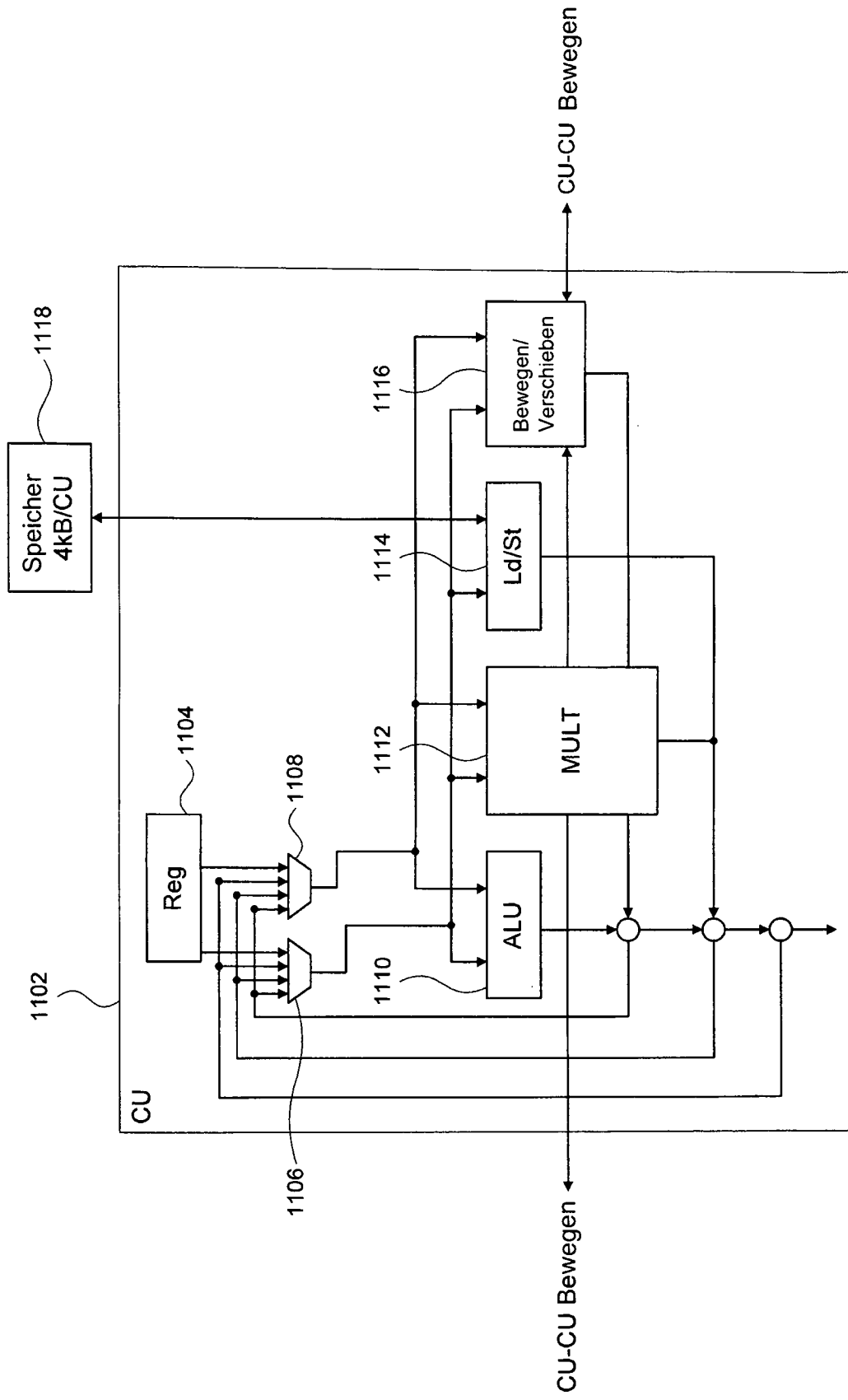
Figur 8



Figur 9



Figur 10



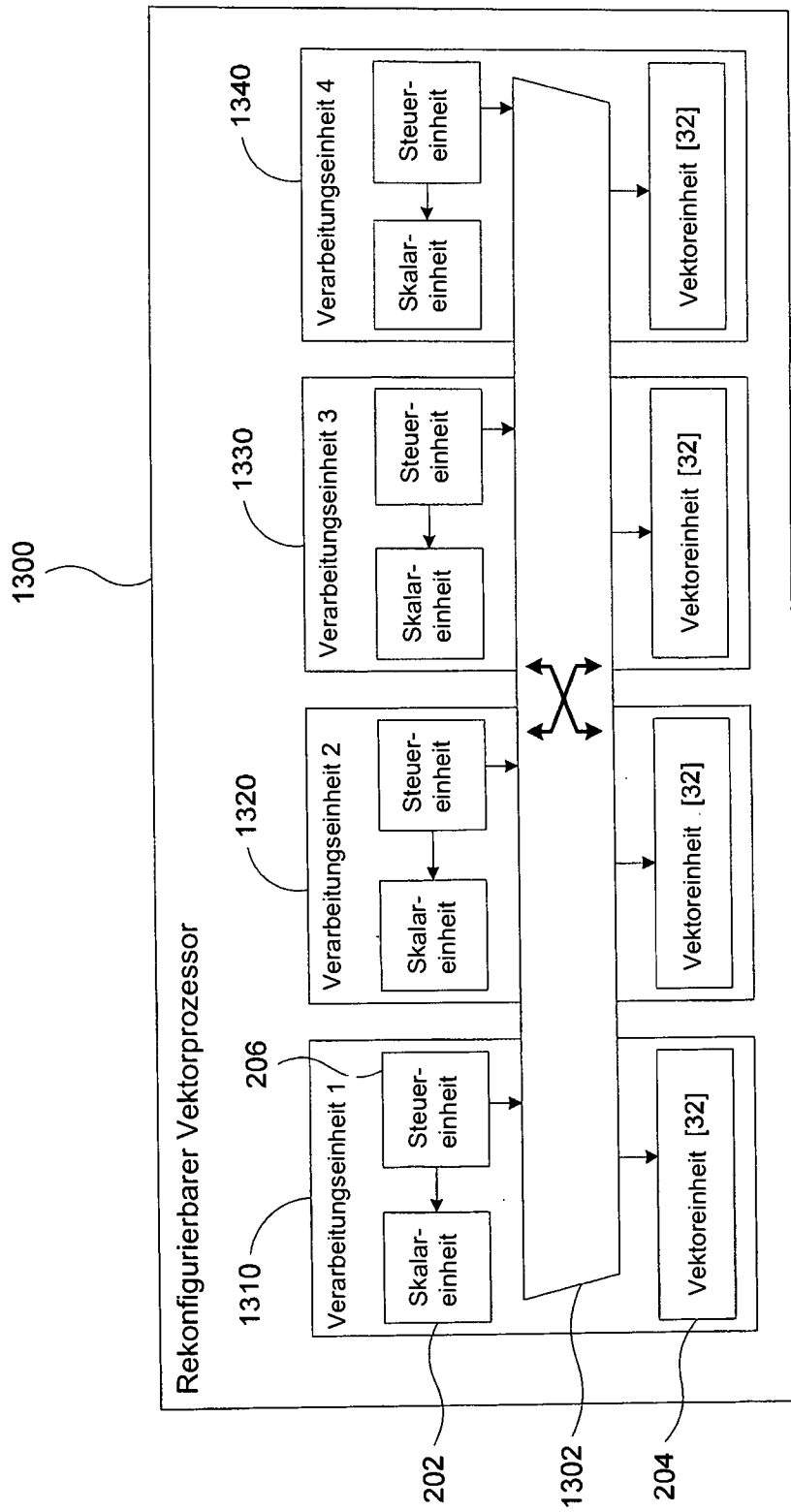
Figur 11

Verarbeitungseinheit 0	Verarbeitungseinheit 1	Verarbeitungseinheit 2	Verarbeitungseinheit 3
32	32	32	32
64		32	32
32	64		32
32	32	64	
	32	32	64
64		64	
	64		64
96			32
32	96		
	32	96	
		32	96
128			
128			
128			
128			

Figur 12A

Verarbeitungseinheit 0	Verarbeitungseinheit 1	Verarbeitungseinheit 2	Verarbeitungseinheit 3
32	32	32	32
64		32	32
32	32	64	
64		64	
96			32
	32	96	
128			
128			

Figur 12B



Figur 13

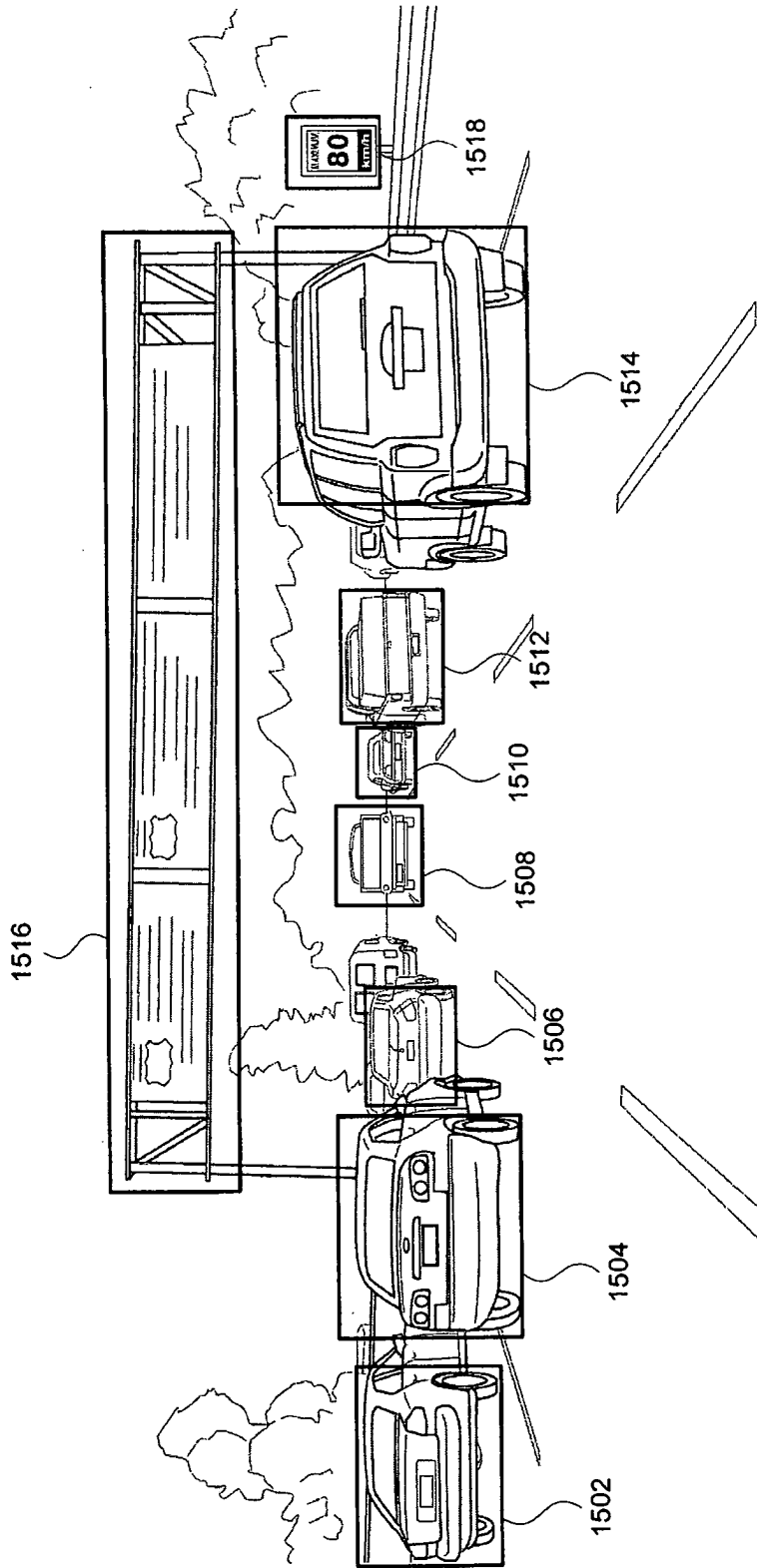
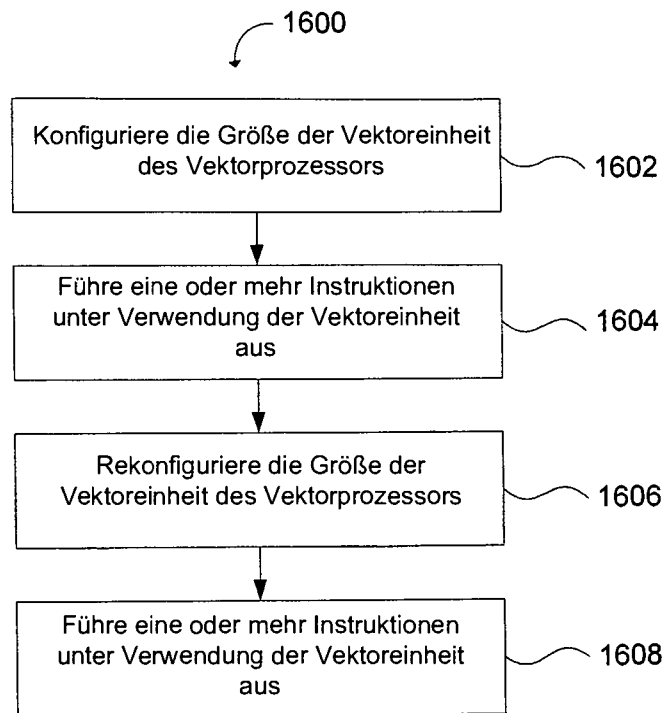
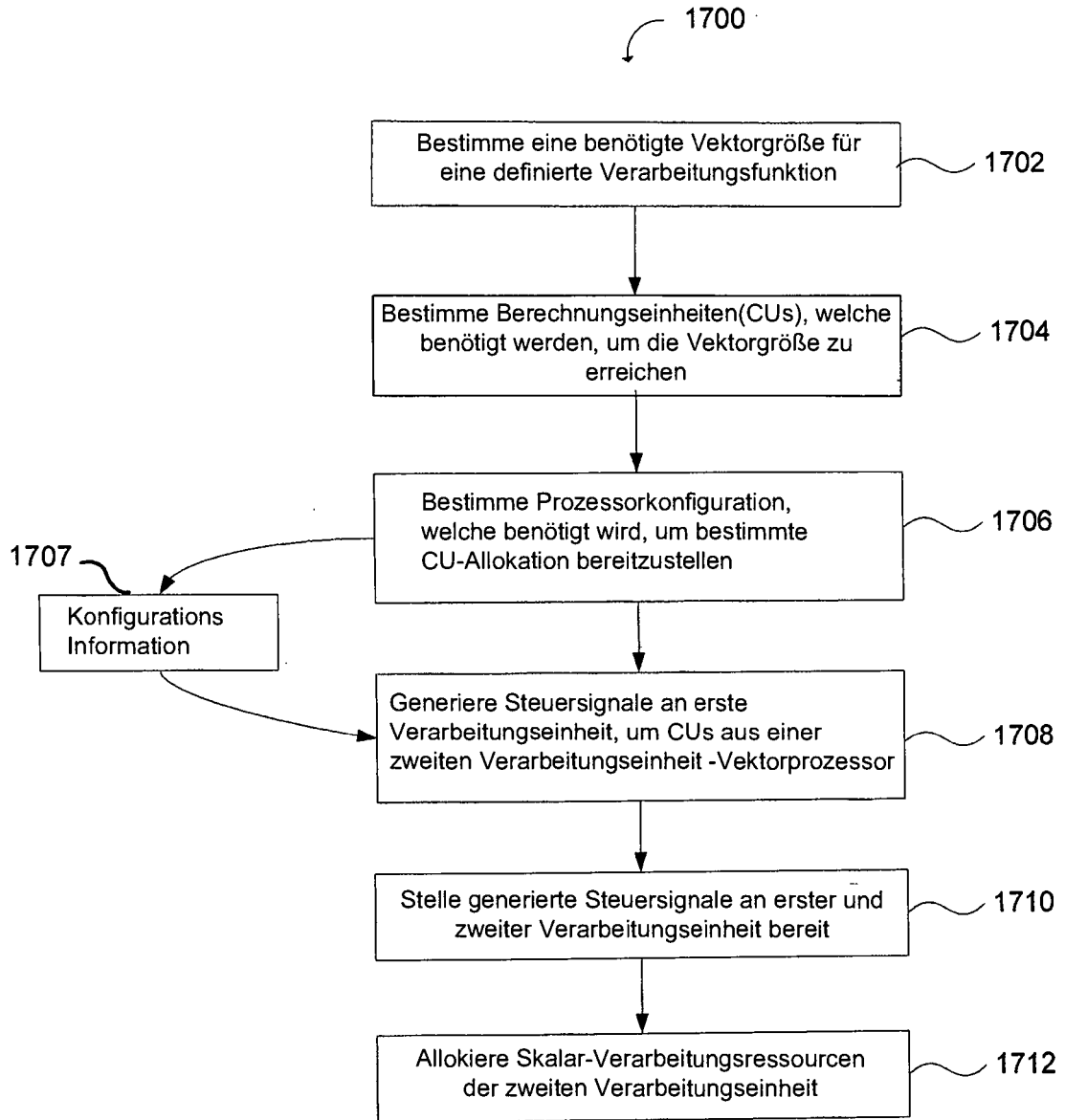


Figure 15



Figur 16



Figur 17