



(10) **DE 11 2008 001 473 B4** 2021.08.26

(12)

Patentschrift

(21) Deutsches Aktenzeichen: **11 2008 001 473.6**
(86) PCT-Aktenzeichen: **PCT/US2008/006746**
(87) PCT-Veröffentlichungs-Nr.: **WO 2008/153799**
(86) PCT-Anmeldetag: **28.05.2008**
(87) PCT-Veröffentlichungstag: **18.12.2008**
(43) Veröffentlichungstag der PCT Anmeldung
in deutscher Übersetzung: **01.07.2010**
(45) Veröffentlichungstag
der Patenterteilung: **26.08.2021**

(51) Int Cl.: **G06F 12/08** (2006.01)
G06F 9/38 (2006.01)

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

(30) Unionspriorität:
11/754,589 **29.05.2007** **US**

(73) Patentinhaber:
Advanced Micro Devices, Inc., Sunnyvale, Calif., US

(74) Vertreter:
Grünecker Patent- und Rechtsanwälte PartG mbB, 80802 München, DE

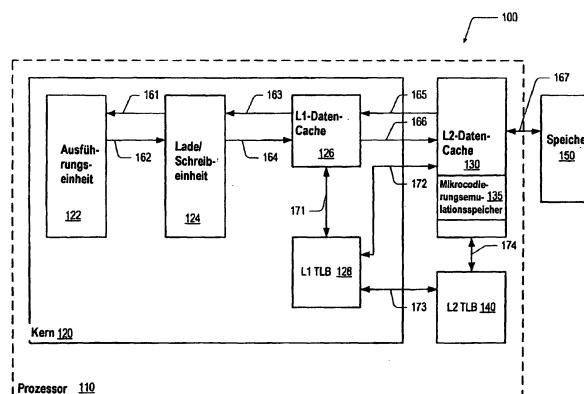
(72) Erfinder:
Lauterbach, Gary, Los Altos, Calif., US; Holloway, Bruce R., Boulder Creek, Calif., US; Butler, Michael Gerard, San Jose, Calif., US; Lie, Sean, Santa Clara, Calif., US

(56) Ermittelter Stand der Technik:

US	5 905 997	A
WO	99/ 60 480	A1

(54) Bezeichnung: **Zwischenspeicherung eines Mikrocodierungsemulationsspeichers**

(57) Hauptanspruch: Prozessor (110) mit:
einer Cache-Speicherhierarchie mit mindestens einem ersten Cache-Speicher (126) der Ebene 1 und einem Cache-Speicher (130) höherer Ebene;
wobei der Prozessor (110) ausgebildet ist, um:
einen ersten Bereich eines physikalischen Speicherraumes (220) einem ersten Bereich des Cache-Speichers (130) höherer Ebene zuzuordnen;
Befehle auszuführen, wovon mindestens einige eine Mikrocodierung enthalten;
einen ersten Befehl der Befehle zu dekodieren;
wenn ein Mikrocodierungszugriffssignal in Reaktion auf das Dekodieren des ersten Befehls erfasst wird, zu bestimmen, dass der erste Befehl eine Mikrocodierung darstellt und es dem ersten Befehl zu ermöglichen, auf den ersten Bereich des Cache-Speichers (130) höherer Ebene unter Verwendung von Lade- und Speicherbefehlen zuzugreifen; und
wenn kein Mikrocodierungszugriffssignal in Reaktion auf das Dekodieren des ersten Befehls erfasst wird, zu bestimmen, dass der erste Befehl keine Mikrocodierung darstellt und zu verhindern, dass der erste Befehl auf den ersten Bereich des Cache-Speichers (130) höherer Ebene zugreift.



Beschreibung

Hintergrund der Erfindung

Technisches Gebiet

[0001] Diese Erfindung betrifft Mikroprozessoren und betrifft insbesondere die Emulation komplexer Befehle durch Mikrocodierung und betrifft die Zwischenspeicherung eines Speicherraumes, der während einer derartigen Emulation verwendet wird.

Hintergrund

[0002] Obwohl es wünschenswert ist, bei Mikroprozessoren die Kompatibilität zu einer komplexen Befehlssatzcomputer- (CISC) Architektur beizubehalten, bieten andere Architekturen eine bessere Ausführungsgeschwindigkeit und ein höheres Leistungsverhalten. Die Gestalter von Mikroprozessoren haben bislang versucht, sowohl eine CISC-Kompatibilität als auch ein gutes Leistungsverhalten durch das Simulieren von CISC-Befehlen zu erreichen. Beispielsweise enthalten superskalare Computerarchitekturen mit einem reduzierten Befehlssatz (RISC) eine Mikrocodierung, die die CISC-Befehlsemulation ausführt. Während des Emulationsprozesses verwendet die Mikrocodierung einen Zwischenspeicher, um Zwischenwerte abzulegen. Um ein hohes Leistungsverhalten beizubehalten, ist es wünschenswert, dass die Mikrocodierung des Mikroprozessors auf den Emulationsspeicher möglichst schnell zugreifen kann.

[0003] Des weiteren enthalten Mikroprozessoren im Allgemeinen mehrere Pufferspeicher bzw. Cache-Speicher, die hierarchisch angeordnet sind und von mehreren Kernen oder Ausführungseinheiten gemeinsam benutzt werden. Es wird eine Vielzahl von Cache-Speicherarchitekturen eingesetzt und diese enthalten Kombinationen aus chipinternen Cache-Speichern und chipexternen Cache-Speichern. Speicheroperationen, die Daten aus einem Cache-Speicher oder Speicher auslesen, werden hierin kürzer als „Ladeoperationen“ bezeichnet. Speicheroperationen, die Daten in den Cache-Speicher oder den Speicher schreiben, werden hierin kürzer als „Schreiboperationen“ bezeichnet. Eine Ladeoperation oder eine Schreiboperation können auf eine spezielle Cache-Zeile (oder einen Teil einer Cache-Zeile) zielen und können eine Adresse enthalten, die die angezielte Zeile angibt, und können auch Daten enthalten, die aus der Cache-Zeile auszulesen oder in diese zu schreiben sind. Da Cache-Speicherzugriffe schneller sind als Hauptspeicherzugriffe werden diverse Cache-Speicherzugriffstechniken eingesetzt, um die Wahrscheinlichkeit zu erhöhen, dass die Daten in einem Cache-Speicher vorhanden sind, wenn ein Kern oder eine Ausführungseinheit auf diese Daten zugreifen muss, wodurch die Ausführungsgeschwindigkeit

verbessert wird. Folglich bietet das Zwischenspeichern des Mikrocodierungsemulationsspeichers im Cache-Speicher den Leistungsvorteil einer relativ kürzeren Zugriffszeit des Cache-Speichers im Vergleich zum Systempeicher. Die kürzesten Zugriffszeiten sind im Allgemeinen jene, die mit der tiefsten Ebene der Cache-Hierarchie verknüpft sind, wobei diese üblicherweise als L1-Cache- bzw. einfach als L1 bezeichnet wird. Daher ist es wünschenswert, den Mikrocodierungsemulationsspeicher im L1 zwischenzuspeichern. Derartige Leistungsvorteile wurden häufig verstärkt, indem ein Teil des L1 permanent für den Mikrocodierungsemulationsspeicher reserviert wurde.

[0004] Selbstverständlich wären die Leistungsvorteile der Verwendung des L1-Cache-Speichers auch für andere Prozesse von Vorteil. Es ist daher wünschenswert, den L1-Cache-Speicher möglichst groß zu gestalten, um damit die Verfügbarkeit des L1-Cache-Speicherraumes für jeden Prozess zu erhöhen. Die Vergrößerung des L1 steigert jedoch die Kosten und die Komplexität des Mikroprozessors. Wenn ferner der Mikrocodierungsemulationsspeicher permanent im L1 abgelegt ist, ist dieser Bereich des L1 nicht für andere Prozesse verfügbar. Um die zuvor genannten Probleme zu lösen, ist eine Möglichkeit erforderlich, die Verfügbarkeit des Speicherraumes in einem L1-Cache-Speicher mit vorgegebener Größe für alle Prozesse zu verbessern, während die Vorteile der Zwischenspeicherung des Mikrocodierungsemulationsspeichers beibehalten werden.

[0005] In der US 5 905 997 A wird ein Verfahren zur Verwaltung eines Cache-Speichers beschrieben, wobei ein Hypercode und Speicheradressplatz in einer logischen Bank eines Cache-Speichers gespeichert werden können, wenn die logische Bank nicht durch Tags besetzt wird. In der WO 99/60480 A1 wird ein Verfahren zur Verwaltung eines Cache-Speichers beschrieben, wobei ein Controller einen Mikrocode ausführt, der auf einem Board des lokalen Controllers in einem separaten Nurlese-Speicher oder in einem Datenbereich des Cache-Speichers gespeichert werden kann.

Überblick über die Erfindung

[0006] Es sind hier diverse Ausführungsformen eines Prozessors, eines Computersystems und Verfahren offenbart. Der Prozessor umfasst eine Cache-Hierarchie mit mindestens einem ersten Cache-Speicher der Ebene 1 und mindestens einem Cache-Speicher höherer Ebene. Der Prozessor ist ausgebildet, einen ersten Bereich eines physikalischen Speicherraumes einem ersten Bereich des Cache-Speichers höherer Ebene zuzuordnen, Befehle auszuführen, wovon mindestens einige eine Mikrocodierung enthalten einen ersten Befehl der Befehle zu dekodieren; wenn ein Mikrocodierungszugriffssignal in

Reaktion auf das Dekodieren des ersten Befehls erfasst wird, zu bestimmen, dass der erste Befehl eine Mikrocodierung darstellt und es dem ersten Befehl zu ermöglichen, auf den ersten Bereich des Cache-Speichers höherer Ebene unter Verwendung von Lade- und Speicherbefehlen zuzugreifen; und wenn kein Mikrocodierungszugriffssignal in Reaktion auf das Dekodieren des ersten Befehls erfasst wird, zu bestimmen, dass der erste Befehl keine Mikrocodierung darstellt und zu verhindern, dass der erste Befehl auf den ersten Bereich des Cache-Speichers höherer Ebene zugreift.

[0007] In einer Ausführungsform ist der Cache-Speicher höherer Ebene ein Cache-Speicher der Ebene 2. In einer weiteren Ausführungsform wird der erste Bereich des physikalischen Speicherraumes permanent für die Verwendung durch die Mikrocodierung reserviert.

[0008] In einer weiteren Ausführungsform ist der Prozessor ausgebildet, eine oder mehrere Cache-Zeilen des ersten Bereichs des Cache-Speichers höherer Ebene von dem Cache-Speicher höherer Ebene zu einem ersten Bereich des ersten Cache-Speichers der Ebene 1 zu verschieben. Der Prozessor ist ferner ausgebildet, es der Mikrocodierung zu ermöglichen, auf den ersten Bereich des ersten Cache-Speichers der Ebene 1 zuzugreifen und zu verhindern, dass Befehle, die keine Mikrocodierung enthalten, auf den ersten Bereich des ersten Cache-Speichers der Ebene 1 zugreifen.

[0009] In einer noch weiteren Ausführungsform ist der Prozessor ausgebildet, ein Mikrocodierungszugriffssignal zu erkennen. Der Prozessor ist ferner ausgebildet zu verhindern, dass Befehle auf den ersten Bereich des physikalischen Speicherraumes zugreifen, wenn das Mikrocodierungszugriffssignal nicht gesetzt ist, und zu erlauben, dass Befehle auf den ersten Bereich des physikalischen Speicherraumes zugreifen, wenn das Mikrocodierungszugriffssignal gesetzt ist.

[0010] In einer noch weiteren Ausführungsform umfasst der Prozessor einen Translationsnebenpuffer (TLB), wobei der Prozessor zur Verhinderung, dass Befehle, die keine Mikrocodierung enthalten, auf den ersten Bereich des physikalischen Speicherraumes zugreifen, ausgebildet ist, TLB-Auffüllungen in den ersten Bereich des physikalischen Speicherraumes zu verhindern.

Figurenliste

Fig. 1 ist eine allgemeine Blockansicht einer Ausführungsform eines Computersystems.

Fig. 2 zeigt eine Ausführungsform eines virtuellen Speichers und einer Cache-Architektur.

Fig. 3 zeigt eine Ausführungsform eines Prozesses zum Zugreifen auf eine Speicherhierarchie, die einen Mikrocodierungsemulationsspeicher enthält.

Fig. 4 zeigt eine Ausführungsform eines Prozessors zum Zugreifen auf einen Mikrocodierungsemulationsspeicher in einem Cache-Speicher der Ebene 1.

Fig. 5 ist eine Blockansicht einer Ausführungsform des Computersystems mit einem Daten-cache-Speicher L2 und einem Mikrocodierungsemulationsspeicher, der mit mehreren Systemkomponenten verbunden ist.

Art bzw. Arten zum Ausführen der Erfindung

[0011] **Fig. 1** ist eine verallgemeinerte Blockansicht einer Ausführungsform eines Computersystems **100**. In der gezeigten Ausführungsform ist ein Prozessor **110** gezeigt, der mit einem Speicher **150** gekoppelt ist. Der Speicher **150** umfasst SDRAM-, SRAM-, ROM-, DRAM- und/oder andere konventionelle Speichereinrichtungen. Der Prozessor **110** enthält einen Kern **120**, einen L2-Daten-cache-Speicher **130** und einen L2-Translationsnebenpuffer (Translations-lookaside-Puffer TLB) **140**. Der Kern **120** enthält eine Ausführungseinheit **122**, eine Lade/Schreib-Einheit **124**, einen L1-Daten-Cache-Speicher **126** und einen L1-TLB **128**. Der L2-Daten-Cache-Speicher **120** enthält einen Mikrocodierungsemulationsspeicher **135**. In alternativen Ausführungsformen umfasst der Prozessor **110** mehr als einen Kern, wobei jeder Kern einen Daten-Cache-Speicher der Ebene 1 enthält und die Kerne gemeinsam einen einzelnen Daten-Cache-Speicher der Ebene 2 verwenden. In einer alternativen Ausführungsform ist der Daten-Cache-Speicher **L1 126** separat zu dem Kern **120** vorgesehen. In anderen alternativen Ausführungsformen sind zusätzliche Cache-Ebenen im Computersystem **100** enthalten, etwa als ein Cache-Speicher der Ebene 3, der intern im Prozessor **110** oder separat zum Prozessor **110** vorgesehen ist. In diesen und anderen alternativen Ausführungsformen ist der Mikrocodierungsemulationsspeicher **135** in einer beliebigen Cache-Ebene oberhalb der Ebene 1 enthalten. Es sind auch viele weitere Ausführungsformen mit eingeschlossen. Für ein einfacheres Verständnis wird jedoch in dem folgenden Beispiel angenommen, dass der Speicherraum permanent in einem Daten-Cache-Speicher der Ebene 2 für den Mikrocodierungsemulationsspeicher **135** reserviert ist.

[0012] Während des Betriebs empfängt die Ausführungseinheit **122** den Datenbereich von Ladeoperationen, die von der Lade/Schreibereinheit **134** auszuführen sind, über eine Verbindung **161** und übermittelt den Datenbereich von Schreiboperationen an die Lade/Schreib-Einheit **124** über eine Verbindung **162**. Die Lade/Schreibereinheit **124** empfängt den Daten-

bereich von Ladeoperationen, die von dem L1-Daten-Cache-Speicher 126 auszuführen sind, über eine Verbindung 163 und übermittelt den Datenbereich für Schreiboperationen in den L1-Daten-Cache-Speicher 126 über eine Verbindung 164. Der L1-Daten-Cache-Speicher 126 empfängt den Datenbereich von Ladeoperationen für den L2-Daten-Cache-Speicher 120 über eine Verbindung 165 und überträgt den Datenbereich für Schreiboperationen in den L2-Daten-Cache-Speicher 130 über eine Verbindung 166. Der L2-Daten-Cache-Speicher 130 empfängt den Datenbereich von Ladeoperationen aus dem Speicher 150 und überträgt den Datenbereich für Schreiboperationen in diesen Speicher über eine Verbindung 167. Der L1-TLB 128 ist so gezeigt, dass dieser mit dem L1-Daten-Cache-Speicher 126 über eine Verbindung 171, mit dem L2-Daten-Cache-Speicher 130 über eine Verbindung 172 und mit dem L2-TLB 140 über eine Verbindung 173 verbunden ist. Der L2-TLB 140 ist ferner so gezeigt, dass er mit dem L2-Daten-Cache-Speicher 130 über eine Verbindung 174 verbunden ist.

[0013] Der L1-Daten-Cache-Speicher 126, der L1-TLB 128, der L2-Daten-Cache-Speicher 130 und der L2-TLB 140 können konventionelle Adressenübersetzungs- und Zwischenspeicherungsfunktionen ausführen. Beispielsweise kann der L1-TLB 128 Zuordnungen virtueller Adressen zu physikalischen Adressen zwischenspeichern. Wenn eine Speicherzugriffsanforderung auftritt, wird der L1-TLB 128 überprüft, um zu erkennen, ob eine Zuordnung der gewünschten virtuellen Adresse zu einer physikalischen Adresse abgespeichert ist. Zuordnungen, die in dem L1-TLB 128 abgelegt sind, können verwendet werden, um zu bestimmen, ob eine gewünschte Cache-Zeile in dem L1-Daten-Cache-Speicher 126 vorhanden ist. Wenn eine gewünschte Cache-Zeile nicht im L1-Daten-Cache-Speicher 126 vorhanden ist, d. h. es gibt einen L1-Cache-Fehlertreffer, wird der L2-TLB1 überprüft, um zu erkennen, ob eine Zuordnung der gewünschten virtuellen Adresse zu einer physikalischen Adresse gespeichert ist. Zuordnungen, die in dem L2-TLB 140 abgelegt sind, können verwendet werden, um zu bestimmen, ob eine gewünschte Cache-Zeile im L2-Daten-Cache-Speicher 130 vorhanden ist. Wenn ein Cache-Fehlertreffer im L1-Daten-Cache-Speicher 126 auftritt, wird, um Platz für einen neuen Eintrag zu schaffen, eine Cache-Zeile aus dem L1-Daten-Cache-Speicher 126 in den L2-Daten-Cache-Speicher 130 ausgelagert. Ein entsprechender Eintrag in dem L1-TLB 128 wird in den L2-TLB 140 verschoben. Um für einen neuen Eintrag in dem L2-Daten-Cache-Speicher 130 Platz zu schaffen, kann es erforderlich sein, eine Cache-Zeile von dem L2-Daten-Cache-Speicher 130 in den Hauptspeicher 150 zu verschieben. Es wird unter Umständen eine neue Adressenzuordnung bzw. Übersetzung für gewünschte Cache-Zeile ausgeführt und das Ergebnis wird im L1 TLB 128 abgelegt, wobei die-

ser Vorgang als eine TLB-Wiederauffüllung bezeichnet werden kann. Weitere Details und Funktionsweisen dieser Daten-Cache-Speicher 126 und 120 und der TLB's 128 und 140 im Hinblick auf das Vermeiden einer Schädigung des Mikrocodierungsemulationsspeicherraums 135 werden im Weiteren angegeben.

[0014] Fig. 2 zeigt eine Ausführungsform einer virtuellen Speicher- und Cache-Architektur, die in dem Prozessor) 10 eingesetzt werden kann. In der Darstellung ist ein virtueller Speicherraum 210 gezeigt, wovon Bereiche einem physikalischen Speicheradressenraum 220 zugeordnet sind. Bereiche des physikalischen Speicheradressenraumes 220 sind so gezeigt, dass sie dem L2-Cache-Speicherraum 230 zugeordnet sind, von welchem Bereiche wiederum dem L1-Cache-Speicherraum 240 zugeordnet sind. Jede Anwendung, die in dem Prozessor 110 abgearbeitet wird, kann einen separaten virtuellen Speicheradressenraum verwenden. Der virtuelle Speicheradressenraum 210, der in Fig. 2 gezeigt ist, enthält Blöcke 211 bis 215, die Bereiche des virtuellen Speichers repräsentieren, die dem physikalischen Speicheradressenraum 220 zugeordnet sind und die für den Zugriff durch eine Anwendung zu einem gegebenen Zeitpunkt verfügbar sind. In ähnlicher Weise enthält der physikalische Speicheradressenraum 220 Blöcke 221 bis 224, die die Bereiche an physikalischem Speicher repräsentieren, die in dem L2-Cache-Speicherraum 230 zwischengespeichert sind. In ähnlicher Weise enthält der L2-Cache-Speicherraum 230 Blöcke 231 bis 233, die die Bereiche des L2-Cache-Speichers repräsentieren, die in dem L1-Cache-Speicherraum 240 abgelegt sind. Insbesondere sind die Blöcke 231, 232 und 233 des L2-Cache-Speicherraums 230 entsprechend den Blöcken 242, 243 und 241 des Cache-Speicherraums 240 zugeordnet. In den diversen Ausführungsformen kann jeder zuvor beschriebene Block eine Gruppe aus Cache-Zeilen, Blöcke mit gleicher Größe, eine Gruppe aus Cache-Zeilen oder Blöcke mit variierender Größe repräsentieren. In alternativen Ausführungsformen können jeweils der virtuelle Speicheradressenraum 120, der physikalische Speicheradressenraum 220, der L2-Cache-Speicher 230 und der L1-Cache-Speicher 240 mehr oder weniger Blöcke aufweisen, als dies in Fig. 2 gezeigt ist.

[0015] In einer Ausführungsform wird der Block 221 in dem physikalischen Speicherraum 220 als ein Mikrocodierungsemulationsspeicher reserviert. Des weiteren kann der Block 231 des L2-Cache-Speicherraums 230 permanent für das Speichern des Inhalts des Mikrocodierungsemulationsspeichers reserviert werden. Während des Betriebs wird, wenn der Prozessor 110 einen Zugriff auf den Mikrocodierungsemulationsspeicher fordert, der Block 231 in dem Cache-Speicher der Ebene 1 zwischengespeichert, etwa im Block 242, wie dies in Fig. 2 gezeigt ist. Jedoch

ist der Block **242** ggf. nicht permanent für die Verwendung als Mikrocodierungsemulationsspeicher reserviert, wie dies für Block **231** der Fall ist. Die Blöcke, die im L1 zwischengespeichert sind, können sich von Zeit zu Zeit abhängig von der Programmausführung ändern. Folglich kann der Mikrocodierungsemulationsspeicher aus dem L1 in den L2 verschoben werden, in welchem der Block **231** für die Verwendung reserviert ist. In einer Ausführungsform wird ein Zugriff auf den Mikrocodierungsemulationsspeicher durch Anwendungen oder Prozesse, die keine Mikrocodierungen sind, verhindert, indem eine TLB-Wiederauffüllung des L1 unter Beteiligung des Blocks **221** des physikalischen Speicherraums nicht gestattet wird.

[0016] Fig. 3 zeigt eine Ausführungsform eines Prozessors **300** zum Zugreifen auf eine Speicherhierarchie, die den Mikrocodierungsemulationsspeicher enthält. Ein Speicherzugriff kann mit einer Prüfung hinsichtlich der Anwesenheit eines Mikrocodierungszugriffssignals (nicht gezeigt) beginnen, die mit jedem Befehl verknüpft ist, der von einer Ausführungseinheit decodiert wird (Entscheidungsblock **310**). Beispielsweise wird in einer Ausführungsform ein Bit jedes decodierten Befehls als ein Mikrocodierungszugriffssignal verwendet. In einer alternativen Ausführungsform besitzen Mikrocodierungsbefehle eine spezielle Op-Codierung bzw. Operationscodierung, die als ein Mikrocodierungssignal dient und mit dieser als Mikrocodierung erkannt werden kann. Es kann jedes aus einer Vielzahl alternativer Mikrocodierungszugriffssignale von einer Ausführungseinheit zu einer Cache-Steuerung übermittelt werden, um anzugeben, ob ein Befehl ein Mikrocodierungsbefehl ist oder nicht. Wenn ein Mikrocodierungszugriffssignal erkannt wird, wird ein Zugriff auf den Mikrocodierungsemulationsspeicher zugelassen (Block **320**) und der Zugriff wird abgeschlossen.

[0017] Wenn das Mikrocodierungszugriffssignal nicht erkannt wird, geht der Prozess **300** in der folgenden Weise weiter. Es werden ein oder mehrere TLB's abgesucht, um einen Eintrag zu finden, der mit der von dem Zugriff betroffenen Cache-Zeile übereinstimmt (Block **330**). Wenn ein übereinstimmender Eintrag in einem L1-TLB gefunden wird (Entscheidungsblock **340**), dann wird auf die angezielte Cache-Zeile zugegriffen (Block **390**) und der Zugriff wird abgeschlossen. Wenn ein übereinstimmender Eintrag nicht in einem L1-TLB nicht gefunden wird, und dieser in einem L2-TLB gefunden wird (Entscheidungsblock **340**), dann wird die angesprochene Cache-Zeile von dem L2-Cache-Speicher in den L1-Cache-Speicher verschoben (Block **360**), es wird auf die gezielte Cache-Zeile zugegriffen (Block **390**) und der Zugriff wird abgeschlossen. Wenn ein übereinstimmender Eintrag weder im L1 noch im L2-Cache-Speicher gefunden wird, dann wird eine Adressenübersetzung oder Zuordnung ausgeführt (Block **370**). Wenn das Ergeb-

nis der Adressenübersetzung eine Zieladresse liefert, die in dem Mikrocodierungsemulationsspeicher liegt (Entscheidungsblock **380**), dann wird ein Zugriff verhindert (Block **380**), wobei der Zugriffsversuch dann beendet wird. Wenn das Ergebnis der Adressenübersetzung eine Zieladresse ist, die nicht in dem Mikrocodierungsemulationsspeicher liegt (Entscheidungsblock **380**), dann wird eine TLB-Wiederauffüllung ausgeführt (Block **382**), es wird auf die angesprochene Cache-Zeile zugegriffen (Block **390**), und der Zugriff wird abgeschlossen.

[0018] Fig. 4 zeigt eine Ausführungsform eines Prozessors **400** zum Zugreifen auf einen Mikrocodierungsemulationsspeicher in einem Cache-Speicher der Ebene 1. Eine Zugriffsanforderung, die auf den Mikrocodierungsemulationsspeicher zielt, beginnt mit einer Prüfung, um zu erkennen, ob die angesprochene Cache-Zeile in einem L1-Cache-Speicher enthalten ist (Entscheidungsblock **410**). Wenn dies der Fall ist, wird der Zugriff auf die angesprochene Cache-Zeile zugelassen (Block **420**) und der Zugriff wird abgeschlossen. Wenn die angesprochene Cache-Zeile nicht in einem L1-Cache-Speicher abgelegt ist, dann wird die reservierte Stelle der angesprochenen Cache-Zeile im L2-Cache-Speicher erhalten (Block **430**). Die angezielte Cache-Zeile wird dann von dem L2-Cache-Speicher in den L1-Cache-Speicher verschoben (Block **440**). Sobald die Ziel-Cache-Zeile in den L1-Cache-Speicher verschoben ist, wird der Zugriff erlaubt (Block **420**) und der Zugriff wird abgeschlossen.

[0019] Fig. 5 ist eine Blockansicht einer Ausführungsform eines Computersystems **500** mit einem L2-Daten-Cache-Speicher **560** und einem Mikrocodierungsemulationsspeicher **135**, die mit mehreren Systemkomponenten gekoppelt sind. In dem dargestellten System ist der Prozessor **510** so gezeigt, dass er mit Peripherieeinrichtungen **510** und einem Speicher **530** verbunden ist. Die Peripherieeinrichtungen **520** umfassen beliebige Einrichtungen, etwa Netzwerkstellen, Zeitgeberschaltungen, Speichermedien, Eingabe/Ausgabe-Einrichtungen, etc., die in einem konventionellen Computersystem vorgefunden werden. Der Speicher **530** umfasst etwa SDRAM-, SRAM-, ROM-, DRAM- und/oder andere konventionelle Speichereinrichtungen. Der Prozessor **510** umfasst Kerne **540a** und **540b**, einen vereinigenden Schreib/Cache-Speicher **550**, einen Daten-Cache-Speicher der Ebene 2 **560** und eine I/O-Schnittstelle **570**. Die I/O-Schnittstelle **570** verbindet etwa alle Kerne **540** mit den Peripherieeinrichtungen **520**. Elemente, die hierin mit einem Bezugszeichen mit einem anschließenden Buchstaben bezeichnet sind, können auch gemeinsam durch lediglich das Bezugszeichen benannt werden. Beispielsweise werden die Kerne **540a** und **540b** als Kerne **540** bezeichnet und ein nicht spezifizierter Kern der Kerne **540** kann einfach als ein Kern **540** bezeichnet werden.

[0020] Jeder Kern **540** enthält eine Daten-Cache-Speicher der Ebene **1 542**, eine Schreiblogikeinheit **544** und eine Lade/Schreibpipeline **546**. Die Schreiblogikeinheit **544** (die alternativ als „Schreibereinheit“ bezeichnet wird) repräsentiert einen Bereich einer Lade/Schreib-Einheit, eine separate Logikeinheit oder eine Kombination davon. Die Schreiblogik **544** ist sowohl mit dem Daten/Cache-Speicher der Ebene **1 542** als auch mit dem vereinigenden Schreib/Cache-Speicher **550** verbunden, um es dem Kern **540** zu ermöglichen, den Speicher jeder Ebene zu beschreiben. Insbesondere überträgt die Schreiblogik **544** Schreiboperationen **584** zu dem Daten-Cache-Speicher **542** der Ebene **1** und überträgt Schreiboperationen **582** zu dem Schreib-Cache-Speicher **550**. Der vereinigende Schreib-Cache-Speicher **550** kann ferner mit einem Daten-Cache-Speicher der Ebene **2 560** mittels Verbindungen für Auffülloperationen **564** und Verbindungen für Verschiebungen **566** gekoppelt sein. Der vereinigende Schreib/Cache-Speicher **550** kann die Schreiboperationen **582** zusammenfügen mit Fülloperationen **564**, um eine geringe Anzahl an Verschiebungen **566** zu erzeugen. Der Daten-Cache-Speicher der Ebene **2 560** ist ferner mit jedem Daten-Cache-Speicher der Ebene **1 542** verbunden. Insbesondere kann der Daten-Cache-Speicher der Ebene **2 560** die Fülloperationen **562** zu dem Daten-Cache-Speicher der Ebene **1 542** übertragen. Der Daten-Cache-Speicher der Ebene **2 560** kann ferner auch bidirektional mit dem Hauptspeicher **530** verbunden sein.

[0021] Während des Betriebs führt der Kern **540** eine Reihe von Befehlen aus, die, wenn sie decodiert sind, Ladeoperationen **586** von dem Daten-Cache-Speicher **L1 542** zur Lade/Schreibpipeline **546** und/oder Schreiboperationen **580** von Lade/Schreib-Pipeline **546** in die Schreiblogik **544** hervorrufen. Die von dem Kern **540** ausgeführten Befehle können auch das Ausführen einer Mikrocodierung beinhalten. Wenn die Ausführung einer Mikrocodierung das Zugreifen auf eine Cache-Zeile in dem Mikrocodierungsemulationsspeicher **135** erfordert, wird auf die angesprochene Cache-Zeile zugegriffen und bei Bedarf wird diese vom L2-Daten-Cache-Speicher **560** zu dem L1-Daten-Cache-Speicher **542** unter Anwendung des in **Fig. 4** vorgeschriebenen Prozesses verschoben. Sobald die angesprochene Cache-Zeile in den L1-Daten-Cache-Speicher **542** verschoben ist, wird auf diese mittels der Ladeoperationen **586** und/oder der Schreiboperationen **580** und **584** zugegriffen. Obwohl das gezeigte System **500** zwei Kerne enthält, sind in alternativen Ausführungsformen mehr als zwei Kerne enthalten und/oder jeder Kern repräsentiert eine Anhäufung aus Ausführungseinheiten. Es können auch zusätzliche Cache-Speicher der Ebene **2** in weiteren alternativen Ausführungsformen vorgesehen sein, in denen mehr als zwei Kerne enthalten sind. Obwohl der Daten-Cache-Speicher der Ebene **2 560** so gezeigt ist, dass dieser direkt mit

dem Speicher **530** gekoppelt ist, und der Speicher **530** als ein Speicher außerhalb des Prozessors gezeigt ist, kann der Prozessor **510** auch eine Speichersteuerung und/oder einen prozessorinternen Speicher umfassen. Ferner kann auch eine prozessor-externe Speichersteuerung den Daten-Cache-Speicher der Ebene **2 560** mit dem Speicher **530** verbinden. Für den Fachmann ergeben sich eine Vielzahl aus Konfigurationen für den Prozessorkern und dem Speicher.

[0022] Zu beachten ist, dass die zuvor beschriebenen Ausführungsformen eine Software enthalten. In einer derartigen Ausführungsform können die Programmbefehle, die die Verfahren und/oder Mechanismen implementieren, von einem computeransprechbaren Medium übertragen oder gespeichert sein. Es sind zahlreiche Arten an Medien verfügbar, die zum Speichern von Programmbefehlen geeignet sind, wozu Festplatten, Laufwerke, CD-ROM-Einrichtungen, DVD, Cache-Speicher, programmierbare ROM's (PROM), Speicher mit wahlfreiem Zugriff (RAM) und diverse andere Formen flüchtiger oder nicht-flüchtiger Speicher gehören. Zu noch weiteren Formen von Medien, die zum Übertragen von Programmbefehlen für den Zugriff durch einen Computer geeignet sind, gehören terrestrische und nicht-terrestrische Kommunikationsverbindungen, etwa Netzwerk, drahtlose Einrichtungen und Satellitenverbindungen, über die elektrische, elektromagnetische, optische oder digitale Signale übertragen werden können. Somit können diverse Ausführungsformen ferner das Empfangen, Senden oder Speichern von Befehlen und/oder von Daten beinhalten, die gemäß der vorhergehenden Beschreibung auf Grund eines durch einen Computer ansprechbaren Medium eingerichtet sind.

Patentansprüche

1. Prozessor (110) mit:
 einer Cache-Speicherhierarchie mit mindestens einem ersten Cache-Speicher (126) der Ebene 1 und einem Cache-Speicher (130) höherer Ebene;
 wobei der Prozessor (110) ausgebildet ist, um:
 einen ersten Bereich eines physikalischen Speicher-
 raumes (220) einem ersten Bereich des Cache-Speichers (130) höherer Ebene zuzuordnen;
 Befehle auszuführen, wovon mindestens einige eine Mikrocodierung enthalten;
 einen ersten Befehl der Befehle zu dekodieren;
 wenn ein Mikrocodierungszugriffssignal in Reaktion auf das Dekodieren des ersten Befehls erfasst wird, zu bestimmen, dass der erste Befehl eine Mikrocodierung darstellt und es dem ersten Befehl zu ermöglichen, auf den ersten Bereich des Cache-Speichers (130) höherer Ebene unter Verwendung von Lade- und Speicherbefehlen zuzugreifen; und
 wenn kein Mikrocodierungszugriffssignal in Reaktion auf das Dekodieren des ersten Befehls erfasst wird,

zu bestimmen, dass der erste Befehl keine Mikrocodierung darstellt und zu verhindern, dass der erste Befehl auf den ersten Bereich des Cache-Speichers (130) höherer Ebene zugreift.

2. Prozessor (110) nach Anspruch 1, wobei der Cache-Speicher (130) höherer Ebene ein Cache-Speicher der Ebene 2 ist.

3. Prozessor (110) nach Anspruch 1, wobei der erste Bereich des physikalischen Speicherraumes (220) permanent für die Verwendung durch die Mikrocodierung reserviert ist.

4. Prozessor (110) nach Anspruch 1, wobei der Prozessor (110) ferner ausgebildet ist, um:
eine oder mehrere Cache-Zeilen des ersten Bereichs des Cache-Speichers (130) höherer Ebene von dem Cache-Speicher höherer Ebene in einen ersten Bereich des ersten Cache-Speichers (126) der Ebene 1 zu verschieben;
es der Mikrocodierung zu ermöglichen, auf den ersten Bereich des ersten Cache-Speichers (126) der Ebene 1 zuzugreifen; und
zu verhindern, dass Befehle, die keine Mikrocodierung enthalten, auf den ersten Bereich des ersten Cache-Speichers (126) der Ebene 1 zugreifen.

5. Prozessor (110) nach Anspruch 1, wobei der Prozessor (110) ausgebildet ist, während des Dekodierens durch Erfassen, dass eines oder mehrere Bits des ersten Befehls einen vorbestimmten Zustand aufweisen, zu bestimmen, dass der erste Befehl eine Mikrocodierung darstellt.

6. Prozessor (110) nach Anspruch 5, der ferner einen Translationsnebenpuffer (TLB) aufweist, wobei der Prozessor (110) zur Verhinderung, dass Befehle, die keine Mikrocodierung enthalten, auf den ersten Bereich des physikalischen Speicherraumes (220) zugreifen, ferner ausgebildet ist, TLB-Wiederauffüllungen in den ersten Bereich des physikalischen Speicherraumes (220) zu verhindern.

7. Prozessor (110) nach Anspruch 1, der ferner mindestens einen ersten Kern und einen zweiten Kern aufweist, wobei der zweite Kern ausgebildet ist, um:
einen zweiten Bereich des physikalischen Speicherraumes (220) einem zweiten Bereich des Cache-Speichers (130) höherer Ebene zuzuordnen;
Befehle auszuführen, wovon mindestens einige eine Mikrocodierung aufweisen;
es der Mikrocodierung zu ermöglichen, auf den zweiten Bereich des Cache-Speichers (130) höherer Ebene zuzugreifen; und
es zu verhindern, dass Befehle, die eine keine Mikrocodierung enthalten, auf den zweiten Bereich des Cache-Speichers (130) höherer Ebene zugreifen.

8. Prozessor (110) nach Anspruch 7, wobei der erste Kern den ersten Cache-Speicher der Ebene 1 enthält und wobei der zweite Kern einen zweiten Cache der Speicherebene 1 enthält; wobei der erste Kern ferner ausgebildet ist, um:

eine oder mehrere Cache-Zeilen, die dem ersten Bereich des Cache-Speichers (130) höherer Ebene zugeordnet sind, von dem Cache-Speicher (130) höherer Ebene in einen Bereich des ersten Cache-Speichers (127) der Ebene 1 zu verschieben;
es der Mikrocodierung zu ermöglichen, auf den Bereich des ersten Cache-Speichers (127) der Ebene 1 zuzugreifen; und

zu verhindern, dass Befehle, die keine Decodierung aufweisen, auf den Bereich des ersten Cache-Speichers (127) der Ebene 1 zugreifen; und

wobei der zweite Kern ferner ausgebildet ist, um:

eine oder mehrere Cache-Zeilen, die dem zweiten Bereich des Cache-Speichers (130) höherer Ebene zugeordnet sind, von dem Cache-Speicher höherer Ebene zu einem Bereich des zweiten Cache-Speichers der Ebene 1 zu verschieben;
es der Mikrocodierung zu ermöglichen, auf den Bereich des zweiten Cache-Speichers der Ebene 1 zuzugreifen; und

zu verhindern, dass Befehle, die keine Mikrocodierung enthalten, auf den Bereich des zweiten Cache-Speichers der Ebene 1 zugreifen; und

zu verhindern, dass Befehle, die keine Mikrocodierung enthalten, auf den Bereich des zweiten Cache-Speichers der Ebene 1 zugreifen.

9. Ein computerimplementiertes Verfahren mit:
Zuordnen eines ersten Bereichs eines physikalischen Speicherraumes (220) zu einem ersten Bereich eines Cache-Speichers (130) höherer Ebene in einer Cache-Speicher-Hierarchie eines Prozessors (110);
Ausführen von Befehlen durch einen ersten Prozessor, wobei mindestens einige Befehle eine Mikrocodierung aufweisen;
Dekodieren eines ersten Befehls der Befehle;

wenn ein Mikrocodierungszugriffssignal in Reaktion auf das Dekodieren des ersten Befehls erfasst wird, Bestimmen, dass der erste Befehl eine Mikrocodierung darstellt und es dem ersten Befehl ermöglichen, auf den ersten Bereich des Cache-Speichers (130) höherer Ebene unter Verwendung von Lade- und Speicherbefehlen zuzugreifen; und

wenn kein Mikrocodierungszugriffssignal in Reaktion auf das Dekodieren des ersten Befehls erfasst wird, Bestimmen, dass der erste Befehl keine Mikrocodierung darstellt und Verhindern, dass der erste Befehl auf den ersten Bereich des Cache-Speichers (130) höherer Ebene zugreift.

10. Verfahren nach Anspruch 9, wobei der Cache-Speicher (130) höherer Ebene ein Cache-Speicher der Ebene 2 ist.

11. Verfahren nach Anspruch 9, das ferner umfasst: permanentes Reservieren des ersten Bereichs des physikalischen Speicherraumes (220) für die Verwendung durch die Mikrocodierung.

12. Verfahren nach Anspruch 9, das ferner umfasst:

Verschieben einer oder mehrerer Cache-Zeilen des ersten Bereichs des Cache-Speichers (130) höherer Ebene von dem Cache-Speicher (139) höherer Ebene zu einem ersten Bereich des ersten Cache-Speichers (127) der Ebene 1 des Prozessors (110);
Zulassen, dass die Mikrocodierung auf den Bereich des ersten Bereichs des ersten Cache-Speichers (127) der Ebene 1 zugreift; und
Verhindern, dass Befehle, die keine Mikrocodierung aufweisen, auf den ersten Bereich des ersten Cache-Speichers (127) der Ebene 1 zugreifen.

auf den Bereich des zweiten Cache-Speichers der Ebene 1 zugreift;

Verhindern, dass Befehle, die keine Mikrocodierung enthalten, auf den Bereich des ersten Cache-Speichers der Ebene 1 zugreifen; und

Verhindern, dass Befehle, die keine Mikrocodierung enthalten, auf den Bereich des zweiten Cache-Speichers der Ebene 1 zugreifen.

17. Computersystem (100) mit einem Prozessor (110) gemäß einem der Ansprüche 1 bis 8.

Es folgen 5 Seiten Zeichnungen

13. Verfahren nach Anspruch 9, das ferner umfasst:

während des Dekodierens durch Erfassen, dass eines oder mehrere Bits des ersten Befehls einen vorbestimmten Zustand aufweisen, Bestimmen, dass der erste Befehl eine Mikrocodierung darstellt.

14. Verfahren nach Anspruch 13, wobei Verhindern, dass Befehle auf den ersten Bereich des physikalischen Speicherraumes (220) zugreifen, wenn das Mikrocodierungssignal nicht gesetzt ist, ferner umfasst: Verhindern von Wiederauffülloperationen an dem ersten Bereich des physikalischen Speicherraumes (220) aus einem Translationsnebenpuffer (TLB).

15. Verfahren nach Anspruch 9, das ferner umfasst:

Zuordnen eines zweiten Bereichs des physikalischen Speicherbereiches zu einem zweiten Bereich des Cache-Speichers (130) höherer Ebene;

Ausführen von Befehlen, wovon zumindest einige eine Mikrocodierung aufweisen, durch einen zweiten Prozessorkern;

Zulassen durch den zweiten Prozessorkern, dass Mikrocodierung auf den zweiten Bereich des Cache-Speichers (130) höherer Ebene zugreift; und

Verhindern durch den zweiten Prozessorkern, dass Befehle, die keine Mikrocodierung enthalten, auf den zweiten Bereich des Cache-Speichers (130) höherer Ebene zugreifen.

16. Verfahren nach Anspruch 15, das ferner umfasst:

Verschieben einer oder mehrerer Cache-Zeilen, die dem ersten Bereich des Cache-Speichers (130) höherer Ebene zugeordnet sind, von dem Cache-Speicher (130) höherer Ebene zu einem Bereich eines ersten Cache-Speichers der Ebene 1;

Verschieben einer oder mehrerer Cache-Zeilen, die dem zweiten Bereich des Cache-Speichers (130) höherer Ebene zugeordnet sind, von dem Cache-Speicher (130) höherer Ebene zu einem Bereich eines zweiten Cache-Speichers der Ebene 1;

Zulassen, dass Mikrocodierung auf den Bereich des ersten Cache-Speichers (127) der Ebene 1 und/oder

Anhängende Zeichnungen

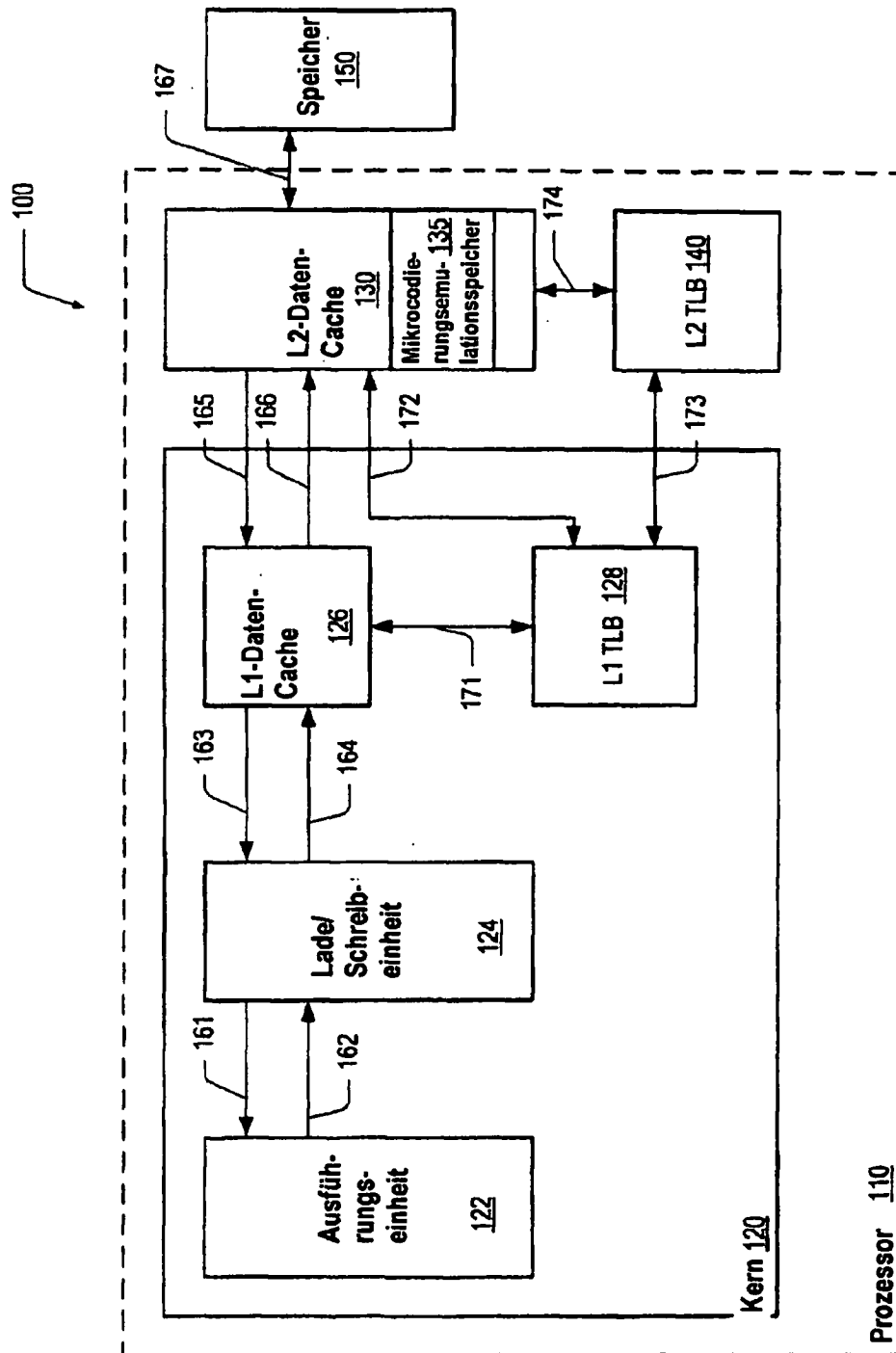


FIG. 1

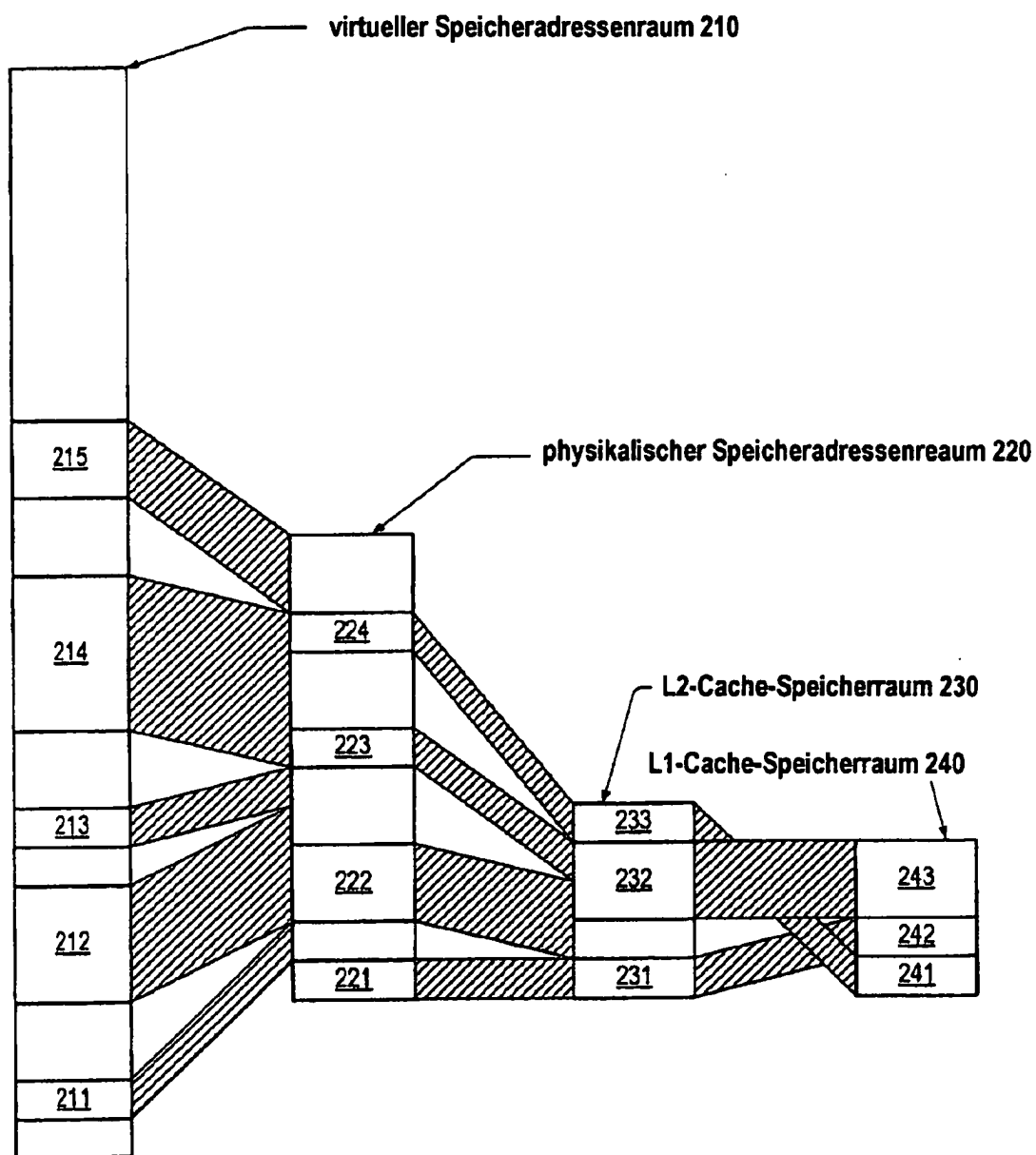


FIG. 2

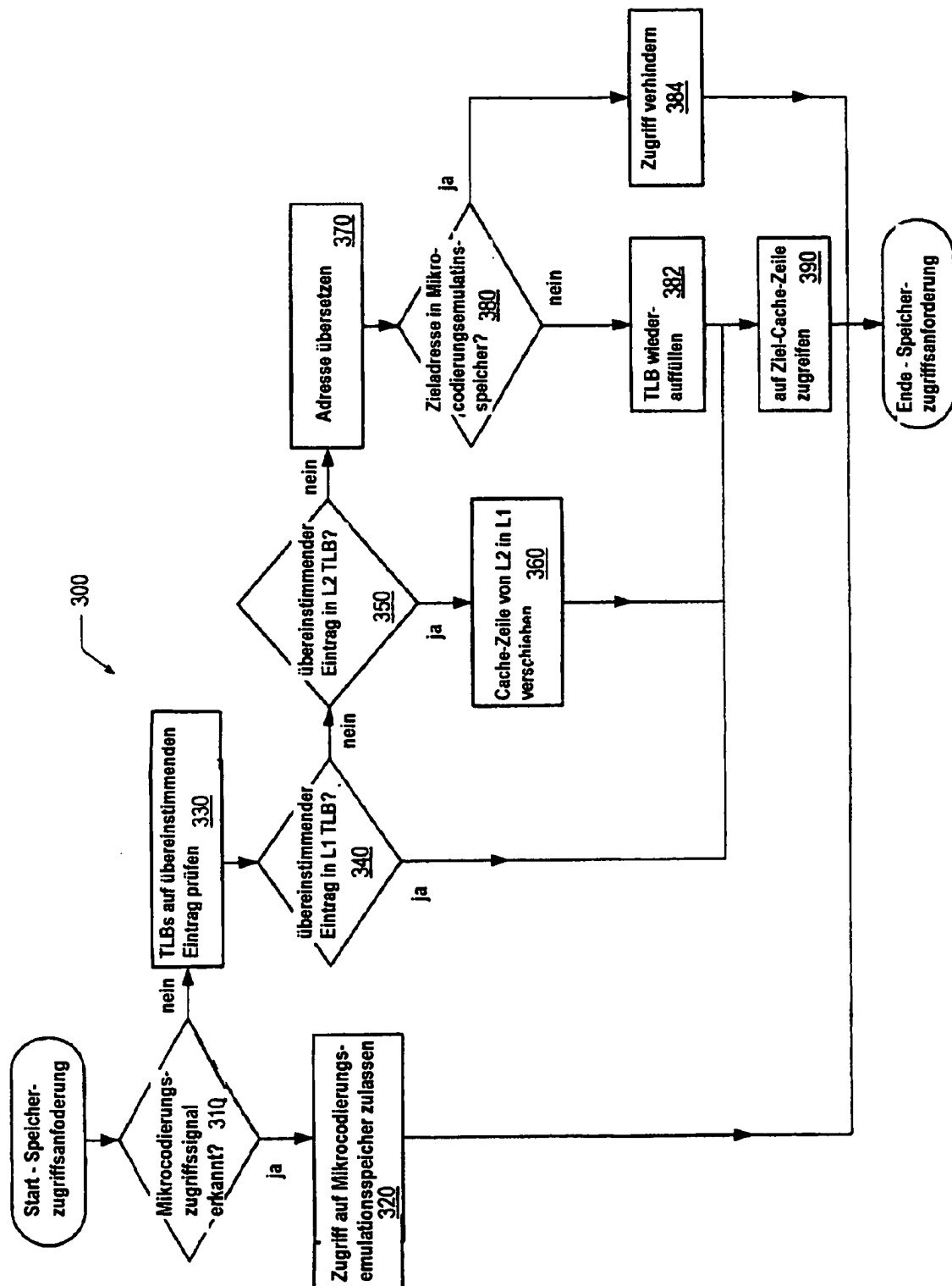


FIG. 3

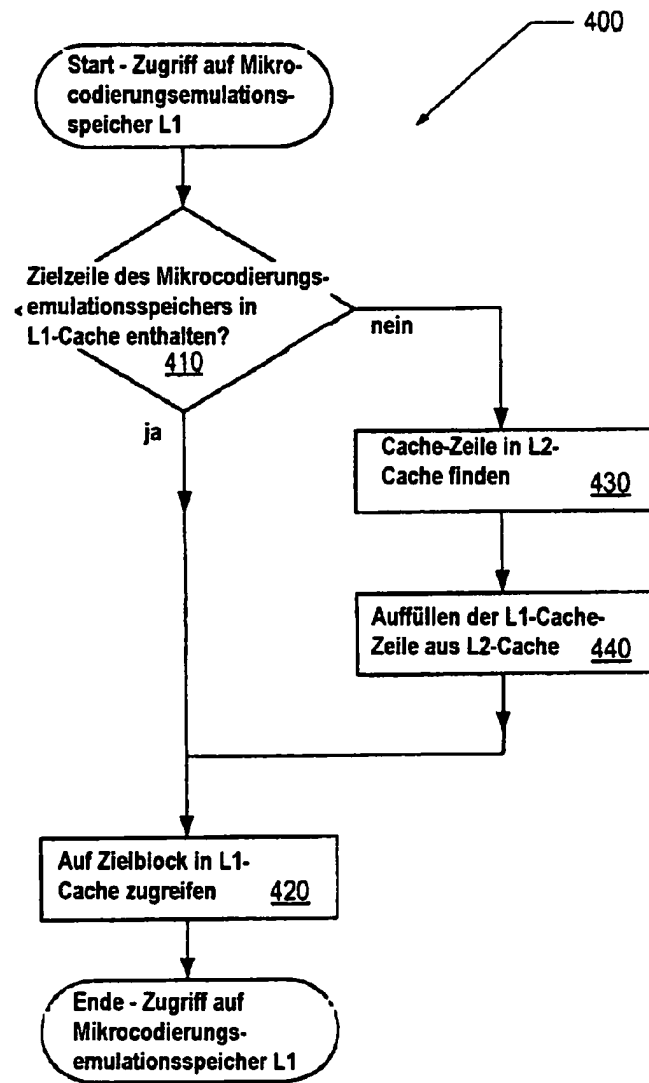


FIG. 4

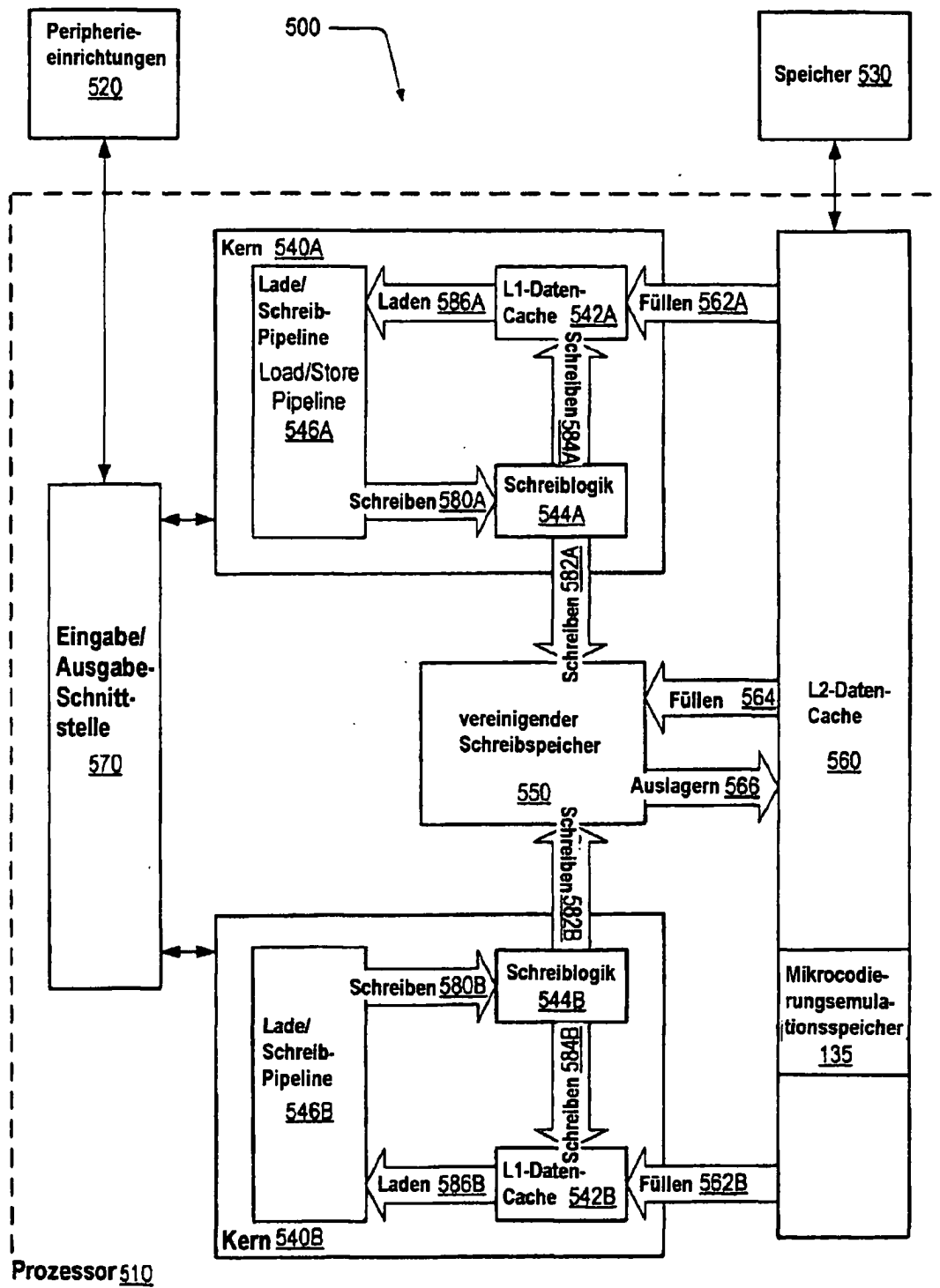


FIG. 5