



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 600 09 151 T2 2004.11.11**

(12)

Übersetzung der europäischen Patentschrift

(97) **EP 1 244 961 B1**

(51) Int Cl.⁷: **G06F 9/38**

(21) Deutsches Aktenzeichen: **600 09 151.1**

(86) PCT-Aktenzeichen: **PCT/US00/21752**

(96) Europäisches Aktenzeichen: **00 951 015.7**

(87) PCT-Veröffentlichungs-Nr.: **WO 01/050252**

(86) PCT-Anmeldetag: **08.08.2000**

(87) Veröffentlichungstag
der PCT-Anmeldung: **12.07.2001**

(97) Erstveröffentlichung durch das EPA: **02.10.2002**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **17.03.2004**

(47) Veröffentlichungstag im Patentblatt: **11.11.2004**

(30) Unionspriorität:

476937 03.01.2000 US

(84) Benannte Vertragsstaaten:

DE, GB

(73) Patentinhaber:

**Advanced Micro Devices, Inc., Sunnyvale, Calif.,
US**

(72) Erfinder:

**KELLER, B., James, Palo Alto, US; GREEN, S.,
Thomas, Sunnyvale, US; LIEN, Wei-Han,
Cupertino, US; HADDAD, W., Ramsey, Cupertino,
US**

(74) Vertreter:

**Patentanwälte von Kreisler, Selting, Werner et col.,
50667 Köln**

(54) Bezeichnung: **VORHERSAGE VON DATENBEFÖRDERUNG VON SPEICHER- ZUM LADEBEFEHL MIT UNTRAI-
NIERUNG**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung**HINTERGRUND DER ERFINDUNG****1. Technisches Gebiet**

[0001] Die vorliegende Erfindung betrifft das Gebiet der Prozessoren und insbesondere Speicher-zu-Lade-Weiterleitungsmechanismen innerhalb von Prozessoren.

2. Stand der Technik

[0002] Prozessoren weisen häufig Speicher-Warteschlangen zum Puffern von Speicher-Abspeicher-Operationen auf, die ausgeführt worden sind, jedoch immer noch spekulativ sind. Die Speicher-Abspeicher-Operationen können in der Speicher-Warteschlange gehalten werden, bis sie zurückgezogen werden. Im Anschluss an das Zurückziehen können die Speicher-Abspeicher-Operationen in den Cache-Speicher und/oder Speicher geleitet werden. Wie hier verwendet, ist eine Abspeicher-Operation eine Operation, die einen Datentransfer zwischen einem Prozessor und einem Hauptspeicher spezifiziert (obwohl der Transfer in dem Cachespeicher beendet sein kann). Speicher-Lade-Speicher-Operationen spezifizieren einen Datentransfer von einem Speicher zu dem Prozessor, und Speicher-Abspeicher-Operationen spezifizieren einen Datentransfer von dem Prozessor zu einem Speicher. Abspeicher-Operationen können impliziter Teil eines Befehls sein, der eine Abspeicher-Operation enthält, oder können explizite Lade-/Speicherbefehle sein. Speicher-Lade-Speicher-Operationen können hier als kürzer als "Ladungen" bezeichnet werden. Ähnlich können Speicher-Abspeicher-Operationen kürzer als "Speicherungen" bezeichnet werden.

[0003] Während eine spekulative Ausführung von Speicherungen und Einreihungen von Speicherungen in die Speicher-Warteschlange eine Erhöhung der Leistungsfähigkeit ermöglichen kann (durch Entfernen von Speicherungen aus der Befehlsausführungs-Pipeline und Ermöglichen der Ausführung weiterer nachfolgender Befehle), können nachfolgende Ladungen auf von den Speicherungen in der Speicher-Warteschlange aktualisierte Speicherstellen zugreifen. Während die Leistungsfähigkeit des Prozessors nicht unbedingt direkt durch in der Speicher-Warteschlange eingereihte Speicherungen beeinträchtigt wird, kann die Leistungsfähigkeit beeinträchtigt werden, wenn nachfolgende Ladungen aufgrund des Zugriffs auf von den Speicherungen in der Speicher-Warteschlange aktualisierte Speicherstellen verzögert werden. Häufig sind Speicher-Warteschlangen derart konfiguriert, dass sie in diesen gespeicherten Daten weiterleiten, wenn eine Ladung die Speicher-Warteschlange trifft. Wie hier verwendet, wird eine Speicher-Abspeicher-Operation spei-

chernde Speicher-Warteschlangen-Eintrag als von einer Speicher-Lade-Speicher-Operation "getroffen" bezeichnet, wenn ein Zugriff durch die Speicher-Lade-Speicher-Operation auf mindestens ein von der Speicher-Abspeicher-Operation aktualisiertes Byte erfolgt.

[0004] Zur weiteren Erhöhung der Leistungsfähigkeit ist es wünschenswert, jüngere Ladungen relativ zu älteren Speicherungen ungeregelt auszuführen. Es ist möglich, dass die jüngeren Ladungen häufig nicht von den älteren Speicherungen abhängig sind und somit nicht die Ausführung der älteren Speicherungen abwarten müssen. Da die Ladungen Operanden für die Ausführung abhängiger Befehle liefern, ermöglicht die Ausführung von Ladungen die Ausführung weiterer Befehle. Das bloße Detektieren von Treffern in der Speicher-Warteschlange bei der Ausführung von Ladungen führt jedoch möglicherweise nicht zu einer korrekten Programmausführung, wenn jüngere Ladungen relativ zu älteren Speicherungen ungeregelt ausgeführt werden können, da bestimmte ältere Speicherungen möglicherweise noch nicht ausgeführt sind (und somit die Speicheradressen dieser Speicherungen möglicherweise nicht bekannt sind und Abhängigkeiten der Ladungen von bestimmten älteren Speicherungen möglicherweise bei der Ausführung der Ladungen nicht detektierbar sind). Entsprechend kann eine Hardware zum Detektieren von Szenarien, bei denen eine jüngere Ladung vor einer älteren Speicherung ausgeführt wird, von der diese jüngere Ladung abhängig ist, erforderlich sein, und dann können in Reaktion auf die Detektierung Korrekturmaßnahmen getroffen werden. Beispielsweise können Befehle eliminiert und wiederabgerufen oder auf eine andere geeignete Weise wiederausgeführt werden. Wie hier verwendet, ist eine Ladung von einer Speicherung "abhängig", wenn die Speicherung mindestens ein Byte des Speichers, auf den die Ladung zugreift, aktualisiert und die Speicherung älter als die Ladung und jünger als die anderen Speicherungen ist, die dieses Byte aktualisieren. Leider können eine unkorrekte ungeregelte Ausführung der Ladung und anschließende Korrekturmaßnahmen zur Erreichung einer korrekten Ausführung die Leistungsfähigkeit reduzieren.

[0005] Es sei darauf hingewiesen, dass Ladungen, Speicherungen und andere Befehlsoperationen hier als älter oder jünger als andere Befehlsoperationen bezeichnet werden können. Ein erster Befehl ist älter als ein zweiter Befehl, wenn der erste Befehl dem zweiten Befehl in der Programmreihenfolge (d. h. der Reihenfolge der Befehle in der Programmausführung) vorangeht. Ein erster Befehl ist jünger als ein zweiter Befehl, wenn der erste Befehl dem zweiten Befehl in der Programmreihenfolge folgt.

[0006] In EP-A-0,709,770 ist eine Vorrichtung zum Steuern von Lade-/Speicher-Operationen beschrie-

ben, bei der ein Abspeicher-Sperr-Cachespeicher einen Speicher-Lade-Konflikt anhand von Entwicklungsbits vorhersagt, die anzeigen, dass bei einer vorangegangenen Ausführung ein Lade-Befehl in der Programmreihenfolge vor einem Speicher-Befehl ausgeführt worden ist, wobei die Lade- und Speicher-Befehle die gleiche reale Adresse haben. Der Abspeicher-Sperr-Cachespeicher speichert Informationen entsprechend den Lade-Abspeicher-Operationen, und ein Treffer in dem Cachespeicher wird dann während des Entsendens des Speicher-Befehls verwendet, um ein Sperrbit zu markieren, das den Ausgang sämtlicher Lade-Befehle verhindert, bis der verletzte Speicher-Befehl die Ausführungsphase der Befehls-Pipeline erfolgreich durchlaufen hat.

[0007] Die oben beschriebenen Probleme werden zum großen Teil mit einem Prozessor nach Anspruch 1 gelöst. Der Prozessor kann generell jüngere Ladungen vor älteren Speicherungen einplanen und/oder ausführen. Ferner kann der Prozessor Szenarien detektieren und Korrekturmaßnahmen hinsichtlich Szenarien ergreifen, bei denen eine ältere Speicherung mit der Ausführung der jüngeren Ladung interferiert. Der Prozessor verwendet eine Speicher-zu-Lade-Weiterleitungs- (STLF-) Vorhersageeinrichtung, die zum Entsenden von Ladungen eine Abhängigkeit von einer Speicherung anzeigen kann. Die Abhängigkeit wird für eine Speicherung angezeigt, die bei einer vorangegangenen Ausführung mit der Ausführung der Ladung interferiert hat. Da eine Abhängigkeit an der Speicherung angezeigt wird, wird verhindert, dass die Ladung vor der Speicherung eingeplant und/oder ausgeführt wird. Die Leistungsfähigkeit kann aufgrund der geringeren Interferenz zwischen Ladungen und Speicherungen erhöht werden.

[0008] Die STLF-Vorhersageeinrichtung wird in Reaktion auf die Ausführung der Ladung und Speicherung und Detektierung der Interferenz einem Training mit Informationen für eine spezielle Ladung und Speicherung unterzogen. Ferner kann die STLF-Vorhersageeinrichtung untrained sein (z. B. können Informationen für eine spezielle Ladung und Speicherung gelöscht sein), wenn eine Ladung von der STLF-Vorhersageeinrichtung als von einer speziellen Speicherung abhängig angezeigt wird und die Abhängigkeit nicht tatsächlich eintritt. Beispielsweise ist die STLF-Vorhersageeinrichtung bei einer Ausführungsform untrained, wenn die Ladung als von einer speziellen Speicherung abhängig angezeigt wird, die Speicherdaten jedoch bei der Ausführung der Ladung nicht von einer Speicher-Warteschlange innerhalb des Prozessors weitergeleitet werden.

[0009] Bei einer Implementierung protokolliert die STLF-Vorhersageeinrichtung mindestens einen Teil der PC einer Speicherung, die mit der Ladung interferiert, in einer ersten Tabelle, welche von der Lade-PC indiziert wird. Eine zweite Tabelle hält einen

entsprechenden Teil der Speicher-PC kürzlich entsendeter Speicherungen zusammen mit Markierungen, die die kürzlich entsendeten Speicherungen identifizieren, aufrecht. Die PC einer Entsende-Ladung wird zum Auswählen einer Speicher-PC aus der ersten Tabelle verwendet. Die ausgewählte Speicher-PC wird mit den in der zweiten Tabelle gespeicherten PCs verglichen. Wenn eine Übereinstimmung detektiert wird, wird die entsprechende Markierung aus der zweiten Tabelle ausgelesen und zum Anzeigen einer Abhängigkeit für die Ladung verwendet.

[0010] Bei einer weiteren Implementierung protokolliert die STLF-Vorhersageeinrichtung eine Differenz zwischen den einer Ladung und einer Speicherung, die mit der Ladung interferiert, zugeordneten Markierungen in einer ersten Tabelle, welche von der Lade-PC indiziert wird. Die PC der Entsende-Ladung wird zum Auswählen einer Differenz aus der Tabelle verwendet, und die Differenz wird der der Ladung zugeordneten Markierung hinzugefügt. Entsprechend kann eine Markierung der Speicherung erzeugt und eine Abhängigkeit der Ladung von der Speicherung angezeigt werden.

[0011] Allgemein gesagt wird davon ausgegangen, dass ein Prozessor eine STLF-Vorhersageeinrichtung und eine mit der STLF-Vorhersageeinrichtung gekoppelte Ausführungs-Pipeline aufweist. Die STLF-Vorhersageeinrichtung ist zum Empfangen einer Anzeige der Entsendung einer ersten Speicher-Lade-Speicher-Operation und zum Ansprechen auf die erste Speicher-Lade-Speicher-Operation zwecks Anzeigen einer Abhängigkeit der ersten Speicher-Lade-Speicher-Operation von einer ersten Speicher-Abspeicher-Operation in Reaktion auf in der STLF-Vorhersageeinrichtung gespeicherte Informationen vorgesehen, wobei die Informationen der Speicher-Lade-Speicher-Operation entsprechen und anzeigen, dass bei einer vorangegangenen Ausführung die erste Speicher-Abspeicher-Operation mit der ersten Speicher-Lade-Speicher-Operation interferiert hat. Die Ausführungs-Pipeline ist zum Sperren der Ausführung der ersten Speicher-Lade-Speicher-Operation vor der ersten Speicher-Abspeicher-Operation in Reaktion auf die Abhängigkeit vorgesehen. Die Ausführungs-Pipeline ist zum Detektieren des Fehlens einer Abhängigkeit bei der Ausführung der ersten Speicher-Lade-Speicher-Operation vorgesehen. Die Ausführungs-Pipeline ist zum Erzeugen eines Untrain-Signals in Reaktion auf das Fehlen einer Abhängigkeit zwecks Aktualisierens der darin gespeicherten Informationen derart vorgesehen, dass nicht angezeigt wird, dass die erste Speicher-Abspeicher-Operation bei der vorangegangenen Ausführung mit der ersten Speicher-Lade-Speicher-Operation interferiert hat. Ferner ist ein Computersystem vorgesehen, das den Prozessor und eine Eingangs-/Ausgangs- (I/O-) Vorrichtung aufweist, die

zur Datenübertragung zwischen dem Computersystem und einem weiteren Computersystem, mit dem die I/O-Vorrichtung koppelbar ist, vorgesehen ist.

[0012] Ferner ist ein Verfahren nach Anspruch 6 vorgesehen. Eine Abhängigkeit einer ersten Speicher-Lade-Speicher-Operation von einer ersten Speicher-Abspeicher-Operation wird in Reaktion auf das Entsenden der ersten Speicher-Lade-Speicher-Operation und in Reaktion auf Informationen, die anzeigen, dass bei einer vorangegangenen Ausführung die erste Speicher-Abspeicher-Operation mit der ersten Speicher-Lade-Speicher-Operation interferiert hat, angezeigt, wobei die Informationen in einer Speicher-zu-Lade-Weiterleitungs(STLF-) Vorhersageeinrichtung gespeichert sind und der ersten Speicher-Lade-Speicher-Operation entsprechen. Ein Einplanen der ersten Speicher-Lade-Speicher-Operation vor dem Einplanen der ersten Speicher-Abspeicher-Operation wird gesperrt. Ein Fehlen der Abhängigkeit wird bei der Ausführung der ersten Speicher-Lade-Speicher-Operation detektiert. Die Informationen, die anzeigen, dass bei der vorangegangenen Ausführung die erste Speicher-Abspeicher-Operation mit der ersten Speicher-Lade-Speicher-Operation interferiert hat, werden derart aktualisiert, dass nicht angezeigt wird, dass bei der vorangegangenen Ausführung die erste Speicher-Abspeicher-Operation mit der ersten Speicher-Lade-Speicher-Operation interferiert hat. Das Aktualisieren erfolgt in Reaktion auf das Detektieren des Fehlens der Abhängigkeit.

[0013] Weitere Aufgaben und Vorteile der Erfindung werden beim Lesen der folgenden detaillierten Beschreibung mit Bezug auf die beiliegenden Zeichnungen offensichtlich. Es zeigen beispielhaft:

[0014] Fig. 1 ein Blockschaltbild einer Ausführungsform eines Prozessors;

[0015] Fig. 2 ein Pipeline-Diagramm einer beispielhaften Pipeline, die in dem in Fig. 1 gezeigten Prozessor verwendet werden kann;

[0016] Fig. 3 ein genaueres Blockschaltbild mit Darstellung einer Ausführungsform einer Abbildungseinheit, eines Schedulers, einer AGU/TLB und einer Lade-/Speichereinheit;

[0017] Fig. 4 ein Blockschaltbild einer Ausführungsform einer in Fig. 3 gezeigten Speicher-zu-Lade-Weiterleitungs- (STLF-) Vorhersageeinrichtung;

[0018] Fig. 5 ein Blockschaltbild einer zweiten Ausführungsform einer in Fig. 3 gezeigten STLF-Vorhersageeinrichtung;

[0019] Fig. 6 ein Ablaufdiagramm mit Darstellung von Training- und Untraining-Vorgängen bezüglich Ladungen bei einer Ausführungsform einer in Fig. 4

oder 5 gezeigten STLF-Vorhersageeinrichtung;

[0020] Fig. 7 ein Blockschaltbild mit Darstellung einer Ausführungsform einer Steuerschaltung, die in einer in Fig. 4 oder 5 gezeigten STLF-Vorhersageeinrichtung verwendet werden kann;

[0021] Fig. 8 ein Blockschaltbild einer Ausführungsform einer in Fig. 3 gezeigten Abhängigkeitseinheit;

[0022] Fig. 9 ein Blockschaltbild einer Ausführungsform eines Computersystems mit dem in Fig. 1 gezeigten Prozessor;

[0023] Fig. 10 ein Blockschaltbild einer zweiten Ausführungsform eines Computersystems mit dem in Fig. 1 gezeigten Prozessor.

ART(EN) ZUM DURCHFÜHREN DER ERFINDUNG

Überblick über den Prozessor

[0024] Fig. 1 zeigt ein Blockschaltbild einer Ausführungsform eines Prozessors **10**. Andere Ausführungsformen sind möglich und werden in Betracht gezogen. Bei der in Fig. 1 gezeigten Ausführungsform weist der Prozessor **10** auf: eine Zeilenvorhersageeinrichtung **12**, einen Befehls-Cachespeicher (I-Cachespeicher) **14**, eine Ausrichteinheit **16**, eine Verzweigungsvorhersage-/abruf-PC-Erzeugungseinheit **18**, mehrere Dekodiereinheiten **24A–24D**, eine Vorhersage-Nichttreffer-Dekodiereinheit **26**, eine Mikrocode-Einheit **28**, eine Abbildungseinheit **30**, eine Zurückzieh-Warteschlange **32**, eine Umbenennungsdateiarchitektur **34**, eine Zukunfts-Datei **20**, einen Scheduler **36**, eine Ganzzahlen-Registerdatei **38A**, eine Gleitkomma-Registerdatei **38B**, einen Ganzzahlen-Ausführungskern **40A**, einen Gleitkomma-Ausführungskern **40B**, eine Lade-/Speichereinheit **42**, einen Daten-Cachespeicher (D-Cachespeicher) **44**, eine Extern-Interface-Einheit **46** und ein PC-Silo **48**. Die Zeilenvorhersageeinrichtung **12** ist mit der Vorhersage-Nichttreffer-Dekodiereinheit **26**, der Verzweigungsvorhersage-/abruf-PC-Erzeugungseinheit **18**, dem PC-Silo **48** und der Ausrichteinheit **16** gekoppelt. Die Zeilenvorhersageeinrichtung **12** kann ferner mit dem I-Cachespeicher **14** gekoppelt sein. Der I-Cachespeicher **14** ist mit der Ausrichteinheit **16** und der Verzweigungsvorhersage-/abruf-PC-Erzeugungseinheit **18** gekoppelt, welche ferner mit dem PC-Silo **48** gekoppelt ist. Die Ausrichteinheit **16** ist ferner mit der Vorhersage-Nichttreffer-Dekodiereinheit **26** und den Dekodiereinheiten **24A–24D** gekoppelt. Die Dekodiereinheiten **24A–24D** sind ferner mit der Abbildungseinheit **30** gekoppelt, und die Dekodiereinheit **24D** ist mit der Mikrocode-Einheit **28** gekoppelt. Die Abbildungseinheit **30** ist mit der Zurückzieh-Warteschlange **32** (welche mit der Umbenennungsdateiarchitektur **34** gekoppelt ist), der Zukunfts-Datei **20**, dem Scheduler **36** und dem PC-Silo

48 gekoppelt. Die Umbenennungsdateiarchitektur **34** ist mit der Zukunftsdatei **20** gekoppelt. Der Scheduler **36** ist mit den Registerdateien **38A–38B** gekoppelt, welche ferner miteinander und mit jeweiligen Ausführungskernen **40A–40B** gekoppelt sind. Die Ausführungskerne **40A–40B** sind ferner mit der Lade-/Speichereinheit **42** und dem Scheduler **36** gekoppelt. Der Ausführungskern **40A** ist ferner mit dem D-Cachespeicher **44** gekoppelt. Die Lade-/Speichereinheit **42** ist mit dem Scheduler **36**, dem D-Cachespeicher **44** und der Extern-Interface-Einheit **46** gekoppelt. Der D-Cachespeicher **44** ist mit den Registerdateien **38** gekoppelt. Die Extern-Interface-Einheit **46** ist mit einem Extern-Interface **52** und mit dem I-Cachespeicher **14** gekoppelt. Elemente, die hier mit einem von einem Buchstaben gefolgt Bezugszeichen versehen sind, werden kollektiv nur mit dem Bezugszeichen bezeichnet. Beispielsweise werden die Dekodiereinheiten **24A–24D** kollektiv als Dekodiereinheiten **24** bezeichnet.

[0025] Bei der in **Fig. 1** gezeigten Ausführungsform verwendet der Prozessor **10** eine Komplexbefehlsatzberechnungs- (CISC-) Befehlsatzarchitektur mit variabler Bytelänge. Beispielsweise kann der Prozessor **10** die x86-Befehlsatzarchitektur (die auch als IA-32 bezeichnet wird) verwenden. Bei anderen Ausführungsformen können andere Befehlsatzarchitekturen verwendet werden, einschließlich Befehlsatzarchitekturen mit fester Länge und Befehlsatzarchitekturen für die Berechnung mit vermindertem Befehlsvorrat (RISC). Bestimmte in **Fig. 1** gezeigte Merkmale können in solchen Architekturen wegfallen.

[0026] Die Verzweigungsvorhersage-/abruf-PC-Erzeugungseinheit **18** ist zum Liefern einer Abrufadressen (Abruf-PC) zu dem I-Cachespeicher **14**, der Zeilenvorhersageeinrichtung **12** und dem PC-Silo **48** vorgesehen. Die Verzweigungsvorhersage-/abruf-PC-Erzeugungseinheit **18** kann einen geeigneten Verzweigungsvorhersagemechanismus zum Unterstützen der Erzeugung von Abrufadressen aufweisen. In Reaktion auf die Abrufadresse liefert die Zeilenvorhersageeinrichtung **12** Ausrichtinformationen entsprechend mehreren Befehlen an die Ausrichteinheit **16**, und sie kann eine nächste Abrufadresse zum Abrufen von Befehlen im Anschluss an die von den gelieferten Befehlsadressen identifizierten Befehle liefern. Die nächste Abrufadresse kann zu der Verzweigungsvorhersage-/abruf-PC-Erzeugungseinheit **18** oder, falls gewünscht, direkt zu dem I-Cachespeicher **14** geliefert werden. Die Verzweigungsvorhersage-/abruf-PC-Erzeugungseinheit **18** kann eine Trap-Adresse von dem PC-Silo **48** empfangen (falls ein Trap detektiert wird), und die Trap-Adresse kann die von der Verzweigungsvorhersage-/abruf-PC-Erzeugungseinheit **18** erzeugte Abruf-PC enthalten. Andernfalls kann die Abruf-PC unter Verwendung der Verzweigungsvorhersageinformationen und Informa-

tionen von der Zeilenvorhersageeinrichtung **12** erzeugt werden. Im wesentlichen speichert die Zeilenvorhersageeinrichtung **12** Informationen entsprechend zuvor von dem Prozessor **10** spekulativ abgerufenen Befehlen. Bei einer Ausführungsform weist die Zeilen-Vorhersageeinrichtung **12** 2K Einträge auf, von denen jede eine Gruppe von einem oder mehreren Befehlen definiert, die hier als "Zeile" von Befehlen bezeichnet wird. Die Zeile von Befehlen kann gleichzeitig dadurch von der Befehlsverarbeitungs-Pipeline des Prozessors **10** verarbeitet werden, dass sie in den Scheduler **36** platziert wird.

[0027] Der I-Cachespeicher **14** ist ein Hochgeschwindigkeits-Cachespeicher zum Speichern von Befehlsbytes. Gemäß einer Ausführungsform kann der I-Cachespeicher **14** beispielsweise eine 128-KByte-Vier-Wege-Satzassoziativ-Organisation aufweisen, die 64-Byte-Cachespeicherzeilen verwendet. Es kann jedoch eine beliebige I-Cachespeicherstruktur geeignet sein (einschließlich Direktabbildungsstrukturen).

[0028] Die Ausrichteinheit **16** empfängt die Befehlsausrichtinformationen von der Zeilenvorhersageeinrichtung **12** und Befehlsbytes entsprechend der Abrufadresse vom I-Cachespeicher **14**. Die Ausrichteinheit **16** wählt Befehlsbytes in jede Dekodiereinheit **24A–24D** entsprechend den gelieferten Befehlsausrichtinformationen. Insbesondere liefert die Zeilenvorhersageeinrichtung **12** einen Befehlszeiger entsprechend jeder Dekodiereinheit **24A–24D**. Der Befehlszeiger lokalisiert einen Befehl innerhalb der abgerufenen Befehlsbytes zum Transportieren zu der entsprechenden Dekodiereinheit **24A–24D**. Bei einer Ausführungsform können bestimmte Befehle zu mehr als einer Dekodiereinheit **24A–24D** transportiert werden. Entsprechend kann bei der dargestellten Ausführungsform eine Zeile von Befehlen von der Zeilenvorhersageeinrichtung **12** bis zu 4 Befehlen enthalten, obwohl bei anderen Ausführungsformen mehr oder weniger Dekodiereinheiten 24 zum Erzeugen von mehr oder weniger Befehlen innerhalb einer Zeile vorgesehen sein können.

[0029] Die Dekodiereinheiten **24A–24D** dekodieren die ihnen zugeführten Befehle, und jede Dekodiereinheit **24A–24D** erzeugt Informationen, die eine oder mehrere Befehlsoperationen (oder ROPs) entsprechend den Befehlen identifizieren. Bei einer Ausführungsform kann jede Dekodiereinheit **24A–24D** bis zu zwei Befehlsoperationen pro Befehl erzeugen. Wie hier verwendet, ist eine Befehlsoperation (oder ROP) eine Operation, die zum Ausführen als einzelne Entität durch eine Ausführungseinheit innerhalb der Ausführungskerne **40A–40B** vorgesehen ist. Einfache Befehle können einer einzelnen Befehlsoperation entsprechen, während komplexere Befehle mehreren Befehlsoperationen entsprechen können. Bestimmte der komplexeren Befehle können als Mikro-

routinen (bei der vorliegenden Ausführungsform über die Dekodiereinheit **24D** aus einem Festwertspeicher abgerufen) in der Mikrocodeeinheit **28** implementiert sein. Ferner kann bei weiteren Ausführungsformen eine einzelne Befehlsoperation für jeden Befehl verwendet werden (d. h. Befehl und Befehlsoperation können bei solchen Ausführungsformen synonym sein).

[0030] Das PC-Silo **48** speichert die Abrufadresse und Befehlsinformationen für jeden Befehlsabruf und ist verantwortlich für das Umleiten des Befehlsabrufs in Ausnahmefällen (wie z. B. von der von dem Prozessor **10** verwendeten Befehlssatzarchitektur definierte Befehls-Traps, Verzweigungs-Falschvorhersagen und andere mikroarchitektonisch definierte Traps). Das PC-Silo **48** kann einen Kreispuffer zum Speichern der Abrufadresse und Befehlsinformationen entsprechend mehreren Zeilen von Befehlen aufweisen, die innerhalb des Prozessors **10** ausstehen können. In Reaktion auf das Zurückziehen einer Zeile von Befehlen kann das PC-Silo **48** den entsprechenden Eintrag streichen. In Reaktion auf eine Ausnahme kann das PC-Silo **48** eine Trap-Adresse zu einer Verzweigungsvorhersage-/abruf-PC-Erzeugungseinheit **18** liefern. Zurückzieh- und Ausnahmeinformationen können von dem Scheduler **36** erzeugt werden. Bei einer Ausführungsform weist die Abbildungseinheit **30** jedem Befehl eine Sequenznummer (R#) zu, um die Reihenfolge der innerhalb des Prozessors **10** ausstehenden Befehle zu identifizieren. Der Scheduler **36** kann R#s zu dem PC-Silo **48** zurückführen, um Befehlsoperationen zu identifizieren, die Ausnahmen oder Zurückziehbefehlsoperationen erfahren haben.

[0031] Bei Detektierung eines Nichttreffers in der Zeilenvorhersageeinrichtung **12** führt die Ausrichteinheit **16** die entsprechenden Befehlsbytes von dem I-Cachespeicher **14** zu der Vorhersage-Nichttreffer-Dekodiereinheit **26**. Die Vorhersage-Nichttreffer-Dekodiereinheit **26** dekodiert den Befehl, wobei einer Zeile von Befehlen die Grenzen aufgezwungen werden, für die der Prozessor **10** ausgelegt ist (z. B. maximale Anzahl von Befehlsoperationen, maximale Anzahl von Befehlen, Beenden an Verzweigungsbefehlen etc.). Bei Beendigung einer Zeile liefert die Vorhersage-Nichttreffer-Dekodiereinheit **26** die Informationen zu Speicherzwecken zu der Zeilenvorhersageeinrichtung **12**. Es sei darauf hingewiesen, dass die Vorhersage-Nichttreffer-Dekodiereinheit **26** zum Entsenden von Befehlen nach deren Dekodierung vorgesehen sein kann. Alternativ kann die Vorhersage-Nichttreffer-Dekodiereinheit **26** die Zeile von Befehlsinformationen dekodieren und diese zu Speicherzwecken zu der Zeilenvorhersageeinrichtung **12** liefern. Anschließend kann die fehlende Abrufadresse in der Zeilenvorhersageeinrichtung **12** erneut ausprobiert werden, und es wird möglicherweise ein Treffer detektiert.

[0032] Außer zum Dekodieren von Befehlen bei einem Nichttreffer in der Zeilenvorhersageeinrichtung **12** kann die Vorhersage-Nichttreffer-Dekodiereinheit **26** zum Dekodieren von Befehlen vorgesehen sein, wenn die von der Zeilenvorhersageeinrichtung **12** gelieferten Befehlsinformationen ungültig sind. Bei einer Ausführungsform versucht der Prozessor **10** nicht, Informationen in der Zeilenvorhersageeinrichtung **12** mit den Befehlen in dem I-Cachespeicher **14** kohärent zu halten (wenn z. B. Befehle in dem I-Cachespeicher **14** ersetzt oder invalidiert werden, können die entsprechenden Befehlsinformationen nicht aktiv invalidiert werden). Die Dekodiereinheiten **24A–24D** können die gelieferten Befehlsinformationen prüfen und der Vorhersage-Nichttreffer-Dekodiereinheit **26** signalisieren, wenn ungültige Befehlsinformationen detektiert werden. Gemäß einer speziellen Ausführungsform werden die folgenden Befehlsoperationen von dem Prozessor **10** unterstützt: Ganzzahlen-Operationen (einschließlich Arithmetik-, Logik-, Schiebe-/Rotations- und Verzweigungsoperationen), Gleitkommaoperationen (einschließlich Multimediaoperationen) und Lade-/Speicheroperationen.

[0033] Die dekodierten Befehlsoperationen und Quellen- und Zielregisternummern werden zu der Abbildungseinheit **30** geliefert. Die Abbildungseinheit **30** ist zum Durchführen der Registerumbenennung durch Zuweisen von Reellregisternummern (PR#s) zu jedem Zielregisteroperanden und Quellenregisteroperanden jeder Befehlsoperation vorgesehen. Die Reellregisternummern identifizieren Register innerhalb der Registerdateien **38A–38B**. Die Abbildungseinheit **30** liefert ferner eine Anzeige der Abhängigkeiten für jede Befehlsoperation durch Liefern der R#s der Befehlsoperationen, die jede einem Quellenoperanden der Befehlsoperation zugewiesene Reellregisternummer aktualisieren. Die Abbildungseinheit **30** aktualisiert ferner die Zukunfts-Datei **20** mit den jedem Zielregister zugewiesenen Reellregisternummern (und der R# der entsprechenden Befehlsoperation) anhand der entsprechenden Logikregisternummer. Ferner speichert die Abbildungseinheit **30** die Logikregisternummern der Zielregister, die zugewiesenen Reellregisternummern und die zuvor zugewiesenen Reellregisternummern in der Zurückzieh-Warteschlange **32**. Wenn Befehle zurückgezogen werden (was der Abbildungseinheit **30** von dem Scheduler **36** angezeigt wird), aktualisiert die Zurückzieh-Warteschlange **32** die Umbenennungsdateiarchitektur **34** und macht Register frei, die nicht mehr verwendet werden. Entsprechend identifizieren die Reellregisternummern in der Registerdateiarchitektur **34** die reellen Register, in denen der festgeschriebene Architekturzustand des Prozessors **10** gespeichert ist, während die Zukunfts-Datei **20** den spekulativen Zustand des Prozessors **10** repräsentiert. Mit anderen Worten: die Umbenennungsdateiarchitektur **34** speichert eine Reellregisternummer entsprechend

jedem Logikregister, wodurch der festgeschriebene Registerzustand für jedes Logikregister repräsentiert wird. Die Zukunfts-Datei **20** speichert eine Reellregisternummer entsprechend jedem Logikregister, wodurch der spekulative Registerzustand jedes Logikregisters repräsentiert wird.

[0034] Die Zeile von Befehlsoperationen, die Quellen-Reellregisternummern und die Ziel-Reellregisternummern werden entsprechend den von der Abbildungseinheit **30** zugewiesenen R#s im Scheduler **36** gespeichert. Ferner können Abhängigkeiten für eine spezielle Befehlsoperation als Abhängigkeiten von anderen Befehlsoperationen, die in dem Scheduler gespeichert sind, erkannt werden. Bei einer Ausführungsform bleiben Befehlsoperationen in dem Scheduler **36**, bis zu zurückgezogen werden.

[0035] Der Scheduler **36** speichert jede Befehlsoperation, bis die für diese Befehlsoperation erkannten Abhängigkeiten erfüllt sind. In Reaktion auf das Einplanen einer speziellen Befehlsoperation zur Ausführung kann der Scheduler **36** bestimmen, bei welchem Taktzyklus diese spezielle Befehlsoperation die Registerdateien **38A–38B** aktualisiert. Unterschiedliche Ausführungseinheiten in den Ausführungskernen **40A–40B** können eine unterschiedliche Anzahl von Pipeline-Phasen (und somit unterschiedliche Latenzen) verwenden. Ferner können bestimmte Befehle eine größere Latenz innerhalb einer Pipeline erfahren als andere. Entsprechend wird eine Rückwärtszählung erzeugt, die die Latenz für die spezielle Befehlsoperation (als Anzahl der Taktzyklen) misst. Der Scheduler **36** erwartet die spezifizierte Anzahl von Taktzyklen (bis die Aktualisierung vor oder gleichzeitig mit dem Lesen der Registerdatei durch die abhängigen Befehlsoperationen erfolgt) und zeigt dann an, dass die von dieser speziellen Befehlsoperation abhängige Operation eingeplant werden kann. Es sei darauf hingewiesen, dass der Scheduler **36** einen Befehl einplanen kann, wenn dessen Abhängigkeiten erfüllt sind (d. h. ungeordnet relativ zu seiner Reihenfolge in der Scheduler-Warteschlange).

[0036] Ganzzahlen- und Lade-/Speicher-Befehlsoperationen lesen Quellenoperanden entsprechend den Quellen-Reellregisternummern aus der Registerdatei **38A** und werden zu Ausführungszwecken zu dem Ausführungskern **40A** transportiert. Der Ausführungskern **40A** führt die Befehlsoperation aus und aktualisiert das dem Ziel in der Registerdatei **38A** zugewiesenen reelle Register. Ferner meldet der Ausführungskern **40A** die R# der Befehlsoperation und Ausnahmeinformationen bezüglich der Befehlsoperation (falls vorhanden) an den Scheduler **36**. Die Registerdatei **38B** und der Ausführungskern **40B** können hinsichtlich Gleitkomma-Befehlsoperationen auf im wesentlichen gleiche Weise arbeiten (und können Speicherdaten für Gleitkommaspeicherungen zu der Lade-/Speichereinheit **42** liefern).

[0037] Bei einer Ausführungsform kann der Ausführungsfern **40A** beispielsweise zwei Ganzzahlen-Einheiten, eine Verzweigungseinheit und zwei Adressenerzeugungseinheiten (mit entsprechenden Translation-Lookaside-Puffern oder TLBs) aufweisen. Der Ausführungskern **40B** kann einen Gleitkomma-/Multimedia-Multiplizierer, einen Gleitkomma-/Multimedia-Addierer und eine Speicherdateneinheit zum Liefern von Speicherdaten zu der Lade-Speichereinheit **42** aufweisen. Andere Konfigurationen der Ausführungseinheiten sind möglich.

[0038] Die Lade-/Speichereinheit **42** bildet ein Interface zu dem D-Cachespeicher **44** zum Durchführen von Speicheroperationen und zum Einplanen von Fülloperationen für Speicheroperationen, die den D-Cachespeicher **44** nicht treffen. Speicher-Lade-Speicher-Operationen können von dem Ausführungskern **40A** abgeschlossen werden, der eine Adressenerzeugung durchführt und Daten (von dem D-Cachespeicher **44** oder einer Speicher-Warteschlange in der Lade-/Speichereinheit **42**) zu den Registerdateien **38A–38B** weiterleitet. Die Speicheradressen können dem D-Cachespeicher **44** bei Erzeugung durch den Ausführungskern **40A** präsentiert werden (direkt über Verbindungen zwischen dem Ausführungskern **40A** und dem D-Cachespeicher **44**). Den Speicheradressen wird ein Speicher-Warteschlangen-Eintrag zugewiesen. Die Speicherdaten können je nach ausgewählter Auslegung gleichzeitig oder anschließend geliefert werden. Beim Zurückziehen des Speicherbefehls werden die Daten in dem D-Cachespeicher **44** gespeichert (obwohl es eine Verzögerung zwischen dem Zurückziehen und dem Aktualisieren des D-Cachespeichers **44** geben kann). Ferner kann die Lade-/Speichereinheit **42** einen Lade-/Speicherpuffer zum Speichern von Lade-/Speicheradressen, die den D-Cachespeicher **44** nicht treffen, für anschließende Cachespeicher-Auffüllungen (über eine Extern-Interface-Einheit **46**) und zum erneuten Versuchen der fehlgeschlagenen Lade-/Speicheroperationen aufweisen. Die Lade-/Speichereinheit **42** ist ferner zum Verarbeiten von Lade-/Speicher-Speicherabhängigkeiten vorgesehen.

[0039] Der D-Cachespeicher **44** ist ein Hochgeschwindigkeits-Cachespeicher zum Speichern von Daten, auf die der Prozessor **10** zugreift. Während der D-Cachespeicher **44** eine beliebige geeignete Struktur haben kann (einschließlich Direktabbildungs- und Satzassoziativstrukturen), kann eine Ausführungsform des D-Cachespeichers **44** einen 128 Kbyte-2-Wege-Satzassoziativ-Cachespeicher mit 64 Bytezeilen aufweisen.

[0040] Die Extern-Interface-Einheit **46** ist zur Datenübertragung über Extern-Interfaces **52** zu anderen Vorrichtungen vorgesehen. Es kann ein beliebiges geeignetes Extern-Interface **52** verwendet werden, einschließlich Interfaces zu L2-Cachespeichern und

einem Externbus oder -Bussen zum Verbinden des Prozessors **10** mit anderen Vorrichtungen. Die Extern-Interface-Einheit **46** ruft Füllungen für den I-Cachespeicher **16** und den D-Cachespeicher **44** auf und schreibt gestrichene aktualisierte Cachespeicher-Zeilen von dem D-Cachespeicher **44** in das Extern-Interface. Ferner kann die Extern-Interface-Einheit **46** auch von dem Prozessor **10** erzeugte nicht pufferbare Lese- und Schreibvorgänge durchführen.

[0041] Fig. 2 zeigt ein beispielhaftes Pipeline-Diagramm mit Darstellung eines beispielhaften Satzes von Pipeline-Phasen, die von einer Ausführungsform des Prozessors **10** verwendet werden können. Bei anderen Ausführungsformen können andere Pipelines verwendet werden, und zwar Pipelines mit mehr oder weniger Pipeline-Phasen als die in Fig. 2 gezeigte Pipeline. Die in Fig. 2 gezeigten Phasen sind durch vertikale Linien begrenzt. Jede Phase ist ein Taktzyklus eines zum Takten von Speicherelementen (z. B. Registern, Latches, Flops u. dgl.) innerhalb des Prozessors **10** verwendeten Taktsignals.

[0042] Gemäß Fig. 2 weist die beispielhafte Pipeline eine CAM0-Phase, eine CAM1-Phase, eine Zeilenvorhersage- (LP-) Phase; eine Befehls-Cachespeicher- (IC-) Phase, eine Ausricht- (AL-) Phase, eine Dekodier- (DEC-) Phase, eine Abbildungs1- (M1-) Phase, eine Abbildungs2- (M2-) Phase, eine Schreib-Scheduler- (WR SC-) Phase, eine Lese-Scheduler- (RD SC-) Phase, eine Registerdatei-Lese- (RF RD-) Phase, eine Ausführungs- (EX-) Phase, eine Registerdatei-Schreib- (RF WR-) Phase und eine Zurückzieh- (RET-) Stufe auf. Einige Befehle verwenden mehrere Taktzyklen in der Ausführungsphase. Beispielsweise sind Speicheroperationen, Gleitkommaoperationen, und Ganzzahlenmultiplizier-Operationen als Explosionsansicht in Fig. 2 gezeigt. Speicheroperationen umfassen eine Adressenerzeugungs- (AGU-) Phase, eine Umsetz- (TLB-) Phase, eine Daten-Cachespeicher1- (DC1-) Phase und eine Daten-Cachespeicher2-(DC2-) Phase. Ähnlich umfassen Gleitkommaoperationen bis zu vier Gleitkomma-Ausführungs- (FEX1-FEX4-) Phasen und die Ganzzahlen-Multiplikationen bis zu vier (IM1-IM4-) Phasen.

[0043] In den CAM0- und CAM1-Phasen vergleicht die Zeilenvorhersageeinrichtung **12** die von der Verzweigungsvorhersage-/abruf-PC-Erzeugungseinheit **18** erzeugte Abrufadresse mit den Adressen der hierin gespeicherten Zeilen. Ferner wird die Abrufadresse in den CAM0- und CAM1-Phasen von einer virtuellen Adresse (z. B. einer linearen Adresse in der x86-Architektur) in eine reale Adresse umgesetzt. In Reaktion auf das Detektieren eines Treffers in den CAM0- und CAM1-Phasen werden die entsprechenden Zeileninformationen in der Zeilenvorhersagephase aus der Zeilenvorhersageeinrichtung ausgelesen. Ferner initiiert der I-Cachespeicher **14** ein Lesen (un-

ter Verwendung der realen Adresse) in der Zeilenvorhersagephase. Das Lesen wird in der Befehls-Cachespeicher-Phase beendet.

[0044] Es sei darauf hingewiesen, dass die in Fig. 2 gezeigte Pipeline zwar zwei Taktzyklen zum Detektieren eines Treffers in der Zeilenvorhersageeinrichtung **12** bei einer Abrufadresse verwendet, bei anderen Ausführungsformen jedoch ein einziger Taktzyklus (und eine einzige Phase) zum Durchführen dieser Operation verwendet werden kann. Ferner liefert bei einer Ausführungsform die Zeilenvorhersageeinrichtung **12** bei einem Treffer eine nächste Abrufadresse für den I-Cachespeicher **14** und ein nächster Eintrag in die Zeilenvorhersageeinrichtung **12**, und daher können die CAM0- und CAM1-Phasen bei Abrufen aufgrund eines vorangegangenen Treffers in der Zeilenvorhersageeinrichtung **12** übersprungen werden.

[0045] Befehlsbytes von dem I-Cachespeicher **14** werden in der Ausrichtphase in Reaktion auf die entsprechenden Zeileninformationen von der Zeilenvorhersageeinrichtung **12** von der Ausrichteinheit **16** auf die Dekodiereinheiten **24A–24D** ausgerichtet. Die Dekodiereinheiten **24A–24D** dekodieren die gelieferten Befehle, wodurch ROPs entsprechend den Befehlen sowie Operandeninformationen in der Dekodierphase identifiziert werden. Die Abbildungseinheit **30** erzeugt in der Abbildungs1-Phase ROPs aus den gelieferten Informationen und führt eine Registerumbenennung durch (wodurch die Zukunftsdatei **20** aktualisiert wird). In der Abbildungs1-Phase werden die ROPs und zugewiesene Umbenennungen in der Zurückzieh-Warteschlange **32** protokolliert. Ferner werden die ROPs, von denen jede ROP abhängig ist, bestimmt. Jede ROP kann von früheren in der Zukunfts-Datei protokollierten ROPs registerabhängig sein und kann ferner andere Arten von Abhängigkeiten aufweisen (z. B. Abhängigkeiten von einem vorangegangenen Serialisierungsbefehl etc.).

[0046] Die erzeugten ROPs werden in der Schreib-Scheduler-Phase in den Scheduler **36** geschrieben. Bis zu dieser Phase durchlaufen die von einer speziellen Informationszeile definierten ROPs die Pipeline als Einheit. Nach dem Schreiben in den Scheduler **36** können die ROPs jedoch unabhängig zu unterschiedlichen Zeiten die restlichen Phasen durchlaufen. Generell bleibt eine spezielle ROP in dieser Phase, bis sie vom Scheduler **36** zur Ausführung ausgewählt wird (z. B. nachdem die ROPs, von denen die spezielle ROP abhängig ist, zur Ausführung ausgewählt worden sind, wie oben beschrieben). Entsprechend kann eine spezielle ROP eine Verzögerung von einem oder mehrere Taktzyklen zwischen der Schreib-Scheduler-Schreib-Phase und der Lese-Scheduler-Phase erfahren. In der Lese-Scheduler-Phase beteiligt sich die spezielle ROP an der Auswahllogik in dem Scheduler **36**, wird zur Ausführung ausgewählt und aus dem Scheduler **36**

ausgelesen. Die spezielle ROP geht dann in der Registerdatei-Lese-Phase zu Lese-Registerdatei-Operationen aus einer der Registerdateien **38A–38B** (je nach ROP-Typ) über.

[0047] Die spezielle ROP und Operanden werden zu dem entsprechenden Ausführungskern **40A** oder **40B** geliefert, und die Befehlsoperation wird in der Ausführungsphase auf den Operanden durchgeführt. Wie oben beschrieben, weisen einige ROPs mehrere Pipeline-Ausführungsphasen auf. Beispielsweise werden Speicherbefehlsoperationen (z. B. Ladungen und Speicherungen) über eine Adressenerzeugungsphase (in der die Datenadresse der Speicherstelle, auf die bei der Speicherbefehlsoperation zugegriffen wird, erzeugt wird), eine Umsetzphase (in der die reale Datenadresse aus der Adressenerzeugungsphase umgesetzt wird) und zwei Daten-Cachespeicher-Phasen, in denen auf den D-Cachespeicher **44** zugegriffen wird, ausgeführt. Gleitkommaoperationen können bis zu 4 Ausführungs-Taktzyklen verwenden, und Ganzzahlen-Multiplikationen können auf ähnliche Weise bis zu 4 Ausführungs-Taktzyklen verwenden.

[0048] Bei Beendigung der Ausführungsphase oder -phasen aktualisiert die spezielle ROP das ihr zugewiesene Register in der Registerdatei-Schreib-Phase. Schließlich wird die spezielle ROP zurückgezogen, nachdem jede vorangegangene ROP (in der Zurückziehphase) zurückgezogen worden ist. Auch hier können bei einer speziellen ROP ein oder mehr Taktzyklen zwischen der Registerdatei-Schreib-Phase und der Zurückziehphase ablaufen. Ferner kann eine spezielle ROP in einer beliebigen Phase aufgrund von Pipeline-Stillstandsbedingungen angehalten werden, wie auf dem Sachgebiet bekannt ist.

Lade-zu-Speicher-Weiterleitung

[0049] **Fig. 3** zeigt ein Blockschaltbild mit genauerer Darstellung einer Ausführungsform der Abbildungseinheit **30**, des Scheduler **36**, einer Adressenerzeugungseinheit/eines Translation-Lookaside-Puffers (AGU/TLB) **40AA** und einer Lade-/Speichereinheit **42**. Andere Ausführungsformen sind möglich und werden in Betracht gezogen. Bei der in **Fig. 3** gezeigten Ausführungsform werden der Scheduler **36**, AGU/TLB **40AA** und die Lade-/Speichereinheit **42** kollektiv als Ausführungs-Pipeline **72** bezeichnet. Die Abbildungseinheit **30** weist eine Lade-zu-Speicher-Weiterleitungs- (STLF-) Vorhersageeinrichtung **60**, eine Abhängigkeitseinheit **62** und eine R#-Zuweisungseinheit **64** (die R#s den Befehlsoperationen zuweist) auf. Der Scheduler **36** weist einen Scheduler-Puffer **66** und einen Realadressen- (PA-) Puffer **70** auf. Die Lade-/Speichereinheit **42** weist eine Speicher-Warteschlange **68** auf. Die Abbildungseinheit **30** ist zum Empfangen von Befehlsoperationen und entsprechenden Programmzähleradressen (PCs) von

der Dekodiereinheit **24**, eines Zurückziehsignals von dem Scheduler **36** und eines Train-/Untrain-Interface (einschließlich Train-/Untrain(T/UT-) Signale, einer Lade-PC (LPC) und eines Speicheridentifizierers (SID) von der Ausführungs-Pipeline **72** vorgesehen. Die Abbildungseinheit **30** ist zum Liefern der Befehlsoperationen, PCs und Abhängigkeitsinformationen an den Scheduler **36** vorgesehen. Insbesondere ist die STLF-Vorhersageeinrichtung **60** zum Empfangen der Befehlsoperationen, PCs, R#s von der R#-Zuweisungseinheit **64** und des Train-/Untrain-Interface und zum Liefern von Informationen bezüglich Abhängigkeiten der Ladung von der Speicherung an die Abhängigkeitseinheit **62** vorgesehen, welche ferner zum Empfangen der Befehlsoperationen und R#s von der R#-Zuweisungseinheit **64** vorgesehen ist. Die Abhängigkeitseinheit **62** ist zum Liefern der Abhängigkeitsinformationen an den Scheduler **36** vorgesehen. Der Scheduler **36** ist zum Liefern von Ladungen und Speicherungen zusammen mit entsprechenden Steuerinformationen zu AGU/TLB **40AA** vorgesehen. AGU/TLB **40AA** ist zum Empfangen entsprechender Operanden von der Registerdatei **38A** und zum Liefern einer realen Adresse und anderer Steuerinformationen zusammen mit Speicherdaten im Falle einer Speicherung an die Speicher-Warteschlange **68** vorgesehen. Ferner ist AGU/TLB **40AA** zum Liefern der realen Adresse und Steuerinformationen an den PA-Puffer **70** vorgesehen, welcher mit dem Scheduler-Puffer **66** gekoppelt ist. Die Speicher-Warteschlange **68** ist zum Empfangen eines Zurückziehsignals von dem Scheduler **36** und zum Liefern einer Speicherung zur Festschreibung und von Speicher-Weiterleitungsdaten zu dem D-Cachespeicher **44** vorgesehen. Bei einer Ausführungsform ist AGU/TLB **40AA** Teil eines Ganzzahlen-Ausführungskerns **40A**.

[0050] Generell empfängt die Abbildungseinheit **30** Befehlsoperationen von den Dekodiereinheiten **24**. Bei jeder Speicher-Lade-Speicher-Operation stellt die STLF-Vorhersageeinrichtung **60** fest, ob sie Informationen besitzt, die anzeigen, dass bei einer vorangegangenen Ausführung eine Speicher-Abspeicher-Operation mit der Speicher-Lade-Speicher-Operation interferiert hat. Wenn eine Speicher-Abspeicher-Operation interferiert hat, liefert die STLF-Vorhersageeinrichtung **60** eine Anzeige dieser Speicher-Abspeicher-Operation an die Abhängigkeitseinheit **62**. Die Abhängigkeitseinheit **62** zeigt eine Abhängigkeit der Speicher-Lade-Speicher-Operation von dieser Speicher-Abspeicher-Operation an (zusätzlich zu Abhängigkeiten der Adressenoperanden etc.), und somit wird die Speicher-Lade-Speicher-Operation nicht vor der Speicher-Abspeicher-Operation eingeplant. Entsprechend kann bei der aktuellen Ausführung der Speicher-Lade-Speicher-Operation die Speicher-Abspeicher-Operation nicht interferieren. Andererseits zeigt, wenn keine Informationen bezüglich einer Interferenz seitens einer

Speicher-Abspeicher-Operation mit einer speziellen Speicher-Lade-Speicher-Operation von der STLF-Vorhersageeinrichtung **60** protokolliert worden ist, die STLF-Vorhersageeinrichtung **60** der Abhängigkeitseinheit **62** keine Abhängigkeit an. Die spezielle Speicher-Lade-Speicher-Operation kann Abhängigkeiten für Quellenregisteroperanden empfangen, jedoch nicht für Speicher-Abspeicher-Operationen.

[0051] Wie hier verwendet, "interferiert" eine Speicher-Abspeicher-Operation mit einer Speicher-Lade-Speicher-Operation, wenn die Speicher-Abspeicher-Operation das Hinzufügen zusätzlicher Taktzyklen zu der Ausführung der Speicher-Lade-Speicher-Operation bewirkt. Die zusätzlichen Taktzyklen können über Pipeline-Stillstände oder über eine Neuausführung der Speicher-Lade-Speicher-Operation hinzugefügt werden. Die restliche Offenbarung konzentriert sich auf eine Ausführungsform, bei der eine Speicher-Abspeicher-Operation mit einer Speicher-Lade-Speicher-Operation interferiert, wenn die Speicher-Abspeicher-Operation älter ist als die Speicher-Lade-Speicher-Operation, die Speicher-Lade-Speicher-Operation von der Speicher-Abspeicher-Operation abhängig ist und die Speicher-Lade-Speicher-Operation vor der Speicher-Abspeicher-Operation eingeplant und/oder ausgeführt wird. Andere Ausführungsformen werden in Betracht gezogen. Beispielsweise wird eine Ausführungsform in Betracht gezogen, bei der Speicher-Lade-Speicher-Operationen nicht vor der Adressenerzeugung einer Speicher-Abspeicher-Operation, jedoch vor der Bereitstellung der Speicherdaten eingeplant werden. Bei einer solchen Ausführungsform kann die Speicherung mit der Ladung interferieren, wenn eine Abhängigkeit besteht und die Speicherdaten bei Ausführung der Speicher-Lade-Speicher-Operation nicht zur Verfügung stehen.

[0052] Die Abbildungseinheit **30** leitet die Befehloperationen, PCs und Abhängigkeiten zu dem Scheduler **36**, der die Befehlsoperationen in den Scheduler-Puffer **66** schreibt. Der Scheduler-Puffer **66** weist mehrere Einträge auf, von denen jeder Eintrag in der Lage ist, Informationen bezüglich einer Befehlsoperation zu speichern. Ein beispielhafter Abspeicheroperationseintrag ist im Scheduler-Puffer **66** dargestellt, einschließlich eines Gültig-Bits und eines Typenfelds (das den Eintrag als Speicherung einer Abspeicheroperation identifiziert und identifiziert, welcher Typ von Abspeicheroperation, entweder Ladung oder Speicherung, gespeichert wird). Bei Speicher-Abspeicher-Operationen wird die PC der Speicher-Lade-Speicher-Operation (LPC) gespeichert. Es können auch weitere Informationen gespeichert werden, um das Training der STLF-Vorhersageeinrichtung **60** mit Informationen bezüglich einer Speicher-Abspeicher-Operation, die mit der Speicher-Lade-Speicher-Operation interferiert, zu unterstützen. Beispielsweise kann ein Speicher-ID-Feld (SID) zum

Speichern einer Anzeige der Speicher-Abspeicher-Operation, die bei der aktuellen Ausführung mit der Speicher-Lade-Speicher-Operation interferiert hat, und einer Wiederholanzeige (R), die anzeigt, dass die Speicher-Lade-Speicher-Operation wiederholt worden ist (wegen einer Interferenz durch die Speicher-Abspeicher-Operation) und somit zur erneuten Ausführung neu eingeplant werden muss, vorgesehen sein. Eine Train-Anzeige (T) ist ebenfalls gespeichert, um anzuzeigen, dass die Ladung von der STLF-Vorhersageeinrichtung **60** bei Entsendung der Ladung zu dem Scheduler **36** als von einer älteren Speicherung abhängig detektiert worden ist. Bei einer Ausführungsform kann die Wiederholanzeige ein eine Wiederholung anzeigendes Bit sein, wenn das Bit gesetzt ist. Ähnlich kann die Train-Anzeige ein Bit sein, das, wenn es gesetzt ist, anzeigt, dass die Abhängigkeit detektiert worden ist. Bei anderen Ausführungsformen kann andersherum vorgegangen werden und können andere Kodierungen verwendet werden. Ferner können, falls gewünscht, weitere Informationen gespeichert werden (z. B. Größeninformationen, Operanden-PR#s etc.), und andere Eintragstypen (z. B. Ganzzahlen, Gleitkomma etc.) können andere Formate haben. Der Scheduler **36** plant die Speicheroperation zur Ausführung im Anschluss an die Erfüllung jeder der protokollierten Abhängigkeiten ein (einschließlich von der STLF-Vorhersageeinrichtung **60** identifizierter Abhängigkeiten) und transportiert den Lade-/Speichertyp der Operation und weitere Steuerinformationen zu AGU/TLB **40AA**.

[0053] Insbesondere können, da der Scheduler **36** eine Abspeicheroperation zur Ausführung einplant, wenn jede der protokollierten Abhängigkeiten erfüllt ist, jüngere Ladungen vor älteren Speicherungen eingeplant werden, wenn die STLF-Vorhersageeinrichtung **60** keine Abhängigkeit der jüngeren Ladung von der älteren Speicherung anzeigt. Die Abbildungseinheit **30** kann jede Quellenregisteroperanden-Abhängigkeit detektieren, ist möglicherweise jedoch nicht in der Lage, sämtliche Lade-Abhängigkeiten von früheren Speicherungen zu detektieren. Die Abhängigkeit einer Ladung von einer Speicherung basiert auf den von der Ladung bzw. der Speicherung beeinflussten Speicheradressen, die von Quellenoperanden der Ladung und Speicherung bei der Ausführung der Ladung und Speicherung erzeugt worden sind. Die STLF-Vorhersageeinrichtung **60** detektiert bestimmte Abhängigkeiten der Ladungen von Speicherungen (wie hier beschrieben), andere werden jedoch möglicherweise nicht detektiert. Entsprechend verwendet der Prozessor **10** den PA-Puffer **70**, wie nachstehend beschrieben, zum Detektieren von Fällen, in denen eine jüngere Speicherung, die vor einer älteren Speicherung eingeplant worden ist, von der älteren Speicherung abhängig ist.

[0054] AGU/TLB **40AA** empfängt die Abspeicheroperation und Operanden (aus der Registerdatei **38A**

in Reaktion auf PR#s von dem Scheduler **36** ausgelesen). AGU/TLB **40AA** fügt die Operanden hinzu, um eine virtuelle Adresse zu erzeugen, und setzt die virtuelle Adresse unter Verwendung von in dem TLB gepufferten Umsetzungen in eine reale Adresse um. AGU/TLB **40AA** liefert die reale Adresse und andere Steuerinformationen an die Speicher-Warteschlange **68**. Speicherdaten werden auch geliefert, wenn die Operation eine Speicherung ist. Unter den von AGU/TLB **40AA** gelieferten Steuerinformationen kann sich der Lade- oder Speichertyp der Operation befinden. Die reale Adresse und andere Steuerinformationen werden ebenfalls von AGU/TLB **40AA** zu dem D-Cachespeicher **44** und dem PA-Puffer **70** geliefert.

[0055] Der PA-Puffer **70** wird bei der vorliegenden Ausführungsform zum Detektieren von Speicherungen, die mit Ladungen interferieren, verwendet. Der PA-Puffer **70** weist mehrere Einträge auf, und zwar einen Eintrag für jeden Eintrag in dem Scheduler-Puffer **66**. Jeder Eintrag ist in der Lage, Informationen über die reale Adresse zu speichern. Wenn eine Ladung ausgeführt wird, wird die dem PA-Puffer **70** gelieferte reale Adresse in einem Eintrag entsprechend dem die Ladung speichernden Scheduler-Puffer-Eintrag gespeichert. Andererseits wird bei Ausführung einer Speicherung die reale Adresse mit der in dem PA-Puffer **70** gespeicherten realen Adresse verglichen. Wenn eine Übereinstimmung festgestellt wird und die entsprechende Befehlsoperation eine Ladung ist, die jünger ist als die Speicherung, wird die Ladung wiederholt. Generell wird eine Abspeicheroperation hier als "wiederholt" bezeichnet, wenn der Zustand der Operation in dem Scheduler **36** auf einen nicht ausgeführten Zustand zurückgesetzt wird. Das anschließende Wiederholen einer Abspeicheroperation führt dazu, dass die Abspeicheroperation neu eingeplant und neu ausgeführt wird.

[0056] Wenn eine Wiederholungsituation in dem PA-Puffer **70** detektiert wird, wird die Wiederholungsanzeige in dem entsprechenden Scheduler-Puffer-Eintrag gesetzt. Ferner wird die von der STLF-Vorhersageeinrichtung **60** zum Identifizieren der Speicherung verwendete Speicher-ID in dem SID-Feld des Scheduler-Puffer-Eintrags gespeichert. Bei einer Ausführungsform kann die Speicher-ID die Speicher-PC sein. Bei einer weiteren Ausführungsform kann die Speicher-ID die R# der Speicherung oder die Differenz zwischen der R# der Speicherung und der R# der Ladung (die Delta-R#) sein. Ausführungsformen, bei denen jede Speicher-ID verwendet wird, werden nachstehend genauer beschrieben. Anschließend, wenn die Ladung neu eingeplant und neu ausgeführt ist, bewirkt die gesetzte Wiederholungsanzeige, dass die Ausführungs-Pipeline **72** die Ladung und die entsprechende Speicherung unter Verwendung des Train-/Untrain-Interface in die STLF-Vorhersageeinrichtung **60** liefert (so dass bei anschließenden Aus-

führungen eine Wiederholung der Ladung dadurch verhindert wird, dass die Ladung von der Speicherung abhängig gemacht wird). Insbesondere kann ein Train-Signal innerhalb des Interface aktiviert werden und können dann die Lade-PC und die Speicher-ID aus dem entsprechenden Scheduler-Eintrag auch zu der STLF-Vorhersageeinrichtung **60** geliefert werden. Es sei darauf hingewiesen, dass entsprechend der gewählten Auslegung das Training aus einer beliebigen Phase der Ausführungs-Pipeline **72** heraus erfolgen kann.

[0057] Durch Anzeigen der Abhängigkeit der Ladung von der Speicherung, die bei einer vorangegangenen Ausführung der Ladung mit der Ladung interferiert hat, kann eine Einplanung der Ladung verhindert werden, bis die Speicherung eingeplant ist. Auf diese Weise kann die Abhängigkeit der Ladung von der Speicherung bei der ersten Ausführung der Speicher-Lade-Speicher-Operation detektiert werden und können Speicherdaten in Reaktion auf die Speicher-Lade-Speicher-Operation weitergeleitet werden. Somit können eine Neueinplanung und Neuausführung der Ladung verhindert werden. Die Leistungsfähigkeit kann aufgrund des Fehlens einer Neueinplanung und Neuausführung der Ladung erhöht werden.

[0058] Andererseits kann, wenn die Ladung von einer Speicherung abhängig gemacht wird und keine tatsächliche Abhängigkeit während einer Ausführung vorhanden ist, die Leistungsfähigkeit aufgrund der verzögerten Einplanung der Ladung verringert werden. Entsprechend kann die Ausführungs-Pipeline **72** auch diese Situationen detektieren und das Train-/Untrain-Interface für ein Untrain der Ladung und der entsprechenden Speicherung von der STLF-Vorhersageeinrichtung **60** verwenden. Insbesondere wenn eine Ladung eingeplant ist und ihre Train-Anzeige in dem Scheduler-Puffer **66** gesetzt ist, stellt die Ausführungs-Pipeline **72** fest, ob die Ladung weitergeleitete Daten von der Speicher-Warteschlange **68** empfängt. Wenn keine Weiterleitung erfolgt, ist eine Abhängigkeit der Ladung von einer Speicherung möglicherweise nicht gewährleistet. Entsprechend kann die Ausführungs-Pipeline **72** ein Untrain-Signal aktivieren und die Lade-PC zu der STLF-Vorhersageeinrichtung **60** liefern. Die STLF-Vorhersageeinrichtung **60** kann ein Untrain der Informationen entsprechend der Ladung durchführen.

[0059] Wie hier verwendet, bedeutet der Ausdruck "Train" das Speichern von Informationen, die das Auftreten einer Speicherung identifizieren, welche mit einer Ladung interferiert, und die Aktualisierungsinformationen enthalten können, die die Wiederkehrwahrscheinlichkeit anzeigen (z. B. wenn die Situation in der Vergangenheit wiederholt eingetreten ist, kann die Wahrscheinlichkeit, dass sie wieder eintritt, grö-

ßer sein). Somit kann das Training das Erzeugen einer stärkeren Korrelation zwischen der Ladung und der Speicherung umfassen. Der Ausdruck "Untrain" bedeutet das Löschen von Informationen, die das Auftreten einer mit einer Ladung interferierenden Speicherung identifizieren und kann das Erzeugen einer schwächeren Korrelation zwischen der Ladung und der Speicherung vor dem Löschen der Informationen umfassen.

[0060] Es sei darauf hingewiesen, dass das Training und Untraining der STLF-Vorhersageeinrichtung **60** aus einer beliebigen Pipeline-Phase heraus erfolgen kann und das Training in einer anderen Phase durchgeführt werden kann als das Untraining. Beispielsweise wird bei der vorliegenden Ausführungsform das Training in Reaktion auf die Wiederholanzeige durchgeführt, wenn die Ladung erneut eingeplant wird, und könnte somit in einer beliebigen Phase nach der in **Fig. 2** gezeigten Scheduler-Lese-Phase durchgeführt werden. Das Untraining erfolgt in Reaktion auf die Train-Anzeige und das Fehlen einer Speicher-Weiterleitung für die Ladung und kann somit an späterer Stelle in der Pipeline erfolgen (z. B. in der in **Fig. 2** gezeigten DC2-Phase).

[0061] Bei der Ausführung von Abspeicheroperationen speichert, wenn die Abspeicheroperation eine Speicherung ist, die Speicher-Warteschlange **68** die von AGU/TLB **40AA** gelieferten Informationen. Andererseits vergleicht, wenn die Abspeicheroperation eine Ladung ist, die Speicher-Warteschlange **68** die Ladeinformationen mit den Informationen in den Speicher-Warteschlangen-Einträgen. Wenn ein Treffer des Speicher-Warteschlangen-Eintrags detektiert wird, werden die entsprechenden Speicher-Warteschlangen-Daten gelesen und zu Weiterleitungszwecken zu dem D-Cachespeicher **44** geliefert (in **Fig. 3** gezeigte Speicher-Weiterleitungs-Daten).

[0062] Die Speicher-Warteschlange **68** hält die Speicherungen zumindest so lange, bis sie von dem Scheduler **36** zurückgezogen werden. Der Scheduler **36** signalisiert der Speicher-Warteschlange **68** über das Zurückziehesignal, dass das Zurückziehen einer oder mehrerer Speicherungen angezeigt werden soll. Die Speicher-Warteschlange **68** transportiert die zurückgezogenen Speicherungen in geordneter Form unter Verwendung des Speicher-Festschreibe-Wegs zu dem D-Cachespeicher **44**. Somit können die Speicherungen in der Speicher-Warteschlange **68** bleiben, bis sie in dem D-Cachespeicher **44** festgeschrieben werden, was sich aufgrund des Zurückziehens der Speicherungen verzögern kann.

[0063] Es sei darauf hingewiesen, dass verschiedene Kombinationen aus AGUs und TLBs möglich sind. Beispielsweise werden bei einer Ausführungsform eine Lade-AGU und eine separate Speicher-AGU in Betracht gezogen. Die Speicher-AGU kann mit ei-

nem Schreib-Port in der Speicher-Warteschlange **68** gekoppelt sein, und die Lade-AGU kann mit einem Vergleich-Port in der Speicher-Warteschlange **68** gekoppelt sein. Weitere Ausführungsformen können eine beliebige gewünschte Anzahl von AGUs für Ladungen, Speicherungen oder Ladungen und Speicherungen aufweisen.

[0064] Es sei darauf hingewiesen, dass zwar bestimmte Details der verschiedenen in **Fig. 3** gezeigten Einheiten dargestellt sind, andere Details und Merkmale, die sich nicht auf das Detektieren von Treffern in der Speicher-Warteschlange landenden Ladungen beziehen, jedoch möglicherweise der Einfachheit halber weggelassen worden sind. Beispielsweise kann die Abbildungseinheit **30** eine Registerumbenennung durchführen, wie oben anhand von **Fig. 1** beschrieben.

[0065] Bei einer Ausführungsform arbeitet die STLF-Vorhersageeinrichtung **60** während der Abbildungs2-Phase der in **Fig. 2** gezeigten Pipeline, und zwar hinsichtlich des Anzeigens von Abhängigkeiten der Ladungen von älteren Speicherungen. Die STLF-Vorhersageeinrichtung **60** kann jedoch bei verschiedenen Ausführungsformen vor dem Auswählen der Ladung zur Ausführung in einer beliebigen Pipeline-Phase arbeiten.

[0066] In der vorstehenden Beschreibung ist das Training während der erneuten Ausführung der Ladung dargestellt. Bei alternativen Ausführungsformen kann das Training jedoch zu unterschiedlichen Zeiten erfolgen. Beispielsweise kann bei einer alternativen Ausführungsform das Training in Reaktion auf das Detektieren der Wiederholsituation erfolgen (z. B. bei der Ausführung der Speicherung, von der die Ladung abhängig ist).

[0067] Die PCs der Ladungen (und bei einer Ausführungsform der Speicherungen) sind in Zusammenhang mit dem Training und Untraining der Ladungen und Speicherungen in der STLF-Vorhersageeinrichtung **60** beschrieben worden. Es sei jedoch darauf hingewiesen, dass bei einigen Ausführungsformen möglicherweise nur ein Teil des PC für das Training und Untraining der Ladungen und Speicherungen in der STLF-Vorhersageeinrichtung **60** verwendet wird. Beispielsweise können bei einer Ausführungsform der STLF-Vorhersageeinrichtung **60** die 10 niederwertigsten Bits der Lade-PC und der Speicher-PC verwendet werden, wie nachstehend beschrieben.

[0068] Wie hier verwendet, bedeutet das Akronym "PC" die Programmzähleradresse eines Befehls. Die PC ist die Adresse eines Befehls in dem Speicher. Mit anderen Worten: die PC ist die zum Abrufen des Befehls aus dem Speicher verwendete Adresse. Bei Ausführungsformen, bei denen mehrere Befehls-

operationen einem Befehl entsprechen können, ist die PC des Befehls auch die PC jeder Befehlsoperation (z. B. Speicher-Lade-Speicher- und Speicher-Abspeicher-Operationen).

[0069] Es sei darauf hingewiesen, dass die R# bei bestimmten vorstehend und nachstehend beschriebenen Ausführungsformen zum Identifizieren von Befehlsoperationen verwendet wird. Generell kann eine beliebige geeignete Markierung verwendet werden. Die R# identifiziert eine relative Programmreihenfolge der Befehlsoperationen und kann den den Befehlsoperationen zugewiesenen Eintrag in dem Scheduler-Puffer **66** identifizieren. Bei anderen Ausführungsformen können Neuordnungs-Puffer-Markierungen oder beliebige andere Markierungen zum Identifizieren der Befehlsoperationen verwendet werden. Ferner können vor oder gleichzeitig mit der Operation der STLF-Vorhersageeinrichtung **60** die R#s oder Markierungen einem beliebigen Punkt in der Pipeline des Prozessors **10** zugewiesen werden.

[0070] Fig. 4 zeigt ein Blockschaltbild einer ersten Ausführungsform der STLF-Vorhersageeinrichtung **60** (STLF-Vorhersageeinrichtung **60a**). Andere Ausführungsformen sind möglich und werden in Betracht gezogen. Bei der in Fig. 4 gezeigten Ausführungsform weist die STLF-Vorhersageeinrichtung **60a** eine Lade-/Speicher-Abhängigkeitstabelle **80**, eine Speicher-PC/R#-Tabelle **82**, eine Lade-/Speicher-Abhängigkeitstabelle- (LSDT-) Steuerschaltung **84**, eine ST/LD-Abhängigkeitsschaltung **86**, eine Speichertabellen-Steuerschaltung **88**, eine Intraline-Abhängigkeits-Prüfschaltung **90** und einen Multiplexer (Mux) **92** auf. Die Lade-/Speicher-Abhängigkeitstabelle **80** ist zum Empfangen der PCs der Entsende-Befehlsoperationen von den Dekodiereinheiten **24** vorgesehen und mit der LSDT-Steuerschaltung **84** gekoppelt. Ferner ist die Lade-/Speicher-Abhängigkeitstabelle **80** zum Empfangen einer Lade-PC und einer Speicher-PC von der Ausführungs-Pipeline **72** zu Training-Zwecken vorgesehen. Die Lade-/Speicher-Abhängigkeitstabelle **80** ist zum Liefern von Speicher-PCs zur Intraline-Abhängigkeits-Prüfschaltung **90** und zur Speicher-PC/R#-Tabelle **82** und von Gültig-Anzeigen zur ST/LD-Abhängigkeitsschaltung **86** vorgesehen. Die Intraline-Abhängigkeits-Prüfschaltung **90** ist zum Empfangen der PCs der Entsende-Befehlsoperationen und einer Anzeige des Lade- oder Speichertyps jeder Befehlsoperation von den Dekodiereinheiten **24** und zum Liefern von Treffersignalen zu der ST/LD-Abhängigkeitsschaltung **86** und einer Selektionssteuerung zum Mux **92** vorgesehen. Die Speicher-PC/R#-Tabelle **82** ist zum Empfangen der PCs der Entsende-Befehlsoperationen und der entsprechenden den Befehlsoperationen zugewiesenen R#s vorgesehen. Die Speicher-PC/R#-Tabelle **82** ist zum Liefern der Speicher-R#s zum Mux **92** vorgesehen, und der Mux **92** ist zum Empfangen der den Entsende-Befehlsoperationen zugewiesenen R#s

und zum Liefern der Speicher-R#s zu der Abhängigkeitseinheit **62** vorgesehen. Die Speicher-PC/R#-Tabelle **82** ist zum Liefern von Treffersignalen zu der Speichertabellen-Steuerschaltung **88** und zum Empfangen von Steuerinformationen von der Speichertabellen-Steuerschaltung **88** vorgesehen. Die ST/LD-Abhängigkeitsschaltung **86** ist mit der Speichertabellen-Steuerschaltung **88** gekoppelt und zum Liefern von Abhängigkeit-gültig- und Abhängig-von-allem-Signalen zu der Abhängigkeitseinheit **62** vorgesehen. Die LSDT-Steuerschaltung **84** ist zum Empfangen von Train-/Untrain-Signalen von der Ausführungs-Pipeline **72** vorgesehen.

[0071] Generell wird die Lade-/Speicher-Abhängigkeitstabelle **80** von einer Lade-PC weitergeschaltet, um einen der mehreren Einträge auszuwählen. Der Eintrag speichert eine Gültig-Anzeige und eine Speicher-PC (SPC in Fig. 4) einer Speicherung, die möglicherweise bei einer vorangegangenen Ausführung mit dieser Ladung interferiert hat. Die Speicher-PC/R#-Tabelle weist mehrere Einträge auf, die die Speicher-PC der kürzlich entsendeten Speicherungen zusammen mit der entsprechenden R# für diese Speicherung speichert. Wenn die Speicher-PC aus dem aus der Lade-/Speicher-Abhängigkeitstabelle **80** ausgewählten Eintrag einen Treffer in der Speicher-PC/R#-Tabelle **82** landet, wird für die Ladung eine Abhängigkeit der Ladung von der Speicherung festgestellt. Auf diese Weise wird verhindert, dass die Ladung vor der Speicherung eingeplant (und somit ausgeführt) wird. Entsprechend kann die Interferenz bei der aktuellen Ausführung vermieden werden.

[0072] Insbesondere werden beim Entsenden von Befehlsoperationen die PCs der Befehlsoperationen zum Weiterschalten in die Lade-/Speicher-Abhängigkeitstabelle **80** verwendet. Die restliche Beschreibung konzentriert sich auf das Ansprechen der STLF-Vorhersageeinrichtung **60a** auf eine Eingangs-PC entsprechend einer Entsende-Befehlsoperation, sofern nichts anderes angegeben ist. Es sei jedoch darauf hingewiesen, dass die STLF-Vorhersageeinrichtung **60a** parallel dazu auf jede PC jeder Entsende-Befehlsoperation ansprechen kann. In Reaktion auf die Eingangs-PC gibt die Lade-/Speicher-Abhängigkeitstabelle **80** eine Gültig-Anzeige und eine Speicher-PC von dem weitergeschalteten Eintrag aus. Die Speicher-PC wird in die Speicher-PC/R#-Tabelle **82** eingegeben und mit den in der PC/R#-Tabelle **82** gespeicherten Speicher-PCs verglichen. Beispielsweise kann die Speicher-PC/R#-Tabelle **82** einen inhaltadressierten Speicher (CAM) aufweisen. Die Speicher-PC/R#-Tabelle **82** gibt Treffersignale für jeden Eintrag aus, die anzeigen, ob dieser Eintrag von der Speicher-PC getroffen worden ist oder nicht. Die Speichertabellen-Steuerschaltung **88** empfängt die Treffersignale und wählt die jüngste in der Speicher-PC/R#-Tabelle

82 repräsentierte Speicherungen aus, die von der Speicher-PC getroffen wird. Der ausgewählte Eintrag gibt eine Speicher-R# an den Mux **92** aus, der generell diese Speicher-R# zur Ausgabe an die Abhängigkeitseinheit **62** auswählt.

[0073] Ferner informiert die Speichertabellen-Steuerschaltung **88** die ST/LD-Abhängigkeitsschaltung **86** mit einer Anzeige darüber, ob ein Treffer in der Speicher-PC/R#-Tabelle **82** für diese Befehlsoperation detektiert worden ist oder nicht. Die ST/LD-Abhängigkeitsschaltung **86** liefert ein Abhängigkeit-gültig-Signal an die Abhängigkeitseinheit **62**. Das Abhängigkeit-gültig-Signal zeigt in aktiviertem Zustand an, dass die Abhängigkeitseinheit **62** eine Abhängigkeit der Befehlsoperation von der Speicherung, die von der von dem Mux **92** gelieferten Speicher-R# identifiziert worden ist, protokollieren soll. Wenn das Abhängigkeit-gültig-Signal deaktiviert ist, zeigt das Signal an, dass keine Abhängigkeit von der Abhängigkeitseinheit **62** zu protokollieren ist. Insbesondere bei einer Ausführungsform kann die ST/LD-Abhängigkeitsschaltung **86** das Abhängigkeit-gültig-Signal aktivieren, wenn: (i) die Befehlsoperation eine Ladung ist (anhand von Lade-/Speicheranzeigen von den Dekodiereinheiten **24** festgestellt); (ii) die Gültig-Anzeige von dem weitergeschalteten Eintrag der Lade-/Speicher-Abhängigkeitstabelle **80** Gültig anzeigt; und (iii) die Speicher-PC von dem weitergeschalteten Eintrag einen Treffer in der Speicher-PC/R#-Tabelle **82** landet.

[0074] Der in **Fig. 1** gezeigte Prozessor **10** versucht das Entsenden mehrerer Befehlsoperationen pro Taktzyklus. Somit ist es möglich, dass die jüngste Speicherung, die mit der Speicher-PC von der Lade-/Speicher-Abhängigkeitstabelle **80** übereinstimmt, gleichzeitig mit der entsprechenden Ladung entsendet wird. Entsprechend vergleicht für eine vorgegebene Speicheroperation die Intraline-Abhängigkeits-Prüfschaltung **90** die von der Lade-/Speicher-Abhängigkeitstabelle **80** ausgegebene Speicher-PC mit den PCs jeder gleichzeitig entsendeten Befehlsoperation, die der vorgegebenen Speicheroperation in der Programmreihenfolge vorangeht. Wenn die PC der vorhergehenden Befehlsoperation mit der Speicher-PC von den Lade-/Speicher-Abhängigkeitstabelle **80** übereinstimmt und die vorhergehende Befehlsoperation eine Speicherung ist (was durch die von den Dekodiereinheiten **24** entsendeten Lade-/Speicheranzeigen angezeigt wird), kann die Intraline-Abhängigkeits-Prüfschaltung **90**: (i) der ST/LD-Abhängigkeitsschaltung **86** einen Treffer für die entsprechende Ladung anzeigen; und (ii) den Mux **92** derart steuern, dass er die von der Speicher-PC/R#-Tabelle **82** gelieferte Speicher-R# mit der R# der Befehlsoperation, auf der der Treffer detektiert wird, übersteuert. Auf diese Weise ist die an die Abhängigkeitseinheit **62** ausgegebene Speicher-R# die R# derjenigen Speicherung, die gleich-

zeitig mit der Ladung entsendet worden ist. Ferner kann die ST/LD-Abhängigkeitsschaltung **86** das Abhängigkeit-gültig-Signal für die Ladung aktivieren, wenn: (i) die Befehlsoperation eine Ladung ist (was durch die von den Dekodiereinheiten **24** entsendeten Lade-/Speicheranzeigen angezeigt wird); (ii) die Gültig-Anzeige von dem weitergeschalteten Eintrag der Lade-/ Speicher-Abhängigkeitstabelle **80** Gültig anzeigt; und (iii) das Treffer-Signal von der Intraline-Abhängigkeits-Prüfschaltung **90** für die Ladung aktiviert ist.

[0075] Bei einer Ausführungsform kann die ST/LD-Abhängigkeitsschaltung **86** ferner das Abhängig-von-allen-Signal für die Befehlsoperation aktivieren. Das Abhängig-von-allen-Signal zeigt in aktiviertem Zustand der Abhängigkeitseinheit **62** an, dass sie die Abhängigkeiten der Befehlsoperation von jeder ausstehenden (entsendeten und nicht zurückgezogenen) Speicherung protokollieren soll. Das Abhängig-von-allen-Signal wird zur Handhabung einer Situation verwendet, in der ein spezieller Eintrag wiederholt mit Speicher-PCs von Speicherungen, die mit Ladungen interferieren, einem Training unterzogen wird. Da die Lade-/Speicher-Abhängigkeitstabelle **80** einen Eintrag in Reaktion auf eine PC einer Befehlsoperation auswählt und der Eintrag eine Speicher-PC speichert, können Ladungen, mit denen unterschiedliche Speicherungen bei unterschiedlichen Ausführungen interferieren, immer noch in Interferenzsituationen geraten, obwohl die STLF-Vorhersageeinrichtung **60a** eine Abhängigkeit von einer Speicherung anzeigt. Zur besseren Handhabung solcher Fälle kann die Gültig-Anzeige in der Lade-/Speicher-Abhängigkeitstabelle **80** ein Bit-Vektor sein. Jedes Mal, wenn ein Eintrag einem Training durch die Ausführungs-Pipeline **72** unterzogen wird, kann ein Bit in dem Bit-uVektor in den Gültig-Zustand gesetzt werden (je nach gewählter Auslegung z. B. gesetzt oder gelöscht werden). Wenn sich jedes Bit im Gültig-Zustand befindet, kann der Eintrag wiederholt einem Training unterzogen werden, da die Ladung bei unterschiedlichen Ausführungen mit unterschiedlichen Speicherungen interferiert. Entsprechend kann das Abhängig-von-allen-Signal aktiviert werden, wenn: (i) jedes Bit in dem Bit-Vektor im Gültig-Zustand ist; und (ii) die Befehlsoperation eine Ladung ist. Eine Ausführungsform des Bit-Vektors und des Setzens der Bits in den Gültig- oder Ungültig-Zustand ist nachstehend genauer beschrieben.

[0076] Vorstehend ist die Verwendung der STLF-Vorhersageeinrichtung **60a** zum Anzeigen von Abhängigkeiten der Ladungen von Speicherungen, die bei vorhergehenden Ausführungen möglicherweise mit den Ladungen interferiert haben, beschrieben worden. Ferner kann die STLF-Vorhersageeinrichtung **60a** einem Training mit Informationen über die Ladungen und Speicherungen unterzogen werden. Die Speicher-PC/R#-Tabelle **82** speichert die Spei-

cher-PCs und R#s der jüngst entsendeten Speicherungen. Somit kann die Speichertabellen-Steuerschaltung **88** Einträge in der Speicher-PC/R#-Tabelle **82** Speicherungen zuordnen, die entsendet werden. Die Speichertabellen-Steuerschaltung **88** empfängt die Lade-/Speicheranzeigen für jede Befehlsoperation von den Dekodiereinheiten **24** und ordnet Einträge jeder Entsende-Speicherung zu. Die zugeordneten Einträge werden mit der Speicher-PC (die von den Dekodiereinheiten **24** empfangen wird) und der entsprechenden R# (die von der R#-Zuweisungseinheit **64** empfangen wird) aktualisiert. Bei einer Ausführungsform kann die Speichertabelle-Steuerschaltung **88** die Speicher-PC/R#-Tabelle **82** als FIFO-Puffer der jüngst entsendeten Speicherungen betreiben. Somit verschieben, wenn die Speicher-PC/R#-Tabelle **82** mit Speicherungen gefüllt ist, anschließend entsendete Speicherungen die ältesten Speicherungen in der Speicher-PC/R#-Tabelle **82**. Ferner ist es möglich, dass eine Speicherung zurückgezogen wird, bevor sie durch anschließend entsendete Speicherungen aus der PC/R#-Tabelle **82** gelöscht werden. Entsprechend kann die Speichertabellen-Steuerschaltung **88** die R#s der Zurückzieh-Speicherungen empfangen und Einträge mit der entsprechenden R# löschen.

[0077] Die Lade-/Speicher-Abhängigkeitstabelle **80** wird in Reaktion auf das Train-/ Untrain-Interface von der Ausführungs-Pipeline **72** einem Training unterzogen. Insbesondere wenn das Train-Signal von der Ausführungs-Pipeline **72** aktiviert wird, bewirkt die LSDT-Steuerschaltung **84**, dass die Lade-/Speicher-Abhängigkeitstabelle **80** aktualisiert wird. Die Ausführungs-Pipeline **72** liefert die FC der einem Training zu unterziehenden Ladung (LPC in **Fig. 4**) und die entsprechende Speicher-PC, die mit der Ladung interferiert, als Eingang zu der Lade-/Speicher-Abhängigkeitstabelle **80**. Die Lade-/Speicher-Abhängigkeitstabelle **80** aktualisiert die von der Lade-PC weitergeschaltete Eingang mit der Speicher-PC, und die LSDT-Steuerschaltung **84** setzt die Gültig-Anzeige in einen Gültig-Zustand. Bei einer Ausführungsform kann die Gültig-Anzeige ein Bit sein und der Gültig-Zustand gesetzt (oder gelöscht) und der Ungültig-Zustand gelöscht (oder gesetzt) sein. Bei einer weiteren oben beschriebenen Ausführungsform kann die Gültig-Anzeige ein Bit-Vektor sein. Bei einer solchen Ausführungsform kann die LSDT-Steuerschaltung **84** ein Bit in dem Bit-Vektor auswählen und dieses Bit während des Trainings in den Gültig-Zustand setzen.

[0078] Ferner kann die LSDT-Steuerschaltung **84** ein Untraining eines Eintrags in Reaktion auf das Aktivieren des Untrain-Signals durch die Ausführungs-Pipeline **72** durchführen. Auch hier kann die Ausführungs-Pipeline **72** die Lade-PC der einem Untraining zu unterziehenden Ladung liefern, die Speicher-PC kann jedoch in dem Untraining-Fall ignoriert

werden. Die Lade-/Speicher-Abhängigkeitstabelle **80** schaltet den von der Lade-PC angezeigten Eintrag weiter, und die LSDT-Steuerschaltung **84** bewirkt, dass die Gültig-Anzeige in dem weitergeschalteten Eintrag in den Ungültig-Zustand gesetzt wird. Bei einer Ausführungsform, bei der ein Gültig-Bit als Gültig-Anzeige verwendet wird, kann das Bit zum Anzeigen von Ungültig gelöscht (oder gesetzt) werden. Bei einer Ausführungsform, bei der der oben beschriebene Bit-Vektor verwendet wird, kann ein ausgewähltes Bit in den Ungültig-Zustand gesetzt werden. Der Eintrag kann im Bit-Vektor-Fall gültig bleiben, wenn andere Bits im Gültig-Zustand bleiben. Mehrere Untrain-Ereignisse können jedoch schließlich bewirken, dass jedes der anderen Bits ebenfalls ungültig wird.

[0079] Wie vorstehend anhand von **Fig. 3** beschrieben, sind PCs zwar derart dargestellt, dass sie die Lade-/Speicher-Abhängigkeitstabelle **80** und die Speicher-PC/R#-Abhängigkeitstabelle **82** weiterzuschalten, verschiedene Ausführungsformen werden jedoch möglicherweise mit nur einem Teil der PCs weitergeschaltet und/oder speichern möglicherweise nur einen Teil der PCs. Der zum Weiterschalten der Lade-/Speicher-Abhängigkeitstabelle **80** verwendete Teil kann durch die Anzahl von in der Tabelle verwendeten Einträgen bestimmt werden. Beispielsweise kann bei einer speziellen Implementierung die Lade-/Speicher-Abhängigkeitstabelle **80** 1K Einträge aufweisen und können somit 10 Bits der PC als Index verwendet werden (z. B. die 10 niederwertigsten Bits). Die Anzahl von Einträgen kann generell als gewählte Auslegung ausgewählt werden, die teilweise auf dem von der Tabelle im Verhältnis zu der Genauigkeit der Tabelle im allgemeinen für die Ladungen in der Target-Software eingenommenen Bereich basiert. Die Anzahl der für die in der Lade-/Speicher-Abhängigkeitstabelle **80** und der Speicher-PC/R#-Abhängigkeitstabelle **82** gespeicherten PCs Speicher-PCs verwendeten Bits kann von der Anzahl von in dem Index verwendeten Bits abweichen und kann auch hier als gewählte Auslegung ausgewählt werden, die teilweise auf dem von den Tabellen im Verhältnis zu der Genauigkeit der Tabellen im allgemeinen für die Ladungen in der Target-Software eingenommenen Bereich basiert. Bei einer speziellen Implementierung werden die 10 niederwertigsten Bits der Speicher-PC gespeichert.

[0080] Ferner kann die Anzahl von Einträgen in der Speicher-PC/R#-Tabelle **82** ebenfalls von der ausgewählten Auslegung abhängen, die teilweise auf dem von der Tabelle im Verhältnis zu der Genauigkeit der Tabelle im allgemeinen für die Ladungen in der Target-Software eingenommenen Bereich basiert. Bei einer speziellen Implementierung können 8–12 Einträge verwendet werden.

[0081] Es sei darauf hingewiesen, dass die oben beschriebene Ausführungsform zwar auf jede PC je-

der Entsende-Befehlsoperation ansprechen kann, bei anderen Ausführungsformen jedoch die Anzahl von gleichzeitig erfolgenden Befehlsoperationen, auf die die STLF-Vorhersageeinrichtung **60a** anspricht, begrenzt sein kann. Bei solchen Ausführungsformen können die in die STLF-Vorhersageeinrichtung **60a** eingegebenen PCs und R#s in Reaktion auf die Lade-/Speicheranzeigen von den Dekodiereinheiten **24** gemultiplext werden, so dass zu Speicherzwecken nur die PCs der Ladungen in die Lade-/Speicher-Abhängigkeitstabelle **80** und nur die PCs der Speicherungen in die Speicher-PC/R#-Tabelle **82** eingegeben werden. Bei einer solchen Ausführungsform kann die Vorhersage-Nichttreffer-Dekodiereinheit **26** eine Zeile von Befehlsoperationen beenden, wenn die Lade- und/oder Speichergrenze erreicht ist.

[0082] Es sei darauf hingewiesen, dass als Alternative zu dem für die Gültig-Anzeige und die Abhängig-von-Allen-Anzeige verwendeten Bit-Vektor zum Verarbeiten der Ladungen, die bei verschiedenen Ausführungen mit verschiedenen Speicherungen interferieren, jeder Eintrag in die Lade-/Speicher-Abhängigkeitstabelle **80** eine Speicherung mehrerer Speicher-PCs und entsprechender Gültig-Bits bewirken kann. Jede Speicher-PC von einem ausgewählten Eintrag kann mit der Speicher-PC/R#-Tabelle **82** verglichen werden, und es kann eine Abhängigkeit der Ladung von jeder Speicherung, bei der es sich um einen Treffer in der Speicher-PC/R#-Tabelle **82** handelt, protokolliert werden.

[0083] Fig. 5 zeigt eine zweite Ausführungsform der STLF-Vorhersageeinrichtung **60** (STLF-Vorhersageeinrichtung **60b**). Andere Ausführungsformen sind möglich und werden in Betracht gezogen. Bei der in Fig. 5 gezeigten Ausführungsform weist die STLF-Vorhersageeinrichtung **60b** eine Lade-/Speicher-Abhängigkeitstabelle **100**, eine Addierschaltung **102**, eine Lade-/Speicher-Abhängigkeitstabelle (LSDT-) Steuerschaltung **104**, eine ST/LD-Abhängigkeitsschaltung **106** und wahlweise eine Speicher-Validierschaltung **108** auf. Die Lade-/ Speicher-Abhängigkeitstabelle **100** ist zum Empfangen von PCs von Entsende-Befehlsoperationen von den Dekodiereinheiten **24** und ferner zum Empfangen einer Lade-PC und Delta-R# von der Ausführungs-Pipeline **72** vorgesehen. Ferner ist die Lade-/Speicher-Abhängigkeitstabelle **100** mit der LSDT-Steuerschaltung **104** gekoppelt und zum Liefern von Gültig-Anzeigen zu der ST/LD-Abhängigkeitsschaltung **106** und Delta-R#s zu der Addierschaltung **102** vorgesehen. Die Addierschaltung **102** ist ferner zum Empfangen von R#s der Entsende-Befehlsoperationen von der R#-Zuweisungseinheit **64** vorgesehen. Die Addierschaltung **102** ist zum Liefern von Speicher-R#s zu der Abhängigkeitseinheit **62** und der Speicher-Validierschaltung **108** vorgesehen, welche zum Empfangen einer gültigen Speicher-R#-Anzeige von der Abhängigkeitseinheit **62** vorgesehen ist. Die Spei-

cher-Validierschaltung **108** ist zum Liefern von Speicher-gültig-Signalen zu der ST/LD-Abhängigkeitsschaltung **106** vorgesehen, welche ferner zum Empfangen von Lade-/Speicheranzeigen entsprechend den Entsende-Befehlsoperationen von den Dekodiereinheiten **24** vorgesehen ist. Die ST/LD-Abhängigkeitsschaltung **106** ist zum Liefern von Abhängigkeit-gültig-Signalen und Abhängig-von-Allen-Signalen zu der Abhängigkeitseinheit **62** vorgesehen. Die LSDT-Steuerschaltung ist zum Empfangen von Train-/Untrain-Signalen von der Ausführungs-Pipeline **72** vorgesehen.

[0084] Generell kann die STLF-Vorhersageeinrichtung **60b** wie folgt auf eine Entsende-Ladung ansprechen. Die Lade-PC wird zum Weiterschalten in die Lade-/ Speicher-Abhängigkeitstabelle **100** verwendet, wodurch einer von mehreren Einträgen ausgewählt wird. Der ausgewählte Eintrag speichert eine Gültig-Anzeige und eine Delta-R#. Die Gültig-Anzeige zeigt an, ob die STLF-Vorhersageeinrichtung **60b** einem Training mit Informationen bezüglich einer Ladung mit der Weiterschalt-PC unterzogen worden ist oder nicht und somit, ob die Delta-R# gültig ist oder nicht. Die Delta-R# ist die Differenz zwischen der R# der Ladung und der R# einer Speicherung, die bei einer vorhergehenden Ausführung mit der Ladung interferiert hat. Da sich Befehlssequenzen während einer Ausführung typischerweise nicht verändern, kann die Differenz zwischen der R# der Ladung und der R# der Speicherung bei der aktuellen Ausführung typischerweise die gleiche sein wie die Differenz bei der vorangegangenen Ausführung. Die Addierschaltung **102** addiert die Delta-R# zu der der Entsende-Ladung zugewiesenen R# hinzu, um eine Speicher-R# zu erzeugen, die zu der Abhängigkeitseinheit **62** geliefert wird. Die Abhängigkeitseinheit **62** kann dann eine Abhängigkeit der Ladung von der Speicherung protokollieren. Auf diese Weise wird verhindert, dass die Ladung vor der Speicherung eingeplant (und somit ausgeführt) wird. Entsprechend kann die Interferenz bei der aktuellen Ausführung vermieden werden.

[0085] Insbesondere werden beim Entsenden von Befehlsoperationen die PCs der Befehlsoperationen zum Weiterschalten in die Lade-/Speicher-Abhängigkeitstabelle **100** verwendet. Die restliche Beschreibung konzentriert sich auf das Ansprechen der STLF-Vorhersageeinrichtung **60b** auf eine Eingangs-PC entsprechend einer Entsende-Befehlsoperation, sofern nichts anderes angegeben ist. Es sei jedoch darauf hingewiesen, dass die STLF-Vorhersageeinrichtung **60b** parallel dazu auf jede PC jeder Entsende-Befehlsoperation ansprechen kann. Die Lade-/Speicher-Abhängigkeitstabelle **100** gibt eine Delta-R# und eine Gültig-Anzeige von dem ausgewählten Eintrag aus. Der Addierer **102** addiert die Delta-R# zu der R#, die der Entsende-Befehlsoperation entspricht, und erzeugt somit eine Speicher-R#,

die zu der Abhängigkeitseinheit **62** transportiert wird. Es sei darauf hingewiesen, dass die Addierschaltung **102** einen Addieren für jede Entsende-Befehlsoperation aufweisen kann, der den entsprechenden Delta-R#-Ausgang von der Lade-/Speicher-Abhängigkeitstabelle **100** und die dieser Entsende-Befehlsoperation von der R#-Zuweisungseinheit **64** zugewiesene R# empfängt.

[0086] Die ST/LD-Abhängigkeitsschaltung **106** empfängt die Gültig-Anzeige darüber, ob die Befehlsoperation eine Ladung oder eine Speicherung ist oder nicht, von den Dekodiereinheiten **24**. Die ST/LD-Abhängigkeitsschaltung **106** liefert ein Abhängigkeit-gültig-Signal an die Abhängigkeitseinheit **62**, die der oben beschriebenen ST/LD-Abhängigkeitsschaltung **86** im wesentlichen gleich ist. Die ST/LD-Abhängigkeitsschaltung **106** kann das Abhängigkeit-gültig-Signal aktivieren, wenn: (i) die Befehlsoperation eine Ladung ist (anhand von Lade-/ Speicheranzeigen von den Dekodiereinheiten **24** festgestellt); (ii) die Gültig-Anzeige von dem weitergeschalteten Eintrag Gültig anzeigt.

[0087] Ferner kann bei einer Ausführungsform die STLF-Vorhersageeinrichtung **60b** die Speicher-Validierschaltung **108** verwenden. Die Speicher-Validierschaltung **108** empfängt eine Anzeige darüber, welche R#s den ausstehenden Speicherungen entsprechen, von der Abhängigkeitseinheit **62**. Bei einer Ausführungsform kann die Anzeige ein Bit-Vektor mit einem Bit pro R# sein, die anzeigt, ob die R# einer Speicherung entspricht oder nicht. Die Speicher-Validierschaltung **108** stellt fest, ob die von der Addierschaltung **102** erzeugte R# einer Speicherung entspricht oder nicht und signalisiert dies der ST/LD-Abhängigkeitsschaltung **106** mit dem Speicher-gültig-Signal. Wenn das Speicher-gültig-Signal aktiviert ist, entspricht die erzeugte R# einer Speicherung. Andererseits entspricht die erzeugte R# nicht einer Speicherung, wenn das Speichergültig-Signal deaktiviert ist. Bei Ausführungsformen, die die Speicher-Validierschaltung **108** verwenden, besteht eine zusätzliche Bedingung für die ST/LD-Abhängigkeitsschaltung **106** zum Aktivieren des Abhängigkeit-gültig-Signals darin, dass das Speicher-gültig-Signal von der Speicher-Validierschaltung **108** aktiviert ist.

[0088] Auch hier kann die ST/LD-Abhängigkeitsschaltung **106** ähnlich wie die ST/LD-Abhängigkeitsschaltung **86** bei Ausführungsformen, bei denen die Gültig-Anzeige ein Bit-Vektor ist, zum Liefern des Abhängig-von-alle-Signals vorgesehen sein. Die Operation der ST/LD-Abhängigkeitsschaltung **106** kann in dieser Hinsicht der ST/LD-Abhängigkeitsschaltung **106** im wesentlichen gleich sein.

[0089] Die Lade-/Speicher-Abhängigkeitstabelle **100** wird in Reaktion auf das Train-/ Untrain-Interface von der Ausführungs-Pipeline **72** einem Training un-

terzogen. Insbesondere wenn das Train-Signal von der Ausführungs-Pipeline **72** aktiviert wird, bewirkt die LSDT-Steuerschaltung **104** eine Aktualisierung der Lade-/Speicher-Abhängigkeitstabelle **100**. Die Ausführungs-Pipeline **72** liefert die PC der einem Training zu unterziehenden Ladung (LPC in **Fig. 5**) und die entsprechende Delta-R# als Eingang zu der Lade-/Speicher-Abhängigkeitstabelle **100**. Die Lade-/Speicher-Abhängigkeitstabelle **100** aktualisiert den von der Lade-PC weitergeschalteten Eintrag mit der Delta-R#, und die LSDT-Steuerschaltung **104** setzt die Gültig-Anzeige in einen Gültig-Zustand. Bei einer Ausführungsform kann die Gültig-Anzeige ein Bit sein und der Gültig-Zustand gesetzt (oder gelöscht) und der Ungültig-Zustand gelöscht (oder gesetzt) werden. Bei einer weiteren oben beschriebenen Ausführungsform kann die Gültig-Anzeige ein Bit-Vektor sein. Bei einer solchen Ausführungsform kann die LSDT-Steuerschaltung **104** ein Bit innerhalb des Bit-Vektors auswählen und dieses Bit während des Trainings in den Gültig-Zustand setzen.

[0090] Ferner kann die LSDT-Steuerschaltung **104** einen Eintrag in Reaktion auf das Aktivieren des Untrain-Signals durch die Ausführungs-Pipeline **72** einem Untraining unterziehen. Auch hier kann die Ausführungs-Pipeline **72** die Lade-PC der einem Untraining zu unterziehenden Ladung erzeugen, die Delta-R# kann im Untraining-Fall jedoch ignoriert werden. Die Lade-/Speicher-Abhängigkeitstabelle **100** schaltet den von der Lade-PC angezeigten Eintrag weiter, und die LSDT-Steuerschaltung **104** bewirkt, dass die Gültig-Anzeige in dem weitergeschalteten Eintrag in den Ungültig-Zustand gesetzt wird. Bei einer Ausführungsform, bei der ein Gültig-Bit als Gültig-Anzeige gesetzt wird, kann das Bit zum Anzeigen von Ungültig gelöscht (oder gesetzt) werden. Bei einer Ausführungsform, bei der der oben beschriebene Bit-Vektor verwendet wird, kann ein ausgewähltes Bit in den Ungültig-Zustand gesetzt werden. Der Eintrag kann in dem Bit-Vektor-Fall gültig bleiben, wenn andere Bits in dem Gültig-Zustand verbleiben. Mehrere Untrain-Ereignisse können jedoch schließlich bewirken, dass jedes andere Bit ebenfalls ungültig wird.

[0091] Wie oben anhand von **Fig. 4** beschrieben, sind PCs zwar derart dargestellt, dass sie die Lade-/Speicher-Abhängigkeitstabelle **100** weiterschalten, verschiedene Ausführungsformen werden jedoch möglicherweise mit nur einem Teil der PCs weitergeschaltet. Der zum Weiterschalten der Lade-/Speicher-Abhängigkeitstabelle **100** verwendete Teil kann durch die Anzahl von in der Tabelle verwendeten Einträgen bestimmt werden. Beispielsweise kann bei einer speziellen Implementierung die Lade-/Speicher-Abhängigkeitstabelle **100** 1K Einträge aufweisen und können somit 10 Bits der PC als Index verwendet werden (z. B. die 10 niederwertigsten Bits). Die Anzahl von Einträgen kann generell als gewählte Auslegung ausgewählt werden, die teilweise

auf dem von der Tabelle im Verhältnis zu der Genauigkeit der Tabelle im allgemeinen für die Ladungen in der Target-Software eingenommenen Bereich basiert.

[0092] Es sei darauf hingewiesen, dass bei der vorliegenden Ausführungsform die Delta-R# zwar während des Trainings zu der STLF-Vorhersageeinrichtung **60b** geliefert wird, bei anderen Ausführungsformen jedoch die Lade- und Speicher-R#s geliefert werden können und die Delta-R# in der STLF-Vorhersageeinrichtung **60b** berechnet werden kann. Ferner können bei Ausführungsformen die Delta-R# und die R# der Ladung entweder addiert oder subtrahiert werden, um die R# der Speicherung zu erzeugen. Außerdem kann eine alternative Konfiguration der Speicher-Validierschaltung **108** darin bestehen, dass die von der Addierschaltung **102** erzeugte Speicher-R# in dem Scheduler **36** gesucht wird, um festzustellen, ob die Befehlsoperation eine Speicherung ist.

[0093] Fig. 6 zeigt ein Ablaufdiagramm mit Darstellung der Operation einer Ausführungsform der Ausführungs-Pipeline **72** mit Bezug auf Speicher-Lade-Speicher-Operationen. Andere Ausführungsformen sind möglich und werden in Betracht gezogen. Obwohl die in Fig. 6 dargestellten Schritte zum besseren Verständnis in einer speziellen Reihenfolge gezeigt sind, kann eine beliebige geeignete Reihenfolge angewendet werden. Insbesondere können parallel Schritte von der kombinatorischen Logik in der Ausführungs-Pipeline **72** durchgeführt werden. Ferner können verschiedene Schritte in unterschiedlichen Phasen in der Ausführungs-Pipeline **72** ausgeführt werden. Informationen über weitere Schritte können in Form einer Pipeline zu Phasen weitergeleitet werden, in denen Schritte durchgeführt werden.

[0094] Die Ausführungs-Pipeline **72** stellt fest, ob eine Ladung zur Ausführung eingeplant ist (Entscheidungsblock **110**). Wenn eine Ladung nicht eingeplant ist, sind bei dieser Ausführungsform keine Training-Operationen möglich. Wenn eine Ladung eingeplant ist, stellt die Ausführungs-Pipeline **72** fest, ob die Ladung bei einem Treffer in dem Realadressen-Puffer **70** wiederholt worden ist (Entscheidungsblock **112**). Insbesondere kann die Ausführungs-Pipeline **72** die Wiederholanzeige von dem der Ladung zugewiesenen Scheduler-Puffer-Eintrag prüfen. Wenn die Ladung aufgrund eines Treffers des Realadressen-Puffers wiederholt worden ist, aktiviert die Ausführungs-Pipeline **72** das Train-Signal zu der STLF-Vorhersageeinrichtung **60** und liefert die Lade-PC und die Speicher-ID der einem Training zu unterziehenden Ladung und Speicherung in die STLF-Vorhersageeinrichtung **60** (Block **114**).

[0095] Andererseits stellt, wenn die Ladung nicht aufgrund eines Treffers des Realadressen-Puffers

wiederholt worden ist, die Ausführungs-Pipeline **72** fest, ob die Ladung aufgrund der Operation der STLF-Vorhersageeinrichtung **60** von einer Speicherung abhängig gemacht worden ist (Entscheidungsblock **116**). Mit anderen Worten: die Ausführungs-Pipeline **72** stellt fest, ob die Train-Anzeige in dem der Ladung zugewiesenen Scheduler-Puffer-Eintrag anzeigt, dass die Ladung einem Training unterzogen worden ist. Wenn die Ladung einem Training unterzogen worden ist, stellt die Ausführungs-Pipeline **72** fest, ob Daten von der Speicher-Warteschlange für die Ladung weitergeleitet werden (Entscheidungsblock **118**). Wenn keine Daten weitergeleitet werden, ist es wahrscheinlich, dass die Ladung nicht mit einer Speicherung interferiert hat. Entsprechend kann in diesem Fall die Ausführungs-Pipeline **72** das Untrain-Signal zu der STLF-Vorhersageeinrichtung **60** aktivieren und die Lade-PC der Ladung zu Untraining-Zwecken liefern (Block **120**).

[0096] Es sei darauf hingewiesen, dass das Training auch bei Ausführung einer Speicherung, die mit einer Ladung interferiert, statt bei einer erneuten Ausführung der Ladung aufgrund der Wiederholung durch geführt werden kann.

[0097] Fig. 7 zeigt ein Blockschaltbild eines Teils einer Ausführungsform einer LSdT-Steuerschaltung **130**. Die LSdT-Steuerschaltung **130** kann bei verschiedenen Ausführungsformen als LSdT-Steuerschaltung **84** und/oder LSdT-Steuerschaltung **104** verwendet werden. Andere Ausführungsformen sind möglich und werden in Betracht gezogen. Bei der in Fig. 7 gezeigten Ausführungsform weist die LSdT-Steuerschaltung **130** eine Steuerschaltung **132** und eine mit der Steuerschaltung gekoppelte Zähler-schaltung **134** auf. Die Steuerschaltung **132** ist zum Empfangen der Train- und Untrain-Signale von der Ausführungs-Pipeline **72** und zum Liefern von Set_V[3:0]-Signalen und Clear_V[3:0]-Signalen zu der Lade-/Speicher-Abhängigkeitstabelle **80** oder **100** (je nach Ausführungsform) vorgesehen.

[0098] Die LSdT-Steuerschaltung **130** ist bei Ausführungsformen, bei denen die Gültig-Anzeigen Bit-Vektoren sind, zum Verwalten der Gültig-Anzeigen in der Lade-/Speicher-Abhängigkeitstabelle während des Trainings und Untrainings vorgesehen. Bei der vorliegenden Ausführungsform ist jedes Bit in dem Bit-Vektor im Gültig-Zustand, wenn es gesetzt ist, und im Ungültig-Zustand, wenn es gelöscht ist, obwohl bei alternativen Ausführungsformen jedes Bit in dem Bit-Vektor im Gültig-Zustand sein kann, wenn es gelöscht ist, und im Ungültig-Zustand sein kann, wenn es gesetzt ist. Bei weiteren Ausführungsformen können die Gültig-Zustände in den Bits kodiert sein.

[0099] Wenn ein Eintrag einem Training unterzogen wird, wählt die Steuerschaltung **132** ein Bit aus dem

Bit-Vektor aus, um dieses in Reaktion auf den in der Zählerschaltung **134** aufrechterhaltenen Wert zu setzen. Ähnlich wählt, wenn ein Eintrag keinem Training unterzogen wird, die Steuerschaltung **132** ein Bit aus dem Bit-Vektor aus, um dieses in Reaktion auf den von der Zählerschaltung **134** aufrechterhaltenen Wert zu löschen. Jeder Wert der Zählerschaltung **134** wählt eines der Bits in dem Bit-Vektor. Die Zählerschaltung **134** weist ein Zählerregister und eine Inkrementiereinrichtung auf, der den Wert in dem Zählerregister inkrementiert. Somit inkrementiert die Zählerschaltung **134** jeden Taktzyklus. Entsprechend kann das ausgewählte Bit für ein vorgegebenes Training oder Untraining bei der bevorzugten Ausführungsform pseudozufällig sein.

[0100] Bei der vorliegenden Ausführungsform sind die Gültig-Anzeigen **4** Bit-Vektoren. Entsprechend entspricht ein Signal in Set_V[3:0] und Clear_V[3:0] jedem Bit in dem Vektor. Wenn ein Eintrag einem Training unterzogen wird, aktiviert die Steuerschaltung **132** das Set_V[3:0]-Signal entsprechend dem anhand der Zählerschaltung **134** ausgewählten Bit. In Reaktion darauf setzt die Lade-/Speicher-Abhängigkeitstabelle **80** oder **100** das entsprechende Bit in dem Bit-Vektor des weitergeschalteten Eintrags. Andererseits aktiviert, wenn ein Eintrag keinem Training unterzogen wird, die Steuerschaltung **132** das Clear_V[3:0]-Signal entsprechend dem anhand der Zählerschaltung **134** ausgewählten Bit. In Reaktion darauf löscht die Lade-/Speicher-Abhängigkeitstabelle **80** oder **100** das entsprechende Bit in dem Bit-Vektor des weitergeschalteten Eintrags. Die Steuerschaltung **132** kann ferner ein Schreib-Freigabe-Signal zum Freigeben der Aktualisierung des weitergeschalteten Eintrags liefern, falls dies gewünscht ist.

[0101] Statt der Inkrementierung des Zählerstands jedes Taktzyklus kann bei alternativen Konfigurationen der Zählerstand nach jedem Train- oder Untrain-Ereignis inkrementieren, falls dies gewünscht ist. Ferner kann bei alternativen Konfigurationen ein im Ungültig-Zustand befindliches Bit zum Umschalten in den Gültig-Zustand während des Trainings und ein im Gültig-Zustand befindliches Bit zum Umschalten zu Ungültig während des Trainings ausgewählt werden.

[0102] Fig. 8 zeigt ein Blockschaltbild eines Teils einer Ausführungsform der Abhängigkeitseinheit **62**. Andere Ausführungsformen sind möglich und werden in Betracht gezogen. Der in Fig. 8 gezeigte Teil kann sich auf das Aufrechterhalten eines ausstehende Speicherungen anzeigenden Speicherbit-Vektors beziehen. Andere (nicht gezeigte) Teile können zum Protokollieren von Abhängigkeiten von Befehlsoperationen zum Entsenden zu dem Scheduler **36** vorgesehen sein. Bei der in Fig. 8 gezeigten Ausführungsform weist die Abhängigkeitseinheit **62** eine Steuerschaltung **140** und einen mit der Steuerschaltung **140**

gekoppelten Bit-Vektor-Speicher **142** auf. Die Steuerschaltung **140** ist ferner zum Empfangen einer Anzeige des Lade-/Speichertyps der Entsende-Befehlsoperationen von den Dekodiereinheiten **24** und zugeordneten R#s von der R#-Zuweisungseinheit **64** vorgesehen. Ferner ist die Steuerschaltung **140** zum Empfangen zurückgezogener R#s und einer Abbruchanzeige von dem Scheduler **36** vorgesehen. Der Speicherbit-Vektor aus dem Bit-Vektor-Speicher **142** wird zu der Speicher-Validierschaltung **108** transportiert.

[0103] Generell empfängt beim Entsenden von Befehlsoperationen die Steuerschaltung **140** Anzeigen der Speicher-Abspeicher-Operationen von den Dekodiereinheiten **24**. Die entsprechenden R#s werden von der R#-Zuweisungseinheit **64** geliefert. Der Speicherbit-Vektor in dem Bit-Vektor-Speicher **142** weist ein Bit für jede R# auf. Die Steuerschaltung **140** setzt die Bits in dem Steuerbit-Vektor, welche Entsende-Speicherungen entsprechen. Ähnlich setzt, wenn Speicherungen von dem Scheduler **36** zurückgesetzt und über die Rückzieh-R#s angezeigt werden, die Steuerschaltung **140** die entsprechenden Bits in dem Speicherbit-Vektor zurück. Schließlich setzt, wenn ein Abbruch signalisiert wird, die Steuerschaltung **140** die Bits der abgebrochenen Speicherungen zurück. Bei einer Ausführungsform können die Abbrüche signalisiert werden, wenn die den Abbruch bewirkende Befehlsoperation zurückgezogen wird. Somit kann die Abbruchanzeige ein zum Löschen des Speicherbit-Vektors verwendetes Signal sein. Bei anderen Ausführungsformen kann die Abbruchanzeige die R# des Abbruchbefehls identifizieren und können nur jüngere Speicherungen abgebrochen werden.

[0104] Wie hier verwendet, bedeutet der Ausdruck "Steuerschaltung" eine Schaltungsanordnung, die zum Erzeugen von Ausgängen auf Eingängen arbeitet, wie oben beschrieben. Generell kann die Steuerschaltung eine beliebige Kombination aus kombinatorischer Logik (statisch oder dynamisch), Zustandsmaschinen, auf den Kundenwunsch zugeschnittener Schaltungsanordnung und getakteten Speichervorrichtungen (wie z. B. Flops, Register etc.) aufweisen.

Computersysteme

[0105] Fig. 9 zeigt ein Blockschaltbild einer Ausführungsform eines Computersystems **200** mit einem über eine Busbrücke **202** mit einer Vielzahl von Systemkomponenten gekoppelten Prozessor **10**. Andere Ausführungsformen sind möglich und werden in Betracht gezogen. Bei dem dargestellte System ist ein Hauptspeicher **204** über einen Speicherbus **206** mit der Busbrücke **202** gekoppelt und ein Grafikkontrollen **208** über einen AGP-Bus **210** mit der Busbrücke **202** gekoppelt. Schließlich sind mehrere PCI-Vorrichtungen **212A-212B** über einen PCI-Bus **214** mit der Busbrücke **202** gekoppelt. Eine Sekundärbusbrücke

216 kann ferner zum Aufnehmen eines elektrischen Interface über einen EISA-/ ISA-Bus **220** zu einer oder mehreren EISA- oder ISA-Vorrichtungen **218** vorgesehen sein. Der Prozessor **10** ist über einen CPU-Bus **224** mit einer Busbrücke **202** und mit einem wahlweise vorgesehenen L2-Cachespeicher **228** gekoppelt. Zusammen können der CPU-Bus **224** und das Interface zu dem L2-Cachespeicher **228** das Extern-Interface **52** aufweisen.

[0106] Die Busbrücke **202** bildet ein Interface zwischen dem Prozessor **10**, dem Hauptspeicher **204**, dem Grafik-Kontroller **208** und mit dem PCI-Bus **214** verbundenen Vorrichtungen. Wenn eine Operation von einer der mit der Busbrücke **202** verbundenen Vorrichtungen empfangen wird, identifiziert die Busbrücke **202** das Target der Operation (z. B. eine spezielle Vorrichtung oder im Falle der PCI-Busbrücke **214**, dass sich das Target auf dem PCI-Bus **214** befindet. Die Busbrücke **202** leitet die Operation zu der Target-Vorrichtung. Die Busbrücke **202** setzt generell eine Operation von dem von der Quellen-Vorrichtung oder dem Quellen-Bus verwendeten Protokoll in das von der Target-Vorrichtung oder dem Target-Bus verwendete Protokoll um.

[0107] Zusätzlich zu dem Bilden eines Interface zu einem ISA/EISA-Bus für den PCI-Bus **214** kann die Sekundärbusbrücke **216** ferner eine zusätzliche Funktionalität aufweisen, falls dies gewünscht ist. Ein (nicht gezeigter) Eingangs-/Ausgangskontroller, der entweder außerhalb der Sekundärbusbrücke **216** angeordnet oder in diese integriert ist, kann auch in dem Computersystem **200** vorgesehen sein, um eine Operationsunterstützung für eine Tastatur und eine Maus **222** und für verschiedene Seriell- und Parallelports zu bieten, falls dies gewünscht ist. Eine (nicht gezeigte) externe Cachespeichereinheit kann bei anderen Ausführungsformen ferner zwischen dem Prozessor **10** und der Busbrücke **202** mit dem CPU-Bus **224** gekoppelt sein. Alternativ kann der externe Cachespeicher mit der Busbrücke **202** gekoppelt sein und kann die Cachespeicher-Steuerlogik für den externen Cachespeicher in die Busbrücke **202** integriert sein. Der L2-Cachespeicher **228** ist ferner als an der Rückseite des Prozessors **10** angeordnet dargestellt. Es sei darauf hingewiesen, dass der L2-Cachespeicher **228** von dem Prozessor **10** getrennt, mit dem Prozessor **10** in eine Cartridge (z. B. Schlitz **1** oder Schlitz A) integriert oder sogar mit dem Prozessor **10** in ein Halbleitersubstrat integriert sein kann.

[0108] Der Hauptspeicher **204** ist ein Speicher, in dem Anwendungsprogramme gespeichert sind und aus dem heraus der Prozessor **10** hauptsächlich ausführt.

[0109] Ein geeigneter Hauptspeicher **204** weist einen DRAM (dynamischen Direktzugriffsspeicher) auf. Beispielsweise kann eine Vielzahl von SDRAM-

(synchronen DRAM-) oder Rambus-DRAM-(RDRAM-) Banken geeignet sein.

[0110] Die PCI-Vorrichtungen **212A-212B** sind als Vielzahl von Peripherievorrichtungen, wie beispielsweise Netz-Interface-Karten, Videobeschleunigern, Audiokarten, Festplatten- oder Diskettenlaufwerken oder Laufwerkskontrollern, SCSI- (Kleincomputersystem-Interface-) Adaptionen und Telefonkarten, dargestellt. Ähnlich ist die ISA-Vorrichtung **218** in Form verschiedener Peripherievorrichtungen, wie z. B. eines Modem, einer Soundkarte und einer Vielzahl von Datenerfassungskarten, wie z. B. GPIB- oder Feldbus-Interface-Karten, dargestellt.

[0111] Der Grafikkontroller **208** ist zum Steuern der Wiedergabe von Text und Bildern auf einer Anzeige **226** vorgesehen. Der Grafikkontroller **208** kann ein typischer Grafikbeschleuniger sein, der beim Stand der Technik zum Wiedergeben dreidimensionaler Datenstrukturen, die auf effektive Weise in den und aus dem Hauptspeicher **204** verschoben werden können, bekannt ist. Der Grafikkontroller **208** kann daher dahingehend ein Master des AGP-Busses **210** sein, dass er Zugriff auf ein Target-Interface in der Busbrücke **202** anfordern und erhalten kann, um dadurch Zugriff auf den Hauptspeicher **204** zu erhalten. Ein zweckbestimmter Grafikbus führt eine schnelle Wiedergewinnung von Daten aus dem Hauptspeicher **204** durch. Bei bestimmten Operationen kann der Grafikkontroller **208** ferner zum Erzeugen von PCI-Protokoll-Transaktionen auf dem AGP-Bus **210** vorgesehen sein. Das AGP-Interface der Busbrücke **202** kann somit eine Funktionalität zum Unterstützen sowohl von AGP-Protokoll-Transaktionen als auch PCI-Protokoll-Target- und -Initiator-Transaktionen aufweisen. Die Anzeige **226** ist eine beliebige elektronische Anzeige, auf der ein Bild oder Text dargestellt werden kann. Eine geeignete Anzeige **226** weist eine Kathodenstrahlröhre ("CRT"), eine Flüssigkristallanzeige ("LCD") etc. auf.

[0112] Es sei darauf hingewiesen, dass die AGuP-, PCI- und ISA- oder EISA-Busse in der vorstehenden Beschreibung zwar beispielhaft dargestellt sind, stattdessen jedoch beliebige Busarchitekturen verwendet werden können, falls dies gewünscht ist. Es sei ferner darauf hingewiesen, dass das Computersystem **200** ein Multiprozessor-Computersystem mit zusätzlichen Prozessoren (z. B. Prozessor **10a**, der als wahlweise vorgesehene Komponente des Computersystems **200** gezeigt ist) sein kann. Der Prozessor **10a** kann dem Prozessor **10** im wesentlichen gleich sein. Insbesondere kann der Prozessor **10a** eine identische Kopie des Prozessors **10** sein. Der Prozessor **10a** kann über einen unabhängigen Bus mit der Busbrücke **202** verbunden sein (wie in Fig. 9 gezeigt) oder kann den CPU-Bus **224** gemeinsam mit dem Prozessor **10** nutzen. Ferner kann der Prozessor **10a** mit einem wahlweise vorgesehenen L2-Ca-

chespeicher **228a**, der dem L2-Cachespeicher **228** im wesentlichen gleich ist, gekoppelt sein.

[0113] Fig. 10 zeigt eine weitere Ausführungsform eines Computersystems **300**. Andere Ausführungsformen sind möglich und werden in Betracht gezogen. Bei der in Fig. 10 gezeigten Ausführungsform weist das Computersystem **300** mehrere Verarbeitungs-Schaltungspunkte **312A**, **312B**, **312C** und **312D** auf. Jeder Verarbeitungs-Schaltungspunkt ist über einen Speicherkontroller **316A–316D** in den jeweiligen Verarbeitungs-Schaltungspunkten **312A–312D** mit einem jeweiligen Speicher **314A–314D** gekoppelt. Ferner weisen die Verarbeitungs-Schaltungspunkte **312A–312D** eine Interface-Logik zum Kommunizieren zwischen den Verarbeitungs-Schaltungspunkten **312A–312D** auf. Beispielsweise weist der Verarbeitungs-Schaltungspunkt **312A** eine Interface-Logik **318A** zum Kommunizieren mit dem Verarbeitungs-Schaltungspunkt **312B**, eine Interface-Logik **318B** zum Kommunizieren mit dem Verarbeitungs-Schaltungspunkt **312C** und eine dritte Interface-Logik **318C** zum Kommunizieren mit einem weiteren (nicht gezeigten) Verarbeitungs-Schaltungspunkt auf. Ähnlich weist der Verarbeitungs-Schaltungspunkt **312B** eine Interface-Logik **318D**, **318E** und **318F** auf; weist der Verarbeitungs-Schaltungspunkt **312C** eine Interface-Logik **318G**, **318H** und **318I** auf; und weist der Verarbeitungs-Schaltungspunkt **312D** eine Interface-Logik **318J**, **318K** und **318L** auf. Der Verarbeitungs-Schaltungspunkt **312D** ist zum Kommunizieren über die Interface-Logik **318L** mit mehreren Eingangs-/Ausgangsvorrichtungen (z. B. Vorrichtungen **320A–320B** in einer Daisy-Chain-Konfiguration) vorgesehen. Weitere Verarbeitungs-Schaltungspunkte können auf im wesentlichen gleiche Weise mit weiteren I/O-Vorrichtungen kommunizieren.

[0114] Die Verarbeitungs-Schaltungspunkte **312A–312B** implementieren eine Verknüpfung auf Paketbasis für die Kommunikation zwischen den Verarbeitungs-Schaltungspunkten. Bei der vorliegenden Ausführungsform wird die Verknüpfung in Form von Sätzen unidirektionaler Leitungen implementiert (z. B. werden Leitungen **324A** zum Übermitteln von Paketen von dem Verarbeitungs-Schaltungspunkt **312A** zu dem Verarbeitungs-Schaltungspunkt **312B** und Leitungen **324B** zum Übermitteln von Paketen von dem Verarbeitungs-Schaltungspunkt **312B** zu dem Verarbeitungs-Schaltungspunkt **312A** verwendet). Weitere Sätze von Leitungen **324C–324H** werden zum Übermitteln von Paketen zwischen weiteren Verarbeitungs-Schaltungspunkten verwendet, wie in Fig. 10 gezeigt. Generell kann jeder Satz von Leitungen **324** eine oder mehrere Datenleitungen, eine oder mehrere Taktleitungen entsprechend den Datenleitungen und eine oder mehrere Steuerleitungen, die den Typ des transportierten Pakets anzeigen, aufweisen. Die Verknüpfung kann zur Kommunikation

zwischen Verarbeitungs-Schaltungspunkten in einer cachespeicher-kohärenten Weise und zur Kommunikation zwischen einem Verarbeitungs-Schaltungspunkt und einer I/O-Vorrichtung (oder einer Busbrücke herkömmlicher Konstruktion, wie z. B. dem PCI- oder ISA-Bus) in einer nichtkohärenten Weise arbeiten. Ferner kann die Verknüpfung, wie dargestellt, in einer nichtkohärenten Weise unter Verwendung einer Daisy-Chain-Struktur zwischen I/O-Vorrichtungen arbeiten. Es sei darauf hingewiesen, dass ein von einem Verarbeitungs-Schaltungspunkt zu einem anderen zu übermittelndes Paket einen oder mehreren Zwischen-Schaltungspunkten durchlaufen kann. Beispielsweise kann ein von dem Verarbeitungs-Schaltungspunkt **312A** zu dem Verarbeitungs-Schaltungspunkt **312D** übermitteltes Paket entweder den Verarbeitungs-Schaltungspunkte **312B** oder den Verarbeitungs-Schaltungspunkt **312C** durchlaufen, wie in Fig. 10 gezeigt. Es kann ein beliebiger geeigneter Leit-Algorithmus verwendet werden. Andere Ausführungsformen des Computersystems **300** können mehr oder weniger Verarbeitungs-Schaltungspunkte als die in Fig. 10 gezeigte Ausführungsform aufweisen.

[0115] Generell können die Pakete als Ein- oder Mehrbitzeiten auf den Leitungen **324** zwischen den Schaltungspunkten übermittelt werden. Eine Bitzeit kann die ansteigende oder abfallende Flanke eines Taktsignals auf den entsprechenden Taktleitungen sein. Die Pakete können Befehls-Pakete zum Initiieren von Transaktionen, Sondier-Pakete zum Aufrechterhalten der Cachespeicher-Kohärenz und Ansprech-Pakete zum Ansprechen auf Sondierungen und Befehle sein.

[0116] Die Verarbeitungs-Schaltungspunkte **312A–312D** können zusätzlich zu einem Speicherkontroller und einer Interface-Logik einen oder mehrere Prozessoren aufweisen. Im weiteren Sinne weist ein Verarbeitungs-Schaltungspunkt mindestens einen Prozessor auf und kann wahlweise mit einem Speicherkontroller zum Kommunizieren mit einem Speicher und einer weiteren Logik versehen sein, falls dies gewünscht ist. Insbesondere kann ein Verarbeitungs-Schaltungspunkt **312A–312D** den Prozessor **10** aufweisen. Die Extern-Interface-Einheit **46** kann die Interface-Logik **318** in dem Schaltungspunkt sowie den Speicherkontroller **316** aufweisen.

[0117] Die Speicher **314A–314D** können beliebige geeignete Speichervorrichtungen aufweisen. Beispielsweise kann ein Speicher **314A–314D** einen oder mehrere RAMBUS DRAMs (RDRAMs), Synchron-DRAMs (SDRAMs), statische RAM etc. aufweisen. Der Adressenraum des Computersystems **300** ist unter den Speichern **314A–314D** aufgeteilt. Jeder Verarbeitungs-Schaltungspunkt **312A–312D** kann einen Speicherbelegungsplan aufweisen, anhand dessen bestimmt wird, welche Adressen auf

welche Speicher **314A–314D** abgebildet werden und somit zu welchen Verarbeitungs-Schaltungspunkt **312A–312D** eine Speicheranforderung für eine spezielle Adresse geleitet werden soll. Bei einer Ausführungsform ist der Kohärenzpunkt für eine Adresse in dem Computersystem **300** der Speichercontroller **316A–316D**, der mit dem der Adresse entsprechenden Bytes speichernden Speicher gekoppelt ist. Mit anderen Worten: der Speichercontroller **316A–316D** muss gewährleisten, dass jeder Speicherzugriff auf den entsprechenden Speicher **314A–314D** in einer cachespeicher-kohärenten Weise erfolgt. Die Speichercontroller **316A–316D** können Steuerschaltungsanordnungen zum Anschließen an die Speicher **314A–314D** aufweisen. Ferner können die Speichercontroller **316A–316D** Warteschlangen zum Einreihen von Speicheranforderungen aufweisen.

[0118] Generell kann die Interface-Logik **318A–318L** eine Vielzahl von Puffern zum Empfangen von Paketen von der Verknüpfung und zum Puffern von auf der Verknüpfung zu übermittelnden Paketen aufweisen. Das Computersystem **300** kann einen beliebigen geeigneten Ablaufsteuermechanismus zum Übermitteln von Paketen verwenden. Beispielsweise speichert bei einer Ausführungsform jede Interface-Logik **318** einen Zählerstand der Anzahl jedes Puffertyps innerhalb des Empfängers am anderen Ende der Verknüpfung, mit dem diese Interface-Logik verbunden ist. Die Interface-Logik übermittelt kein Paket, solange die Empfangs-Interface-Logik keinen freien Puffer zum Speichern des Pakets hat. Wenn ein Empfangspuffer durch Weiterleiten eines Pakets frei geworden ist, übermittelt die Empfangs-Interface-Logik eine Mitteilung an die Send-Interface-Logik, um anzuzeigen, dass der Puffer frei geworden ist. Ein solcher Mechanismus kann als System "auf Coupon-Basis" bezeichnet werden.

[0119] Die I/O-Vorrichtungen **320A–320B** können beliebige geeignete I/O-Vorrichtungen sein. Beispielsweise können die I/O-Vorrichtungen **320A–320B** Netz-Interface-Karten, Videobeschleuniger, Audiokarten, Festplatten- oder Diskettenlaufwerke oder Laufwerkscontroller, SCSI- (Kleincomputersystem-Interface-) Adapter und Telefonkarten, Modems, Soundkarten und eine Vielzahl von Datenerfassungskarten, wie z. B. GPIB- oder Feldbus-Interface-Karten, aufweisen.

[0120] Zahlreiche Variationen und Modifikationen werden für Fachleute auf dem Sachgebiet anhand der vorstehenden Offenbarung offensichtlich. Die nachfolgenden Patentansprüche müssen derart ausgelegt werden, dass sie sämtliche Variationen und Modifikationen umfassen.

INDUSTRIELLE ANWENDBARKEIT

[0121] Die vorliegende Erfindung ist auf Prozesso-

ren und Computersysteme anwendbar.

Patentansprüche

1. Prozessor (**10**) mit:
einer Speicher-zu-Lade-Weiterleitungs-STLF-Vorhersageeinrichtung (**60**) zum Empfangen einer Entsende-Anzeige einer ersten Speicher-Lade-Speicher-Operation, wobei die STLF-Vorhersageeinrichtung (**60**) in Reaktion auf die erste Speicher-Lade-Speicher-Operation folgende Abhängigkeit anzeigt:
der ersten Speicher-Lade-Speicher-Operation von einer ersten Speicher-Abspeicher-Operation in Reaktion auf in der STLF-Vorhersageeinrichtung (**60**) gespeicherte Informationen, die der Speicher-Lade-Speicher-Operation entsprechen und anzeigen, dass während einer vorangegangenen Ausführung die erste Speicher-Abspeicher-Operation mit der ersten Speicher-Lade-Speicher-Operation interferiert hat; und einer mit der STLF-Vorhersageeinrichtung (**60**) gekoppelten Ausführungs-Pipeline (**72**), die zum Sperren der Ausführung der ersten Speicher-Lade-Speicher-Operation vor der ersten Speicher-Abspeicher-Operation in Reaktion auf die Abhängigkeit und zum Detektieren eines Fehlens der Abhängigkeit während der Ausführung der ersten Speicher-Lade-Speicher-Operation und zum Erzeugen eines Untrain-Signals in Reaktion auf das Fehlen der Abhängigkeit vorgesehen ist;
wobei die STLF-Vorhersageeinrichtung (**60**) zum Empfangen des Untrain-Signals und zum Aktualisieren der darin gespeicherten Informationen derart vorgesehen ist, dass nicht angezeigt wird, dass die erste Speicher-Abspeicher-Operation während der vorangegangenen Ausführung mit der ersten Speicher-Lade-Speicher-Operation interferiert hat.
2. Prozessor nach Anspruch 1, bei dem die Informationen eine Gültig-Anzeige entsprechend der ersten Speicher-Lade-Speicher-Operation enthalten und die STLF-Vorhersageeinrichtung (**60**) zum Setzen der Gültig-Anzeige in einen Ungültig-Zustand zwecks Aktualisierung der Informationen vorgesehen ist.
3. Prozessor nach Anspruch 1, bei dem die Informationen eine Gültig-Anzeige entsprechend der ersten Speicher-Lade-Speicher-Operation enthalten und die Gültig-Anzeige einen Bit-Vektor aufweist und die Gültig-Anzeige in einem Gültig-Zustand ist, wenn mindestens ein Bit in einem Gültig-Zustand ist, und die STLF-Vorhersageeinrichtung (**60**) zum Setzen eines ersten Bits des Bit-Vektors in einen Ungültig-Zustand zwecks Aktualisierung der Informationen vorgesehen ist.
4. Prozessor nach Anspruch 1, bei dem die erste Speicher-Abspeicher-Operation mit der ersten Speicher-Lade-Speicher-Operation interferiert, wenn die

erste Speicher-Lade-Speicher-Operation vor dem Einplanen der ersten Speicher-Abspeicher-Operation eingeplant ist und die erste Speicher-Lade-Speicher-Operation von der ersten Speicher-Abspeicher-Operation abhängig ist.

5. Prozessor nach Anspruch 1, bei dem die Ausführungs-Pipeline (72) eine Speicher-Warteschlange (68) aufweist und zum Detektieren des Fehlens einer Abhängigkeit vorgesehen ist, wenn das Weiterleiten von Daten von der Speicher-Warteschlange (68) für die erste Speicher-Lade-Speicher-Operation nicht erfolgt.

6. Verfahren mit folgenden Schritten:
Entsenden einer ersten Speicher-Lade-Speicher-Operation und in Reaktion auf das Entsenden der ersten Speicher-Lade-Speicher-Operation Anzeigen einer Abhängigkeit der ersten Speicher-Lade-Speicher-Operation von einer ersten Speicher-Abspeicher-Operation in Reaktion auf Informationen in einer Speicher-zu-Lade-Weiterleitungs-STLF-Vorhersageeinrichtung (60) und entsprechend der ersten Speicher-Lade-Speicher-Operation, wobei die Informationen anzeigen, dass während einer vorangegangenen Ausführung die erste Speicher-Abspeicher-Operation mit der ersten Speicher-Lade-Speicher-Operation interferiert hat; Sperren des Einplanens der ersten Speicher-Lade-Speicher-Operation vor dem Einplanen der ersten Speicher-Abspeicher-Operation; Detektieren eines Fehlens der Abhängigkeit während der Ausführung der ersten Speicher-Lade-Speicher-Operation; und Aktualisieren der Informationen, die anzeigen, dass während der vorangegangenen Ausführung die erste Speicher-Abspeicher-Operation derart mit der ersten Speicher-Lade-Speicher-Operation interferiert hat, dass nicht angezeigt wird, dass während der vorangegangenen Ausführung die erste Speicher-Abspeicher-Operation in Reaktion auf das Detektieren mit der ersten Speicher-Lade-Speicher-Operation interferiert hat.

7. Verfahren nach Anspruch 6, bei dem die Informationen eine Gültig-Anzeige entsprechend der ersten Speicher-Lade-Speicher-Operation enthalten und das Aktualisieren das Setzen der Gültig-Anzeige in einen Ungültig-Zustand umfasst.

8. Verfahren nach Anspruch 6, bei dem die Informationen eine Gültig-Anzeige entsprechend der ersten Speicher-Lade-Speicher-Operation enthalten und die Gültig-Anzeige einen Bit-Vektor aufweist und das Anzeigen durchgeführt wird, wenn mindestens ein Bit in dem Bit-Vektor in einem Gültig-Zustand ist, und das Aktualisieren das Selektieren eines ersten Bits des Bit-Vektors und das Setzen des ersten Bits in einen Ungültig-Zustand umfasst.

9. Verfahren nach Anspruch 6, bei dem das Detektieren das Detektieren eines Fehlens des Weiterleitens von Daten von einer Speicher-Warteschlange (68) für die erste Speicher-Lade-Speicher-Operation umfasst.

10. Speicher-zu-Lade-Weiterleitungs-STLF-Vorhersageeinrichtung (60) zum Anzeigen einer Abhängigkeit einer ersten Speicher-Lade-Speicher-Operation von einer ersten Speicher-Abspeicher-Operation, wobei die STLF-Vorhersageeinrichtung Informationen entsprechend der ersten Speicher-Lade-Speicher-Operation speichert und die Abhängigkeit in Reaktion auf eine Anzeige der ersten Speicher-Lade-Speicher-Operation erzeugt.

Es folgen 10 Blatt Zeichnungen

Anhängende Zeichnungen

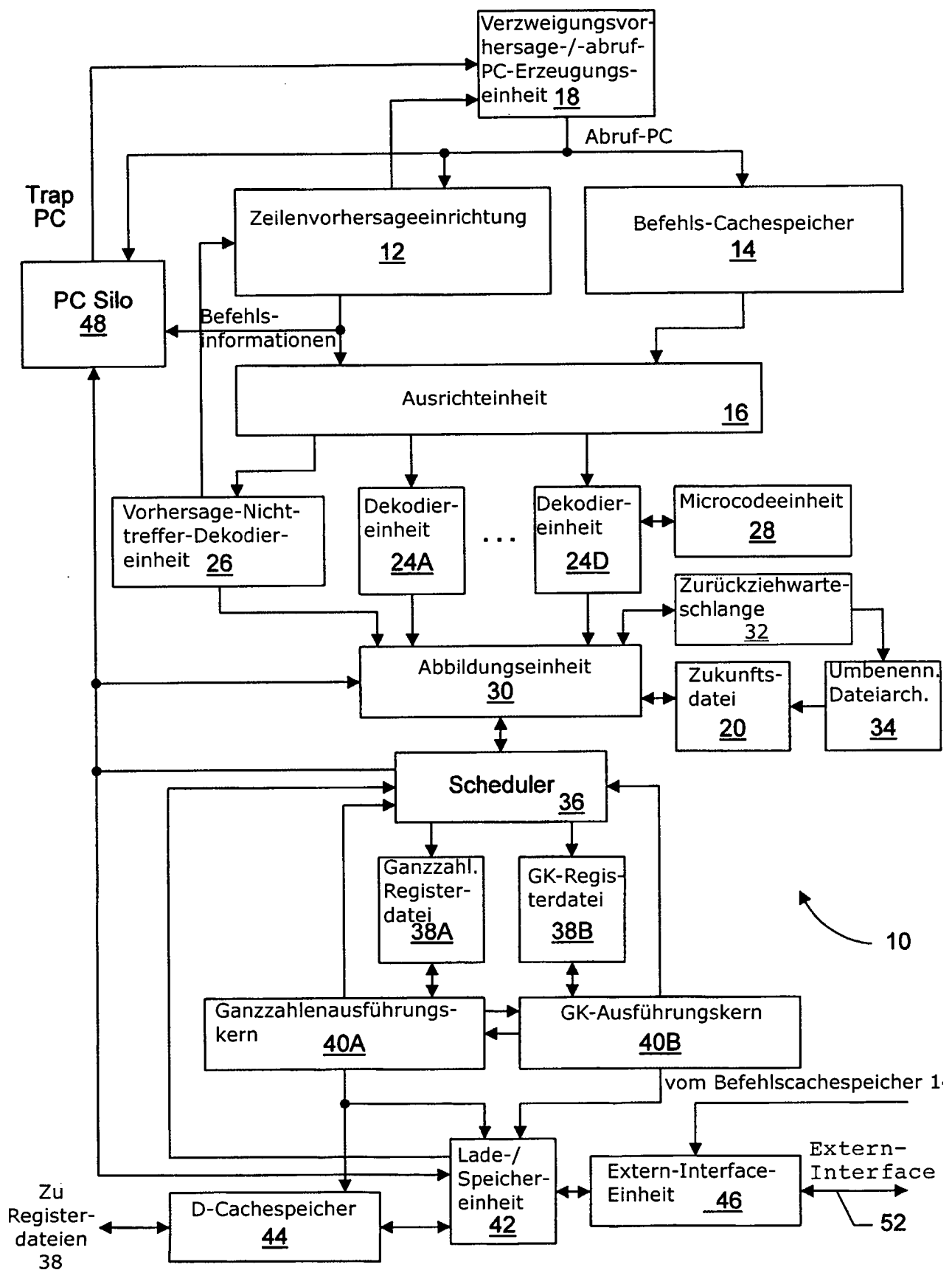


Fig. 1

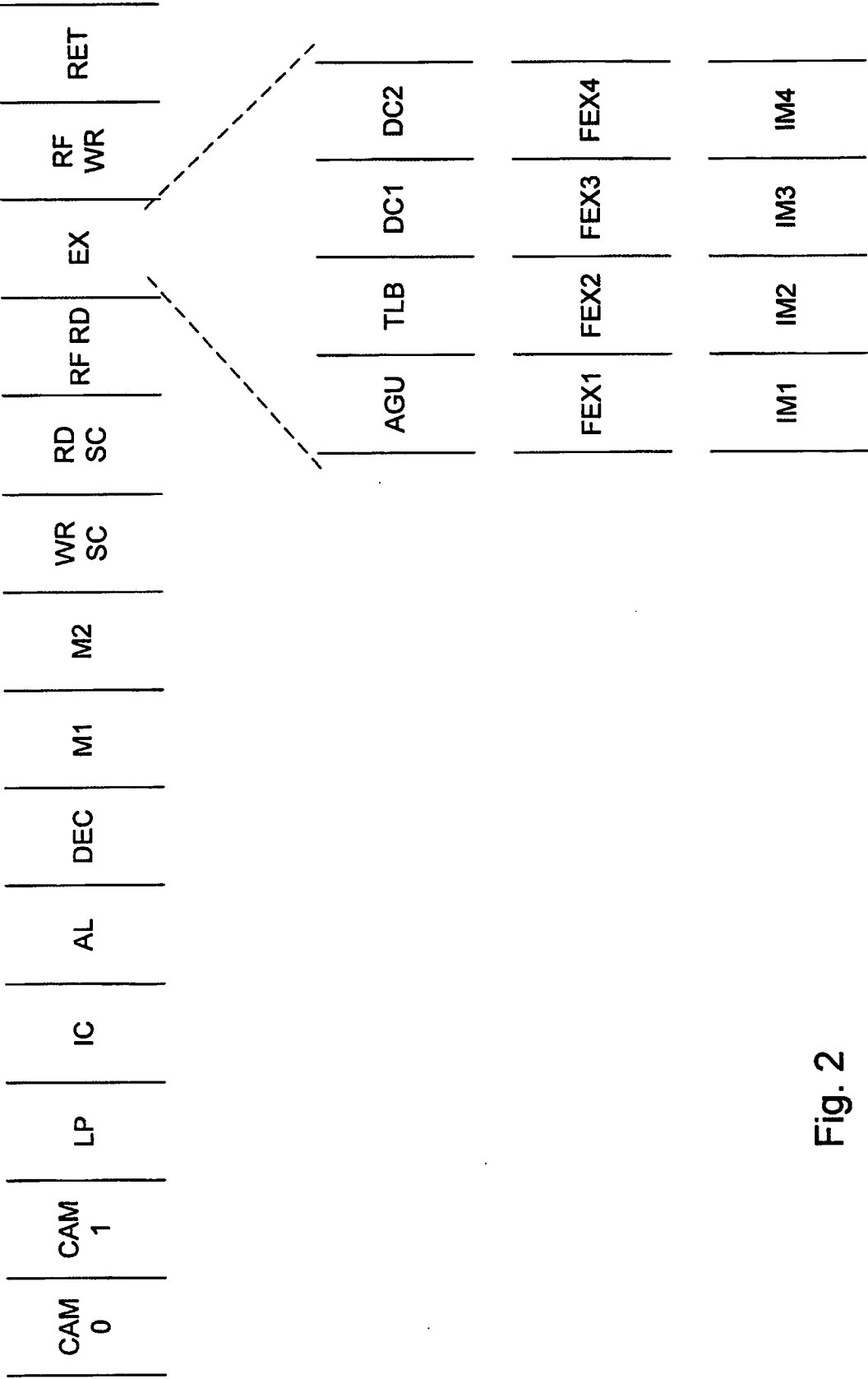


Fig. 2

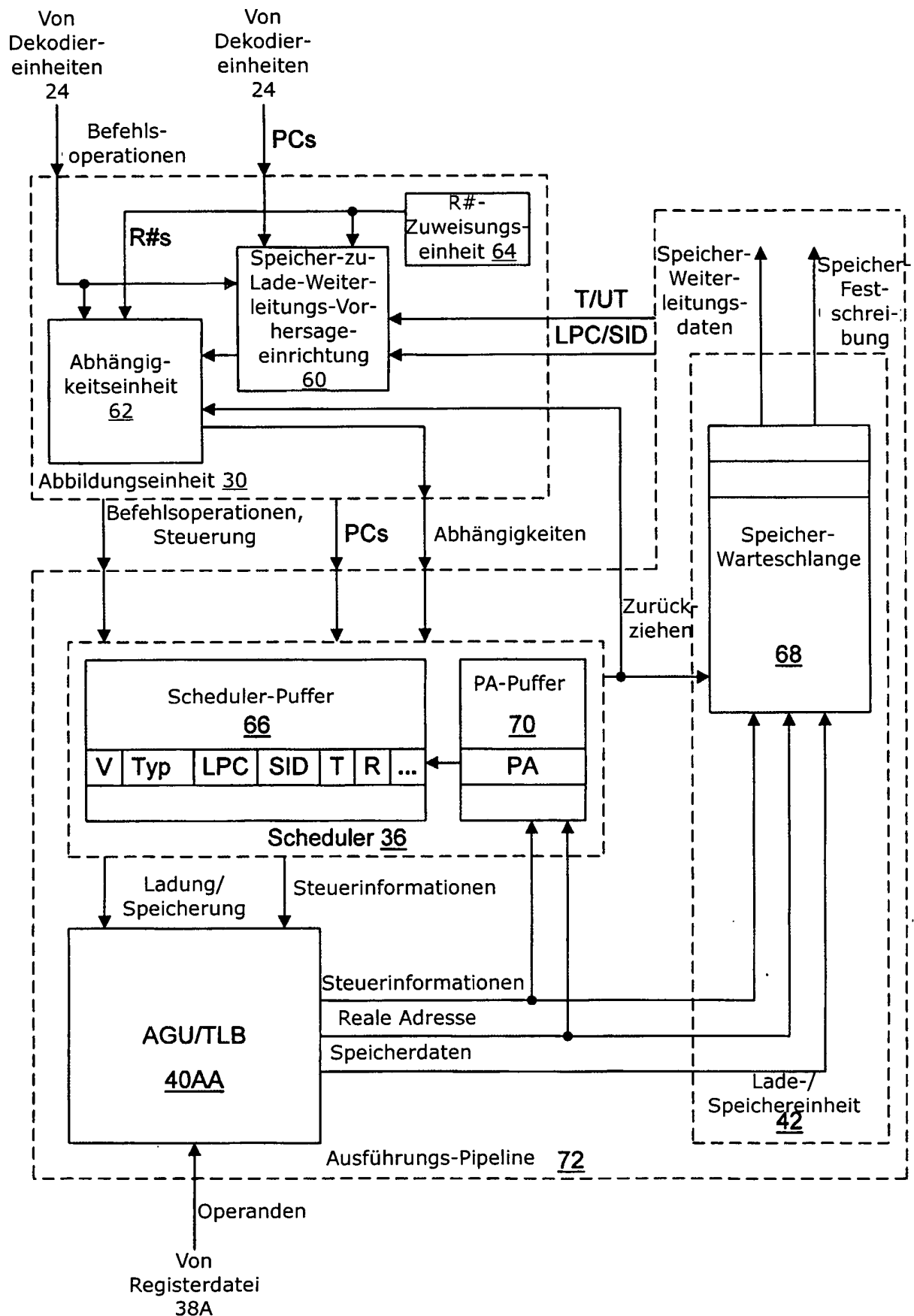


Fig. 3

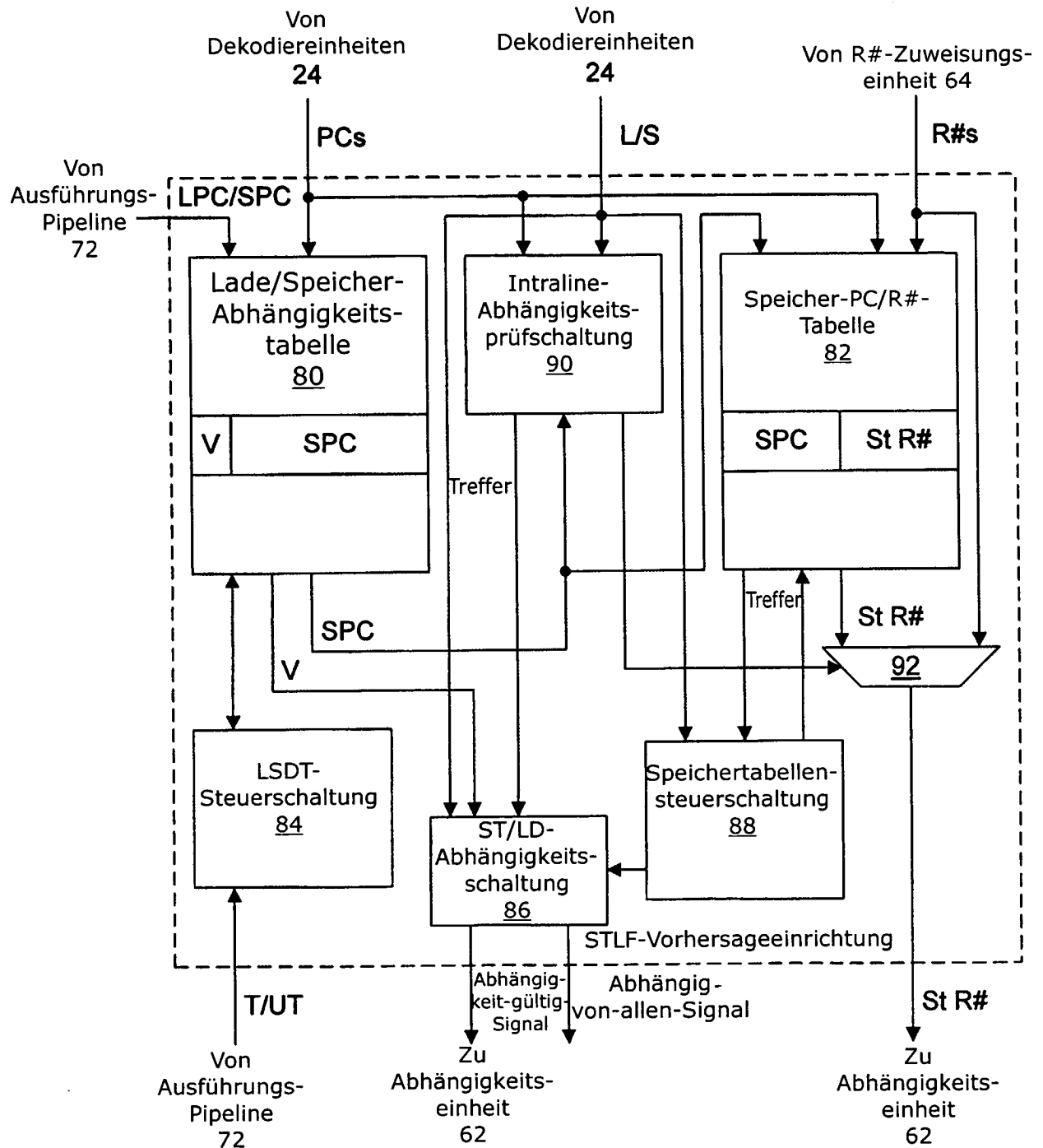


Fig. 4

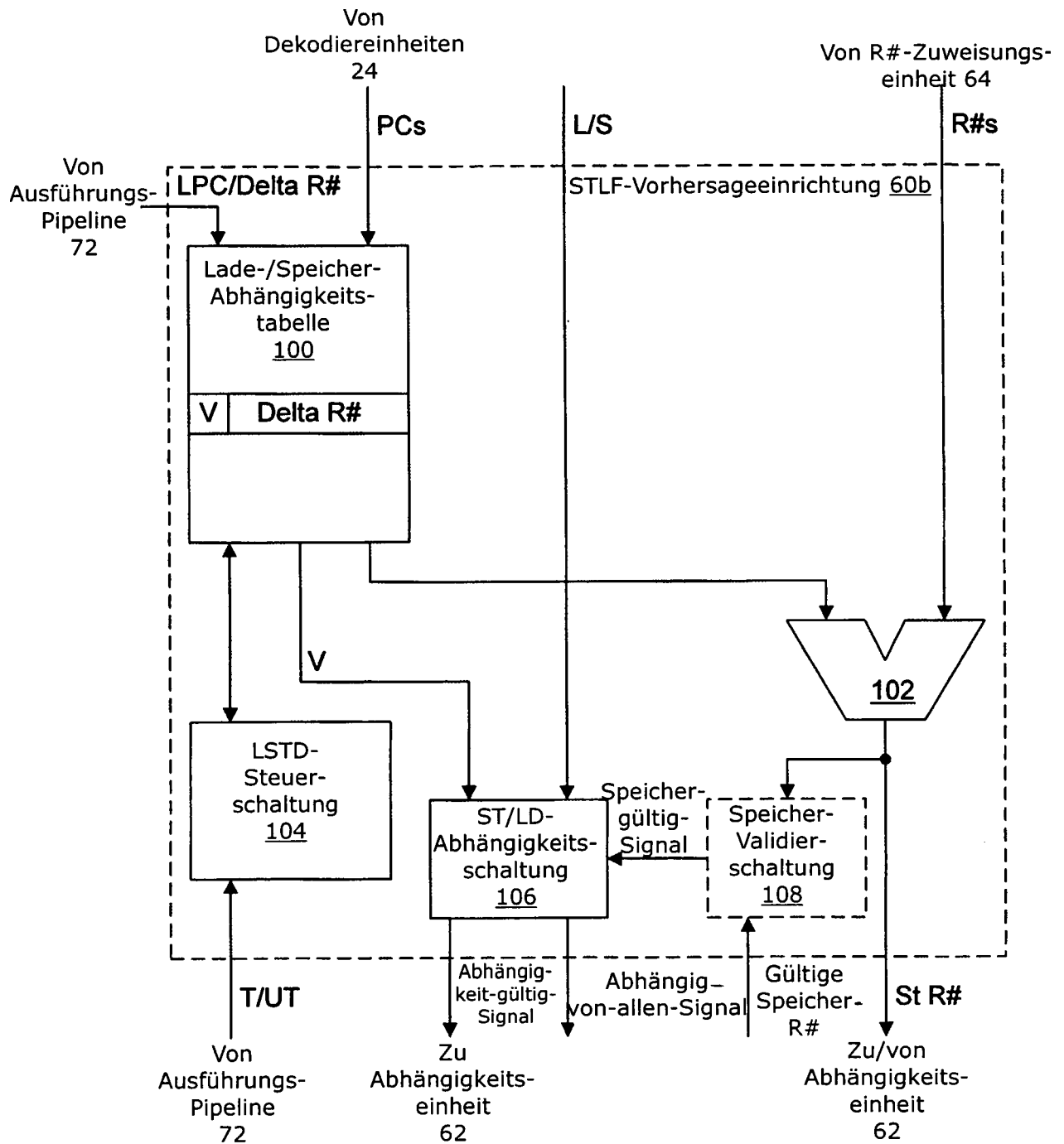


Fig. 5

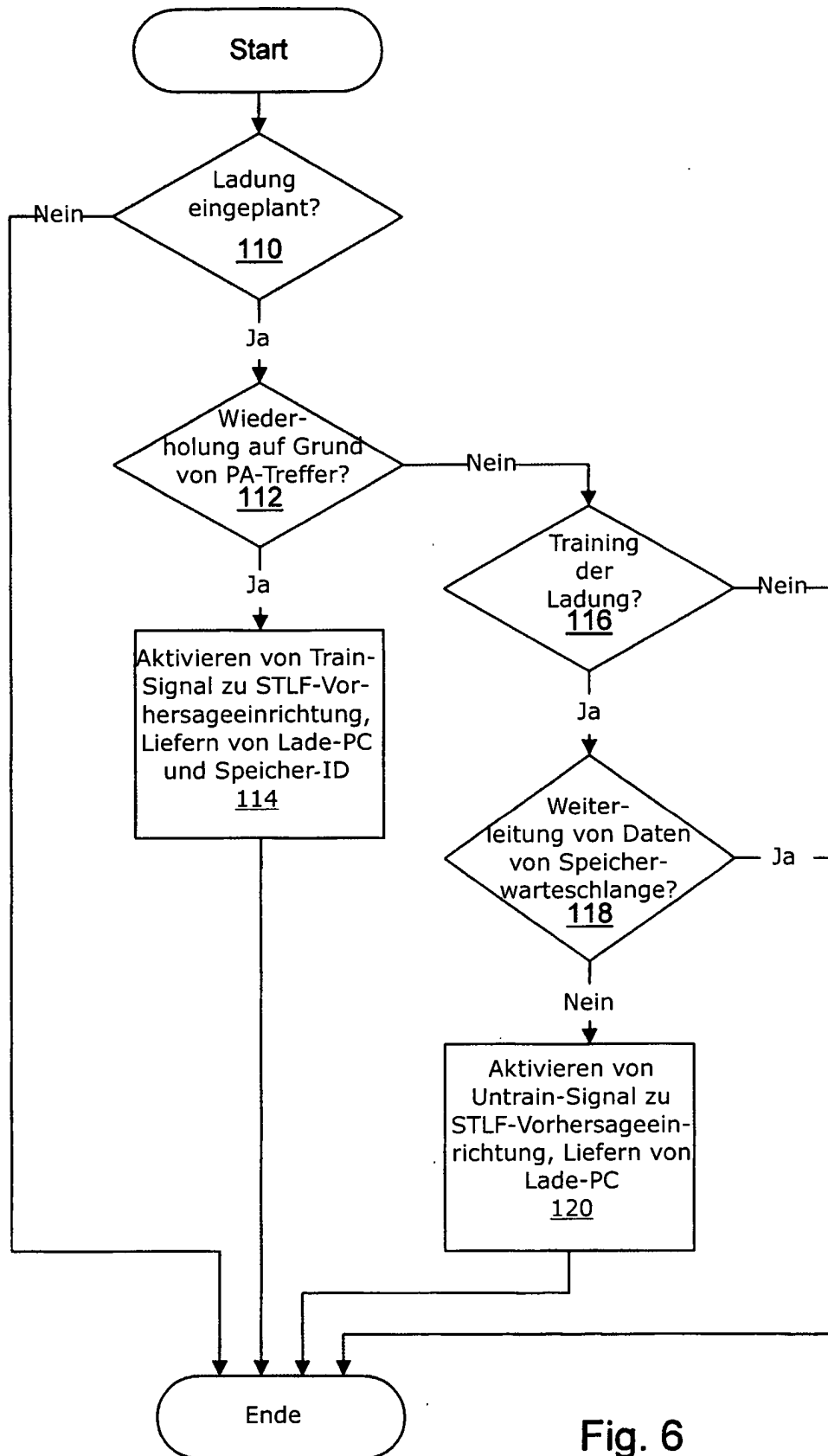


Fig. 6

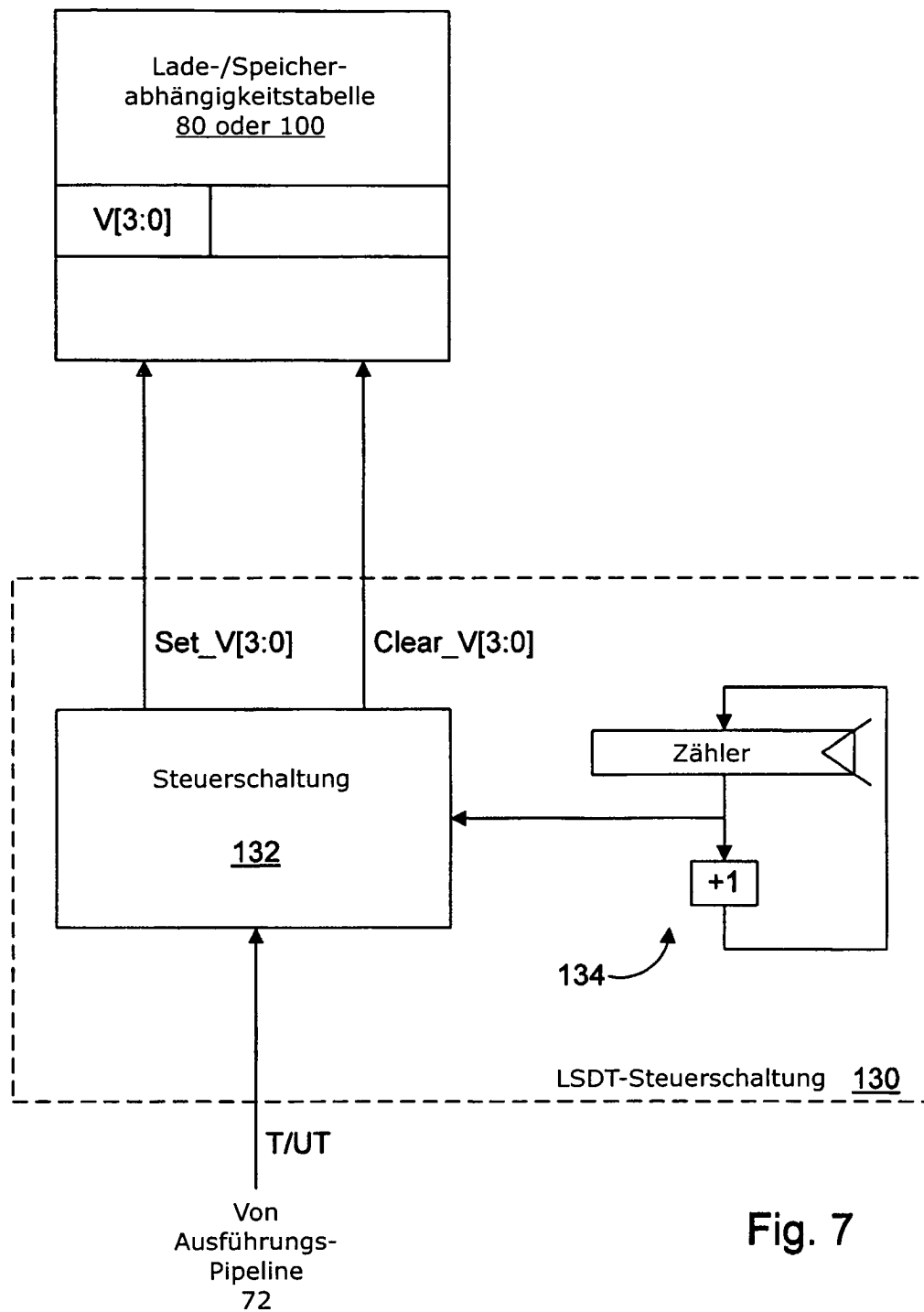


Fig. 7

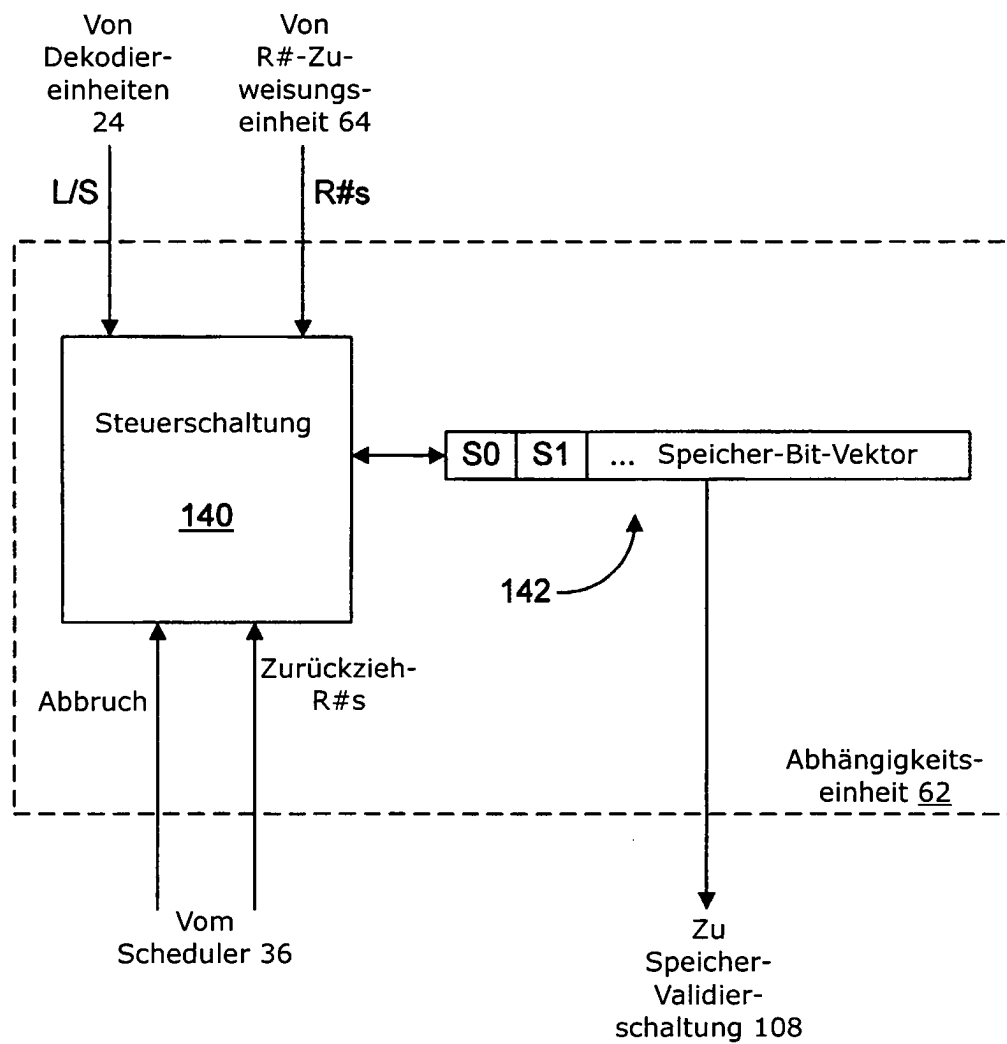


Fig. 8

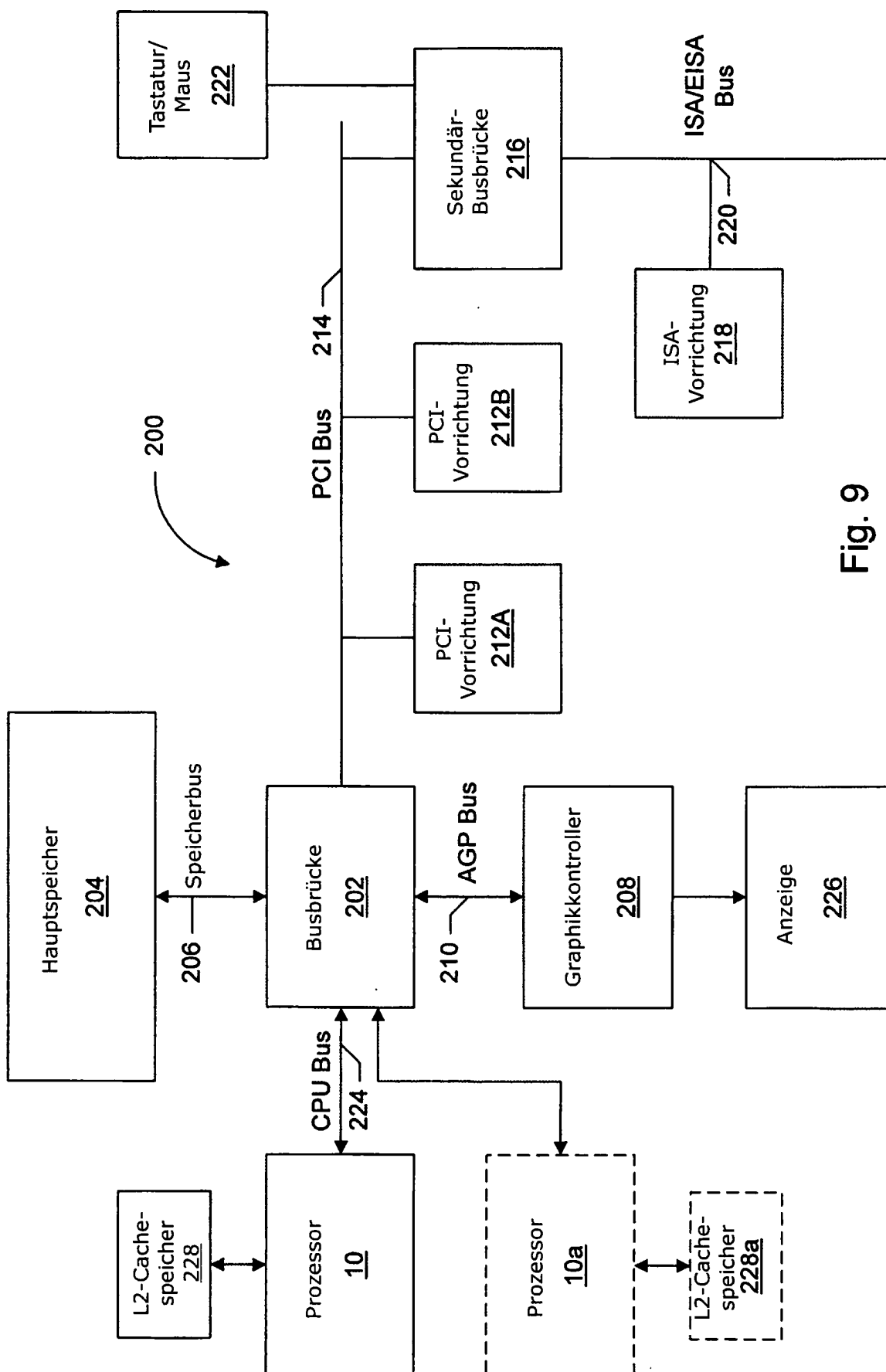


Fig. 9

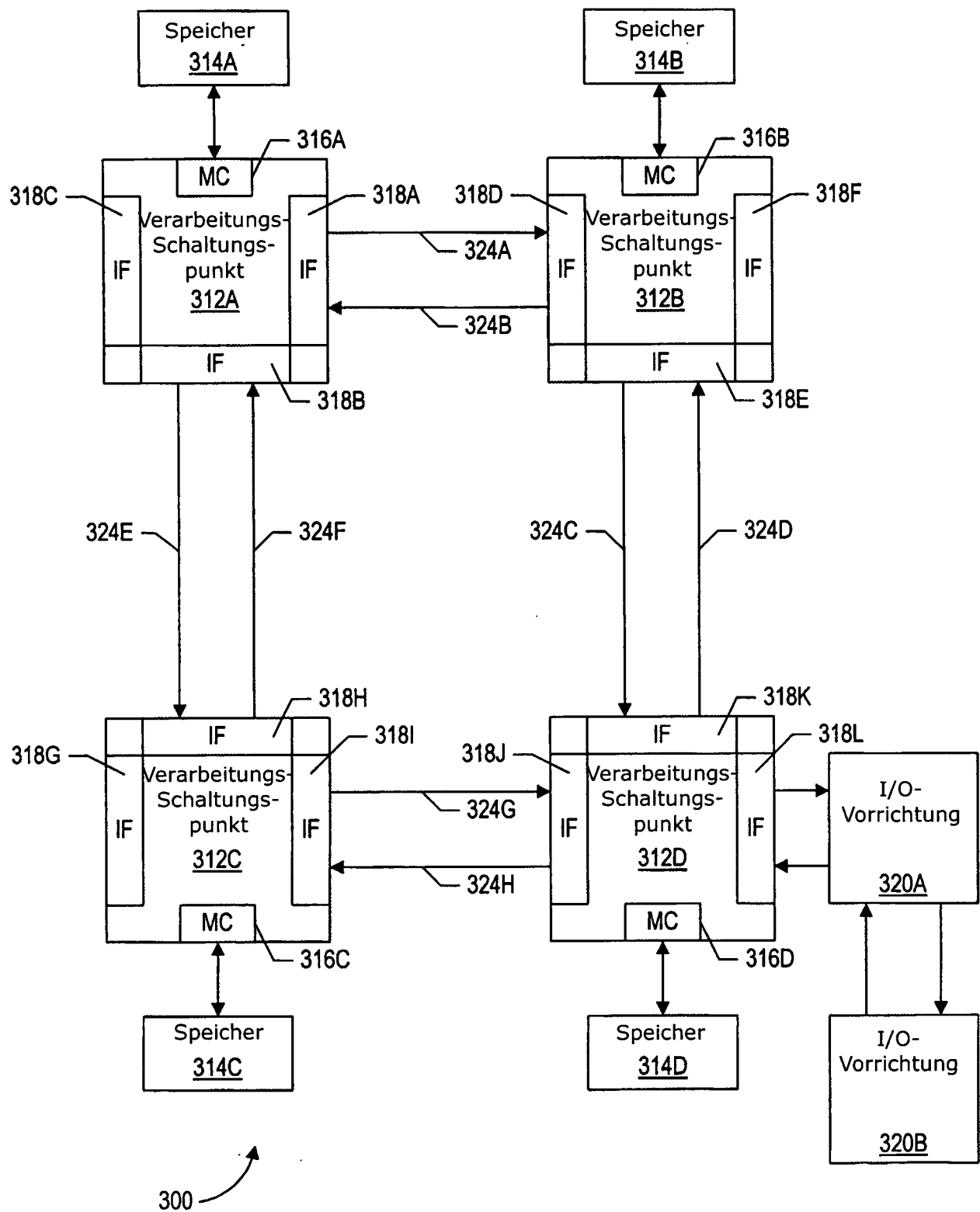


Fig. 10