



BUNDESPATENTGERICHT

18 W (pat) 21/17

(AktENZEICHEN)

Verkündet am
29. November 2019

...

BESCHLUSS

In der Beschwerdesache

betreffend die Patentanmeldung 10 2012 222 552.8

...

hat der 18. Senat (Technischer Beschwerdesenat) des Bundespatentgerichts auf die mündliche Verhandlung vom 29. November 2019 durch die Vorsitzende Richterin Dipl.-Ing. Wickborn sowie die Richter Kruppa, Dipl.-Ing. Veit und Dipl.-Ing. Altvater

beschlossen:

Auf die Beschwerde der Anmelderin wird der Beschluss der Prüfungsstelle für Klasse G 06 F des Deutschen Patent- und Markenamts vom 23. Mai 2017 aufgehoben und das Patent auf der Grundlage der folgenden Unterlagen erteilt:

- Patentansprüche 1 bis 8 gemäß Hilfsantrag 2, eingereicht in der mündlichen Verhandlung,
- Beschreibung, Seiten 1 bis 39, eingegangen am 1. März 2013,
- Figuren 1, 2, 3A bis 3C, 4 und 5, eingegangen am 1. März 2013.

Gründe

I.

Die am 7. Dezember 2012 beim Deutschen Patent- und Markenamt unter Inanspruchnahme der Priorität US 13/329,066 vom 16. Dezember 2011 eingereichte Patentanmeldung 10 2012 222 552.8 mit der Bezeichnung

„Aufeinanderfolgende Wiederholungen von divergenten Operationen“

wurde durch Beschluss der Prüfungsstelle für Klasse G06F des Deutschen Patent- und Markenamts in der Anhörung vom 23. Mai 2017 zurückgewiesen, da der Gegenstand des Patentanspruchs 1 nach dem Hauptantrag sowie den Hilfsanträgen 1 und 2 mit Rücksicht auf den aus den Druckschriften

- D3** Collange, S.: Multi-threading or SIMD? - How GPU architectures exploit regularity. In: Archi'11, 14.06.2011 [Auszug: S. 1, 3, 6, 8, 10, 13, 15, 29, 32, 43, 45, 47, 50, 51, 63, 65];
und
- D4** US 2004 / 0 172 523 A1.

bekanntem Stand der Technik nicht auf einer erfinderischen Tätigkeit beruhen.

Im Prüfungsverfahren wurden als Stand der Technik zudem die folgenden Druckschriften genannt:

- D1** US 2010 / 0 262 781 A1
- D2** Batten, C.; u. a.: Cache Refill/Access Decoupling for Vector Machines. In: Proceedings of the 37th International Symposium on Microarchitecture, 2004. S. 331-342.

Gegen den vorstehend genannten Beschluss richtet sich die Beschwerde der Anmelderin. Sie beantragt,

den Beschluss der Prüfungsstelle für Klasse G 06 F des Deutschen Patent- und Markenamts vom 23. Mai 2017 aufzuheben und das Patent auf der Grundlage der folgenden Unterlagen zu erteilen:

- Patentansprüche 1 bis 8, eingereicht in der mündlichen Verhandlung,
hilfsweise gemäß Hilfsantrag 1
- Patentansprüche 1 bis 8, eingereicht in der mündlichen Verhandlung,
hilfsweise gemäß Hilfsantrag 2
- Patentansprüche 1 bis 8, eingereicht in der mündlichen Verhandlung,

- Beschreibung, Seiten 1 bis 39, eingegangen am 1. März 2013,
- Figuren 1, 2, 3A bis 3C, 4 und 5, eingegangen am 1. März 2013.

Der seitens des Senats mit einer Gliederung versehene **Patentanspruch 1 nach Hauptantrag** lautet:

- 1.1** „Computer-implementiertes Verfahren zum Wiederholen einer Zugriffsoperation auf gemeinsam genutzte Ressourcen, wobei das Verfahren aufweist:
- 1.2** Auswählen eines ersten Threads und eines zweiten Threads von einer Gruppe von Threads, welche konfiguriert sind, eine Anweisung in einer Mehr-Stufen-Pipeline simultan auszuführen,
- 1.2a** wobei der erste Thread und der zweite Thread nicht bedient sind;
- 1.3** Auswählen, aus der Gruppe von Threads, eines ersten Satzes von Threads, welche mit dem ersten Thread zu assoziieren sind,
- 1.3a** wobei der erste Thread und jeder Thread in dem ersten Satz von Threads konfiguriert sind, auf eine erste gemeinsam genutzte Ressource zuzugreifen;
- 1.4** Auswählen, aus der Gruppe von Threads, eines zweiten Satzes von Threads, welche mit dem zweiten Thread zu assoziieren sind,
- 1.4a** wobei der zweite Thread und jeder Thread in dem zweiten Satz von Threads konfiguriert sind, auf eine zweite gemeinsam genutzte Ressource zuzugreifen;
- 1.5** Einfügen einer ersten Wiederholung-Operation, welche mit dem ersten Thread und dem ersten Satz von Threads assoziiert ist, in die Mehr-Stufen-Pipeline, um die Anweisung auszuführen; und

- 1.6 Einfügen einer zweiten Wiederholung-Operation, welche mit dem zweiten Thread und dem zweiten Satz von Threads assoziiert ist, in die Mehr-Stufen-Pipeline, um die Anweisung auszuführen,
- 1.7 wobei die zweite Wiederholung-Operation in die Mehr-Stufen-Pipeline seriell relativ zu der ersten Wiederholung-Operation eingefügt ist.“

Wegen des Wortlauts des nach Hauptantrag geltenden nebengeordneten Patentanspruchs 2 und der Unteransprüche 3 bis 8 wird auf die Akte verwiesen.

Der seitens des Senats mit einer Gliederung versehene **Patentanspruch 1 nach Hilfsantrag 1** lautet unter Hervorhebung der Änderungen gegenüber dem Hauptantrag:

- 1.1^{H1} „Computer-implementiertes Verfahren zum Wiederholen einer divergenten Zugriffsoperation auf gemeinsam genutzte Ressourcen, wobei das Verfahren aufweist:
- 1.2 Auswählen eines ersten Threads und eines zweiten Threads von einer Gruppe von Threads, welche konfiguriert sind, eine Anweisung in einer Mehr-Stufen-Pipeline simultan auszuführen,
- 1.2a^{H1} wobei der erste Thread auf eine erste gemeinsam genutzte Ressource zugreift,
- 1.2b^{H1} wobei der zweite Thread auf eine zweite gemeinsam genutzte Ressource zugreift,
- 1.2c^{H1} wobei auf die erste und die zweite gemeinsam genutzte Ressource nicht in derselben Zugriffsoperation zugegriffen werden kann,
- 1.2d^{H1} wodurch der erste Thread und der zweite Thread nicht bedient sind;

- 1.3 Auswählen, aus der Gruppe von Threads, eines ersten Satzes von Threads, welche mit dem ersten Thread zu assoziieren sind,
- 1.3a^{H1}** wobei ~~der erste Thread und~~ jeder Thread in dem ersten Satz von Threads konfiguriert ~~sind~~ ist, auf ~~eine~~ die erste gemeinsam genutzte Ressource zuzugreifen;
- 1.4 Auswählen, aus der Gruppe von Threads, eines zweiten Satzes von Threads, welche mit dem zweiten Thread zu assoziieren sind,
- 1.4a^{H1}** wobei ~~der zweite Thread und~~ jeder Thread in dem zweiten Satz von Threads konfiguriert ~~sind~~ ist, auf ~~eine~~ die zweite gemeinsam genutzte Ressource zuzugreifen;
- 1.5 Einfügen einer ersten Wiederholung-Operation, welche mit dem ersten Thread und dem ersten Satz von Threads assoziiert ist, in die Mehr-Stufen-Pipeline, um die Anweisung auszuführen; und
- 1.6 Einfügen einer zweiten Wiederholung-Operation, welche mit dem zweiten Thread und dem zweiten Satz von Threads assoziiert ist, in die Mehr-Stufen-Pipeline, um die Anweisung auszuführen,
- 1.7 wobei die zweite Wiederholung-Operation in die Mehr-Stufen-Pipeline seriell relativ zu der ersten Wiederholung-Operation eingefügt ist,
- 1.8^{H1}** wobei die erste und die zweite Wiederholung-Operation in der Mehr-Stufen-Pipeline simultan voranschreiten.“

Wegen des Wortlauts des nach Hilfsantrag 1 geltenden nebengeordneten Patentanspruchs 2 und der Unteransprüche 3 bis 8 wird auf die Akte verwiesen.

Der seitens des Senats mit einer Gliederung versehene **Patentanspruch 1 nach Hilfsantrag 2** lautet unter Hervorhebung der Unterschiede zur ursprünglich eingereichten deutschsprachigen Anspruchsfassung:

- 1.1^{H1} „Computer-implementiertes Verfahren zum Wiederholen einer ~~gemeinsame-Ressource-Zugriffs-Operation~~ divergenten Zugriffsoperation auf gemeinsam genutzte Ressourcen, wobei das Verfahren aufweist:
- 1.2 Auswählen eines ersten Threads und eines zweiten Threads von einer Gruppe von Threads, welche konfiguriert sind, eine Anweisung in einer Mehr-Stufen-Pipeline simultan auszuführen,
- 1.2a^{H1} wobei der erste Thread auf eine erste gemeinsam genutzte Ressource zugreift,
- 1.2b^{H1} wobei der zweite Thread auf eine zweite gemeinsam genutzte Ressource zugreift,
- 1.2c^{H1} wobei auf die erste und die zweite gemeinsam genutzte Ressource nicht in derselben Zugriffsoperation zugegriffen werden kann,
- 1.2d^{H1} ~~wobei weder wodurch der erste Thread noch und der zweite Thread die Anweisung bereits ausgeführt hat~~ nicht bedient sind;
- 1.3 Auswählen, aus der Gruppe von Threads, eines ersten Satzes von Threads, welche mit dem ersten Thread zu assoziieren sind,
- 1.3a^{H1} wobei jeder Thread in dem ersten Satz von Threads konfiguriert ist, auf die erste gemeinsam genutzte Ressource zuzugreifen;
- 1.4 Auswählen, aus der Gruppe von Threads, eines zweiten Satzes von Threads, welche mit dem zweiten Thread zu assoziieren sind,
- 1.4a^{H1} wobei jeder Thread in dem zweiten Satz von Threads konfiguriert ist, auf die zweite gemeinsam genutzte Ressource zuzugreifen;

- 1.5 Einfügen einer ersten Wiederholung-Operation, welche mit dem ersten Thread und dem ersten Satz von Threads assoziiert ist, in die Mehr-Stufen-Pipeline, um die Anweisung auszuführen; und
- 1.6 Einfügen einer zweiten Wiederholung-Operation, welche mit dem zweiten Thread und dem zweiten Satz von Threads assoziiert ist, in die Mehr-Stufen-Pipeline, um die Anweisung auszuführen,
- 1.7 wobei die zweite Wiederholung-Operation in die Mehr-Stufen-Pipeline seriell relativ zu der ersten Wiederholung-Operation eingefügt ist,
- 1.8^{H2} und um einen Taktzyklus versetzt die Mehr-Stufen-Pipeline durchläuft, wobei maximal zwei Wiederholungs-Operationen aufeinanderfolgend durchgeführt werden unabhängig von der Anzahl der erforderlichen Wiederholung-Operationen.“

Der seitens des Senats mit einer Gliederung versehene **Patentanspruch 2 nach Hilfsantrag 2** lautet unter Hervorhebung der Unterschiede zu Patentanspruch 3 in der ursprünglich eingereichten deutschsprachigen Anspruchsfassung:

- 2.0^{H1} „Subsystem zum Wiederholen einer divergenten Zugriffsoperation auf gemeinsam genutzte Ressourcen ~~gemeinsame-Ressource-Zugriffs-Operation~~, aufweisend:
 - 2.1 eine Lade-Speicher-Einheit (LSU), welche konfiguriert ist, um:
 - 2.2 einen ersten Thread und einen zweiten Thread von einer Gruppe von Threads auszuwählen, welche konfiguriert sind, eine Anweisung in einer mehrstufigen Pipeline simultan auszuführen,
 - 2.2a^{H1} wobei der erste Thread auf eine erste gemeinsam genutzte Ressource zugreift,

- 2.2b^{H1}** wobei der zweite Thread auf eine zweite gemeinsam genutzte Ressource zugreift.
- 2.2c^{H1}** wobei auf die erste und die zweite gemeinsam genutzte Ressource nicht in derselben Zugriffsoperation zugegriffen werden kann.
- 2.2d^{H1}** ~~wobei weder wodurch der erste Thread noch und der zweite Thread die Anweisung bereits ausgeführt hat~~ nicht bedient sind;
- 2.3** einen ersten Satz von Threads aus der Gruppe von Threads auszuwählen, welche mit dem ersten Thread zu assoziieren sind ~~um mit dem ersten Thread assoziiert zu werden,~~
- 2.3a** wobei der erste Thread und jeder Thread in dem ersten Satz von Threads konfiguriert sind, auf die erste gemeinsam genutzte Ressource zuzugreifen;
- 2.4** einen zweiten Satz von Threads aus der Gruppe von Threads auszuwählen, welche mit dem zweiten Thread zu assoziieren sind ~~um mit dem zweiten Thread assoziiert zu werden,~~
- 2.4a** wobei der zweite Thread und jeder Thread in dem zweiten Satz von Threads konfiguriert sind, auf die zweite gemeinsam genutzte Ressource zuzugreifen;
- 2.5** eine erste Wiederholung-Operation, welche mit dem ersten Thread und dem ersten Satz von Threads assoziiert ist, in die Mehr-Stufen-Pipeline einzufügen, um die Anweisung auszuführen; und
- 2.6** eine zweite Wiederholung-Operation, welche mit dem zweiten Thread und dem zweiten Satz von Threads assoziiert ist, in die Mehr-Stufen-Pipeline einzufügen, um die Anweisung auszuführen,
- 2.7** wobei die zweite Wiederholung-Operation in die Mehr-Stufen-Pipeline seriell relativ zu der ersten Wiederholung-Operation eingefügt ist,

2.8^{H2} und um einen Taktzyklus versetzt die Mehr-Stufen-Pipeline durchläuft, wobei maximal zwei Wiederholungs-Operationen aufeinanderfolgend durchgeführt werden unabhängig von der Anzahl der erforderlichen Wiederholung-Operationen.“

Wegen des Wortlauts der nach Hilfsantrag 2 geltenden Unteransprüche 3 bis 8 wird auf die Akte verwiesen.

Die Beschwerdeführerin macht geltend, dass die Anspruchsfassungen gemäß Hauptantrag und den Hilfsanträgen 1 und 2 jeweils zulässig und die Gegenstände der jeweiligen Patentansprüche im Lichte des Standes der Technik neu seien und auf einer erfinderischen Tätigkeit beruhen.

Wegen der weiteren Einzelheiten wird auf den Akteninhalt verwiesen.

II.

Die zulässige Beschwerde hat insoweit Erfolg, als sie zur Aufhebung des angefochtenen Beschlusses und zur Patenterteilung mit geänderten Unterlagen gemäß Hilfsantrag 2 führt. Im Übrigen war die Beschwerde bezüglich des Hauptantrags und des Hilfsantrags 1 zurückzuweisen. Denn der Gegenstand des Patentanspruchs 1 nach Hauptantrag und nach Hilfsantrag 1 beruht jeweils nicht auf einer erfinderischen Tätigkeit. Die Fragen der Zulässigkeit der geltenden Ansprüche gemäß Hauptantrag und Hilfsantrag 1 sowie der Neuheit dieser Anspruchsgegenstände können somit dahinstehen (vgl. BGH, Urteil vom 18. September 1990 – X ZR 29/89, GRUR 1991, 120, 121 li. Sp. Abs. 3 – Elastische Bandage).

1. Die Patentanmeldung betrifft im Allgemeinen Computer-Architekturen und insbesondere aufeinander folgende (*batched*) Wiederholungen (*replays*) von divergenten Operationen (vgl. geltende Beschreibung, Abs. 0001).

Die Anmeldung geht davon aus, dass es eine gewöhnliche Praxis in Parallel-Verarbeitungs-Systemen sei, einen Prozessor zu entwerfen, welcher eine gewisse Anzahl von Threads simultan ausführe. Jeder Thread könne innerhalb einer separaten Ausführungs-Pipeline innerhalb des Prozessors ausgeführt werden. Wenn alle solche Threads dieselbe Anweisungs-Sequenz (typischerweise mit verschiedenen Daten für jeden Thread) ausführten, sei es von Vorteil, die Steuerstrukturen der Threads gemeinsam zu benutzen bzw. zu teilen (*sharing*). Zum Beispiel müsse nur eine Anweisung geholt werden und alle Threads führten dann diese selbe Anweisung aus. Dieser Typ von Operation finde sich auf Einzel-Anweisung-Mehrprozess-gestützte-Prozessoren (*Single Instruction, Multiple Threads* bzw. *SIMT*) und Einzel-Anweisung-Mehrdaten-Prozessoren (*Single Instruction, Multiple Data* bzw. *SIMD*).

Wenn eine Ausführung parallel fortschreite, könnten verschiedene Threads auf eine gemeinsame Ressource zugreifen, wie etwa auf einen gemeinsamen Speicher. Dies könne für die Threads zu einem Ressourcen-Konflikt führen. Zum Beispiel könnten die Threads eine Speicherzugriffs-Operation ausführen, wie etwa eine Lade-Anweisung, bei der sich der Satz von Speicherstellen über zwei oder mehr Zwischenspeicher-Zeilen erstrecke (*spans*). Solch eine Lade-Anweisung könne eine „divergente“ Operation genannt werden, wenn die Speicherstellen, welche für die verschiedenen Threads erforderlich sind, divergente Zwischenspeicher-Zeilen seien. In solchen Situationen transferiere die Pipeline eine der Zwischenspeicher-Zeilen, auf welche einige der Threads zugriffen, und diese Threads seien in der Lage, die Speicherzugriffs-Operationen zu vollenden. Die anderen Threads, welche auf Stellen innerhalb einer anderen Zwischenspeicher-Zeile wiesen, seien nicht in der Lage, die Speicherzugriffs-Operationen zu vollenden und blieben unbedient. Somit seien einige Threads in der Lage, die Speicherzugriffs-Operationen mit einem einzelnen Durchgang (*pass*) durch die Pipeline zu vollenden, während andere Threads nicht dazu in der Lage seien. Ohne ein Mittel, mehrere Ausführungszyklen zu verarbeiten, sei die Operation nicht in der Lage, erfolgreich beendet zu werden.

Eine Vorgehensweise, mehrere Ausführungszyklen zu implementieren, sei es, die Anweisung in eine vorherige Stufe der Verarbeitungs-Pipeline erneut einzusetzen, und die Lade-Anweisung erneut für die Threads auszuführen, welche nicht auf die Daten aus ihren Ziel-Speicheradress-Stellen zugreifen konnten. Solch eine Technik werde als „Wiederholung-Operation“ (*replay operation*) bezeichnet. Ein Nachteil dieser Vorgehensweise sei, dass die Pipeline angehalten werde (*stalled*), bis alle Wiederholung-Operationen abgeschlossen seien, was die Performanz des Gesamt-Systems vermindere. Ein zusätzlicher Nachteil sei, dass das Parallel-Verarbeitungs-System nicht in der Lage sein könne, alle Pipeline-Stufen innerhalb einer Pipeline-Stufe-Verzögerung anzuhalten. Dadurch könnten eine oder mehrere neue Anweisungen, welche in die Pipeline einträten, inkorrekt verworfen werden oder die Wiederholung-Operation werde ähnlich verworfen. In jedem Fall werde die neue Anweisung oder die Wiederholung-Operation nicht korrekt abgeschlossen.

Eine andere Vorgehensweise, um mehrere Ausführungszyklen zu implementieren, sei es, die Anweisung weiter zurück in der Pipeline erneut einzusetzen. Mit dieser Vorgehensweise würden die Anweisungen, welche wiederholt werden, in der Pipeline zusammen mit neuen Anweisungen verschachtelt (*interleaved*), wodurch die Häufigkeit von Pipeline-Unterbrechungen (*pipeline stalls*) vermindert und die Pipeline-Performanz erhöht werde. Ein Nachteil dieser Vorgehensweise sei jedoch eine erhöhte Latenz der Anweisungen, welche wiederholt würden (vgl. geltende Beschreibung, Abs. 0002-0005).

In der Patentanmeldung ist als Aufgabe genannt, einen effizienteren Weg anzugeben, um Wiederholung-Operationen für divergente Operationen auszuführen (vgl. geltende Beschreibung, Abs. 0006).

Der zuständige Fachmann weist ein abgeschlossenes Hochschulstudium der Fachrichtung Elektrotechnik oder Informationstechnik auf und verfügt über eine mehrjährige Berufserfahrung im Entwurf und der Programmierung von SIMD- bzw. SIMT-Prozessoren.

Die Aufgabe soll gemäß Hauptantrag und den Hilfsanträgen 1 und 2 durch ein Computer-implementiertes Verfahren nach dem jeweiligen Patentanspruch 1 und ein Subsystem nach dem jeweiligen Patentanspruch 2 gelöst werden.

2. Der Fachmann legt dem Anspruchsgegenstand der jeweiligen unabhängigen Patentansprüche 1 nach Hauptantrag und den Hilfsanträgen 1 und 2 das folgende Verständnis zugrunde:

Das Verfahren nach **Patentanspruch 1** gemäß Hauptantrag ist auf ein computer-implementiertes Verfahren zum Wiederholen einer Zugriffsoperation auf gemeinsam genutzte Ressourcen gerichtet (vgl. **Merkmal 1.1**), wobei der Patentanspruch die Art der Zugriffsoperation und der Ressourcen offenlässt. Im jeweiligen Patentanspruch 1 nach den Hilfsanträgen 1 und 2 ist präzisiert, dass es sich um eine divergente Zugriffsoperation handelt (vgl. **Merkmal 1.1^{H1}**). Als Hintergrund hierzu beschreibt die Anmeldung die parallele Ausführung von Threads, d. h. die parallele Ausführung einer Serie von Anweisungen in separaten Ausführungs-Pipelines innerhalb eines Prozessors. Dabei können Steuerstrukturen für Threads, welche dieselbe Anweisungs-Sequenz typischerweise mit verschiedenen Daten für jeden Thread parallel ausführen, gemeinsam benutzt werden. Somit muss bspw. nur eine Anweisung geholt werden, welche alle Threads dann ausführen (vgl. geltende Beschreibung, Abs. 0040 i. V. m. Abs. 0002). Als Beispiel für eine divergente Zugriffsoperation nennt die Anmeldung eine Lade-Anweisung, bei der die Speicherstellen, deren Inhalte von den verschiedenen Threads jeweils beim Ausführen dieser Speicherzugriffs-Operationen angefordert werden, auf verschiedenen und somit „divergenten“ Zwischenspeicher-Zeilen liegen (vgl. geltende Beschreibung, Abs. 0003).

Das beanspruchte Verfahren sieht vor, einen ersten und einen zweiten Thread einer Gruppe von Threads auszuwählen, die zur Ausführung einer Anweisung in einer Mehr-Stufen-Pipeline konfiguriert sind, ohne dass die Threads bedient wurden. Dies bedeutet, dass die Ausführung der einen Anweisung für die genannten Threads nicht erfolgreich war (**Merkmale 1.2, 1.2a**). Die Ursache hierfür lässt

Patentanspruch 1 gemäß Hauptantrag offen, wobei der Beschreibung zu entnehmen ist, dass dies aufgrund einer divergenten Operation erfolgen kann (vgl. geltende Beschreibung, Abs. 0060 ff). Eine solche divergente Operation ist in Patentanspruch 1 der Hilfsanträge 1 und 2 näher erläutert. Demnach greift der erste Thread auf eine erste und der zweite Thread auf eine zweite gemeinsam genutzte Ressource zu, wobei auf die beiden Ressourcen nicht in derselben Zugriffsoption zugegriffen werden kann, was dazu führt, dass die Threads nicht bedient werden (vgl. **Merkmale 1.2a^{H1} bis 1.2d^{H1}**).

Zu dem ersten und zweiten Thread wird dann ein erster bzw. zweiter Satz von Threads ausgewählt, die jeweils mit dem ersten bzw. zweiten Thread assoziiert sind. Der erste und zweite Satz von Threads ist Bestandteil jeweils einer Gruppe von Threads, die konfiguriert sind, um auf dieselbe gemeinsam genutzte erste bzw. zweite Ressource zuzugreifen wie der jeweils erste bzw. zweite Thread (**Merkmale 1.3 und 1.3a bzw. 1.3a^{H1}, 1.4 und 1.4a bzw. 1.4a^{H1}**). Die Beschreibung spricht in diesem Zusammenhang auch von „provokanten“ Threads (unbediente Threads) und einer jeweils assoziierten „Thread-Familie“ (vgl. Abs. 0060). Eine erste bzw. zweite „Wiederholung-Operation“ bezeichnet die zugehörige Anweisung zu dem jeweils ausgewählten ersten bzw. zweiten Thread. In die Mehr-Stufen-Pipeline werden (durch eine Lade-Speicher-Einheit / LSU) eine erste und eine zweite Wiederholung-Operation zum Ausführen der zu wiederholenden Anweisung eingefügt, welche mit dem ersten Thread und dem ersten Satz von Threads bzw. mit dem zweiten Thread und dem zweiten Satz von Threads assoziiert ist (**Merkmale 1.5, 1.6**).

Dabei wird die zweite Wiederholung-Operation seriell relativ zur ersten Wiederholung-Operation in die Mehr-Stufen-Pipeline eingefügt, was bedeutet, dass die beiden Operationen nacheinander in die Pipeline eingefügt werden (**Merkmal 1.7**).

In Patentanspruch 1 nach Hilfsantrag 1 ist dazu ergänzt, dass die beiden nacheinander eingefügten Wiederholung-Operationen in der Mehr-Stufen-Pipeline simultan voranschreiten (**Merkmal 1.8^{H1}**). Patentanspruch 1 nach Hilfsantrag 2

enthält an Stelle dieses Merkmals die Präzisierung, dass die beiden nacheinander eingefügten Wiederholung-Operationen die Mehr-Stufen-Pipeline um einen Taktzyklus versetzt durchlaufen, sowie die Einschränkung, dass unabhängig von der Anzahl der erforderlichen Wiederholung-Operationen maximal zwei Wiederholungs-Operationen aufeinanderfolgend durchgeführt werden (vgl. **Merkmal 1.8^{H2}**). Dies bedeutet, dass aufeinanderfolgend in die Mehr-Stufen-Pipeline allenfalls zwei Wiederholung-Operationen für die zu wiederholenden Gruppen von nicht bedienten Threads eingefügt werden, auch wenn weitere nicht bediente Threads vorliegen, die nicht den beiden Gruppen angehören.

Das Subsystem nach **Patentanspruch 2** gemäß Hauptantrag und den Hilfsanträgen 1 und 2 weist eine Lade-Speicher-Einheit (LSU) auf, die zur Ausführung von Verfahrensschritten konfiguriert ist, die den Merkmalen des jeweiligen Verfahrensanspruchs 1 entsprechen. Die Anordnung der LSU im System ist in Figur 3C dargestellt, ihre Funktion ist beispielgebend im Rahmen des Verfahrens zu Figur 4 beschrieben (vgl. geltende Beschreibung, Abs. 0058 ff)

3. Der Gegenstand des **Patentanspruchs 1 nach Hauptantrag** beruht für den Fachmann ausgehend von Druckschrift D3 und seinem Fachwissen nicht auf einer erfinderischen Tätigkeit (§ 4 PatG).

Aus **Druckschrift D3** ist ein Computer-implementiertes Verfahren zum Wiederholen von Speicherzugriffs-Operationen bekannt, wobei der Fachmann mitliest, dass nicht erfolgreiche Anweisungen, die zu Wiederholungen führen (*instruction replay*; vgl. S. 50, 51), auf divergente Operationen, also den Zugriff auf gemeinsam genutzte Ressourcen, zurückzuführen sind (vgl. *memory divergence*, S. 51 und *Multiple threads, one shared resource*, S. 63 i. V. m. S. 47 / **Merkmal 1.1**). Druckschrift D3 ist das Wiederholen von Zugriffsoperationen für Threads (*replay*) zu entnehmen, welche eine bestimmte Anweisung noch nicht erfolgreich ausgeführt haben, d.h. deren Zugriffsanweisung jeweils nicht bedient wurde (*Threads that miss...*; vgl. S. 50, 51). Dies ist im Kontext des parallelen Ausführens einer

Anweisung in einer Mehr-Stufen-Pipeline in mehreren Threads zu sehen (*SIMT*, vgl. S. 15, i. V. m. *memory divergence*, vgl. S. 51), aus dem sich implizit auch ein simultanes Ausführen der einen Anweisung ergibt. Eine Betrachtung von Wiederholungs-Operationen für einen ersten und einen zweiten Thread ist den allgemeinen Ausführungen von Druckschrift D3 zur Wiederholung von Zugriffsoptionen nicht explizit zu entnehmen (**Merkmal 1.2a, teilweise Merkmal 1.2**).

Die Wiederholung-Operation, die in Druckschrift D3 als eine zu wiederholende Speicherzugriffs-Operation zu verstehen ist (vgl. S. 50), wird der Mehr-Stufen-Pipeline jeweils wieder zugeführt, also in die Mehr-Stufen-Pipeline eingefügt, um die Anweisung auszuführen (*instruction replay*). Diese Anweisung ist zumindest mit einem Thread assoziiert, nämlich dem Thread, der nicht bedient wurde (*Threads that miss...*) (vgl. S. 50, S. 51 / **Merkmal 1.5**). Das Einfügen ist dabei nicht auf eine einzige Wiederholung-Operation oder zwei explizit genannte Threads beschränkt, sondern ist allgemein für jeden nicht bedienten Thread vorgesehen (**Merkmal 1.6**).

Ein serielles Einfügen mehrerer Wiederholungs-Operationen folgt implizit aus dem auf Seite 50 dargestellten logischen Aufbau, da dieser nur eine einzige „Replay-Queue“ für die zu wiederholenden Operationen (*instruction replay*) zeigt und keinen Hinweis auf separate Replay-Queues für einzelne Threads gibt (**Merkmal 1.7**).

Wie die Anmelderin zutreffend ausführt, ist das Bilden von Gruppen von Threads, welche mit den zwei Threads nach den Merkmalen 1.3 und 1.4 assoziiert sind, und denen jeweils der Zugriff auf eine Ressource gemein ist (vgl. Merkmale 1.3a, 1.4a), in Druckschrift D3 nicht genannt. Auf Seite 63 findet sich in Druckschrift D3 allerdings der Hinweis darauf, die Ressourcen durch Threads opportunistisch gemeinsam zu nutzen (*threads opportunistically share the resource*), sowie ein allgemeiner Hinweis auf Gruppierungs-Strategien (*grouping strategies*) bei der Verarbeitung der Anweisungen. Ausgehend von der zugrundeliegenden Single Instruction, Multiple Threads-Architektur (SIMT), welche die

parallele Verarbeitung einer Anweisung in mehreren Threads beschreibt, versteht dies der Fachmann als Anregung, die Verarbeitung auch im Hinblick auf Ressourcenzugriffe, also bspw. Speicherzugriffsoperationen, zu optimieren und Gruppen von zusammengehörigen bzw. assoziierten Threads zu bilden, die eine gemeinsame Ressource adressieren, bspw. die gleiche Cache-Zeile. Für den Fachmann, der gemäß der zugrunde liegenden Aufgabe divergente Operationen betrachtet, bietet sich dabei einerseits an, gemeinsam genutzte Ressourcen bereits jeweils vor jedem Ressourcenzugriff zu identifizieren, was jedoch nachteilig einen Eingriff in den Verarbeitungsablauf und eine Verzögerung aller Threads bedeutet, oder andererseits, bei nicht bedienten Threads, die aufgrund divergenter Operationen zu erwarten sind, diese nicht erfolgreich und somit in ihrer Ausführung auf jeden Fall verzögerten Threads zur Wiederholung hinsichtlich gemeinsamer Ressourcenzugriffe zusammenzufassen. Dem Fachmann steht damit eine begrenzte Anzahl von Maßnahmen zur Verfügung, aus denen er aufgrund seines Fachwissens die im Hinblick auf den Verfahrensablauf vorteilhaftere Variante der Zusammenfassung nicht bedienter Threads auswählt. Daher gelangt der Fachmann ausgehend von den Anregungen der Druckschrift D3 und seinen Fachkenntnissen naheliegend zum Bilden von Gruppen von Threads, die mit einem Thread assoziiert sind, da sie jeweils konfiguriert sind, auf eine gemeinsam genutzte Ressource (bspw. gemeinsam auf eine Cache-Zeile) zuzugreifen (**Merkmale 1.3a, 1.4a**).

Druckschrift D3 unterscheidet sich vom Gegenstand des Patentanspruchs 1 nach Hauptantrag außerdem darin, dass eine Wiederholungsstrategie nicht explizit für jeweils zwei Threads nach Merkmal 1.2 vorgesehen ist, also für einen ersten und einen zweiten Thread. Wie vorstehend zu den Merkmalen 1.3a und 1.4a erläutert, besteht ausgehend von Druckschrift D3 für den Fachmann allgemein die Veranlassung, assoziierte Threads im Hinblick auf Wiederholungsoperationen zu identifizieren. Da der Fachmann bei divergenten Operationen davon ausgeht, dass Threads wiederholt nicht bedient werden (vgl. auch S. 50), ist es selbstverständlich, ein Gruppieren von assoziierten Threads nicht als

einmalige Maßnahme vorzusehen, sondern immer dann, wenn nicht bediente Threads auftreten. Der vorliegende Patentanspruch nennt zwar nur zwei zu bildende Gruppen assoziierter Threads. Das Verfahren nach Patentanspruch 1 des Hauptantrags ist aber dadurch weder im Hinblick auf eine wiederholte Ausführung des Verfahrens (also die wiederholte Bildung von Thread-Gruppen) noch im Hinblick auf die Anzahl der zu wiederholenden Operationen beschränkt. Der beanspruchte Fall der Bildung zweier Gruppen von Threads, die konfiguriert sind, jeweils auf eine gemeinsam genutzte Ressource zuzugreifen, ist daher von der Bildung von Thread-Gruppen mit umfasst, die sich für den Fachmann naheliegend aus den Anregungen der Druckschrift D3 ergibt und kann daher eine erfinderische Tätigkeit nicht begründen (**Merkmale 1.2, 1.3, 1.4**).

Den Ausführungen der Anmelderin, dass Druckschrift D3, wie bspw. auf Seite 50 dargestellt, nur den auch in der vorliegenden Anmeldung beschriebenen Stand der Technik mit einer Wiederholung jedes nicht erfolgreichen bzw. nicht bedienten Threads beschreibe, kann aufgrund der weiteren enthaltenen Hinweise an den Fachmann nicht gefolgt werden. Der Gegenstand des Patentanspruchs 1 nach Hauptantrag ergibt sich für den Fachmann vielmehr naheliegend aus Druckschrift D3 und seinem einschlägigen Fachwissen.

Patentanspruch 1 gemäß Hauptantrag beruht daher nicht auf einer erfinderischen Tätigkeit und ist daher nicht patentfähig.

Mit dem nicht patentfähigen Patentanspruch 1 gemäß Hauptantrag sind auch die weiteren Patentansprüche 2 bis 8 des Hauptantrags nicht schutzfähig, da auf diese Ansprüche kein eigenständiges Schutzbegehren gerichtet war (vgl. BGH, Beschluss vom 27. Juni 2007 - X ZB 6/05; GRUR 2007, 862 Abschnitt III 3. a) aa) - Informationsübermittlungsverfahren II).

4. Der Gegenstand des **Patentanspruchs 1 nach Hilfsantrag 1** beruht für den Fachmann ausgehend von Druckschrift D3 und seinem Fachwissen ebenfalls nicht auf einer erfinderischen Tätigkeit (§ 4 PatG).

Der Patentanspruch 1 gemäß Hilfsantrag 1 unterscheidet sich von Patentanspruch 1 des Hauptantrags unter anderem darin, dass die Zugriffsoperation auf gemeinsam genutzte Ressourcen als divergente Zugriffsoperation präzisiert ist (vgl. Merkmal 1.1^{H1}). Die Merkmale 1.2a^{H1} bis 1.2d^{H1} beschreiben eine solche divergente Zugriffsoperation, bei welcher der erste Thread auf eine erste gemeinsam genutzte Ressource zugreift, der zweite Thread auf eine zweite gemeinsam genutzte Ressource zugreift, wobei auf die erste und die zweite gemeinsam genutzte Ressource nicht in derselben Zugriffsoperation zugegriffen werden kann, wodurch der erste Thread und der zweite Thread nicht bedient sind.

Von einer solchen divergenten Zugriffsoperation wurde bereits in den vorstehenden Ausführungen zu Patentanspruch 1 gemäß Hauptantrag ausgegangen (vgl. Abschnitt II. 3.). Die Merkmale 1.2a^{H1} bis 1.2d^{H1} geben dabei nichts Anderes als das fachmännische Verständnis einer divergenten Operation am Beispiel des Zugriffs auf zwei Ressourcen wieder, die im Kontext einer Single Instruction, Multiple Thread-Architektur (SMT) mit parallel ausgeführten Anweisungen auftritt (vgl. Druckschrift D3, S. 15 i. V. m. *memory divergence*, vgl. S. 51), wenn beispielsweise bei der parallelen Ausführung einer Speicherzugriffsoperation (*single instruction*) auf verschiedene erste und zweite Speicher- bzw. Cache-Zeilen als erste und zweite Ressourcen zugegriffen wird (**Merkmal 1.1^{H1}, Merkmale 1.2a^{H1} bis 1.2d^{H1}**).

Der Patentanspruch 1 gemäß Hilfsantrag 1 unterscheidet sich vom Gegenstand des Patentanspruchs 1 nach Hauptantrag außerdem darin, dass die erste und die zweite Wiederholung-Operation in der Mehr-Stufen-Pipeline simultan voranschreiten (vgl. Merkmal 1.8^{H1}). Ein solches simultanes Fortschreiten der Opera-

tionen in einer Mehr-Stufen-Pipeline ergibt sich zwangsläufig aus dem fachmännischen Verständnis der in Druckschrift D3 auf Seite 50 dargestellten Systemarchitektur (*instruction queue*), in welche die Wiederholung-Operationen nacheinander eingefügt werden und in der alle Operationen getaktet fortschreiten (vgl. Darstellung auf S. 50). Hinweise auf eine abweichende, bspw. eine verzögerte oder beschleunigte Abarbeitung einzelner Operationen in der dargestellten Pipeline sind in Druckschrift D3 nicht ersichtlich (**Merkmal 1.8^{H1}**).

Die **Merkmale 1.3a^{H1} und 1.4a^{H1}** sind gegenüber dem Patentanspruch 1 des Hauptantrags sprachlich an die vorausgehenden geänderten Merkmale 1.2a^{H1} bis 1.2d^{H1} angepasst und enthalten keine inhaltlichen Änderungen. Für diese Merkmale, sowie die weiteren, gegenüber Patentanspruch 1 nach Hauptantrag unveränderten Merkmale des Patentanspruchs 1 nach Hilfsantrag 1 gelten die Ausführungen zum Hauptantrag in gleicher Weise (vgl. Abschnitt II. 3.).

Der Gegenstand des Patentanspruchs 1 gemäß Hilfsantrag 1 beruht daher ebenfalls nicht auf einer erfinderischen Tätigkeit und ist nicht patentfähig.

Mit dem nicht patentfähigen Patentanspruch 1 gemäß Hilfsantrag 1 sind auch die weiteren Ansprüche des Hilfsantrags 1 nicht schutzfähig, da auf diese Ansprüche kein eigenständiges Schutzbegehren gerichtet war (vgl. BGH, a.a.O. - Informationsübermittlungsverfahren II).

5. Der Hilfsantrag 2 erfüllt die Voraussetzungen für eine Patenterteilung.

Der zweifelsfrei gewerblich anwendbare Gegenstand des **Patentanspruchs 1 gemäß Hilfsantrag 2** ist gegenüber dem im Verfahren befindlichen Stand der Technik neu und beruht auf einer erfinderischen Tätigkeit. Auch die weiteren Voraussetzungen zur Patenterteilung sind erfüllt (§§ 1 bis 5, § 34 und § 38 PatG).

a) Die Patentansprüche 1 bis 8 nach Hilfsantrag 2 sowie die Beschreibungsunterlagen mitsamt Figuren sind zulässig (§ 38 PatG).

Die Änderungen in den Patentansprüchen 1 bis 8 gegenüber der ursprünglich in englischer Sprache eingereichten Fassung liegen im Rahmen der ursprünglichen Offenbarung. Beschreibung und Zeichnungen sind gegenüber der ursprünglich eingereichten deutschen Übersetzung unverändert.

Der unabhängige Patentanspruch 1 basiert auf der deutschen Übersetzung der ursprünglich in englischer Sprache eingereichten Ansprüche 1 und 2 (vgl. Merkmale 1.1^{H1}, 1.2, 1.3, 1.4, 1.5 bis 1.7) und der Beschreibung zu Fig. 4 in den Absätzen 0058 bis 0069 (vgl. Merkmale 1.3a^{H1}, 1.4a^{H1}), wobei die Bezeichnung der betrachteten Threads bzw. Gruppen von Threads gemäß den ursprünglichen Ansprüchen beibehalten wurde. Die Beschreibung einer „divergenten Operation“ mit daraus resultierenden, nicht bedienten Threads wurde basierend auf den Absätzen 0068 bis 0073 in Verbindung mit Absatz 0060 der ursprünglich eingereichten Beschreibung präzisiert (vgl. Merkmale 1.1^{H1}, 1.2a^{H1} bis 1.2d^{H1}). Die Beschränkung auf maximal zwei Wiederhol-Operationen basiert auf Absätzen 0064 und 0065 der ursprünglich eingereichten Beschreibung (vgl. Merkmal 1.8^{H2}).

Patentanspruch 2 basiert auf den ursprünglich eingereichten Patentansprüchen 3 und 8 und wurde entsprechend den Änderungen des Patentanspruchs 1 gemäß Hilfsantrag 2 gegenüber der ursprünglichen Anspruchsfassung angepasst.

Die Unteransprüche 3 bis 8 entsprechen der deutschen Übersetzung der ursprünglich in englischer Sprache eingereichten Ansprüche 4 bis 7, 9 und 10 unter Anpassung der Nummerierung und Rückbezüge.

b) Die unabhängigen Patentansprüche 1 und 2 gemäß Hilfsantrag 2 sind neu gegenüber dem im Verfahren befindlichen Stand der Technik (§ 3 PatG).

Keine der im Verfahren befindlichen Druckschriften D1 bis D4 sieht eine systematische Begrenzung der Wiederholung-Operationen für nicht bediente Threads vor, insbesondere keine aufeinanderfolgende Durchführung von maximal zwei Wiederholung-Operationen, unabhängig von der Anzahl der erforderlichen Wiederholung-Operationen, also unabhängig davon, wie viele nicht bediente Gruppen von Threads vorliegen, die jeweils auf eine gemeinsame Ressource zugreifen (vgl. Merkmal 1.8^{H2} bzw. Merkmal 2.8^{H2}).

Druckschrift D1 betrifft Lade-Anweisungen für Speicherzugriffe, die über Cache-Zeilen hinausgehen (vgl. Abstract und Fig. 4c). Hierzu wird insbesondere der Ablauf bei zwei Zugriffen auf zwei Cache-Zeilen in Bezug auf ein einziges Vektor-Register für das Speichern der ausgelesenen Daten betrachtet (vgl. Abs. 0006-0008). Wenn einer der Zugriffe auf die beiden Cache-Zeilen kein Cache-Treffer ist, wird die ganze Anweisung wiederholt. Dabei wird die (zeilenübergreifende) Lade-Anweisung in zwei Lade-Anweisungen aufgeteilt und ein temporäres Register für die Ergebnisse der beiden Zugriffe verwendet (vgl. Fig. 6 und 7; Abs. 0077 ff). Damit steht die Vorgehensweise nach Druckschrift D1 in keinem Zusammenhang zur Gruppierung und Wiederholung parallel ausgeführter Threads bei divergenten Cache-Zugriffen für eine in parallelen Threads ausgeführte Ladeanweisung. Strategien für die Wiederholung der Anweisungen, insbesondere eine Beschränkung auf zwei Wiederholung-Operationen, sind Druckschrift D1 nicht zu entnehmen.

Der Gegenstand der Patentansprüche 1 und 2 nach Hilfsantrag 2 ist daher neu gegenüber Druckschrift D1.

Druckschrift D2 befasst sich unter anderem mit der Realisierung einer Wiederholung von Speicherzugriffen bei einem „Cache-Miss“ durch mehrere Warteschlangen (*replay queues*, vgl. zw. Seite, Abschnitt 3.1). Unabhängig davon, dass divergente Speicheroperationen im Sinne der vorliegenden Anmeldung

nicht angesprochen werden, sind Strategien, wie hierbei Threads zusammengefasst oder die Wiederholungen in die Verarbeitungspipeline eingefügt werden, der Druckschrift nicht zu entnehmen.

Der Gegenstand der Patentansprüche 1 und 2 nach Hilfsantrag 2 ist daher neu gegenüber Druckschrift D2.

Druckschrift D3, welche dem Anspruchsgegenstand am nächsten kommt (vgl. Ausführungen zum Hauptantrag, Abschnitt II. 3.), sieht nur allgemein eine Wiederholung von Operationen aufgrund nicht erfolgreich ausgeführter Speicherzugriffe vor (vgl. S. 50). Das Bilden von zwei Gruppen von Threads für assoziierte Threads, die jeweils auf eine gemeinsam genutzte Ressource zugreifen, ist in Druckschrift D3 nicht explizit genannt. Damit ist Druckschrift D3 auch keine Begrenzung der aufeinanderfolgenden Durchführung auf maximal zwei Wiederholungs-Operationen zu entnehmen, unabhängig von der Anzahl der erforderlichen Wiederholung-Operationen.

Damit ist der Gegenstand der Patentansprüche 1 und 2 nach Hilfsantrag 2 neu gegenüber Druckschrift D3.

Druckschrift D4 sieht abweichend vom vorliegenden Anspruchsgegenstand für jeden Thread eigene Wiederholungs-Warteschlangen vor (*thread-specific replay queue* bzw. *processed independently*, vgl. Abstract, Abs. 0021, 0022, 0090, und Patentansprüche 11, 15), außerdem werden keine assoziierten Threads (welche die gleiche Operation ausführen) betrachtet, sondern abhängige Operationen (*dependent instructions*, bspw. arithmetische Operationen, die vom Laden der zugehörigen Daten abhängen). Zudem wird im Unterschied zur vorliegenden Anmeldung keine SIMT-Architektur betrachtet, da beispielsweise mehrere Anweisungen in einem Taktschritt ausgegeben werden (vgl. Abs. 0095).

Ein serielles Einfügen von Wiederholung-Operationen kann in Druckschrift D4 allenfalls für eine gemeinsame Wiederholungs-Warteschlange eines Threads mitgelesen werden. Hieraus folgt jedoch nicht zwangsläufig ein aufeinanderfolgendes Einfügen zweier Wiederholung-Operationen und insbesondere keine

Beschränkung auf zwei Wiederholung-Operationen. Dies ergibt sich auch weder aus dem Zurückhalten von Operationen in jeweils eigenen, separaten Warteschlangen, bis die Voraussetzungen zur Ausführung erfüllt sind (vgl. bspw. *staging queues*, Abs. 0035, 0040; Abs. 0074-0076), noch aus den Prioritäts-Schemata (vgl. Abs. 0066, 0068).

Daher ist der Gegenstand der Patentansprüche 1 und 2 nach Hilfsantrag 2 neu gegenüber Druckschrift D4.

c) Der Gegenstand der unabhängigen Patentansprüche 1 und 2 nach Hilfsantrag 2 beruht jeweils gegenüber dem im Verfahren befindlichen Stand der Technik auch auf einer erfinderischen Tätigkeit (§ 4 PatG).

Eine systematische Begrenzung der Wiederholungs-Operationen für nicht bediente Threads, insbesondere eine aufeinanderfolgende Durchführung von maximal zwei Wiederholung-Operationen, unabhängig von der Anzahl der erforderlichen Wiederholung-Operationen entsprechend Merkmal 1.8^{H2} des **Patentanspruchs 1 gemäß Hilfsantrag 2** ist dem Fachmann aus dem vorliegenden Stand der Technik weder in Verbindung mit seinem Fachwissen noch in einer Zusammenschau der Druckschriften D1 bis D4 nahegelegt.

Die Vorgehensweise nach **Druckschrift D1** steht in keinem Zusammenhang mit einer Gruppierung und Wiederholung parallel ausgeführter Threads bei divergenten Cache-Zugriffen für eine in parallelen Threads ausgeführte Ladeanweisung. Insbesondere werden keine Strategien zur Durchführung von Wiederholungen für solche divergenten Cache-Zugriffe betrachtet. Druckschrift D1 ist daher ebenfalls kein Hinweis darauf zu entnehmen, die Wiederholung von nicht bedienten Operationen im Sinne des Merkmals 1.8^{H2} des Patentanspruchs 1 zu begrenzen.

Druckschrift D2 sind keine Strategien zu entnehmen, wie Threads bei einer Wiederholung zusammengefasst oder die Wiederholungen in die Verarbeitungspipelines eingefügt werden. Auch Druckschrift D2 ist daher kein Hinweis darauf zu entnehmen, die Wiederholung von nicht bedienten Operationen im Sinne des Merkmals 1.8^{H2} des Patentanspruchs 1 zu begrenzen.

Druckschrift D3 liefert zwar neben dem Vorsehen einer Wiederholung-Operationen bei nicht erfolgreich abgeschlossenen bzw. nicht bedienten Threads einen Hinweis darauf, Thread-Gruppen zu bilden und Speicherzugriffe opportunistisch zu nutzen, wie vorstehend zum jeweiligen Patentanspruch 1 des Hauptantrags und des Hilfsantrags 1 näher ausgeführt (vgl. Abschnitte II.3. und II.4.). Druckschrift D3 befasst sich aber nicht mit der Frage, welche Auswirkungen solche Wiederholungen auf die Performanz des Systems haben, und gibt somit keinen Hinweis darauf, die Wiederholung von Operationen im Sinne des Merkmals 1.8^{H2} des Patentanspruchs 1 zu begrenzen.

Druckschrift D4 befasst sich mit Wiederholungsstrategien für nicht erfolgreich abgeschlossene Threads für den Fall eines „Cache-Miss“ bei einem Speicherzugriff, der ein Nachladen erfordert. Die im Hinblick darauf gewählte Lösung der verzögerten Wiederholung von Operationen mit eigenen Warteschlangen lässt sich aufgrund der verschiedenen Randbedingungen nicht auf den vorliegend betrachteten Fall divergenter Zugriffsoptionen übertragen, zumal die Lösung nach Druckschrift D4 aufgrund der gezielten Verzögerung auch von der vorliegend beanspruchten Vorgehensweise weg weist. Druckschrift D4 ist darüber hinaus auch kein Hinweis darauf zu entnehmen, die Wiederholung von nicht bedienten Operationen im Sinne des Merkmals 1.8^{H2} des Patentanspruchs 1 zu begrenzen.

Somit ergibt sich aus keiner der im Verfahren befindlichen Druckschriften ein Anlass für den Fachmann, bei der Verarbeitung divergenter Operationen zwei Wiederholung-Operationen zu bestimmen, welche einen Taktzyklus versetzt die

Mehr-Stufen-Pipeline durchlaufen, wobei unabhängig von der Anzahl der erforderlichen Wiederholung-Operationen maximal zwei Wiederholung-Operationen aufeinanderfolgend durchgeführt werden (vgl. Merkmal 1.8^{H2}).

Es ist daher anzuerkennen, dass der Gegenstand des Patentanspruchs 1 gemäß Hilfsantrag 2 auf einer erfinderischen Tätigkeit beruht und patentfähig ist.

Dies gilt in gleicher Weise für den auf ein Subsystem zum Wiederholen einer divergenten Zugriffsoperation auf gemeinsam genutzte Ressourcen gerichteten **Patentanspruch 2 gemäß Hilfsantrag 2**. Dieses Subsystem ist durch die funktionalen Merkmale 2.1 bis 2.8^{H2} charakterisiert, die den Verfahrensmerkmalen 1.2 bis 1.8^{H2} nach Patentanspruch 1 des Hilfsantrags 2 entsprechen. Insbesondere ist auch das Subsystem nach Patentanspruch 2 des Hilfsantrags 2 entsprechend Merkmal 1.8^{H2} des Patentanspruchs 1 dazu eingerichtet, dass maximal zwei Wiederholung-Operationen aufeinanderfolgend durchgeführt werden unabhängig von der Anzahl der erforderlichen Wiederholung-Operationen (vgl. Merkmale 2.8^{H2}). Es wird daher auf die Ausführungen zu Patentanspruch 1 des Hilfsantrags 2 verwiesen, die für den Patentanspruch 2 in gleicher Weise gelten.

Der Gegenstand des Patentanspruchs 2 gemäß Hilfsantrag 2 beruht daher ebenfalls auf einer erfinderischen Tätigkeit und ist patentfähig.

d) Gleichfalls patentfähig sind die über das Selbstverständliche hinausgehenden Ausführungsformen gemäß den Patentansprüchen 3 bis 8 nach Hilfsantrag 2, die auf Patentanspruch 2 rückbezogen sind.

6. Da die vorgelegten geltenden Unterlagen auch den weiteren Voraussetzungen zur Patenterteilung (§§ 1, 2, 5, 34 PatG) genügen, war auf die Beschwerde der Anmelderin der Zurückweisungsbeschluss der Prüfungsstelle für Klasse G 06 F des Deutschen Patent- und Markenamts aufzuheben und ein Patent gemäß Hilfsantrag 2 zu erteilen.

III.

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 einen beim Bundesgerichtshof zugelassenen Rechtsanwalt als Bevollmächtigten schriftlich einzulegen.

Wickborn

Kruppa

Veit

Altvater

prä