



BUNDESPATENTGERICHT

17 W (pat) 33/20

(Aktenzeichen)

Verkündet am
16. Januar 2024

...

BESCHLUSS

In der Beschwerdesache

...

betreffend die Patentanmeldung 11 2012 002 465.6

hat der 17. Senat (Technischer Beschwerdesenat) des Bundespatentgerichts auf die mündliche Verhandlung vom 16. Januar 2024 unter Mitwirkung des Vorsitzenden Richters Dipl.-Phys. Dr. Morawek, der Richterin Akintche, des Richters Dipl.-Phys. Dr. Forkel und des Richters Dipl.-Phys. Dr. Städele

beschlossen:

Die Beschwerde wird zurückgewiesen.

Gründe

I.

Die vorliegende Patentanmeldung 11 2012 002 465.6 trägt die Bezeichnung

„Grafikprozessor mit nicht blockierender gleichzeitiger Architektur“

und geht aus einer PCT-Anmeldung hervor, die am 15. Juni 2012 unter Inanspruchnahme einer US-amerikanischen Priorität vom 16. Juni 2011 (US 61/497,915) eingereicht und in englischer Sprache als WO 2012 / 174 334 A1 und in deutscher Übersetzung als DE 11 2012 002 465 T5 veröffentlicht worden ist. Die Patentanmeldung wurde durch Beschluss der Prüfungsstelle für Klasse G06F des Deutschen Patent- und Markenamts vom 7. Oktober 2020 zurückgewiesen, da die Gegenstände der Patentansprüche 1 nach dem Hauptantrag sowie den Hilfsanträgen 1 bis 3 nicht auf einer erfinderischen Tätigkeit des Fachmanns beruhten, soweit sie ein technisches Problem mit technischen Mitteln lösten und damit eine Erfindung auf Gebieten der Technik betreffen.

Gegen diesen Beschluss ist die Beschwerde der Anmelderin gerichtet.

Die Anmelderin stellt den Antrag,

den Beschluss der Prüfungsstelle für Klasse G06F des Deutschen Patent- und Markenamts vom 7. Oktober 2020 aufzuheben und das Patent auf der Grundlage folgender Unterlagen zu erteilen:

Hauptantrag:

Patentansprüche 1 bis 23, eingegangen am 25. September 2020,
Beschreibung Seiten 1 bis 47, eingegangen am 16. Dezember 2013
sowie
12 Blatt Zeichnungen mit Figuren 1 bis 16, eingegangen am 16.
Dezember 2013;

Hilfsantrag 1:

Patentansprüche 1 bis 23, eingegangen am 25. September 2020,
Beschreibung und Zeichnungen wie Hauptantrag;

Hilfsantrag 2:

Patentansprüche 1 bis 26, eingegangen am 25. September 2020,
Beschreibung und Zeichnungen wie Hauptantrag;

Hilfsantrag 3:

Patentansprüche 1 bis 23, eingegangen am 7. Oktober 2020,
Beschreibung und Zeichnungen wie Hauptantrag;

Hilfsantrag 4:

Patentanspruch 1, überreicht in der mündlichen Verhandlung,
Patentansprüche 2 bis 11 basierend auf Hilfsantrag 3,
Beschreibung und Zeichnungen wie Hauptantrag.

Der geltende **Patentanspruch 1** gemäß **Hauptantrag** lautet (mit einer Gliederung versehen, die an die Gliederung aus dem Zurückweisungsbeschluss angelehnt ist):

A1 Verarbeitungssystem, das in der Lage ist, gleichzeitige
Grafikberechnung durchzuführen, umfassend:

- A2** eine Vielzahl von Berechnungselementen, wobei jedes Berechnungselement einen lokalen Speicher und eine Vielzahl von arithmetisch-logischen Einheiten (ALU) umfasst,
- A2.1** wobei die jeweiligen arithmetisch-logischen Einheiten jedes der Vielzahl von Berechnungselementen von einem gemeinsamen Programmzähler aus betrieben werden
- A3** einen Scheduler für die Berechnungselemente,
- A3a** wobei der Scheduler einen Grob-Scheduler (44) und eine Paketeinheit (105, 205) umfasst und
- A3b** betriebsfähig ist, Eingaben zu empfangen, die Instanzen einer ersten an der Vielzahl von Berechnungselementen durchzuführenden Berechnungsart definieren,
- A3.1** wobei jede Berechnungsinstanz mit einem jeweiligen Datenelement, das in einem lokalen Speicher eines der Vielzahl von Berechnungselementen gespeichert ist, und einem jeweiligen Scheduling-Schlüssel assoziiert ist,
- A3.2** wobei der Scheduler betriebsfähig ist, die Berechnungsinstanzen nach ihren Scheduling-Schlüsseln zu Paketen zu gruppieren und ein Paket auszugeben, das eine Vielzahl von Berechnungsinstanzen umfasst; und
- A4** einen Verteiler, der betriebsfähig ist, das Paket zu empfangen und die Berechnungsinstanzen auf die Berechnungselemente des Clusters zur Durchführung auf diesen zu verteilen,
- A4.1** wobei jedes der Berechnungselemente betriebsfähig ist, Berechnungsinstanzen zur gleichzeitigen Durchführung durch ihre

ALUs zu kombinieren, für welche dieselbe Art von Berechnung durchzuführen ist,

A4.2 und die sich auf verschiedene Datenelemente beziehen, die in ihrem lokalen Speicher gespeichert sind, um gleichzeitig die Berechnung für die kombinierten Berechnungsinstanzen durchzuführen.

Patentanspruch 1 nach **Hilfsantrag 1** beruht auf Patentanspruch 1 nach Hauptantrag, wobei Merkmal **A3.2** durch das Merkmal

A3.2^{Hi1} wobei der Scheduler betriebsfähig ist, die Berechnungsinstanzen nach ihren Scheduling-Schlüsseln zu Paketen zu gruppieren und ein Paket auszugeben, das eine Vielzahl von Berechnungsinstanzen umfasst, die einen Scheduling-Schlüssel teilen; und

ersetzt wird (Änderungen gegenüber Hauptantrag markiert).

Patentanspruch 1 nach **Hilfsantrag 2** unterscheidet sich von Patentanspruch 1 nach Hauptantrag dadurch, dass zwischen den Merkmalen **A3.1** und **A3.2** das Merkmal

A3.3^{Hi2} wobei jede Berechnungsinstanz einen Identifikator für ein als Scheduling-Schlüssel verwendetes Datenelement umfasst,

eingeschoben wird.

Patentanspruch 1 nach **Hilfsantrag 3** geht aus Patentanspruch 1 nach Hilfsantrag 2 hervor, indem die zwischen den Merkmalen **A3.3^{Hi2}** und **A3.2** das Merkmal

A3.4^{Hi3} wobei das als Scheduling-Schlüssel verwendete Datenelement einen Offset des globalen Speichers, eine Verzweigungsbedingung, eine Speicherreferenzen einer Berechnungsinstanz, eine Speicheradresse

einer Hauptspeicherleseoperation und/oder eine Ausgangsspeicher-
adresse eines ersten Elements einer Beschleunigerstruktur umfasst.

eingefügt wird.

Bei **Patentanspruch 1** nach **Hilfsantrag 4** fehlt gegenüber Patentanspruch 1 nach Hauptantrag das Merkmal **A3a**; ferner wurden die Merkmale **A3.1** und **A3.2** durch die Merkmale

A3.1^{Hi4} wobei jede Berechnungsinstanz ein erstes und ein zweites Datenelement aufweist,
wobei eines der Datenelemente als Scheduling-Schlüssel verwendet wird, um zu steuern, wie diese Arbeitsbelastung mit anderen Arbeitsbelastungen gruppiert wird und das andere steuert, wo unter der Vielzahl von Berechnungselementen diese Berechnungsinstanz ausgeführt wird,

und **A3.2^{Hi1}** ersetzt sowie das Merkmal **A3.4^{Hi3}** in den Anspruch aufgenommen.

Zu den nebengeordneten Patentansprüchen sowie zu den Unteransprüchen aller Anträge wird auf die Akte verwiesen.

Im Prüfungsverfahren vor dem deutschen Patent- und Markenamt sind u.a. die Druckschriften

P1 US 7 925 860 B1

und

P2 US 2008 / 0 074 433 A1

genannt worden.

II.

Die Beschwerde wurde rechtzeitig eingelegt und ist auch sonst zulässig. Sie hat jedoch keinen Erfolg, da der Gegenstand des jeweiligen Patentanspruchs 1 nach Hauptantrag sowie nach den Hilfsanträgen 1 bis 4 nicht auf erfinderischer Tätigkeit beruht (§ 1 Abs. 1 i. V. m. § 4 Satz 1 PatG).

1. Die vorliegende Patentanmeldung betrifft die Verarbeitung von Daten mittels paralleler Verarbeitungsressourcen.

In der Beschreibung der Anmeldung wird ausgeführt, dass sich eine parallele Datenverarbeitung nicht analog zur Verfügbarkeit von Berechnungsressourcen skalieren lasse. Ein Grund dafür sei, dass die Speicherbandbreite nicht im Verhältnis zur verfügbaren Rechenleistung skaliere. Berechnungsprobleme, die keine inhärente Datenlokalität aufweisen, könnten daher dazu führen, dass Berechnungselemente auf Daten aus dem Speicher, auf den sie gerade zugreifen, warten müssten (vgl. DE 11 2012 002 465 T5 (im Folgenden „DE-T5“ genannt), Absatz [0043]).

Außerdem ergebe sich aus der SIMD-Beschaffenheit von Berechnungsclustern, dass das gleichzeitige Ausführen von Threads identischen Programmausführungspfaden folgen müsse, um einen maximalen Berechnungsdurchsatz zu erzielen. Wenn sich beispielsweise die Hälfte der Threads in einem SIMD-Cluster in die eine und der Rest der Threads in die andere Richtung verzweige, müsse die Hardware diese beiden Pfade serialisieren. In einer solchen Situation erreiche die Berechnungseinheit nur 50 % des möglichen Durchsatzes. In Situationen, in denen eine ausgeführte Codebasis zahlreiche Zweige enthalte, könne sich die Leistung im schlechtesten Fall um den Faktor $1/\text{SIMD-Breite}$ reduzieren, was bei einer SIMD-Architektur der Breite 32 etwa 3% Wirkungsgrad entspreche. Es gebe eine große Anzahl von Berechnungsproblemen,

die zwar theoretisch parallelisierbar seien, aber nicht effizient an solche breiten SIMD-Architekturen angepasst worden seien (DE-T5, Absatz [0044]).

Eine **Aufgabe** wird in der vorliegenden Patentanmeldung nicht ausdrücklich genannt. Jedoch lassen die vorstehend zitierten Absätze [0043] und [0044] der Druckschrift DE-T5 auf die Aufgabenstellung schließen, ein Grafikdatenverarbeitungssystem bereitzustellen, mit dem sich ein höherer Berechnungsdurchsatz und demzufolge eine höhere Datenverarbeitungsgeschwindigkeit erzielen lassen, als dies bei aus dem Stand der Technik bekannten Grafikdatenverarbeitungssystemen der Fall ist.

Als **Fachmann**, der mit der Lösung dieser Aufgabe betraut wird, ist ein Informatiker oder Ingenieur der Fachrichtung Elektrotechnik mit mehrjähriger Erfahrung in der Entwicklung paralleler, insbesondere für Grafikverarbeitungssysteme verwendbarer Datenverarbeitungsarchitekturen anzusehen.

2. Zur Lehre von Patentanspruch 1

Patentanspruch 1 nach Hauptantrag sieht ein Verarbeitungssystem vor, das in der Lage ist, gleichzeitige Grafikberechnung durchzuführen (Merkmal **A1**) und eine Vielzahl von Berechnungselementen umfasst, von denen jedes einen lokalen Speicher und eine Vielzahl von arithmetisch-logischen Einheiten (ALU) umfasst (Merkmal **A2**).

Gemäß Merkmal **A2.1** werden die ALUs jedes dieser Berechnungselemente von einem gemeinsamen Programmzähler aus betrieben. Dieses Merkmal ist aus Sicht des Fachmanns insbesondere dann erfüllt, wenn die Befehle, die eine jeweilige ALU zur Ausführung erhält, mittels desselben - gemeinsamen - Programmzählers aus einem Speicher abgerufen werden wie die Befehle mindestens einer weiteren ALU, die von der Vielzahl von Berechnungselementen umfasst wird. Nach dem Wortlaut von Merkmal **A2.1** ist nicht ausgeschlossen, dass das Verarbeitungssystem noch

weitere Programmzähler zum Abruf von Befehlen umfasst, die an ALUs übermittelt werden, die nicht unter die in Merkmal **A2** definierte Vielzahl von ALUs eines Berechnungselements zu subsumieren sind.

Des Weiteren umfasst das Verarbeitungssystem laut Merkmal **A3** i. V. m. Merkmal **A1** einen Scheduler für die Berechnungselemente, d.h. eine Einheit, die die zeitliche Ausführung von Arbeitsaufgaben durch die Berechnungselemente regelt oder beeinflusst, z.B. indem sie die Aufgaben auf verschiedene Ressourcen (z.B. auf Prozessoren) verteilt.

Der Scheduler umfasst seinerseits einen Grob-Scheduler und eine Paketeinheit (Merkmal **A3a**). Was unter einem Grob-Scheduler - ein solcher ist in den Figuren 1 und 2 der DE-T5 abgebildet - konkret zu verstehen sein soll, lässt die vorliegende Anmeldung offen. Da in ihr aber auch von einem „lokalen Scheduler“ und einem „ALU-Cluster-Scheduler“ die Rede ist (vgl. DE-T5, Absätze [0025], [0091]; Fig. 6), wird der Fachmann einen Grob-Scheduler insbesondere als einen Scheduler ansehen, der Arbeitsaufgaben auf die einzelnen Berechnungselemente verteilt, aber nicht direkt an die einzelnen ALUs innerhalb eines jeweiligen Berechnungselements weitergibt. Weiterhin wird der Fachmann eine anspruchsgemäße Paketeinheit vor dem Hintergrund der Absätze [0078] und [0080] der DE-T5 als eine Hardware- und/oder Softwarekomponente ansehen, die Berechnungsinstanzen in Form von „Paketen“ (d.h. Gruppen) verarbeitet.

Der Scheduler soll ferner betriebsfähig sein, Eingaben zu empfangen, die Instanzen einer ersten an der Vielzahl von Berechnungselementen durchzuführenden Berechnungsart definieren (Merkmal **A3b**). Der Fachmann versteht unter derartigen Instanzen insbesondere Threads, die eine bestimmte Datenverarbeitungsoperation - z.B. eine numerische Operation wie z.B. eine Addition oder Multiplikation, die von einem Vertex- oder Pixelshader festgelegt wird - ausführen (vgl. DE-T5, Absätze [0073] und [0074]).

Ferner soll jede Berechnungsinstanz mit einem jeweiligen Datenelement, das in einem lokalen Speicher eines der Vielzahl von Berechnungselementen gespeichert ist, und einem jeweiligen Scheduling-Schlüssel assoziiert sein (Merkmal **A3.1**), wobei der Scheduler entsprechend dem Merkmal **A3.2** die Berechnungsinstanzen nach ihren Scheduling-Schlüsseln zu Paketen gruppieren und ein Paket ausgeben kann, das eine Vielzahl von Berechnungsinstanzen umfasst.

Laut Beschreibung der vorliegenden Patentanmeldung können solche Scheduling-Schlüssel nicht nur zur Gruppierung von Berechnungsinstanzen eingesetzt werden, sondern auch, um Kriterien anzuzeigen, die beim Scheduling der Ausführung einer „fibre“ bzw. Faser (d.h. eines speziellen Threads) verwendet werden, oder aber auch, um Fasern zu sammeln oder zu sortieren (vgl. DE-T5, Absätze [0063], [0069], [0086], [0087], [0089], [0090], [0113], [0119]). Ein Scheduling-Schlüssel kann insbesondere eine Speicherreferenz, eine Speicheradresse oder ein Teil einer solchen Adresse sein (DE-T5, Absätze [0108], [0112]).

Der Fachmann versteht daher unter einem anspruchsgemäßen Scheduling-Schlüssel eine Information, anhand derer Berechnungsinstanzen beim Scheduling zu Paketen zusammengestellt werden, d.h. insbesondere zu Gruppen von Berechnungsinstanzen, die zusammen verarbeitet werden sollen.

Weiterhin soll der Verteiler betriebsfähig sein, das von dem Scheduler ausgegebene Paket zu empfangen und die Berechnungsinstanzen auf die Berechnungselemente „des Clusters“ zur Durchführung auf diesen zu verteilen (Merkmal **A4**). Da von einem Cluster an keiner anderen Stelle des Patentanspruchs 1 die Rede ist, wird der Fachmann Merkmal **A4** dahingehend interpretieren, dass der Verteiler imstande sein soll, die Berechnungsinstanzen des Pakets an Berechnungselemente eines (beliebigen) Clusters weiterzugeben, zu übertragen oder auf diese aufzuteilen (vgl. DE-T5, Absätze [0078] bis [0080] i. V. m. Figur 2, Bezugszeichen 225; Absatz [0091] i. V. m. Figur 6; Absatz [0100] i. V. m. Figur 8). Ein „Cluster“ ist laut Absatz [0081]

der DE-T5 ein Mittel, um gleichzeitige Berechnungen auf einer Vielzahl von Datenelementen mittels eines einzelnen Steuerflusses durchzuführen.

Jedes der Berechnungselemente soll betriebsfähig sein, Berechnungsinstanzen, für die dieselbe Art von Berechnung durchzuführen ist, zur gleichzeitigen Durchführung durch ihre ALUs zu kombinieren (Merkmal **A4.1**). Dabei sollen sich die kombinierten Berechnungsinstanzen auf verschiedene Datenelemente beziehen, die im lokalen Speicher eines jeweiligen Berechnungselements gespeichert sind, um gleichzeitig die Berechnung für die kombinierten Berechnungsinstanzen durchzuführen (Merkmal **A4.2**).

3. Der **Hauptantrag** hat keinen Erfolg, weil der Gegenstand seines Patentanspruchs 1 nicht auf erfinderischer Tätigkeit beruht.

3.1 Zur Beurteilung der beanspruchten Lehre ist die Druckschrift **P1** von besonderer Bedeutung.

So beschreibt die Druckschrift **P1** ein Computersystem 100 mit einem Grafikverarbeitungssystem 112 und N Grafikverarbeitungseinheiten 122, von denen jede ein Core-Array 202 besitzt, das mehrere Threads gleichzeitig abarbeiten kann. Ein Core-Array 202 umfasst mindestens einen Verarbeitungscluster 302, der wiederum mindestens einen Core 310 enthält, mit dem Threads für allgemeine Berechnungen oder auch für spezielle Bildverarbeitungsberechnungen wie z. B. ein Vertex-, Geometrie- oder Pixel-Shading ausgeführt werden (Spalte 4, Zeilen 36 bis 60; Spalte 5, Zeile 42 bis 61; Figuren 1 bis 3). Jeder Core 310 besteht aus P Parallelprozessoren 320, die jeweils eine Reihe von arithmetisch-logischen Einheiten sowie ein lokales Registerfile 325 enthalten, SIMD-Befehle von einer Befehlseinheit 312 entgegennehmen und G Threads gleichzeitig („concurrently“) ausführen können (Spalte 6, Zeilen 4 bis 9 sowie 20 bis 23; Spalte 7, Zeile 11 bis 15; Figur 3).

Damit offenbart die Druckschrift **P1** ein Verarbeitungssystem, welches in der Lage ist, gleichzeitige Grafikberechnungen auszuführen, und welches mit den Cores 310 eine Vielzahl von Berechnungselementen enthält, die mit dem lokalen Registerfile 325 neben einer Vielzahl von arithmetisch-logischen Einheiten jeweils auch einen lokalen Speicher enthalten - Merkmale **A1** und **A2**.

Jeder Core 310 kann die Threads nach dem sogenannten SIMD-Prinzip („SIMD“ = „single-instruction, multiple-data“) abarbeiten, um die Ressourcenallokation zu vereinfachen. Dazu werden jeweils P Threads von einem Core-Array 202 zu einer SIMD-Gruppe zusammengestellt, indem diesen ein Gruppenindex GID zugewiesen wird (Spalte 5, Zeilen 55 bis 65; Spalte 6, Zeilen 50 bis 56 - „[...] a “group index“ (GID) for the associated thread may be included with the instruction“ i. V. m. Spalte 20, Zeilen 8 bis 11 - „organizing the threads [...] into a plurality of SIMD groups“). Die Core-Schnittstelle 308 gibt eine SIMD-Gruppe an einen der Cores 310 weiter und übermittelt diesem den Befehl, mit der Ausführung der SIMD-Gruppe zu beginnen (Spalte 6, Zeile 61 bis Spalte 7, Zeile 10 - „core interface 308 [...] loads a SIMD group into core 310, then instructs core to launch the group“). Dazu erhalten alle P Parallelprozessoren 320 eines Cores 310 von der Befehlseinheit 312 in jedem Taktzyklus denselben (SIMD-)Befehl, der auf unterschiedlichen Datenelementen ausgeführt wird (Spalte 6, Zeilen 29 bis 33 sowie 38 bis 44 und 50 bis 56 - „On each clock cycle, one instruction is issued to all P Threads making up a selected one of the G SIMD groups“; Spalte 9, Zeilen 28 bis 30; Spalte 10, Zeilen 3 bis 10).

Hieraus ergibt sich unmittelbar, dass ein Gruppenindex GID und eine SIMD-Gruppe einen Scheduling-Schlüssel bzw. ein aus einer Vielzahl von Berechnungsinstanzen bestehendes Paket im Sinne der Merkmale **A3.1** und **A3.2** darstellen.

Ferner stellt die Core-Schnittstelle 308 einen anspruchsgemäßen Grob-Scheduler dar, der die SIMD-Gruppen erstellt und an die Cores 310 weitergibt. Da die Cores 310 die SIMD-Gruppen verarbeiten und insbesondere deren Ausführung veranlassen (vgl. das obenstehende Zitat „core interface 308 [...] instructs core to

launch the group“), kann die Gesamtheit der Cores 310 als eine Paketeinheit angesehen werden, die zusammen mit der Core-Schnittstelle 308 einen Scheduler für die Threads bildet. Damit liegen die Merkmale **A3**, **A3a** und **A3.2** vor.

Weiterhin zeigt die Druckschrift **P1**, dass Threads zur Vereinfachung der Ressourcenzuteilung innerhalb der Grafikprozessoren auch in Form sogenannter kooperativer Thread-Arrays („cooperative thread arrays“ = CTAs) abgearbeitet werden können. Ein CTA besteht aus mehreren Threads oder SIMD-Gruppen (Spalte 7, Zeilen 39 bis 44 sowie 52 bis 56; Spalte 8, Zeile 35; Spalte 9, Zeilen 39 bis 43) und wird von einem Core-Array 202 gebildet, indem dieses jedem Thread des CTA eine CTA-ID zuordnet (Spalte 18, Zeilen 46 bis 48; Spalte 8, Zeile 31). Insoweit stellen auch die Größen CTA-ID bzw. CTA einen Scheduling-Schlüssel bzw. ein eine Vielzahl von Berechnungsinstanzen umfassendes Paket im Sinne der Merkmale **A3.1** und **A3.2** dar.

Der **P1** ist ferner zu entnehmen, dass ein Core-Array 202 ein CTA an einen der Cores 310 weitergibt (Spalte 17, Zeilen 50/51). Daraus folgt, dass Bestandteile des Core-Arrays 202, die außerhalb der Cores 310 angeordnet sind, die Threads zu den CTAs gruppieren. Da diese Bestandteile auch die zeitliche Ausführung der Threads beeinflussen, indem sie diese zu einem definierten Zeitpunkt in Form einzelner CTAs an einen jeweiligen Core 310 weitergeben, bilden sie einen Grob-Scheduler für die Cores 310, welcher zusammen mit der Gesamtheit der Cores 310 einen anspruchsgemäßen Scheduler repräsentiert (ebenfalls Merkmale **A3**, **A3a**). Da der Grob-Scheduler die Threads zu CTAs gruppiert und an einen jeweiligen Core weitergibt (s.o.), wird das Merkmal **A3.2** auch auf diese Weise in der Lehre der **P1** verwirklicht.

Des Weiteren ist es selbstverständlich, dass das Core-Array 202 zu verarbeitende Daten in Gestalt von Eingaben empfängt, welche von der Rendering-Pipeline 200 und dem Shading-Modul 210 stammen (vgl. **P1**, Figur 2), bestimmte, von den

Threads auszuführende Berechnungen (z.B. Shader-Berechnungen) festlegen und damit auch die Threads selbst definieren (Merkmal **A3b**).

Da die von einem Thread verarbeiteten Datenelemente in dem lokalen Registerfile 325, also einem lokalen Speicher, gespeichert sein können und ferner einem Thread ein Gruppenindex GID einer SIMD-Gruppe (Spalte 6, Zeilen 51 bis 54; Spalte 12, Zeile 12 bis 17) oder auch eine CTA-ID (Spalte 8, Zeile 31; Spalte 18, Zeile 46 bis 48) als Scheduling-Schlüssel zugeordnet sein kann, liegt auch Merkmal **A3.1** vor.

Ferner müssen die Threads eines CTA von einem jeweiligen Core 310 empfangen und in geeigneter Weise auf die Parallelprozessoren 320 verteilt werden, um dort ausgeführt werden zu können. Die Hardware- und/oder Softwareeinheiten eines Cores 310, die diese Verteilung bewerkstelligen, bilden einen Verteiler im Sinne von Merkmal **A4**. Da die Parallelprozessoren 320 Berechnungselemente eines Verarbeitungsclusters 302 sind (vgl. Figur 3), ist auch Merkmal **A4** gemäß der vorstehenden Auslegung aus Abschnitt **II.2** erfüllt.

Wie vorstehend ausgeführt, können die Parallelprozessoren 320 nach dem SIMD-Prinzip arbeiten und G Threads, die beispielsweise Shader-Berechnungen - und damit eine bestimmte Berechnungsart - repräsentieren, gleichzeitig ausführen. Das bedeutet, dass jeder Core 310 diese G Threads zur gleichzeitigen Ausführung durch die ALUs, auf denen die Threads abgearbeitet werden, kombiniert (Merkmal **A4.1**).

Da die Datenelemente, die von diesen Threads verarbeitet werden, in dem lokalen Registerfile 325 gespeichert sein können, liegt auch Merkmal **A4.2** vor.

Der **P1** ist allerdings nicht ausdrücklich zu entnehmen, dass die arithmetisch-logischen Einheiten eines Cores 310 gemäß dem verbleibenden Merkmal **A2.1** von einem Programmzähler aus betrieben werden, der diesen Einheiten gemeinsam ist.

3.2 Ausgehend von der Druckschrift **P1** gelangt der Fachmann jedoch ohne erfinderisches Zutun zum Gegenstand des Patentanspruchs 1 gemäß Hauptantrag.

So ist in **P1** die im Zusammenhang mit einer parallelen Datenverarbeitung gemäß dem SIMD-Prinzip fachübliche Verfahrensweise beschrieben, dass die Befehlseinheit 312 allen Parallelprozessoren 320 in jedem Taktzyklus - also gleichzeitig - dieselbe SIMD-Instruktion übermittelt (Spalte 6, Zeilen 29 bis 31, 38 bis 40 und 50/51) und zum Abruf dieser Instruktion nur eine einzige Steuereinheit verwendet wird (Spalte 5, Zeilen 61 bis 65). Für den Fachmann liegt es daher auf der Hand, auch nur einen einzigen Programmzähler zum Abruf der SIMD-Instruktion zu verwenden. In diesem Fall werden die ALUs der Parallelprozessoren 320, die die SIMD-Instruktion verarbeiten, von einem gemeinsamen Programmzähler aus betrieben, wie es laut Merkmal **A2.1** beansprucht ist.

3.3 Die Anmelderin ist sinngemäß der Auffassung, die vorliegende Erfindung erlaube die Möglichkeit einer flexiblen Mischung zwischen paralleler und serieller Ausführung von Fasern in Abhängigkeit von einem Scheduling-Schlüssel oder dem Umstand, dass sich eine Berechnung auf dieselben oder verschiedene Datenelemente bezieht. Diese Möglichkeit sei nicht in **P1** gezeigt, da ein CTA mehrere vollständige Threads umfasse und daher nicht mit einem Scheduling-Schlüssel gleichgesetzt werden könne. Ferner ermögliche es die Erfindung, Berechnungsinstanzen mit verschiedenen Scheduling-Schlüsseln zur gleichzeitigen Durchführung zu kombinieren.

Diese Argumentation hält einer näheren Überprüfung nicht stand.

So bezieht sich Patentanspruch 1 bereits nicht ausdrücklich auf Fasern, sondern auf beliebige Berechnungsinstanzen (z.B. auf jede Art von Threads). Dass Berechnungsinstanzen in Abhängigkeit eines Scheduling-Schlüssels seriell oder parallel verarbeitet werden, ist ebensowenig Gegenstand von Patentanspruch 1 wie der Umstand, dass Berechnungsinstanzen mit verschiedenen Scheduling-

Schlüsseln gleichzeitig ausgeführt werden müssen. Patentanspruch 1 fordert nur, dass Berechnungsinstanzen gemäß ihren Scheduling-Schlüsseln zu Paketen gruppiert werden (vgl. Merkmal **A3.2**) und Berechnungsinstanzen, für die dieselbe Art von Berechnungen durchzuführen ist, die aber ansonsten beliebig sind, zur gleichzeitigen Durchführung kombiniert werden können (vgl. Merkmal **A4.1**). Diese beiden Kriterien sind erfüllt, wenn - wie in **P1** beschrieben - Threads gemäß ihren SIMD-Gruppenindizes oder CTA-IDs zu SIMD-Gruppen bzw. CTAs zusammengestellt werden und die ALUs der Parallelprozessoren 320 Threads verschiedener SIMD-Gruppen bzw. CTAs nach dem SIMD-Prinzip parallel abarbeiten.

3.4 Somit waren für den Fachmann lediglich fachgemäße Überlegungen erforderlich, um in Kenntnis der Druckschrift **P1** zu einem Verfahren mit den Merkmalen des Patentanspruchs 1 in der Fassung des Hauptantrags zu gelangen.

3.5 Die Anmelderin verfolgt ihr Patentbegehren mit dem Hauptantrag und den Hilfsanträgen jeweils in Form eines in sich geschlossenen Anspruchssatzes. Da über einen Antrag nur einheitlich entschieden werden kann, sind daher auch die Patentansprüche 2 bis 23 gemäß Hauptantrag nicht gewährbar (vgl. BGH GRUR 2017, 57 ff. – Datengenerator; GRUR 2007, 862 ff. – Informationsübermittlungsverfahren II; GRUR 1997, 120 – Elektrisches Speicherheizgerät).

4. Hilfsantrag 1 ist nicht günstiger als der Hauptantrag zu bewerten.

4.1 Patentanspruch 1 nach Hilfsantrag 1 beruht auf dem Patentanspruch 1 nach Hauptantrag, wobei Merkmal **A3.2** durch das Merkmal **A3.2^{Hi1}** ersetzt worden ist.

Das letztgenannte Merkmal fügt dem Gegenstand von Patentanspruch 1 nach Hauptantrag hinzu, dass diejenigen Berechnungsinstanzen, die von dem Paket

umfasst sind, welches von dem Scheduler ausgegeben werden kann, einen Scheduling-Schlüssel teilen sollen.

4.2 Dieser Unterschied geht jedoch aus der Druckschrift **P1** hervor.

Denn die CTA-ID - ein anspruchsgemäßer Scheduling-Schlüssel - ist allen Threads eines CTAs gemeinsam (vgl. Spalte 8, Zeilen 25 bis 27 - „a CTA identifier (“CTA ID“) that is common to all threads in the CTA“). Entsprechendes gilt ferner auch für den Gruppenindex GID der Threads einer SIMD-Gruppe (vgl. Spalte 12, Zeile 16/17 - „x is a SIMD group index [...] that is the same for all threads in a SIMD group“).

4.3 Unter Berücksichtigung der Ausführungen zum Hauptantrag sind daher auch die Merkmale des Gegenstandes des Patentanspruchs 1 gemäß Hilfsantrag 1 aus der Druckschrift **P1** nahegelegt.

4.4 Mit dem Patentanspruch 1 gemäß Hilfsantrag 1 fallen auch die Patentansprüche 2 bis 23.

5. Auch dem **Hilfsantrag 2** kann nicht stattgegeben werden, weil die Lehre seines Patentanspruchs 1 für den Fachmann nahegelegen hat.

5.1 Hilfsantrag 2 beruht auf dem Hauptantrag, wobei zwischen den Merkmalen **A3.1** und **A3.2** das Merkmal **A3.3^{Hi2}** eingeschoben worden ist. Dieses Merkmal besagt, dass jede Berechnungsinstanz einen Identifikator für ein als Scheduling-Schlüssel verwendetes Datenelement umfassen soll.

Ein solcher Identifikator ist aus Sicht des Fachmanns eine Größe, die zur Identifikation, Kennzeichnung oder Ermittlung eines als Scheduling-Schlüssel verwendeten Datenelements herangezogen werden kann.

5.2. Auch Merkmal **A3.3^{Hi2}** kann eine Patentfähigkeit nicht begründen.

So wird laut **P1** an alle P Threads einer SIMD-Gruppe ein Befehl weitergegeben, der den Gruppenindex GID enthält, um anzuzeigen, welcher Thread aktuell aktiv ist (Spalte 6, Zeile 50 bis 54). Damit umfasst jeder der Threads diesen Befehl. Da sich der Gruppenindex - ein Datenelement - anhand des Befehls ermitteln lässt, stellt der Befehl einen Identifikator für den als Scheduling-Schlüssel verwendeten Gruppenindex dar.

Somit geht auch das Merkmal **A3.3^{Hi2}** aus **P1** hervor. Unter Berücksichtigung der Ausführungen zum Hauptantrag sind daher auch die Merkmale des Gegenstandes des Patentanspruchs 1 nach Hilfsantrag 2 durch die Lehre der Druckschrift **P1** nahegelegt.

5.3 Mit dem Patentanspruch 1 fällt der gesamte Hilfsantrag 2.

6. Hilfsantrag 3 bleibt ohne Erfolg, weil sein Patentanspruch 1 nichts Zusätzliches enthält, was eine Patentfähigkeit tragen könnte.

6.1 Im Vergleich mit dem Patentanspruch 1 nach Hilfsantrag 2 enthält der Patentanspruch 1 nach Hilfsantrag 3 zwischen den Merkmalen **A3.3^{Hi2}** und **A3.2** das weitere Merkmal

A3.4^{Hi3} wobei das als Scheduling-Schlüssel verwendete Datenelement einen Offset des globalen Speichers, eine Verzweigungsbedingung, eine Speicherreferenzen einer Berechnungsinstanz, eine Speicheradresse einer Hauptspeicherleseoperation und/oder eine Ausgangsspeicheradresse eines ersten Elements einer Beschleunigerstruktur umfasst.

Dabei versteht der Fachmann unter einem globalen Speicher insbesondere einen Hauptspeicher eines Computersystems (vgl. DE-T5, Figur 10, Bezugszeichen 584, 585, 587), und unter einem Offset des globalen Speichers einen Versatz, der in Form eines ganzzahligen Wertes zu einer Referenz- bzw. Basisadresse addiert wird, um eine absolute Speicheradresse festzulegen.

6.2 Auch das Merkmal **A3.4^{Hi3}** entnimmt der Fachmann der Druckschrift **P1**.

So geht aus **P1** hervor, dass die Adressen, über die die Threads einer SIMD-Gruppe auf die zu verarbeitenden Datenelemente zugreifen, zusammenhängenden Speicherbereichen des Grafikspeichers 124 oder des Systemspeichers 104 entsprechen können („Base+TID addressing“, vgl. Spalte 12, Zeilen 1 bis 12 und 36 bis 45 i. V. m. Spalte 5, Zeilen 29 bis 40 sowie Spalte 7, Zeilen 28 bis 31). Die beiden Speicher 104 und 124 sind die Hauptspeicher des Computersystems 100 bzw. des Grafikverarbeitungssubsystems 112 (vgl. **P1**, Figur 1) und somit jeweils globale Speicher.

Falls ein CTA mehrere SIMD-Gruppen umfasst, können die Speicheradressen gemäß den Gleichungen (2d) und (3) aus Spalte 12 der **P1** in der Form „base + y*w = a₀ + x*P*w + y*w“ geschrieben werden. Dabei ist die Größe „base“ eine Basisadresse. Der Parameter a₀ stellt die Adresse des ersten Elements, auf das zugegriffen wird (und damit eine Basisadresse für den gesamten CTA) dar. Ferner bezeichnet die Variable x den Gruppenindex einer aus P Threads bestehenden SIMD-Gruppe, und die Variable y gibt die Größe jedes Datenelements, auf das zugegriffen wird, in Bytes an (Spalte 12, Zeilen 20 bis 31 sowie 49 bis 60).

Der Fachmann wird erkennen, dass die Speicheradressen auch in der zu den Gleichungen (2d) und (3) äquivalenten Form „a₀ + x + x + ... + x + x + y*w“ geschrieben werden können, wobei hier der Gruppenindex P*w-mal addiert wird. Der Gruppenindex x bildet somit einen Versatz, der zu der Größe a₀ - einer Basisadresse für den gesamten CTA - mehrfach addiert wird, und stellt somit nicht nur einen Scheduling-Schlüssel, sondern auch einen Offset des globalen Speichers im Sinne des Merkmals **A3.4^{Hi3}** dar.

Damit geht die Alternative des Merkmals **A3.4^{Hi3}**, nach der das als Scheduling-Schlüssel verwendete Datenelement einen Offset des globalen Speichers umfasst, aus der **P1** hervor.

Mit Rücksicht auf die Ausführungen zu Hilfsantrag 2 ergibt sich der Gegenstand von Patentanspruch 1 nach Hilfsantrag 3 somit für den Fachmann in naheliegender Weise aus der **P1**.

6.3 Mit seinem Patentanspruch 1 fällt der gesamte Hilfsantrag 3.

7. Auch **Hilfsantrag 4** kann nicht günstiger beurteilt werden.

7.1 Hilfsantrag 4 beruht auf dem Hauptantrag, wobei in dessen Patentanspruch 1 die Merkmale **A3.1** und **A3.2** durch die Merkmale

A3.1^{Hi4} wobei jede Berechnungsinstanz ein erstes und ein zweites Datenelement aufweist,
wobei eines der Datenelemente als Scheduling-Schlüssel verwendet wird, um zu steuern, wie diese Arbeitsbelastung mit anderen Arbeitsbelastungen gruppiert wird und
das andere steuert, wo unter der Vielzahl von Berechnungselementen diese Berechnungsinstanz ausgeführt wird,

und **A3.2^{Hi1}** ersetzt wurden. Zudem wurde das Merkmal **A3a** gestrichen und Merkmal **A3.4^{Hi3}** neu in den Patentanspruch 1 aufgenommen.

Mit einer - durch Unterstreichung hervorgehobenen - redaktionellen Anpassung lauten die Merkmale **A3** und **A3b** wie folgt:

A3 einen Scheduler für die Berechnungselemente,

A3b der betriebsfähig ist, Eingaben zu empfangen, die Instanzen einer ersten an der Vielzahl von Berechnungselementen durchzuführenden Berechnungsart definieren,

Gemäß dem neuen Merkmal **A3.1^{Hi4}** weist jede Berechnungsinstanz nunmehr zwei Datenelemente auf. Eines dieser Elemente soll als Scheduling-Schlüssel verwendet

werden, um zu steuern, wie „diese Arbeitsbelastung“ - also das mit der Ausführung einer jeden Berechnungsinstanz verbundene Arbeitspensum, das die Berechnungselemente bewältigen müssen - mit anderen Arbeitsbelastungen gruppiert - d.h. zur gemeinsamen Abarbeitung zusammengestellt - wird.

Das andere Datenelement steuert, wo unter der Vielzahl von Berechnungselementen diese Berechnungsinstanz ausgeführt wird. Dies bedeutet insbesondere, dass mittels des anderen Datenelements mindestens ein innerhalb der Menge der Berechnungselemente liegender Ort festgelegt wird, an dem die Berechnungsinstanz ausgeführt werden soll.

7.2 Mit den Merkmalen **A3.1^{Hi4}**, **A3.2^{Hi1}** und **A3.4^{Hi3}** kann eine Patentfähigkeit nicht begründet werden.

7.2.1 Zu den Merkmalen **A3.2^{Hi1}** und **A3.4^{Hi3}** wird auf die vorstehenden Ausführungen zu den Hilfsanträgen 1 und 3 verwiesen.

7.2.2 Ferner wurde in Abschnitt **II.3** dieses Beschlusses im Zusammenhang mit Merkmal **A3.1** erläutert, dass jeder Thread einen Index GID einer SIMD-Gruppe und eine CTA-ID aufweisen kann. Diese Größen können jeweils als Scheduling-Schlüssel - d.h. als anspruchsgemäßes erstes Datenelement - verwendet werden, um - selbstverständlich mittels entsprechender Programmbefehle, die auf einem Rechner ausgeführt werden - zu steuern, wie die Threads zu SIMD-Gruppen bzw. CTAs gruppiert werden.

Dadurch, dass die Threads in Form von SIMD-Gruppen oder CTAs abgearbeitet werden, fallen die mit den Threads verbundenen Arbeitsbelastungen gleichzeitig an, d.h. die Gruppierung der Threads überträgt sich auf die Gruppierung der Arbeitsbelastungen. Diese sind somit ebenfalls gruppiert.

Daher ist der **erste Teil** von Merkmal **A3.1^{Hi4}** „wobei ~~eines der~~ Datenelemente als Scheduling-Schlüssel verwendet wird, um zu steuern, wie diese Arbeitsbelastung mit anderen Arbeitsbelastungen gruppiert wird“ der **P1** zu entnehmen.

7.2.3 Der verbleibende Teil von Merkmal **A3.1^{Hi4}**, gemäß dem eine Berechnungsinstanz neben dem als Scheduling-Schlüssel verwendeten Datenelement noch ein anderes, zweites Datenelement aufweist, das steuert, wo unter der Vielzahl von Berechnungselementen diese Berechnungsinstanz ausgeführt wird, ergibt sich aus dem zitierten Stand der Technik in mehrfacher Hinsicht:

(1) Laut **P1** wird jeder Thread einer SIMD-Gruppe oder eines CTA auf einen bestimmten Parallelprozessor 320 abgebildet (**P1**, Spalte 10, Zeile 6 bis 10 - „SIMD groups get scheduled onto the processing engines such that each thread is mapped to a particular processing engine“). Demnach wird jedem Thread eine bestimmte Information zugeordnet, die festlegt, wo - d.h. auf welchem Parallelprozessor 320 - der Thread abgearbeitet wird. Diese Information muss in dem Verarbeitungssystem der **P1** in Form eines entsprechenden (weiteren) Datenelements vorliegen.

(2) Aus der Angabe, dass die P Parallelprozessoren 320 eine Anzahl von P*G Threads gleichzeitig ausführen können (Spalte 6, Zeile 33 bis 35 - „each processing engine 320 is also multithreaded, supporting up to G threads, core 310 [...] can have up to P*G threads in process concurrently“), schließt der Fachmann, dass die ALUs eines Parallelprozessors 320 G Verarbeitungsbahnen bilden können, auf denen G Threads in demselben Taktzyklus parallel ausgeführt werden. In diesem Fall muss festgelegt werden, wo - d.h. auf welcher Verarbeitungsbahn - ein jeweiliger Thread in einem jeweiligen Taktzyklus abgearbeitet wird. Zur Festlegung der Verarbeitungsbahn eines Threads bietet sich die Thread-ID an, da diese laut **P1** zur Steuerung der Threadverarbeitung verwendet wird und für einen Thread während seiner Ausführung verfügbar ist (vgl. **P1**, Spalte 7, Zeilen 44 bis 51 - „[...] “thread ID“ or “TID“ [...] that is accessible to the thread during its execution. The TID controls

various aspects of the thread's processing behavior“). Somit spielt auch eine Thread-ID die Rolle des zweiten, anderen Datenelements aus Merkmal **A3.1**^{Hi4}.

(3) Im Übrigen ist dem Fachmann geläufig, dass Threads zur Vereinfachung der Ressourcenzuweisung eine Hardware-ID umfassen können, die eine dedizierte Hardwareeinheit kennzeichnet, an die ein Thread gesendet bzw. auf der der Thread ausgeführt werden soll (vgl. **P2**, Figur 2 mit Claims 24/25 sowie Absätzen [0046] - „To simplify scheduling, instructions may be associated with hardware identifiers (HW IDs) that indicate which hardware units to use for the instructions“, [0058] - „pre-decoder 450 [...] may determine the hardware unit to use for each instruction [...], [0059] - „The HW IDs may be stored in the context register for the thread and used for scheduling“).

Der Fachmann wird bei Bedarf die aus **P1** bekannten Cores 310 auch als dedizierte Hardwareeinheiten ausgestalten, die jeweils an allgemeine Berechnungen oder alternativ dazu an verschiedene Shader-Berechnungen angepasst sind, und in diesem Fall einem Thread zur Vereinfachung der Ressourcenzuweisung eine entsprechende Hardware-ID zuweisen. Diese kann ebenfalls als ein anderes, zweites Datenelement eines Threads angesehen werden, das steuert, wo der Thread ausgeführt wird.

7.2.4 Nachdem somit auch mit Merkmal **A3.1**^{Hi4} eine erfinderische Tätigkeit nicht begründet werden kann, ergibt sich der Gegenstand von Patentanspruch 1 nach Hilfsantrag 4 unter Berücksichtigung der Ausführungen zum Hauptantrag und den Hilfsanträgen 1 und 3 in naheliegender Weise aus dem Stand der Technik.

7.3 Mit dem Patentanspruch 1 nach Hilfsantrag 4 fallen auch die übrigen Patentansprüche dieses Antrags.

III.

Nachdem keiner der gestellten Anträge Erfolg hatte, war die Beschwerde der Anmelderin gegen den Zurückweisungsbeschluss der Prüfungsstelle für Klasse G06F des Deutschen Patent- und Markenamtes zurückzuweisen.

Rechtsmittelbelehrung

Gegen diesen Beschluss steht den am Beschwerdeverfahren Beteiligten das Rechtsmittel der Rechtsbeschwerde zu. Da der Senat die Rechtsbeschwerde nicht zugelassen hat, ist sie nur statthaft, wenn gerügt wird, dass

1. das beschließende Gericht nicht vorschriftsmäßig besetzt war,
2. bei dem Beschluss ein Richter mitgewirkt hat, der von der Ausübung des Richteramtes kraft Gesetzes ausgeschlossen oder wegen Besorgnis der Befangenheit mit Erfolg abgelehnt war,
3. einem Beteiligten das rechtliche Gehör versagt war,
4. ein Beteiligter im Verfahren nicht nach Vorschrift des Gesetzes vertreten war, sofern er nicht der Führung des Verfahrens ausdrücklich oder stillschweigend zugestimmt hat,
5. der Beschluss aufgrund einer mündlichen Verhandlung ergangen ist, bei der die Vorschriften über die Öffentlichkeit des Verfahrens verletzt worden sind, oder
6. der Beschluss nicht mit Gründen versehen ist.

Die Rechtsbeschwerde ist innerhalb eines Monats nach Zustellung des Beschlusses beim Bundesgerichtshof, Herrenstr. 45 a, 76133 Karlsruhe durch eine beim Bundesgerichtshof zugelassene Rechtsanwältin oder durch einen beim Bundesgerichtshof zugelassenen Rechtsanwalt einzulegen.

Dr. Morawek

Dr. Forkel

Akintche

Dr. Städele