



# BUNDESPATENTGERICHT

18 W (pat) 57/14

---

(Aktenzeichen)

Verkündet am  
19. August 2015

...

## BESCHLUSS

In der Beschwerdesache

**betreffend die Patentanmeldung 11 2004 002 368.8-53**

...

hat der 18. Senat (Techn. Beschwerdesenat) des Bundespatentgerichts auf die mündliche Verhandlung vom 19. August 2015 durch die Vorsitzende Richterin Dipl.-Ing. Wickborn sowie die Richter Kruppa, Dipl.-Phys. Dr. Schwengelbeck und Dr.-Ing. Flaschke

beschlossen:

Die Beschwerde wird zurückgewiesen.

## **Gründe**

### **I.**

Die vorliegende Patentanmeldung 11 2004 002 368.8 geht aus einer PCT-Anmeldung hervor (Veröffentlichungsnummer WO 2005/057408 A1), die am 3. Dezember 2004 unter Inanspruchnahme einer US-amerikanischen Priorität vom 4. Dezember 2003 eingereicht wurde. Sie trägt in der deutschen Übersetzung die Bezeichnung

„Prozessor und Verfahren zur Unterstützung eines kompilierergerichteten Managements von Multi-Threading“

und wurde durch die Prüfungsstelle für Klasse G 06 F mit Beschluss vom 13. Oktober 2009 mit der Begründung zurückgewiesen, dass die Gegenstände der jeweiligen Ansprüche 1 nach Hauptantrag und den Hilfsanträgen 1 bis 3 keinen Aspekt erkennen ließen, der durch eine technische Überlegung geprägt sei, die über das elementare Grundwissen und Können des Fachmanns hinausgehe. Die Ansprüche 1 nach Hauptantrag und den Hilfsanträgen 1 bis 3 erfüllten damit nicht die Patentierungsvoraussetzungen nach § 1 PatG. Im Übrigen lasse der Gegenstand des Anspruchs 1 nach Hauptantrag auch in einer etwas anderen Interpretation keine erfinderische Tätigkeit erkennen, wobei auf folgende Druckschriften verwiesen wurde:

**D1:** US 6 061 709 A

**D2:** US 5 740 451 A.

Gegen diesen Beschluss ist die Beschwerde der Anmelderin gerichtet.

Mit Schriftsatz vom 20. Juli 2015 hat der Senat u. a. auf die Druckschrift

**D3:** US 6 101 524 A

als Stand der Technik hingewiesen.

Die Anmelderin beantragt,

den Beschluss der Prüfungsstelle für Klasse G 06 F des Deutschen Patent- und Markenamts vom 13. Oktober 2009 aufzuheben und das Patent auf der Grundlage der folgenden Unterlagen zu erteilen:

- Patentansprüche 1 bis 22, eingegangen am 14. August 2015,  
  
hilfsweise gemäß Hilfsantrag 1  
Patentansprüche 1 bis 20, eingegangen am 14. August 2015,  
  
hilfsweise gemäß Hilfsantrag 2  
Patentansprüche 1 bis 20, eingegangen am 14. August 2015,  
  
hilfsweise gemäß Hilfsantrag 3  
Patentansprüche 1 bis 18, eingereicht in der mündlichen Verhandlung,
- Beschreibung, Seiten 1 bis 22, eingegangen am 6. Juni 2006,
- Figuren 1 bis 9, eingegangen am 6. Juni 2006.

Der seitens des Senats mit einer Gliederung versehene nebengeordnete **Patentanspruch 12 nach Hauptantrag** lautet:

- M1** „Ein Verfahren aufweisend:  
Ausführen von einem oder mehreren Threads in einer Ausführungseinheit eines Prozessors;
- M2** Detektieren in einer Detektionseinheit des Prozessors, die mit der Ausführungseinheit verbunden ist, ob ein gegebener Thread einen Identifizierer enthält; und
- M3** selektives Fortsetzen der Ausführung des gegebenen Threads oder Ausführen eines anderen Threads in der Ausführungseinheit des Prozessors, abhängig davon, ob der Identifizierer durch die Detektionseinheit des Prozessors detektiert wird.“

Der seitens des Senats mit einer Gliederung versehene **Patentanspruch 11 nach Hilfsantrag 1** weist die Merkmale des Anspruchs 12 nach Hauptantrag auf unter Hinzufügung des folgenden Merkmals:

- M3a** „wobei in Antwort auf die Detektion des Identifizierers die Ausführung des gegebenen Threads ausgesetzt wird und ein anderer Thread ausgeführt wird.“

Der seitens des Senats mit einer Gliederung versehene **Patentanspruch 11 nach Hilfsantrag 2** lautet (Merkmalsgliederung hinzugefügt; Änderungen gegenüber dem Anspruch 12 nach Hauptantrag hervorgehoben):

- M1** „Ein Verfahren aufweisend:  
Ausführen von einem oder mehreren Threads in einer Ausführungseinheit eines Prozessors,

- M1a** wobei jeder der Threads eine oder mehrere Anweisungen enthält und wobei ein gegebener Thread einen oder mehrere Identifizierer enthalten kann, wobei der Identifizierer eine Markierung ist, die in einer bestimmten Anweisung des gegebenen Threads enthalten ist;
- M1b** Beginnen der Ausführung der mit dem aktuellen Thread verbundenen Anweisungen;
- M2\*** während der Ausführung des aktuellen Threads, Detektieren in einer Detektionseinheit des Prozessors, die mit der Ausführungseinheit verbunden ist, ob ein-gegebener der aktuelle Thread einen Identifizierer enthält; und
- M3** selektives Fortsetzen der Ausführung des gegebenen Threads oder Ausführen eines anderen Threads in der Ausführungseinheit des Prozessors, abhängig davon, ob der Identifizierer durch die Detektionseinheit des Prozessors detektiert wird;
- M3a\*** wobei in Antwort auf die Detektion des Identifizierers die Ausführung des aktuellen Threads ausgesetzt wird, und ein anderer Thread ausgeführt wird.“

Der seitens des Senats mit einer Gliederung versehene **Patentanspruch 10 nach Hilfsantrag 3** weist die Merkmale des Anspruchs 12 nach Hauptantrag auf unter Hinzufügung der folgenden Merkmale:

- M4** „weiter aufweisend die Aussetzung der Ausführung des gegebenen Threads und Ausführung eines anderen Threads in Antwort auf den Empfang eines globalen Ausführungsparameters;
- M5** wobei in Antwort auf die Detektion des Identifizierers der globale Ausführungsparameter übergangen wird, und die Ausführung des gegebenen Threads fortgesetzt wird.“

Wegen des Wortlauts der weiteren nebengeordneten Ansprüche 1, 17, 21 und 22 nach Hauptantrag, der nebengeordneten Ansprüche 1, 15, 19 und 20 nach den Hilfsanträgen 1 und 2, der nebengeordneten Ansprüche 1, 13, 17 und 18 nach Hilfsantrag 3 sowie der Unteransprüche 2 bis 11, 13 bis 16, 18 bis 20 nach Hauptantrag, der Unteransprüche 2 bis 10, 12 bis 14, 16 bis 18 nach den Hilfsanträgen 1 und 2 und der Unteransprüche 2 bis 9, 11, 12 und 14 bis 16 nach Hilfsantrag 3 wird auf den Akteninhalt verwiesen.

Die Beschwerdeführerin macht hierzu sinngemäß geltend, dass die geltenden Ansprüche nach Hauptantrag sowie nach den Hilfsanträgen 1 bis 3 zulässig seien, und die Anspruchsgegenstände gemäß Hauptantrag und den Hilfsanträgen 1 bis 3 neu seien sowie auf einer erfinderischen Tätigkeit beruhen.

Wegen der weiteren Einzelheiten wird auf den Akteninhalt verwiesen.

## II.

Die zulässige Beschwerde hat in der Sache keinen Erfolg. Der Gegenstand des Anspruchs 12 nach Hauptantrag und die Gegenstände der Ansprüche 11 nach den Hilfsanträgen 1 und 2 sowie der Gegenstand des Anspruchs 10 nach Hilfsantrag 3 beruhen nicht auf einer erfinderischen Tätigkeit. Die Fragen der Zulässigkeit der geltenden Ansprüche sowie der Neuheit können 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 Multi-Threading Prozessoren und insbesondere das Managen der Ausführung von mehreren Threads in einem Multi-Threading Prozessor. Multi-Threading bezeichne die Ausführung von mehreren Threads von einem oder mehreren Prozessen zu einer bestimmten Zeit. Das Multi-Threading könne Kapazitäten ähnlich denen eines Multi-Prozessor-Systems bereitstellen,

während nur ein physischer Prozessor verwendet werde. Ein Prozess bezeichne eine Anwendung, die durch einen Prozessor ausgeführt werde. Ein Prozess könne einen oder mehrere Ausführ-Threads haben. Ein Thread sei ein Steuerungsfluss durch einen gegebenen Prozess. Der momentane Status eines Threads, welcher durch einen momentanen Programmzähler repräsentiert wird, erlaube es dem Prozessor, einen Thread einzufrieren (d. h. er beendet die Ausführung desselben) und die Ausführung zu einem späteren Zeitpunkt fortzuführen. Das Teilen eines einzelnen Prozessors zwischen mehreren Threads könne die Zeit minimieren, die zum Wechseln zwischen Threads benötigt wird, da nur eine minimale Menge an Information gespeichert und wiederhergestellt werden müsse, wenn von einem Thread zum anderen gewechselt werde. Die Menge an Information, die gespeichert und wiederhergestellt werden müsse, sei deshalb minimal, weil die Programmausführungsumgebung zwischen den verschiedenen Threads geteilt werde. Dies stehe im Gegensatz zu einem Standardkontext-Wechsel in einem normal ablaufenden Prozess. Programmierer schrieben daher Multi-Threaded Anwendungssoftware, um die Leistung einer Anwendung zu verbessern. Da Multi-Threaded Programme geschrieben werden könnten, um parallel abzulaufen anstelle von nacheinander, könne das Multi-Threading Ausführungszeit einsparen, indem Teile eines Prozesses unabhängig voneinander ausgeführt werden könnten. Obwohl aktuelle Multi-Threading Technologien die Ausführung von Prozessen durch Ausführung mehrerer Threads zur selben Zeit beschleunigen könnten, könne ein erheblicher Anteil an Ausführungszeit an bestimmten Threads verschwendet werden, die Anweisungen enthalten, die noch nicht zur Ausführung bereit seien. Ein Verfahren, um dieses Problem zu überwinden, sei, von einem Thread zum anderen zu wechseln oder die Ausführung eines aktuellen Threads auszusetzen, wenn das System feststelle, dass der aktuelle Thread zu lange zur Ausführung benötige. Wenn die Ausführung eines Befehls zum Beispiel in einer Cache-Fehlzuordnung resultiere, dann könne der Prozessor zu einem anderen Thread wechseln, der zur Ausführung bereit sei. Dieses Verfahren könne allerdings die Ausführung genau des Befehls benötigen, der die Cache-Fehlzuordnung ver-

ursacht habe, was zu verschwendeter Ausführungszeit führen könne (vgl. geltende Beschreibungsunterlagen, S. 1, Z. 9 bis S. 3, Z.3).

Als **Aufgabe** wird im Schriftsatz vom 14. August 2015 (vgl. S. 5 zw. Abs. im Abschnitt c.) angegeben, einen Prozessor stets optimal auszulasten, um die Rechenkapazitäten effizient zu nutzen.

Der **Fachmann**, der mit der Lösung dieser Aufgabenstellung betraut wird, weist eine abgeschlossene Hochschulausbildung auf dem Gebiet der Informationstechnik auf und verfügt über eine mehrjährige Erfahrung im Bereich von Prozesssteuerungen im Zusammenhang mit Threads und höheren Programmiersprachen wie C++ oder Java.

Die genannte Aufgabe soll unter anderem durch die Merkmale des auf ein Verfahren gerichteten **Anspruchs 12 nach Hauptantrag** gelöst werden. Danach ist vorgesehen, einen oder mehrere Threads mittels einer Ausführungseinheit eines Prozessors auszuführen. Abhängig davon, ob ein Identifizierer während der Ausführung eines Threads detektiert wird, wird die Ausführung des gegebenen Threads fortgesetzt oder ein anderer Thread ausgeführt. Das Detektieren erfolgt dabei in einer Detektionseinheit des Prozessors, welche mit der Ausführungseinheit verbunden ist.

In der Fassung des **Hilfsantrags 1** ist zur Lösung der Aufgabe zusätzlich vorgesehen, dass bei Detektion eines Identifizierers die Ausführung des gegebenen Threads ausgesetzt und ein anderer Thread stattdessen ausgeführt wird.

In der Fassung des **Hilfsantrags 2** ist weiter konkretisiert, dass innerhalb eines Threads mehrere Identifizierer enthalten sein können, die während der Ausführung eines Threads detektiert werden. Dabei ist angegeben, dass der Identifizierer eine Markierung innerhalb einer bestimmten Anweisung ist.



Gemäß **Anspruch 10 nach Hilfsantrag 3** ist ein globaler Ausführungsparameter vorgesehen, der anzeigt, dass der Prozessor den aktuellen Thread aussetzen und zu einem anderen Thread wechseln soll. Der globale Ausführungsparameter wird übergangen, wenn der Thread ebenfalls einen Identifizierer enthält, der vorgibt, dass die Ausführung des aktuellen Threads fortgesetzt werden soll.

Einige Merkmale des **Anspruchs 12 nach Hauptantrag** bzw. der Ansprüche 11 nach den Hilfsanträgen 1 und 2 bzw. des Anspruchs 10 nach Hilfsantrag 3 bedürfen der Auslegung:

Es ist gemäß **Merkmal M1** das Ausführen von einem oder mehreren Threads in einer Ausführungseinheit eines Prozessors vorgesehen. Der Prozessor ist als ein Multithreading-fähiger Prozessor aus der SPARC- oder x86-Familie zu verstehen, der in der Lage ist, mehrere Threads eines Prozesses quasi-parallel auf einem physischen Prozessorkern bearbeiten zu können (vgl. S. 1, Z. 9 bis S. 2, Z. 4, S. 6, Z. 5 - 21 sowie S. 8, Z. 7 - 12 der geltenden Beschreibungsunterlagen). Das scheinbar gleichzeitige Abarbeiten der Threads wird durch eine softwareseitige Sequentialisierung der Threads erreicht. Zu diesem Zweck werden sogenannte „Identifizierer“ eingesetzt. Der Identifizierer, bei dem es sich z. B. um eine Anweisung oder eine Markierung (d. h. ein Flag) innerhalb eines Threads handeln kann, zeigt an, dass der aktuell ausgeführte Thread ausgesetzt und stattdessen ein anderer Thread ausgeführt werden kann (vgl. S. 6, Z. 23 bis S. 7, Z. 10 u. Z. 25 - 30 u. S. 10, Z. 8 - 23 der geltenden Beschreibungsunterlagen). Ob ein gegebener Thread einen Identifizierer enthält, prüft eine Detektionseinheit (**Merkmal M2**). Die Detektionseinheit ist im Sinne der Beschreibung der Anmeldung eine funktionale Einheit und kann als ein Scheduler verstanden werden (vgl. geltende Beschreibungsunterlagen, S. 7, Z. 5 - 10, S. 11, Z. 4 - 7 u. S. 13, Z. 7 - 10 i. V. m. Fig. 1). Abhängig davon, ob die Detektionseinheit einen Identifizierer detektiert, setzt die Ausführungseinheit selektiv die Abarbeitung des gegebenen Threads fort oder unterbricht die Abarbeitung des gegebenen Threads und arbeitet stattdessen einen anderen Thread ab (**Merkmal M3**).

2. Das Verfahren gemäß **Anspruch 12 nach Hauptantrag** ergibt sich für den Fachmann in naheliegender Weise aus der Lehre der Druckschrift **D3** und beruht damit nicht auf einer erfinderischen Tätigkeit.

Druckschrift **D3** beschreibt ein Verfahren zur Ausführung von Threads durch einen Multithreading-fähigen Einfachkern-Prozessor (*CPU 116* in Fig. 1 i. V. m. Sp. 2, Z. 43 - 47, Fig. 4 u. 10, sowie Sp. 3, Z. 66 - Sp. 4, Z. 1). Dabei gehört es zum Grundwissen des Fachmanns, dass der Prozessor bzw. die CPU eine Ausführungseinheit aufweist, in der die Befehle und Threads ausgeführt werden (**Merkmal M1**). Für die Steuerung der Threads wird ein Scheduler (*thread scheduler*) verwendet, der eine Detektionseinheit darstellt, da er aufgrund der Detektion von Identifizieren (*thread identifier* i. V. m. *FirstCriticalEvent[i]*, *LastCriticalEvent[i]*) bestimmt, wann welcher Thread jeweils Prozessorzeit erhält (vgl. Sp. 3, Z. 13 - 20, Sp. 5, Z. 64 - Sp. 6, Z. 26 u. Sp. 7, Z. 14 - 33 i. V. m. Fig. 4; **teilweise Merkmal M2**). Nach Aufruf einer *start()*-Methode wird ein Thread vom Scheduler gestartet und solange ausgeführt, bis nach Ablauf der zugeteilten Zeit auf einen anderen Thread umgeschaltet wird (vgl. Sp. 3, Z. 13 - 48). Dabei werden die Threads derart koordiniert, dass bestimmte Threads, bei deren Abarbeitung kritische Zustände zu erwarten sind, zeitweise von einer unterbrechungsfreien Ausführung ausgeschlossen sind. Damit offenbart Druckschrift **D3** auch die von der Anmelderin in der mündlichen Verhandlung als anmeldungsgemäß beschriebenen „Sollbruchstellen“ innerhalb eines Threads (vgl. Threadwechsel nach dem ersten kritischen Abschnitt in Fig. 4 i. V. m. Sp. 5, Z. 14 - 28). Der Fachmann entnimmt Druckschrift **D3**, dass insbesondere bei der Verwendung gemeinsamer Ressourcen die Synchronisation der Threads erforderlich ist, um kritische Situationen zu vermeiden, in denen sich Threads gegenseitig blockieren (*deadlocks*; vgl. Sp. 4, Z. 18 - 33 u. Sp. 8, Z. 39 - 41). Die Abarbeitung eines gegebenen Threads kann vom Scheduler unterbrochen werden, um andere Threads auszuführen. Dabei wird die Ausführung des gegebenen Threads selektiv fortgesetzt oder ein anderer Thread fortgeführt, abhängig davon, ob der als Detektionseinheit fungierende Scheduler einen Identifizierer, hier ein Flag, zur Selbstsuspendierung

(*AnySelfSuspended / suspend(self) operation*) detektiert (vgl. Fig. 4 u. 10 i. V. m. Sp. 4, Z. 44 - 46, Sp. 7, Z. 14 - 33, Sp. 9, Z. 5 - 49 u. Sp. 11, Z. 5 - 13 u. 18 - 20; **teilweise Merkmal M3**).

Der aus Druckschrift D3 bekannte Scheduler ist – in Übereinstimmung mit den Ausführungen der Anmelderin – in Form von Software ausgeführt (vgl. Sp. 3, Z. 24 - 44 u. Sp. 6, Z. 18 - 24). Allerdings weiß der Fachmann, dass dieser Scheduler im Prozessor – welcher die Threads ausführen muss – realisiert ist. Somit umfasst der Prozessor auch eine Detektionseinheit, die der vorliegenden Anmeldung entsprechend als Scheduler konzipiert und mit der Ausführungseinheit funktional verbunden ist (vgl. Sp. 6, Z. 12 - 17; **Merkmale M2, M3**).

Die Auffassung der Anmelderin, die in Druckschrift D3 beschriebene Detektionseinheit wäre in einer Abstraktionsschicht zwischen dem Anwendungsprogramm und dem Betriebssystem ausgeführt, kann dahinstehen. Denn wie vorstehend ausgeführt, ist Druckschrift D3 auch zu entnehmen, dass der als Detektionseinheit zu verstehende Scheduler während der Abarbeitung des Programms letztendlich im physischen Prozessor realisiert ist.

Dem Argument der Anmelderin, dass in der Anmeldung im Gegensatz zur Lehre von Druckschrift D3 ein hardwaregesteuerter Thread-Wechsel offenbart sei, kann nicht beigetreten werden. Denn in den geltenden Anmeldeunterlagen wird in Bezug zu Figur 1 auf Seite 13 im zweiten Absatz offenbart, dass in Figur 1 funktionale Blöcke des Prozessors dargestellt sind. Weiter wird auf Seite 7, erster Absatz der Anmeldeunterlagen explizit offenbart, dass die Funktionalität der Detektionseinheit in der Einteilungseinheit (Scheduler) enthalten sein kann. Bei dieser handelt es sich gemäß den Anmeldeunterlagen (vgl. S. 11, Z. 4 - 7) um Software des Betriebssystems.

Auch das Argument der Anmelderin, in Druckschrift D3 wären nur High-Level-Anweisungen beschrieben, die dem Prozessor nicht als Opcodes zur Verfügung

stehen würden, greift nicht. Denn eine solche Maßnahme ist nicht Gegenstand des Anspruchs.

Somit ist der Gegenstand des Anspruchs 12 nach Hauptantrag dem Fachmann nahegelegt. Der geltende Patentanspruch 12 nach Hauptantrag ist daher nicht patentfähig.

**3.** Auch der Gegenstand des **Anspruchs 11 nach Hilfsantrag 1** beruht gegenüber Druckschrift **D3** nicht auf einer erfinderischen Tätigkeit.

Patentanspruch 11 nach Hilfsantrag 1 unterscheidet sich von Anspruch 1 nach Hauptantrag durch das angefügte Merkmal M3a, wonach „in Antwort auf die Detektion des Identifizierers die Ausführung des gegebenen Threads ausgesetzt wird und ein anderer Thread ausgeführt wird“, und beinhaltet ansonsten die Merkmale gemäß Hauptantrag, so dass hinsichtlich dieser Merkmale auf die Ausführungen unter Abschnitt II.2 verwiesen wird.

Auch das zusätzlich aufgenommene Merkmal M3a kann eine erfinderische Tätigkeit nicht begründen. Denn wie zum Hauptantrag ausgeführt, entnimmt der Fachmann Druckschrift **D3**, dass der gegebene Threads fortgesetzt wird oder ein anderer Thread ausgeführt wird, abhängig davon, ob ein Identifizierer durch die Detektionseinheit des Prozessors detektiert wird. Es wird dabei offenbart, dass während der Ausführung eines Threads zu einem anderen Thread umgeschaltet werden kann (*thread switch*; vgl. Sp. 6, Z. 58 - 65 i. V. m. Fig. 3 u. 4, sowie Sp. 10, Z. 62 - 65). Hierzu werden Methoden genannt, mit denen der Scheduler die Threads in ihre sich wechselnden Zustände bringen kann. So werden die Methoden *suspend()*, *suspend(self)* und *interrupt()* beschrieben, mit denen ein Thread ausgesetzt werden kann, um anderen Threads die Möglichkeit zur Ausführung zu geben. Beim Aufruf der *suspend(self)*-Methode wird im gegebenen Thread ein als Identifi-

zierer zu verstehendes Flag gesetzt, das vom Scheduler detektiert wird (vgl. Fig. 8, Schritte 807 u. 809 i. V. m. Sp. 9, Z. 42 - 47). Daraufhin wird der Thread temporär ausgesetzt und ein anderer Thread ausgeführt (vgl. Sp. 4, Z. 34 - 51, Sp. 7, Z. 14 - 34, Sp. 8, Z. 45 - 51, Sp. 9, Z. 5 u. 6, Sp. 10, Z. 46 - 48, Sp. 11, Z. 18 - 20); Figur 4 zeigt beispielhaft das Umschalten zwischen vier Threads (**Merkmal M3a**).

Somit ist auch der Gegenstand des Anspruchs 11 nach Hilfsantrag 1 dem Fachmann nahegelegt. Der geltende Patentanspruch 11 nach Hilfsantrag 1 ist daher nicht patentfähig.

**4.** Auch der Gegenstand des **Anspruchs 11 nach Hilfsantrag 2** beruht gegenüber Druckschrift **D3** nicht auf einer erfinderischen Tätigkeit.

Patentanspruch 1 nach Hilfsantrag 2 unterscheidet sich vom Anspruch 1 nach Hilfsantrag 1 darin, dass zusätzlich die Merkmale M1a und M1b aufgenommen wurden. Zudem wurden die Merkmale M2\* und M3a\* konkretisiert. Ansonsten beinhaltet der Anspruch die Merkmale gemäß Hauptantrag und Hilfsantrag 1, so dass hinsichtlich dieser Merkmale auf die Ausführungen unter Abschnitt II.2 bzw. zu Merkmal M3a\* unter Abschnitt II.3 verwiesen wird.

Mit den zusätzlich aufgenommenen Merkmalen wird beansprucht, dass „jeder der Threads eine oder mehrere Anweisungen enthält“ und „ein gegebener Thread einen oder mehrere Identifizierer enthalten kann, wobei der Identifizierer eine Markierung ist, die in einer bestimmten Anweisung des gegebenen Threads enthalten ist“ (Merkmal M1a). Zudem wird das Verfahren konkretisiert, indem zusätzlich beansprucht wird, dass mit der Ausführung der mit dem aktuellen Thread verbundenen Anweisungen begonnen wird (Merkmal M1b), und dass „während der Ausführung des aktuellen Threads“ detektiert wird, ob der aktuelle Thread einen Identifi-

zierer enthält (Merkmal M2\*). Wie bereits in den Abschnitten II.2 und II.3 ausgeführt, ist Druckschrift D3 entnehmbar, dass jeder der Threads eine oder mehrere Anweisungen enthält, und dass ein gegebener Thread einen oder mehrere Identifizierer enthalten kann (vgl. Sp. 3, Z. 30 - 44, Sp. 5, Z. 64 - Sp. 6, Z. 8, Sp. 8, Z. 45 - 51 u. Sp. 11, Z. 37 ff. sowie Fig. 10; **Merkmal M1a**). Des Weiteren wird offenbart, dass mit der Ausführung der mit dem aktuellen Thread verbundenen Anweisungen begonnen wird (S0 in Fig. 10; **Merkmal M1b**). Der – wie vorstehend beschrieben – als Detektionseinheit zu verstehende Scheduler detektiert während der Ausführung des aktuellen Threads, ob der aktuelle Thread einen Identifizierer enthält (vgl. Sp. 6, Z. 12 - 23 i. V. m. Sp. 7, Z. 14 - 33 u. Fig. 4 sowie Fig. 8, Schritt 807 u. Fig. 9 mitsamt zugeh. Text in Sp. 10, Z. 19 ff.; **Merkmal M2\***).

Somit ist auch der Gegenstand des Anspruchs 11 nach Hilfsantrag 2 dem Fachmann nahegelegt. Der geltende Patentanspruch 11 nach Hilfsantrag 2 ist daher nicht patentfähig.

**5.** Auch der Gegenstand des **Anspruchs 10 nach Hilfsantrag 3** beruht gegenüber Druckschrift **D3** nicht auf einer erfinderischen Tätigkeit.

Patentanspruch 10 nach Hilfsantrag 3 unterscheidet sich von Anspruch 1 nach Hauptantrag durch die angefügten Merkmale M4 und M5, wonach das Verfahren weiter „die Aussetzung der Ausführung des gegebenen Threads und Ausführung eines anderen Threads in Antwort auf den Empfang eines globalen Ausführungsparameters“ (Merkmal M4), „wobei in Antwort auf die Detektion des Identifizierers der globale Ausführungsparameter übergangen wird, und die Ausführung des gegebenen Threads fortgesetzt wird“ (Merkmal M5), aufweist, und beinhaltet ansonsten die Merkmale gemäß Hauptantrag, so dass hinsichtlich dieser Merkmale auf die Ausführungen unter Abschnitt II.2 verwiesen wird.

Auch die zusätzlich aufgenommenen Merkmale M4 und M5 können eine erfinderische Tätigkeit nicht begründen. Wie zum Hauptantrag ausgeführt, entnimmt der Fachmann Druckschrift **D3**, dass der gegebene Thread fortgesetzt oder ein anderer Thread ausgeführt wird, abhängig davon, ob ein Identifizierer durch die Detektionseinheit des Prozessors detektiert wird. Des Weiteren ist ein globaler Ausführungsparameter entnehmbar (vgl. Sp. 6, Z. 4 - 18: *FirstCriticalEvent[i]* and *LastCriticalEvent[i]* represent a global clock value), welcher einen gemeinsam genutzten Codeabschnitt freigeben (set to '0') oder sperren kann (set to '1'; vgl. Sp. 7, Z. 62 - Sp. 8, Z. 9 i. V. m. Fig. 5: *release GCounter\_Lock, acquire GCounter\_Lock*). In Antwort auf den Empfang des globalen Ausführungsparameters *release GCounter\_Lock* wird der Prozessor durch den Scheduler freigegeben, damit die Ausführung eines gegebenen Threads ausgesetzt und ein anderer Thread ausgeführt werden kann (vgl. Sp. 9, Z. 19 - 27 u. Sp. 1, Z. 62 - 65; **Merkmal M4**). Der beschriebene Scheduler sieht dabei vor, dass der globale Ausführungsparameter zunächst übergangen und die Ausführung des gegebenen Threads fortgesetzt wird. Erst wenn der im gegebenen Thread enthaltene Identifizierer ein zweites Mal detektiert wird, wird der Zugriff auf den kritischen Bereich freigeben, damit die Ausführung eines gegebenen Threads ausgesetzt und der wartende Thread ausgeführt werden kann (*suspend(self)*); vgl. Fig. 9 i. V. m. Sp. 10, Z. 5 - 14; **Merkmal M5**).

Somit ist dem Fachmann auch der Gegenstand des Anspruchs 10 nach Hilfsantrag 3 nahegelegt. Der geltende Patentanspruch 10 nach Hilfsantrag 3 ist daher nicht patentfähig.

6. Mit dem nicht patentfähigen Anspruch 12 nach Hauptantrag, Anspruch 11 nach Hilfsantrag 1 bzw. Hilfsantrag 2 sowie Anspruch 10 nach Hilfsantrag 3 sind auch die auf die Ansprüche direkt oder indirekt rückbezogenen Unteransprüche sowie die jeweiligen nebengeordneten Ansprüche nach Hauptantrag bzw. nach

den Hilfsanträgen 1 bis 3 nicht schutzfähig, da auf diese Ansprüche kein eigenständiges Patentbegehren gerichtet war und über einen Antrag nur einheitlich entschieden werden kann (vgl. BGH, Beschluss vom 27. Juni 2007 – X ZB 6/05, GRUR 2007, 862, Abschnitt III. 3. a) aa) – Informationsübermittlungsverfahren II).

7. Nachdem die jeweiligen Anspruchssätze nach Hauptantrag bzw. den Hilfsanträgen 1, 2 und 3 nicht patentfähig sind, war die Beschwerde zurückzuweisen.

### 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

Dr. Schwengelbeck

Dr. Flaschke

Hu