



(12)

## Offenlegungsschrift

(21) Aktenzeichen: 10 2014 103 183.0

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

(22) Anmeldetag: 11.03.2014

(43) Offenlegungstag: 18.09.2014

(30) Unionspriorität:

13/831,488

14.03.2013 US

(74) Vertreter:

KUHNEN & WACKER Patent- und  
Rechtsanwaltsbüro, 85354 Freising, DE

(71) Anmelder:

Samsung Electronics Co., Ltd., Suwon-si,  
Gyeonggi-do, KR

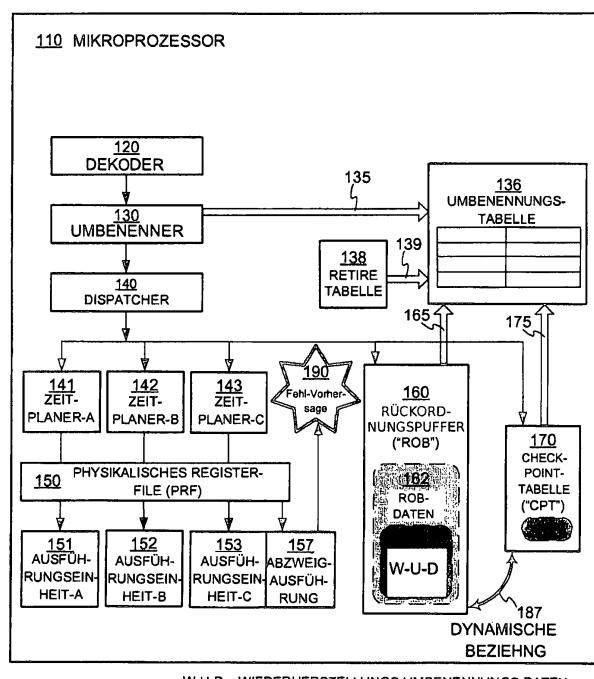
(72) Erfinder:

Iyengar, Ravi, Austin, Tex., US;  
Santhanakrishnan, Prarthna, Austin, Tex., US

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Aufzeichnungspuffer-basiertes, dynamisches Checkpointing zum Wiederherstellen einer Umbenennungstabelle**

(57) Zusammenfassung: Out-of-Order-CPPUs, Vorrichtungen und Verfahren verringern die Zeiteinbuße vom Anhalten der Pipe zum Wiederaufbauen einer Umbenennungstabelle (136, 336) beispielsweise aufgrund einer Fehlvorhersage. Ein Mikroprozessor (110, 310) kann eine Pipe aufweisen, welche einen Dekoder (120, 320), einen Dispatcher (140, 340) und wenigstens eine Ausführungseinheit (151, 152, 153, 351, 352, 353) hat. Eine Umbenennungstabelle (136, 336) speichert Umbenennungsdaten und eine Checkpoint-Tabelle („CPT“) (170, 270, 370) speichert Umbenennungsdaten, welche von dem Dispatcher (140, 340) empfangen werden. Ein Aufzeichnungspuffer („ROB“) (160, 260, 360) speichert ROB-Daten (162, 262A, 362) und hat eine dynamische Mapping-Beziehung (187, 287) mit der CPT (170, 270, 370). Wenn die Umbenennungstabelle (136, 336) entleert wird, wie beispielsweise aufgrund einer Fehlvorhersage, wird die Umbenennungstabelle (136, 336) wenigstens zum Teil durch ein gleichzeitiges Kopieren von Umbenennungsdaten, welche in der CPT (170, 270, 370) gespeichert sind, in Koordination mit Durchlaufen des ROB (160, 260, 360) wiederhergestellt.



**Beschreibung****QUERVERWEIS AUF  
VERWANDTE ANMELDUNGEN**

**[0001]** Es kann bemerkt werden, dass diese Anmeldung auf die U.S.-Patentanmeldung mit der Seriennummer 13/831,488 bezogen ist, welche am 14. März 2013 durch die gleichen Erfinder eingereicht wurde, eingereicht durch denselben Begünstigten an demselben Tag wie die vorliegende Anmeldung.

**HINTERGRUND**

**[0002]** Diese Offenbarung bezieht sich auf Halbleitervorrichtungen und genauer auf Mikroprozessoren, welche den Betrieb von elektronischen Vorrichtungen steuern, sowie elektronische Vorrichtungen, welche solche Mikroprozessoren verwenden.

**[0003]** Ein Mikroprozessor, auch bekannt als eine zentrale Verarbeitungseinheit (CPU = Central Processing Unit) arbeitet durch ein Ausführen von Befehlen. Einige Befehle führen zu Abzweigungspunkten, wo ein Weg der Ausführung gegenüber einem anderen ausgewählt werden kann. Ein Mikroprozessor kann eine erhöhte Geschwindigkeit haben, wenn er eine korrekte spekulative Vorhersage darüber tätigt, welcher Weg gewählt werden wird, und vorab Befehle entlang dieses Weges ausführt. Solche CPUs sind als Out-Of-Order-CPs bekannt. Der Geschwindigkeitsvorteil verringert sich jedoch, wenn es eine Fehlvorhersage gegeben hat, und eine Wiederherstellung bzw. eine Heilung benötigt wird.

**[0004]** Eine Herausforderung bei Out-Of-Order-CPs ist ein Risiko wie beispielsweise Write-After-Write (WAW) und Write-After-Read (WAR). Diese Risiken werden durch eine Register-Umbenennung vermieden, welche mit der Hilfe einer Umbenennungstabelle bewerkstelligt wird, welche einen Überblick über die umbenannten Quell- und Ziel-Register behält.

**[0005]** Ein verbleibendes Problem ist es jedoch, dass jedes Mal, wenn es einen Fehl-vorhergesagten Zweigbefehl bzw. Abzweigbefehl gibt, die Umbenennungstabelle vollständig geleert werden muss. Ein Leeren ist ein Problem, da, weil die Zweige Out-Of-Order bzw. nicht in Reihenfolge ausgeführt werden können, Befehle vorhanden sein könnten, welche darauf warten, sich zurückzuziehen bzw. zu beenden (retire), welche älter sind als der fehl-vorhergesagte Abzweigbefehl bzw. Zweigbefehl. Die Umbenennungsinformationen für diese älteren Befehle müssen in die Umbenennungstabelle wieder aufgebaut werden.

**[0006]** Das Problem manifestiert sich selbst dann als Verzögerung. Während des Wiederherstellungs-

Vorgangs muss die Umbenennungslogik das vordere Ende der Pipeline (pipe) von einem Senden neuer Befehle zum Umbenennen abhalten. Dieses Abhalten bzw. Aussetzen führt zu einer Verzögerung, welche sich zu der Einbuße für eine Abzweig-Fehl-Vorhersage ansammelt. Die Einbuße hängt nicht nur von der Wiederherstellungs-Latenz ab, sondern auch von der Um-Adressier- bzw. Umleit-Latenz und der Tiefe des vorderen Endes der Pipeline.

**[0007]** Zum Verringern des Anhaltens (engl. Stalling) wurden Checkpointing-Schemata, welche vor der Dispatch-Stufe sind, als ein Teil der Umbenennungs-Pipeline vorgeschlagen. In solchen Schemata startet traditionell jeder Abzweigbefehl ein neues Checkpoint-Fenster. Diese Herangehensweise ist arbeitsbereichsintensiv, da sie so viele Checkpoints bzw. Checkpunkte benötigt, wie es In-Flight-Abzweigungen gibt, welche in der Maschine erlaubt sind.

**KURZFASSUNG**

**[0008]** Die vorliegende Beschreibung gibt Beispiele von Mikroprozessoren, Vorrichtungen, welche Mikroprozessoren integrieren, und Verfahren, welche Probleme im Stand der Technik überwinden.

**[0009]** In einer Ausführungsform weist ein Mikroprozessor eine Pipe bzw. Pipeline auf, welche einen Decoder, einen Dispatcher und wenigstens eine Ausführungseinheit hat. Eine Umbenennungstabelle speichert Umbenennungsdaten und eine Checkpoint-Tabelle („CPT“) speichert Umbenennungsdaten, welche von dem Dispatcher empfangen werden. Ein Rückordnungspuffer bzw. Re-Order-Puffer („ROB“) speichert ROB-Daten und hat eine dynamische Mapping-Beziehung mit der CPT. Wenn die Umbenennungstabelle geleert wird, beispielsweise aufgrund einer Fehlvorhersage, wird die Umbenennungstabelle wenigstens teilweise durch ein gleichzeitiges Kopieren von Umbenennungsdaten, welche in der CPT gespeichert sind, wiederhergestellt, in Koordination mit einem Durchlaufen (walking) des ROB.

**[0010]** Ein Vorteil gegenüber dem Stand der Technik ist es, dass die Zeiteinbuße vom Abhalten der Pipe bis zum Wiederherstellen einer Umbenennungstabelle verringert wird. Ein anderer Vorteil entsteht aus der Tatsache, dass die CPT nach dem Dispatcher ist, was die Anzahl von Checkpoints bzw. Checkpunkten drastisch verringert, welche zum erfolgreichen Wiederherstellen der Umbenennungstabelle benötigt werden. Die Erfindung führt zu Ausführungsformen, welche hinsichtlich einer Arbeitsleistung bzw. eines Arbeitsbereichs, einer Leistung und eines Timings effizienter sind, als Vor-Dispatch-Checkpoint-Schemata.

**[0011]** Diese und andere Merkmale und Vorteile dieser Beschreibung werden deutlicher offensichtlich

werden aus der folgenden detaillierten Beschreibung, welche unter Bezugnahme auf die Zeichnungen voranschreitet, in welchen:

#### KURZE BESCHREIBUNG DER ZEICHNUNGEN

**[0012]** **Fig.** 1 ist ein Diagramm von Komponenten bzw. Bestandteilen eines Mikroprozessors, gefertigt gemäß beispielhaften Ausführungsformen.

**[0013]** **Fig.** 2A, **Fig.** 2B sind Diagramme, welche unterschiedliche Betriebsszenarien der Komponenten in **Fig.** 1 zum Veranschaulichen einer dynamischen Beziehung gemäß Ausführungsformen zeigen.

**[0014]** **Fig.** 3A ist ein Blockschaltbild, welches einen Mikroprozessor zeigt, welcher gemäß beispielhaften Ausführungsformen gefertigt ist, wenn er normal arbeitet.

**[0015]** **Fig.** 3B ist ein Diagramm des Mikroprozessors der **Fig.** 3A, welches eine Umbenennungstabelle zeigt, welche nach einem Normalbetrieb geleert wird, wobei das Leeren ein Ergebnis eines Erfassen einer Fehl-Vorhersage ist, in welcher Befehle auszuführen sind.

**[0016]** **Fig.** 3C ist ein Diagramm des Mikroprozessors der **Fig.** 3B, welches eine Umbenennungstabelle zeigt, welche wiederaufgebaut wird, nachdem sie geleert ist.

**[0017]** **Fig.** 4 ist ein Zeitablauf-Diagramm, welches relative Zeitabstimmungs-Pulse zum Wiederherstellen einer Umbenennungstabelle zeigt, wie beispielsweise der Umbenennungstabelle der **Fig.** 3C gemäß Ausführungsformen.

**[0018]** **Fig.** 5 ist ein Diagramm der Komponenten der **Fig.** 3C für ein erstes Beispielsszenario, in dem aus der Erfindung keine Vorzüge resultieren.

**[0019]** **Fig.** 6 ist ein Diagramm der Komponenten der **Fig.** 3C für ein zweites Beispielsszenario, in dem Vorfürze aus der Erfindung resultieren.

**[0020]** **Fig.** 7 ist ein Diagramm der Komponenten der **Fig.** 3C für ein drittes Beispielsszenario, in dem ein maximaler Vorzug aus der Erfindung resultiert.

**[0021]** **Fig.** 8 ist ein Blockschaltbild zum Veranschaulichen eines Systems, welches einen Mikroprozessor gemäß beispielhaften Ausführungsformen aufweist.

**[0022]** **Fig.** 9 ist ein Flussdiagramm zum Veranschaulichen von Verfahren gemäß beispielhaften Ausführungsformen.

#### DETAILLIERTE BESCHREIBUNG

**[0023]** Wie erwähnt wurde, bezieht sich die vorliegende Beschreibung auf Mikroprozessoren, Vorrichtungen und Verfahren. Ausführungsformen werden nun detaillierter beschrieben werden.

**[0024]** **Fig.** 1 ist ein Diagramm von Komponenten bzw. Bestandteilen eines Mikroprozessors **110**, welcher gemäß beispielhaften Ausführungsformen gefertigt ist. Der Mikroprozessor **110** weist einen Dekoder **120** zum Empfangen von Instruktionen und zum Dekodieren dieser Instruktionen in Mikro-Operationen auf, welche auch als Mikrobefehle bekannt sind.

**[0025]** Der Mikroprozessor **110** weist auch eine oder mehrere Ausführungseinheiten zum Ausführen der Mikrobefehle auf. In **Fig.** 1 sind drei Ausführungseinheiten gezeigt, nämlich die Ausführungseinheit-A **151**, die Ausführungseinheit-B **152** und die Ausführungseinheit-C **153**, obwohl diese Anzahl von Einheiten im Wege eines Beispiels und nicht als Beschränkung gezeigt ist.

**[0026]** Der Dekoder **120** und die Einheiten **151**, **152** und **153** sind Teil der sogenannten Pipe bzw. Pipeline, welche zusätzliche Komponenten aufweist. Eine solche Komponente ist ein Dispatcher **140**, welcher zum letztendlichen Dispatchen der Mikrobefehle der Ausführungseinheiten vorgesehen ist. In dem Beispiel der **Fig.** 1 obliegt das Dispatchen zuerst einem der drei Zeitplaner bzw. Scheduler, nämlich Zeitplaner-A **141**, Zeitplaner-B **142** und Zeitplaner-C **143**. Die Zeitplaner übertragen die Mikrobefehle zu Ausführungseinheiten **151**, **152**, **153**. Diese Ausführungseinheiten lesen die Quellen bzw. Sources von einem physikalischen Register-File (PRF = Physical Register File) und führen dann die Mikrobefehle aus. Der Fachmann wird anerkennen, dass das Obige nur eine beispielhafte Architektur für diesen Teil der Pipe ist, und verschiedene Architekturen auch möglich sind.

**[0027]** Der Mikroprozessor **110** weist zusätzlich einen Umbenener **130** auf, welcher die Mikrobefehle vom Dekoder **120** empfängt. Der Umbenener **130** erzeugt Umbenennungsdaten gemäß den Mikrobefehlen.

**[0028]** Der Mikroprozessor **110** weist darüber hinaus eine Umbenennungstabelle **136** auf. Die Umbenennungstabelle **136** speichert die Umbenennungsdaten, welche sie von dem Umbenener **130** gemäß einem Pfeil **135** empfängt.

**[0029]** Der Mikroprozessor **110** weist auch einen Rückordnungspuffer bzw. Re-Order-Puffer („ROB“) **160** auf. Der ROB **160** hat ROB-Einträge, welche in **Fig.** 1 nicht individuell gezeigt sind. Wie später detaillierter gesehen werden wird, weist der ROB

**160** einen Retire-Zeiger und einen Flush-Zeiger auf, von welchen jeder auf einen variablen einen der ROB-Einträge zeigt. Der ROB **160** kann zirkular sein oder nicht. Die ROB-Einträge sind zum Speichern von ROB-Daten **162**, welche aus den Mikrobefehlen erzeugt werden. In einigen Ausführungsformen speichert jeder ROB-Eintrag einen microop, obwohl mehr Mikrobefehle in einem einzelnen ROB-Eintrag gespeichert werden könnten. ROB-Daten **162** weisen Wiederherstellungs-Umbenennungsdaten **182** auf, welche vorzugsweise Umbenennungsdaten entsprechen, welche in der Umbenennungstabelle **136** gespeichert sind. In der Ausführungsform der **Fig. 1** werden ROB-Daten **162** von dem Dispatcher **140** empfangen.

**[0030]** Der Mikroprozessor **110** weist weiterhin wenigstens eine Checkpoint-Tabelle bzw. Checkpunkt-Tabelle („CPT“) **170** auf. Die CPT **170** ist so breit wie die Umbenennungstabelle **136**. Die CPT **170** hat CPT-Einträge, welche in **Fig. 1** nicht individuell gezeigt sind. Die CPT-Einträge sind so viele wie diejenigen der Umbenennungstabelle **136**, was bedeutet, dass die CPT so tief ist wie die Umbenennungstabelle **136**. Die Anzahl der CPT-Einträge definiert die CPT-Tiefe. Die CPT-Einträge sind zum Speichern einer Checkpoint-Version (checkpointed version) **180** der Umbenennungsdaten, welche in der Umbenennungstabelle **136** gespeichert sind. Die Checkpoint-Version **180** der Umbenennungsdaten kann eine beliebige Version sein, welche die Umbenennungsdaten tragen bzw. ergeben kann, oder sie können genau die Umbenennungsdaten sein, in welchem Fall sie Checkpoint-Umbenennungsdaten **180** genannt werden, da sie in der Checkpoint-Tabelle **170** gespeichert werden. In der Ausführungsform der **Fig. 1** wird die Checkpoint-Version **180** der Umbenennungsdaten von dem Dispatcher **140** empfangen.

**[0031]** Wie untenstehend detaillierter beschrieben werden wird, mappt das ROB-Checkpoint-Fenster in die CPT gemäß einer dynamischen Beziehung **187**, wobei das ROB-Checkpoint-Fenster definiert ist basierend auf dem ROB-Eintrag, auf welchen der Retire-Zeiger zeigt. Da der Retire-Zeiger auf einen allgemein unterschiedlichen ROB-Eintrag zu unterschiedlichen Zeiten zeigt, ist die Beziehung **187** demnach variabel. Als solches wird, wenn ein ROB-Eintrag mit einem neuen micro-op alloziert wird, ein ausgewählter CPT-Eintrag ebenso mit den Ziel-Umbenennungsinformationen für diesen micro-op aktualisiert, wenn dieser micro-op ein Ziel hat.

**[0032]** Wie voranstehend beschrieben ist, kann die Umbenennungstabelle **136** geleert werden, was genauer bedeutet, dass die Umbenennungsdaten, welche in der Umbenennungstabelle gespeichert sind, geleert werden können. In der Ausführungsform der **Fig. 1** weist der Mikroprozessor **110** auch eine Zweig-Ausführungslogik bzw. Abzweig-Ausführungs-

logik **157** auf. Die Logik **157** kann eine Fehl-Vorhersage, welche durch ein Flag **190** gezeigt wird, beim Ausführen der Mikrobefehle gemäß dem Obenstehenden erfassen. Die Umbenennungstabelle **136** kann in Antwort auf ein Erfassen einer Fehl-Vorhersage **190** geleert werden.

**[0033]** Wenn die Umbenennungstabelle **136** entleert wird, kann sie wiederhergestellt werden. Für Zwecke des Wiederherstellens kann der Mikroprozessor **110** auch eine Retire-Tabelle **138** aufweisen, welche Retire-Checkpoint-Daten speichern kann. Die Retire-Checkpoint-Daten können kopiert werden, um die Tabelle **136** gemäß einem Pfeil **139** als den anfänglichen Teil des Wiederherstellens umzubenennen.

**[0034]** Weiterhin können für Zwecke des Wiederherstellens die geleerten Umbenennungsdaten zu der Umbenennungstabelle **136** von zwei Datenquellen wiederhergestellt werden, zusätzlich zudem anfänglichen Kopieren von der Retire-Tabelle. Als erstes kann die Checkpoint-Version **180** der Umbenennungsdaten als ein Massenimport von der CPT **170** zu der Umbenennungstabelle **136** gemäß einem Pfeil **175** kopiert werden. Der Massenimport wird implementiert durch ein gleichzeitiges Kopieren, falls es durch einen Aspekt der dynamischen Beziehung **187** erlaubt ist, wie später in diesem Dokument erklärt wird. Als zweites kann wenigstens ein Abschnitt der Wiederherstellungs-Umbenennungsdaten **182** innerhalb von ROB-Daten **162** von dem ROB **160** zu der Umbenennungstabelle **136** gemäß einem Pfeil **165** kopiert werden, was ein Vorgang ist, welcher anderweitig bekannt ist als „Walking the ROB“ bzw. „Durchlaufen des ROB“.

**[0035]** Implementierungsdetails werden nun vorgenommen. Die **Fig. 2A** und **Fig. 2B** zeigen einen ROB **260** und eine CPT **270**, welche in einem Mikroprozessor **110** der **Fig. 1** substituiert werden könnten. Es wird anerkannt werden, dass die Auswählen von Größen und funktionellen Beziehungen zwischen dem ROB **260** und einer CPT **270** getätigten werden können mit einer Sichtweise eines Favorisierens eines Wiederherstellens mittels des gleichzeitigen Kopieren des Pfeiles **175** gegenüber dem Kopieren des Pfeiles **165**, um Prozessorzeit zu sparen.

**[0036]** In diesem Beispiel hat der ROB **260** 96 Einträge, benannt von 0 bis 95. Der ROB **260** könnte eine beliebige Anzahl von ROB-Einträgen gehabt haben. Zusätzlich wird ein ROB-Checkpoint-Fenster definiert als eine Gruppe von ROB-Einträgen habend. Die Gruppe kann eine beliebige Anzahl sein, welche optimiert sein kann. Eine Anzahl, welche für die Gruppe gut arbeitet, ist ungefähr die Hälfte der Größe der Tiefe des ROB, in diesem Falle 48 ROB-Einträge. Wie aus dem Untenstehenden verstanden werden wird, wird ein ROB-Checkpoint-Fenster, welches zu viel größer als das Optimum ist, die CPT nicht oft-

mals genug aktivieren, um einen guten Unterschied zu bereiten, während wenn es zu viel kleiner ist als das Optimum, es die CPT genügende Male aktivieren wird, jedoch nicht für genug Umbenennungsdaten. In beiden Fällen wird es, wenn die Gruppengröße zu weit von dem Optimum abweicht, zu einem Walking the ROB bzw.

**[0037]** Durchlaufen des ROB mehrere Male als es notwendig ist führen, wie untenstehend gesehen werden wird.

**[0038]** In den **Fig. 2A** und **Fig. 2B** sind der ROB **260** und die CPT **270** bei verschiedenen Szenarien gezeigt, während welcher die Umbenennungstabelle geleert wurde. Selbstverständlich wird das ROB-Checkpoint-Fenster jedes Mal eine unterschiedliche Gruppe von 48 ROB-Einträgen umfassen bzw. aufspannen. Zusätzlich wird das ROB-Wiederherstellungs-Fenster jedes Mal allgemein eine unterschiedliche Anzahl von ROB-Einträgen haben, wenn sie zu unterschiedlichen Zeiten durch unterschiedliche Ereignisse des Leerens der Umbenennungstabelle definiert werden.

**[0039]** Genauer wird in **Fig. 2A** das ROB-Checkpoint-Fenster definiert als beginnend von dem Retire-Zeiger bei ROB-Eintrag 20 und umfassend bis zum ROB-Eintrag 67. Zusätzlich ist das ROB-Wiederherstellungs-Fenster **263A** definiert, welches 37 ROB-Einträge umfasst, nämlich die ROB-Einträge 20 bis 56, da der Eine nach 56 57 ist, der Eine, auf welchen durch den Flush-Zeiger gezeigt wird. Diese 37 ROB-Einträge speichern ROB-Daten **262A**, welche Wiederherstellungs-Umbenennungsdaten **282A** aufweisen. Die CPT **270** speichert Checkpoint-Umbenennungsdaten **280A**.

**[0040]** Darüber hinaus ist in **Fig. 2B** das ROB-Checkpoint-Fenster definiert als beginnend von dem Retire-Zeiger bei ROB-Eintrag 40 und aufweisend bis zu ROB-Eintrag 87. Zusätzlich ist das ROB-Wiederherstellungs-Fenster **263B** definiert, welches nur 27 ROB-Einträge umfasst, nämlich ROB-Einträge 40 bis 66. Diese 27 ROB-Einträge speichern ROB-Daten **262B**, welche wiederhergestellte Umbenennungsdaten **282B** aufweisen. Die CPT **270** speichert Checkpoint-Umbenennungsdaten **280B**.

**[0041]** Die dynamische Beziehung **287** ist ein Weg wie der ROB, und genauer das ROB-Checkpoint-Fenster, in die CPT mappt. Das Mapping wird manchmal eine Assoziation bzw. Verbindung genannt. Die Beziehung **287** wird dynamisch genannt, da sie sich ändert, da das ROB-Checkpoint-Fenster basierend auf dem ROB-Eintrag definiert ist, auf welchen der Retire-Zeiger zeigt. In der Tat ist in den **Fig. 2A** und **Fig. 2B** das ROB-Checkpoint-Fenster an unterschiedlichen Stellen bzw. Orten definiert.

**[0042]** Für Zwecke des Wiederherstellens existiert die Option, dass ein bestimmtes Element zu der Umbenennungstabelle wiederhergestellt werden wird von entweder dem ROB-Eintrag, in dem es gespeichert ist, oder der CPT **270**, in der es gespeichert ist. Wenn ein Kopieren von der CPT erlaubt ist, wird es schneller sein.

**[0043]** Die dynamische Beziehung **287** regelt zum Teil, ob und in welchem Umfang ein Wiederherstellen von der CPT **270** zusätzlich zu von dem ROB **260** sein wird. In der Tat findet ein Kopieren von der CPT **270** statt, wenn es anderweitig durch einen Aspekt der Beziehung **287** erlaubt ist. Beispiele dafür, wann dies erlaubt ist, werden in diesem Dokument später beschrieben.

**[0044]** Die **Fig. 3A**, **Fig. 3B**, **Fig. 3C** werden nun aufeinanderfolgend verwendet, um eine Beispielsequenz von Ereignissen für einen Beispiels-Mikroprozessor **310**, welcher gemäß Ausführungsformen gefertigt ist, zu beschreiben. Der Mikroprozessor **310** weist einen Dekoder **320**, einen Umbenener **330**, eine Umbenennungstabelle **336**, eine Retire-Tabelle **338**, einen Dispatcher **340**, Scheduler bzw. Zeitplaner **341**, **342**, **343**, eine PRF **350**, Ausführungseinheiten **351**, **352**, **353**, eine Abzweigungs-Ausführungslogik **357** und einen ROB **360** auf, welche alle wie oben unter Bezugnahme auf ähnliche Komponenten in **Fig. 1** beschrieben ist, sein können. Der ROB **360** speichert ROB-Daten **362** ähnlich zu ROB-Daten **162**. Die ROB-Daten **362** weisen Wiederherstellungs-Umbenennungsdaten **382** ähnlich zu Wiederherstellungs-Umbenennungsdaten **182** auf.

**[0045]** Zusätzlich weist der Mikroprozessor **310** eine CPT **370** auf, welche eine dynamische Beziehung mit dem ROB **360** hat. Die CPT **370** und der ROB **360** könnten dieselbe dynamische Beziehung haben wie die Beziehung **187** der **Fig. 1**, **Fig. 2A**, **Fig. 2B**. Die CPT **370** speichert Checkpoint-Umbenennungsdaten **380** gemäß der dynamischen Beziehung.

**[0046]** In **Fig. 3A** arbeitet der Mikroprozessor **310** normal. Befehle werden im Dekoder **320** empfangen, treten durch die Pipe hindurch und werden bei Ausführungseinheiten **351**, **352**, **353** ausgeführt.

**[0047]** In **Fig. 3B** ist eine Fehl-Vorhersage erfasst, wie durch Flag **390** angezeigt wird. Als ein Ergebnis wird die Umbenennungstabelle **336** geleert, angezeigt durch einen Kommentar **392**.

**[0048]** In **Fig. 3C** wird die Umbenennungstabelle **336** wiederhergestellt, wie durch einen Kommentar **393** angezeigt ist. Als ein Ergebnis gibt es ein Anhalten an dem vorderen Ende der Pipe, angezeigt durch einen Kommentar **391**.

**[0049]** Ein Wiederherstellen findet wie folgt statt: Als erstes wird ein Retire-Checkpoint von der Retire-Tabelle **338** gemäß einem Pfeil **339** kopiert. Dann wird durch Daten, welche allgemein von zwei unterschiedlichen Quellen kommen wiederhergestellt. Checkpoint-Umbenennungsdaten **380** werden gemäß einem Pfeil **375** kopiert, falls es durch einen Aspekt der anwendbaren dynamischen Beziehung erlaubt ist. Zusätzlich werden Wiederherstellungs-Umbenennungsdaten **282** durch ein Durchlaufen des ROB **360** gemäß einem Pfeil **365** kopiert. In einigen bestimmten Fällen jedoch mag eine dieser zwei Quellen nicht beitragen, wie in beispielhaften Szenarien untenstehend gesehen werden wird.

**[0050]** Der Mikroprozessor **310** weist typischerweise auch einen Taktgeber auf, was nicht gezeigt ist. Der Taktgeber gibt Pulse aus, welche Taktzyklen definieren. Die Taktzyklen sind ein guter Weg zum Messen der Vorteile der Erfindung. Beispiele werden nun beschrieben.

**[0051]** **Fig. 4** ist ein Zeitablauf-Diagramm, welches relative Zeitabstimmungs-Pulse zum Wiederherstellen einer Umbenennungstabelle zeigt, wie beispielsweise in **Fig. 3C**. Taktzyklen **412** weisen eine Gruppe **415** von N Zyklen auf, hier als ein einzelner Puls gezeigt. Die Anzahl N wird später detaillierter beschrieben werden.

**[0052]** **Fig. 4** veranschaulicht ein Auftreten des Erfassens einer Fehl-Vorhersage und die nachfolgenden Einbußen. Ein Puls **492** entspricht einem Entleeren der Umbenennungstabelle **396** gemäß Kommentar **392**. Ein verlängerter Anhaltepuls **491** entspricht der Zeit, die die Pipe angehalten wird, gemäß Kommentar **391**; die Dauer des Pulses **491** bezieht sich auf die Zeiteinbuße für die Fehl-Vorhersage und die nachfolgende Verschlechterung in der Leistungsfähigkeit des Mikroprozessors. Die Einbuße ist größer, wenn N groß ist sowohl in Puls **491** als auch in der Gruppe von Zyklen **415**. Pulse **493A** und **493B** zeigen den Beginn und das Ende des Wiederherstellens an, entsprechend Kommentar **393**.

**[0053]** Im Allgemeinen weist ein Umbenennen von Zyklen **436** auf: a) einen Zyklus **439** zum Kopieren des Retire-Checkpoint gemäß Operation **339**, b) einen Zyklus **475** zum gleichzeitigen Kopieren von einer CPT gemäß einem Pfeil **375** und c) eine Gruppe **465** von so vielen N-Zyklen zum Durchlaufen der ROB gemäß Pfeil **365**. Ähnlich zu der Gruppe von Zyklen **415** ist Zyklus **465** als ein Einzelner gezeigt, während tatsächlich beide den Wert von N annehmen. Icons von Pfeilen **339**, **375** und **365** sind in **Fig. 4** wiederholt, an Stellen, welche die mentale Assoziation für den Leser verbessern werden.

**[0054]** Insbesondere sind jedoch einige dieser Umbenennungszyklen **436** nicht jedes Mal enthalten.

Die Frage, welche enthalten sind und welche nicht, hängt von dem Szenario ab, welches zu dem Moment definiert ist, zu dem eine Fehl-Vorhersage erfasst wird, und die Umbenennungstabelle demnach geleert wird. Bestimmte Betriebsszenarien werden nun in diesem Dokument präsentiert und sind auf **Fig. 4** bezogen.

**[0055]** Als erstes sollte für Zwecke des Verstehens der Szenarien erkannt werden, dass ein ROB-Wiederherstellungsfenster definiert wird, wenn die Umbenennungstabelle geleert wird. Das ROB-Wiederherstellungsfenster hat ROB-Einträge beginnend von dem Retire-Zeiger und endend einen vor dem Flush-Zeiger, wie bereits obenstehend in **Fig. 2A** und **Fig. 2B** gesehen wurde. Es sind die ROB-Einträge innerhalb des ROB-Wiederherstellungsfensters, welche definieren, welche Daten zu der Umbenennungstabelle wiederhergestellt werden müssen, entweder durch gleichzeitiges Kopieren von der CPT oder durch Durchlaufen des ROB. In jedem Fall ist der Abschnitt der Wiederherstellungs-Umbenennungsdaten, welcher von dem ROB zu der Umbenennungstabelle kopiert wird, innerhalb des ROB-Wiederherstellungsfensters.

**[0056]** Die Szenarien sind darin unterschiedlich, dass jedes Mal, wenn die Umbenennungstabelle geleert wird, das ROB-Wiederherstellungsfenster allgemein einen unterschiedlichen Satz von ROB-Einträgen umfassen wird. Als solches sondieren bzw. erkunden die unterschiedlichen Szenarien, wie das ROB-Wiederherstellungsfenster unterschiedliche Größen relativ zu der Tiefe des voranstehend erwähnten ROB-Checkpoint-Fensters hat. Es ist auch dieser Aspekt in den folgenden beispielhaften Szenarien, welcher bestimmen wird, ob es der CPT-Tabelle erlaubt ist oder nicht, gleichzeitig zu der Umbenennungstabelle kopiert zu werden, oder ob anstelle dessen dieselben Inhalte durch Durchlaufen des ROB kopiert werden.

**[0057]** Beispiele von verschiedenen solchen Szenarien werden nun untersucht. Für den Zweck dieser Szenarien wird angenommen, dass der ROB **360** und die CPT **370** verwendet werden, ähnlich zu dem, was in den **Fig. 2A**, **Fig. 2B** gezeigt ist. Zusätzlich trifft eine dynamische Beziehung **587** ähnlich zu der Beziehung **287** ebenso zu, angezeigt durch die Klammern, welche für das ROB-Checkpoint(CP)-Fenster und die CPT **370** gezeigt sind.

**[0058]** **Fig. 5** ist ein Diagramm von Komponenten der **Fig. 3C** für ein erstes Beispielsszenario, wo kein Vorzug aus der Erfindung resultiert. In dem ROB **360** beginnt das Checkpoint-Window von ROB-Eintrag 20, auf welchen durch den Retire-Zeiger gezeigt wird, und endet bei ROB-Eintrag 67. Zusätzlich beginnt ein ROB-Wiederherstellungsfenster **563**, welches durch einen Pfeil bezeichnet ist, von dem

ROB-Eintrag 20, auf welchen durch den Retire-Zeiger gezeigt wird, und endet bei ROB-Eintrag 56, welcher eins weniger ist als der ROB-Eintrag 57, auf welchen durch den Flush-Zeiger gezeigt wird. ROB-Daten **562** innerhalb des ROB-Wiederherstellungsfensters **563** weisen Wiederherstellungs-Umbenennungsdaten **582** auf, welche zu der Umbenennungstabelle **336** wiederherzustellen sind, entweder durch gleichzeitiges Kopieren von der CPT **370** oder direkt durch ein Durchlaufen des ROB **360**.

**[0059]** In diesem Beispiel ist das ROB-Wiederherstellungsfenster **563** kleiner als das ROB-Checkpoint-Fenster. Demzufolge würden die Inhalte der ROB-Einträge 57 bis 67 entleert und demnach werden Checkpoint-Umbenennungsdaten **580**, welche in der CPT **370** gespeichert sind, nicht in eine Umbenennungstabelle als ein Teil des Wiederherstellungs-Vorgangs kopiert werden, da die gesamte CPT **370** zu der Umbenennungstabelle **336** kopiert werden müsste. Das ist der Grund, warum Checkpoint-Umbenennungsdaten **580** ausgekreuzt gezeigt sind. Demnach werden gemäß Kommentar **567** Wiederherstellungs-Umbenennungsdaten **582** in allen ROB-Einträgen 20 bis 56 gemäß Pfeil **565** wiederhergestellt werden, d. h. durch ein Durchlaufen des ROB **360**.

**[0060]** In anderen Worten gesagt kann die CPT der **Fig. 5** nicht für ein Wiederherstellen verwendet werden, und demnach bot die Erfindung keinen Vorteil in dem Szenario der **Fig. 5**. Kurz Bezug nehmend auf **Fig. 4** für das Szenario der **Fig. 5**, wären Umbenennungszyklen **475** enthalten und die Anzahl N nimmt einen Wert von bis zu m an, wobei m die maximale Anzahl von Schritten ist, welche der ROB **360** unter einem solchen Szenario durchlaufen werden könnte. Die Dauer von Anhaltepulsen **491** wird nicht von dem verringert, was sie ohne die Erfindung wäre. So etwas ist bei den verbleibenden Beispiel-Szenarien nicht der Fall.

**[0061]** **Fig. 6** ist ein Diagramm von Komponenten der **Fig. 3C** für ein zweites Beispielsszenario. In dem ROB **360** beginnt das Checkpoint-Fenster von ROB-Eintrag 20, auf welchen durch den Retire-Zeiger gezeigt wird, und endet bei ROB-Eintrag 67. Zusätzlich umfasst ein ROB-Wiederherstellungsfenster **663** innerhalb des ROB **360** ROB-Einträge 20 bis 76. In diesem Beispiel ist das ROB-Wiederherstellungsfenster **663** größer als das ROB-Checkpoint-Fenster. Innerhalb des ROB-Wiederherstellungsfensters **663** weisen ROB-Daten **662** wiederhergestellte Umbenennungsdatenabschnitte **682A** und **682B** auf, welche zur Umbenennungstabelle **336** wiederherzustellen sind. Es wird anerkannt werden, dass die Unterteilung der Wiederherstellungs-Umbenennungsdaten in Abschnitte **682A** und **682B** innerhalb des ROB **360** von den Grenzen des ROB-Checkpoint-Fensters entsteht. In der Tat mappen ROB-Einträge 20 bis 67 in

den Checkpoint-Umbenennungsdatenabschnitt **680**, welcher in der CPT **370** gespeichert ist, während ROB-Einträge 68 bis 76 nicht in die CPT **370** mappen.

**[0062]** Der Checkpoint-Umbenennungsdatenabschnitt **680** ist verwendbar und gemäß Kommentar **677** wird er gemäß einem Pfeil **675** wiederhergestellt werden. Demzufolge wird der Wiederherstellungs-Umbenennungsdatenabschnitt **682A** nicht benötigt werden, und er ist ausgekreuzt gezeigt. Der Abschnitt **682B** kann jedoch nicht effizient über eine CPT kopiert werden und gemäß Kommentar **667** wird der Wiederherstellungs-Umbenennungsdatenabschnitt **682B** in den ROB-Einträgen 48 bis 54 gemäß einem Pfeil **665** durch ein Durchlaufen des ROB **360** wiederhergestellt werden. Unter Bezugnahme auf die **Fig. 4** kann ein Wiederherstellen gemäß Pfeil **675** in einem einzelnen Taktpuls **475** stattfinden. N wird nach wie vor größer als Null sein, jedoch geringer als m, und die Dauer des Anhalte- bzw. Verzögerungspulses **491** wird aufgrund der Erfindung verringert werden. Es wird beobachtet werden, dass ein Wiederherstellen in dem Szenario der **Fig. 6** ein Kopieren von mehr Daten als in demjenigen der **Fig. 5** benötigt, jedoch dank der Erfindung weniger Zeit benötigt.

**[0063]** **Fig. 7** ist ein Diagramm von Komponenten der **Fig. 3C** für ein drittes Beispielsszenario. In dem ROB **360** startet das Checkpoint-Fenster von ROB-Eintrag 20, auf welchen durch den Retire-Zeiger gezeigt wird, und ended bei ROB-Eintrag 67. Zusätzlich umfasst ein ROB-Wiederherstellungsfenster **763** die ROB-Einträge 20 bis 67. In diesem Beispiel ist das ROB-Wiederherstellungsfenster **763** genau dasselbe wie das ROB-Checkpoint-Fenster. Dies ist ein Spezialfall, welcher statistisch mit einiger Häufigkeit auftritt. Die ROB-Daten **762** weisen Wiederherstellungs-Umbenennungsdaten **182** auf.

**[0064]** Checkpoint-Umbenennungsdaten **780** sind verwendbar und gemäß Kommentar **777** werden sie gemäß einem Pfeil **775** wiederhergestellt werden. Darüber hinaus wird es gemäß Kommentar **767** kein Durchlaufen des ROB **360** in diesem Szenario geben. Keine Wiederherstellungs-Umbenennungsdaten werden von dem ROB **360** zu der Umbenennungstabelle **336** kopiert werden. Demzufolge werden Wiederherstellungs-Umbenennungsdaten **782** nicht benötigt werden, und sie sind ausgekreuzt gezeigt. Unter Bezugnahme auf **Fig. 4** kann ein Wiederherstellen gemäß Pfeil **775** in einem einzelnen Taktpuls **475** stattfinden. N wird Null sein, und es wird keine Gruppe von Pulsen **415**, **465** geben. Die Dauer des Anhaltepulses **491** wird aufgrund der Erfindung minimiert sein.

**[0065]** Bezug nehmend auf **Fig. 8** weist eine elektronische Vorrichtung ein System **800** auf, welches mit einer Halbleitervorrichtung gemäß beispielhaften

Ausführungsformen arbeitet. Das System **800** kann in einem eines persönlichen digitalen Assistanten (PDA = Personal Digital Assistant), eines Laptop-Computers, eines mobilen Computers, eines Web-Tablet, eines drahtlosen Telefons, eines Mobiltelefons, eines digitalen Musikspielers, einer verdrahteten oder drahtlosen elektronischen Vorrichtung oder einer komplexen elektronischen Vorrichtung, die wenigstens zwei davon aufweist, verwendet werden. Das System **800** kann einen Controller bzw. eine Steuerung **810**, eine Eingabe-/Ausgabe-(I/O)-Vorrichtung **820** wie beispielsweise ein Keypad, eine Tastatur, eine Anzeige, einen Speicher **830** und eine Schnittstelle **840** aufweisen, welche miteinander über einen Bus **850** kommunizieren. Der Controller **810** kann beispielsweise wenigstens einen Mikroprozessor aufweisen, welcher gemäß Ausführungsformen gefertigt ist, einen digitalen Signalprozessor, einen Mikrocontroller oder dergleichen. Der Speicher **830** kann konfiguriert sein, um Befehle, welche durch den Controller **810** zu verwenden sind, und/oder Verwenderdaten, auf welche über die I/O-Vorrichtung zugegriffen werden kann, zu speichern. Die elektronische Vorrichtung **800** kann die Schnittstelle **840** verwenden, welche konfiguriert ist, um Daten zu einem Kommunikationsnetzwerk übertragen oder um Daten von diesem zu empfangen. Die Übertragung kann über Drähte, beispielsweise über Kabel oder eine USB-Schnittstelle erfolgen. Alternativ kann das Kommunikationsnetzwerk drahtlos sein, und die Schnittstelle **840** kann drahtlos sein und beispielsweise eine Antenne, einen drahtlosen Transceiver usw. aufweisen. Das elektronische System **800** kann in einem Kommunikations-Schnittstellenprotokoll eines Kommunikationssystems wie beispielsweise CDMA, GSM, NADC, E-TDMA, WCDMA, CDMA2000, Wi-Fi, Muni Wi-Fi, Bluetooth, DECT, Wireless USB, Flash-OFDM, IEEE 802.20, GPRS, iBurst, WiBro, WiMAX, WiMAX Advanced, UMTS-TDD, HSPA, EVDO, LTE-Advanced, MMDS usw. verwendet werden.

**[0066]** Fig. 9 zeigt ein Flussdiagramm **900** zum Beschreiben von Verfahren gemäß den Ausführungsformen. Die Verfahren des Flussdiagramms **900** können auch durch Ausführungsformen, welche obenstehend beschrieben sind, praktiziert werden, wie Mikroprozessoren und elektronische Vorrichtungen.

**[0067]** Gemäß einer Operation **910** wird ein Befehl in Mikrobefehle dekodiert. Gemäß einer nächsten Operation **920** werden Umbenennungsdaten gemäß den Mikrobefehlen erzeugt. Gemäß einer nächsten Operation **930** werden die Umbenennungsdaten in einer Umbenennungstabelle gespeichert.

**[0068]** Gemäß einer nächsten Operation **940** wird eine Checkpoint-Version der Umbenennungsdaten in einer CPT gespeichert. Gemäß einer anderen Operation **950** werden ROB-Daten in einem ROB gespeichert, welcher aus den Mikrobefehlen erzeugt wur-

de. Die ROB-Daten weisen Wiederherstellungs-Umbenennungsdaten auf, welche für ein ultimatives Wiederherstellen der Umbenennungstabelle verwendet werden können, falls Bedarf besteht. Der ROB erhält eine dynamische Beziehung mit der CPT aufrecht. Gemäß einer Operation **960** werden die Mikrobefehle ausgeführt.

**[0069]** Gemäß einer nächsten Operation **970** wird bestimmt, ob die Umbenennungstabelle geleert wurde; in anderen Worten gesagt wurden die Umbenennungsdaten aus der Umbenennungstabelle herausgenommen bzw. herausgespült. Falls nicht, schreitet die Ausführung zu der obigen Operation **910** zurück. In einigen Ausführungsformen wurde die Umbenennungstabelle in Antwort auf ein Erfassen einer Fehl-Vorhersage gemäß Obigem geleert.

**[0070]** Wenn die Umbenennungstabelle geleert wurde, dann werden gemäß einer nächsten Operation **980** Umbenennungsdaten zu der Umbenennungstabelle wiederhergestellt; in anderen Worten gesagt wird die Umbenennungstabelle wiederhergestellt. Eine Operation **980** weist eine oder beide von Operationen **982** und **984** auf. Für Operation **980** kann ein ROB-Wiederherstellungsfenster von ROB-Einträgen definiert werden, wenn die Umbenennungstabelle geleert wird. Es sind die Wiederherstellungs-Umbenennungsdaten dieser ROB-Einträge innerhalb des ROB-Wiederherstellungsfensters, welche zu der Umbenennungstabelle wiederhergestellt werden, entweder durch Operation **982** oder durch Operation **984**.

**[0071]** Gemäß Operation **982** wird die Checkpoint-Version der Umbenennungsdaten von der CPT zu der Umbenennungstabelle kopiert, wenn es durch die Beziehung der Operation **950** erlaubt ist. In einigen Ausführungsformen wird die Checkpoint-Version der Umbenennungsdaten so in einem einzelnen Taktzyklus kopiert.

**[0072]** Gemäß Operation **984** werden Wiederherstellungs-Umbenennungsdaten von dem ROB zu der Umbenennungstabelle kopiert. Vorzugsweise wird dies für jeglichen Rest, welcher nicht durch die Operation **982** umfasst ist, getan, und dies wird durch ein Durchlaufen des ROB getan.

**[0073]** In dem Obigen ist die Reihenfolge der Operationen nicht auf das beschränkt, was gezeigt ist, und unterschiedliche Reihenfolgen können gemäß den unterschiedlichen Ausführungsformen möglich sein. Zusätzlich können in bestimmten Ausführungsformen neue Operationen hinzugefügt werden oder individuelle Operationen können modifiziert oder gelöscht werden.

**[0074]** Ein Fachmann wird in der Lage sein, die vorliegende Erfindung hinsichtlich dieser Beschreibung zu praktizieren, welche als ein Ganzes zu nehmen

ist. Details wurden eingeschlossen, um ein vollständiges Verständnis vorzusehen. In anderen Beispielen wurden wohlbekannte Aspekte nicht beschrieben, um nicht unnötigerweise die vorliegende Erfindung zu verschleiern.

**[0075]** Diese Beschreibung weist ein oder mehrere Beispiele auf, doch dies beschränkt nicht, wie die Erfindung praktiziert werden kann. In der Tat können Beispiele oder Ausführungsformen der Erfindung gemäß dem, was beschrieben ist, praktiziert werden, oder auch unterschiedlich und ebenso in Zusammenhang mit anderen gegenwärtigen oder zukünftigen Technologien.

**[0076]** Die folgenden Ansprüche definieren bestimmte Kombinationen und Unterkombinationen von Elementen, Merkmalen und Schritten oder Operationen, welche als neu und nicht offensichtlich betrachtet werden. Zusätzliche Ansprüche für andere solche Kombinationen und Unterkombinationen können in diesem oder einem verwandten Dokument präsentiert werden.

**ZITATE ENTHALTEN IN DER BESCHREIBUNG**

*Diese Liste der vom Anmelder aufgeführten Dokumente wurde automatisiert erzeugt und ist ausschließlich zur besseren Information des Lesers aufgenommen. Die Liste ist nicht Bestandteil der deutschen Patent- bzw. Gebrauchsmusteranmeldung. Das DPMA übernimmt keinerlei Haftung für etwaige Fehler oder Auslassungen.*

**Zitierte Nicht-Patentliteratur**

- IEEE 802.20 [0065]

## Patentansprüche

1. Mikroprozessor (**110, 310**), der Folgendes aufweist:  
 eine Umbenennungstabelle (**136, 336**) zum Speichern von Umbenennungsdaten,  
 einen Dispatcher (**140, 340**) und  
 eine Checkpoint-Tabelle („CPT“) (**170, 270, 370**) zum Speichern von Umbenennungsdaten, welche von dem Dispatcher (**140, 340**) empfangen werden,  
 bei welchem, wenn die Umbenennungstabelle (**136, 336**) geleert wird, die Umbenennungstabelle (**136, 336**) unter Verwendung der Umbenennungsdaten, welche in der CPT (**170, 270, 370**) gespeichert sind, wiederhergestellt wird.
2. Mikroprozessor (**110, 310**) nach Anspruch 1, bei welchem die Umbenennungstabelle (**136, 336**) nur unter Verwendung der Umbenennungsdaten, welche in der CPT (**170, 270, 370**) gespeichert sind, wiederhergestellt wird.

3. Mikroprozessor (**110, 310**) nach Anspruch 1, weiterhin aufweisend:  
 eine Abzweigungs-Ausführungslogik (**157, 357**) zum Erfassen einer Fehl-Vorhersage, und  
 bei welchem, in Antwort darauf, dass eine Fehl-Vorhersage erfasst wird, die Umbenennungstabelle (**136, 336**) geleert wird.

4. Mikroprozessor (**110, 310**) nach Anspruch 1, weiterhin aufweisend:  
 einen Taktgeber zum Ausgeben von Pulsen, welche Taktzyklen definieren, und  
 bei welchem die Umbenennungstabelle (**136, 336**) in einem einzelnen der Taktzyklen wiederhergestellt wird.

5. Mikroprozessor (**110, 310**) nach Anspruch 1, weiterhin aufweisend:  
 einen Rückordnungspuffer (ROB) (**160, 260, 360**), welcher ROB-Einträge und  
 einen Retire-Zeiger hat, welcher auf einen variablen der ROB-Einträge zeigt, und in welchem  
 eine Gruppe der ROB-Einträge als ein ROB-Checkpoint-Fenster definiert ist,  
 wobei das ROB-Checkpoint-Fenster gemäß einer dynamischen Beziehung (**187, 287**) in die CPT (**170, 270, 370**) mappt, wobei das ROB-Checkpoint-Fenster definiert wird basierend auf dem ROB-Eintrag, auf welchen der Retire-Zeiger zeigt, und  
 wobei die Umbenennungstabelle (**136, 336**) unter Verwendung der Umbenennungsdaten, welche in der CPT (**170, 270, 370**) gespeichert sind, wiederhergestellt wird nur wenn es durch einen Aspekt der dynamischen Beziehung (**187, 287**) erlaubt ist.

6. Verfahren für einen Mikroprozessor (**110, 310**), das Folgendes aufweist:

ein Erzeugen von Umbenennungsdaten gemäß Mikrobefehlen;  
 ein Speichern der Umbenennungsdaten in einer Umbenennungstabelle (**136, 336**);  
 einen Dispatcher (**140, 340**), welcher einen der Mikrobefehle zu einer Ausführungseinheit (**151, 152, 153, 351, 352, 353**) absendet;  
 wobei der Dispatcher (**140, 340**) die Umbenennungsdaten zu einer Checkpoint-Tabelle („CPT“) (**170, 270, 370**) zum Speichern überträgt; und  
 wenn die Umbenennungstabelle (**136, 336**) geleert wird, die Umbenennungstabelle (**136, 336**) durch ein Kopieren der Umbenennungsdaten, welche in der CPT (**170, 270, 370**) gespeichert sind, von der CPT (**170, 270, 370**) zu der Umbenennungstabelle (**136, 336**) wiederhergestellt wird.

7. Verfahren nach Anspruch 6, weiterhin aufweisend:  
 ein Erfassen einer Fehl-Vorhersage, und  
 bei welchem in Antwort darauf, dass eine Fehl-Vorhersage erfasst wird, die Umbenennungstabelle (**136, 336**) geleert wird.

8. Mikroprozessor (**110, 310**), der Folgendes aufweist:  
 einen Dekoder (**120, 320**) zum Dekodieren eines Befehls in Mikrobefehle;  
 eine Ausführungseinheit (**151, 152, 153, 351, 352, 353**) zum Ausführen der Mikrobefehle;  
 einen Umbenener (**130, 330**) zum Erzeugen von Umbenennungsdaten gemäß den Mikrobefehlen;  
 eine Umbenennungstabelle (**136, 336**) zum Speichern der Umbenennungsdaten derart, dass die Umbenennungsdaten von der Umbenennungstabelle (**136, 336**) geleert werden können;  
 eine Checkpoint-Tabelle („CPT“) (**170, 270, 370**) zum Speichern einer Checkpoint-Version der Umbenennungsdaten;  
 einen Rückordnungspuffer („ROB“) (**160, 260, 360**), welcher ROB-Einträge zum Speichern von ROB-Daten (**162, 262A, 362**), welche von den Mikrobefehlen erzeugt werden, und einen Retire-Zeiger hat, welcher auf einen variablen der ROB-Einträge zeigt, wobei die ROB-Daten (**162, 262A, 362**) Wiederherstellungs-Umbenennungsdaten (**182, 282A, 382**) aufweisen, wobei eine Gruppe der ROB-Einträge als ein ROB-Checkpoint-Fenster definiert ist, wobei das ROB-Checkpoint-Fenster gemäß einer dynamischen Beziehung (**187, 287**) in die CPT (**170, 270, 370**) mappt, wobei das ROB-Checkpoint-Fenster definiert wird basierend auf dem ROB-Eintrag, auf welchen der Retire-Zeiger zeigt, und  
 in welchem, wenn Umbenennungsdaten, welche in der Umbenennungstabelle (**136, 336**) gespeichert sind, von der Umbenennungstabelle (**136, 336**) geleert werden, die entleerten Umbenennungsdaten zu der Umbenennungstabelle (**136, 336**) wiederhergestellt werden können durch:

die Checkpoint-Version der Umbenennungsdaten, welche von der CPT (170, 270, 370) zu der Umbenennungstabelle (136, 336) kopiert wird, wenn es durch einen Aspekt der dynamischen Beziehung (187, 287) erlaubt ist, wobei zusätzlich wenigstens ein Abschnitt der Wiederherstellungs-Umbenennungsdaten (182, 282A, 382) in dem ROB-Wiederherstellungsfenster von dem ROB (160, 260, 360) zu der Umbenennungstabelle (136, 336) kopiert wird.

9. Mikroprozessor (110, 310) nach Anspruch 8, weiterhin aufweisend:  
einen Dispatcher (140, 340) zum Absenden der Mikrobefehle zu der Ausführungseinheit (151, 152, 153, 351, 352, 353), und  
bei welchem die CPT (170, 270, 370) die Checkpoint-Version der Umbenennungsdaten von dem Dispatcher (140, 340) empfängt.

10. Mikroprozessor (110, 310) nach Anspruch 8, weiterhin aufweisend:  
eine Abzweigungs-Ausführungslogik (157, 357) zum Erfassen einer Fehl-Vorhersage beim Ausführen der Mikrobefehle, und  
bei welchem in Antwort darauf, dass eine Fehl-Vorhersage erfasst wird, die Umbenennungsdaten von der Umbenennungstabelle (136, 336) entleert werden.

11. Mikroprozessor (110, 310) nach Anspruch 8, weiterhin aufweisend:  
einen Taktgeber zum Ausgeben von Pulsen, welche Taktzyklen definieren, und  
bei welchem die Checkpoint-Version der Umbenennungsdaten von der Checkpoint-Tabelle zu der Umbenennungstabelle (136, 336) in einem einzelnen der Taktzyklen kopiert wird.

12. Verfahren für einen Mikroprozessor (110, 310), das Folgendes aufweist:  
ein Dekodieren eines Befehls in Mikrobefehle;  
ein Ausführen der Mikrobefehle;  
ein Erzeugen von Umbenennungsdaten gemäß den Mikrobefehlen;  
ein Speichern der Umbenennungsdaten in einer Umbenennungstabelle (136, 336) derart, dass die Umbenennungsdaten aus der Umbenennungsdaten entleert werden können;  
ein Speichern einer Checkpoint-Version der Umbenennungsdaten in einer Checkpoint-Tabelle („CPT“) (170, 270, 370), welche eine Tiefe hat; und  
ein Speichern in einem Rückordnungspuffer („ROB“) (160, 260, 360) von Einträgen eines ROB (160, 260, 360), ROB-Daten (162, 262A, 362), welche aus den Mikrobefehlen erzeugt werden, wobei die ROB-Daten (162, 262A, 362) Wiederherstellungs-Umbenennungsdaten, einen Retire-Zeiger des ROB (160, 260, 360), welcher auf einen variablen der ROB-Einträge zeigt, aufweist, wobei eine Gruppe der ROB-Einträge

ge als ein ROB-Checkpoint-Fenster definiert ist, wobei das ROB-Checkpoint-Fenster gemäß einer dynamischen Beziehung (187, 287) in die CPT (170, 270, 370) mappt, wobei das ROB-Checkpoint-Fenster definiert wird basierend auf dem ROB-Eintrag, auf welchen der Retire-Zeiger zeigt; und  
wenn Umbenennungsdaten, welche in der Umbenennungstabelle (136, 336) gespeichert sind, von der Umbenennungstabelle (136, 336) entleert werden, die entleerten Umbenennungsdaten zu der Umbenennungstabelle (136, 336) wiederhergestellt werden durch:  
ein Kopieren der Checkpoint-Version der Umbenennungsdaten von der CPT (170, 270, 370) zu der Umbenennungstabelle (136, 336), wenn es durch einen Aspekt der dynamischen Beziehung (187, 287) erlaubt ist, sowie  
ein Kopieren wenigstens eines Abschnitts der wiederhergestellten Umbenennungsdaten von dem ROB (160, 260, 360) zu der Umbenennungstabelle (136, 336).

13. Verfahren nach Anspruch 12, bei welchem die Mikrobefehle für die Ausführung von dem Dekoder (120, 320) über einen Dispatcher (140, 340) empfangen werden, und die CPT (170, 270, 370) die Checkpoint-Version der Umbenennungsdaten von dem Dispatcher (140, 340) empfängt.

14. Verfahren nach Anspruch 12, weiterhin aufweisend:  
ein Erfassen einer Fehl-Vorhersage beim Ausführen der Mikrobefehle, und  
bei welchem in Antwort darauf, dass eine Fehl-Vorhersage erfasst wird, die Umbenennungsdaten von der Umbenennungstabelle (136, 336) entleert werden.

15. Verfahren nach Anspruch 12, bei welchem Pulse ausgegeben werden, welche Taktzyklen definieren, und  
die Checkpoint-Version der Umbenennungsdaten von der Checkpoint-Tabelle zu der Umbenennungstabelle (136, 336) in einem einzelnen der Taktzyklen kopiert wird.

16. Mikroprozessor (110, 310), der Folgendes aufweist:  
einen Umbenener (130, 330) zum Erzeugen von Umbenennungsdaten gemäß Mikrobefehlen;  
eine Umbenennungstabelle (136, 336) zum Speichern der Umbenennungsdaten;  
einen Rückordnungspuffer („ROB“) (160, 260, 360) zum Speichern von ROB-Daten (162, 262A, 362), welche aus den Mikrobefehlen erzeugt werden, wobei die ROB-Daten (162, 262A, 362) Wiederherstellungs-Umbenennungsdaten (182, 282A, 382) aufweisen; und

genau eine Checkpoint-Tabelle („CPT“) (170, 270, 370) zum Speichern einer Checkpoint-Version der Umbenennungsdaten, und bei welchem, wenn Umbenennungsdaten, welche in der Umbenennungstabelle (136, 336) gespeichert sind, von der Umbenennungstabelle (136, 336) entleert werden, die entleerten Umbenennungsdaten wiederhergestellt werden durch:  
die Checkpoint-Version der Umbenennungsdaten, welche von der CPT (170, 270, 370) zu der Umbenennungstabelle (136, 336) kopiert werden,  
jedoch ohne irgendeinen Abschnitt der Wiederherstellungs-Umbenennungsdaten (182, 282A, 382), welche von dem ROB (160, 260, 360) zu der Umbenennungstabelle (136, 336) kopiert werden.

17. Mikroprozessor (110, 310) nach Anspruch 16, weiterhin aufweisend:  
einen Dispatcher (140, 340) zum Absenden der Mikrobefehle zu der Ausführungseinheit (151, 152, 153, 351, 352, 353), und  
bei welchem die CPT (170, 270, 370) die Checkpoint-Version der Umbenennungsdaten von dem Dispatcher (140, 340) empfängt.

18. Mikroprozessor (110, 310) nach Anspruch 16, weiterhin aufweisend:  
eine Abzweigungs-Ausführungslogik (157, 357) zum Erfassen einer Fehl-Vorhersage bei einer Ausführung der Mikrobefehle, und  
bei welchem die Umbenennungstabelle (136, 336) so wiederhergestellt wird in Antwort auf eine Fehl-Vorhersage, welche erfasst wird.

19. Mikroprozessor (110, 310) nach Anspruch 16, weiterhin aufweisend:  
einen Taktgeber zum Ausgeben von Pulsen, welche Taktzyklen definieren, und  
bei welchem die Checkpoint-Version der Umbenennungsdaten von der Checkpoint-Tabelle zu der Umbenennungstabelle (136, 336) in einem einzelnen der Taktzyklen kopiert wird.

20. Verfahren für einen Mikroprozessor (110, 310), das Folgendes aufweist:  
ein Erzeugen von Umbenennungsdaten gemäß Mikrobefehlen;  
ein Speichern der Umbenennungsdaten in einer Umbenennungstabelle (136, 336);  
ein Speichern von Rückordnungspuffer („ROB“)-Daten, welche von den Mikrobefehlen erzeugt werden, in einem ROB (160, 260, 360);  
ein Speichern einer Checkpoint-Version der Umbenennungsdaten in genau einer Checkpoint-Tabelle („CPT“) (170, 270, 370), wobei die ROB-Daten (162, 262A, 362) Wiederherstellungs-Umbenennungsdaten (182, 282A, 382) aufweisen; und wobei wenn Umbenennungsdaten, welche in der Umbenennungstabelle (136, 336) gespeichert sind, von der Umbenennungstabelle (136, 336) entleert werden,

die entleerten Umbenennungsdaten zu der Umbenennungstabelle (136, 336) wiederhergestellt werden durch:  
ein Kopieren der Checkpoint-Version der Umbenennungsdaten von der CPT (170, 270, 370) zu der Umbenennungstabelle (136, 336),  
jedoch ohne ein Kopieren beliebiger Wiederherstellungs-Umbenennungsdaten (182, 282A, 382) von dem ROB (160, 260, 360) zu der Umbenennungstabelle (136, 336).

21. Verfahren nach Anspruch 20, weiterhin aufweisend:  
ein Erfassen einer Fehl-Vorhersage beim Ausführen der Mikrobefehle, und  
bei welcher, in Antwort darauf, dass eine Fehlvorhersage erfasst wird, die Umbenennungsdaten von der Umbenennungstabelle (136, 336) entleert werden.

22. Elektronische Vorrichtung, die Folgendes aufweist:  
einen Bus (850),  
eine Schnittstelle (840), welche konfiguriert ist zum Übertragen von Daten zu oder Empfangen von Daten von einem Kommunikationsnetzwerk, welches mit dem Bus (850) verbunden ist,  
eine I/O-Vorrichtung (820), welche mit dem Bus (850) verbunden ist,  
einen Speicher (830), welcher mit dem Bus (850) verbunden ist, welcher konfiguriert ist, um Verwender-Daten zu speichern, auf welche über die I/O-Vorrichtung (820) oder Befehle zugegriffen werden kann; und  
einen Controller (810), welcher mit dem Bus (850) verbunden ist und konfiguriert ist, um die Befehle zu verwenden, wobei der Controller (810) wenigstens einen Mikroprozessor (110, 310) enthält, der Folgendes aufweist:  
eine Umbenennungstabelle (136, 336) zum Speichern von Umbenennungsdaten, einen Dispatcher (140, 340), und  
eine Checkpoint-Tabelle („CPT“) (170, 270, 370) zum Speichern von Umbenennungsdaten, welche von dem Dispatcher (140, 340) empfangen werden, bei welcher, wenn die Umbenennungstabelle (136, 336) entleert wird, die Umbenennungstabelle (136, 336) unter Verwendung der Umbenennungsdaten, welche in der CPT (170, 270, 370) gespeichert sind, wiederaufgebaut wird.

23. Vorrichtung nach Anspruch 22, in welcher die Schnittstelle (840) drahtlos ist und das Kommunikationsnetzwerk drahtlos ist.

24. Vorrichtung nach Anspruch 22, in welcher die Umbenennungstabelle (136, 336) unter Verwendung nur der Umbenennungsdaten, welche in der CPT (170, 270, 370) gespeichert sind, wiederhergestellt wird.

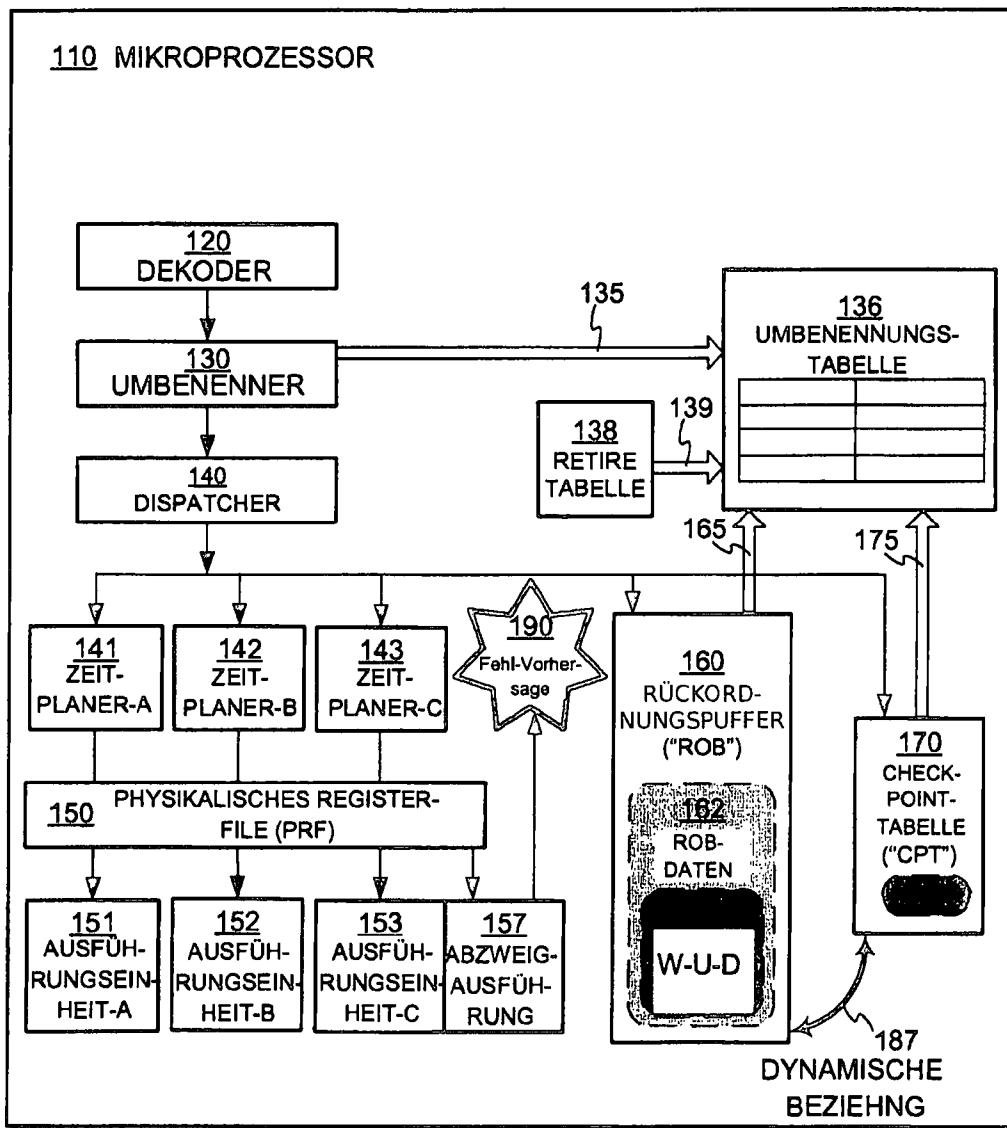
25. Vorrichtung nach Anspruch 22, weiterhin aufweisend:  
eine Abzweigungs-Ausführungslogik (**157, 357**) zum Erfassen einer Fehl-Vorhersage, und bei welcher in Antwort darauf, dass eine Fehl-Vorhersage erfasst wird, die Umbenennungstabelle (**136, 336**) entleert wird.

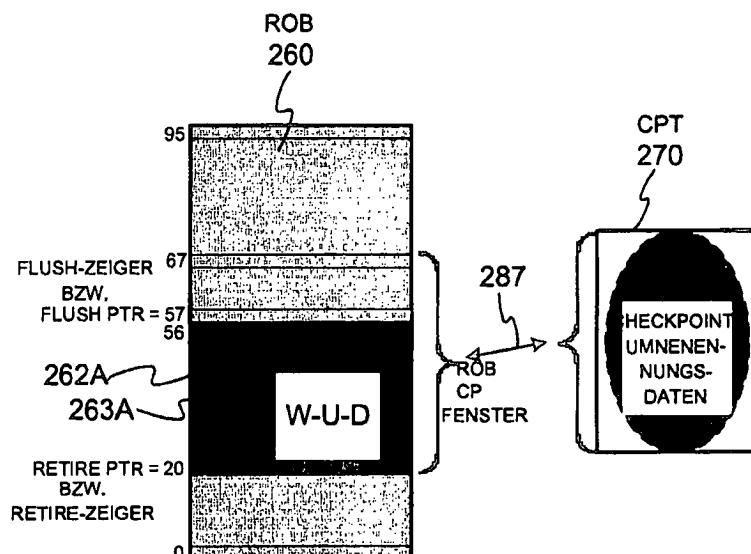
26. Vorrichtung nach Anspruch 22, weiterhin aufweisend:  
einen Taktgeber zum Ausgeben von Pulsen, welche Taktzyklen definieren, und bei welcher die Umbenennungstabelle (**136, 336**) in einem einzelnen der Taktzyklen wiederaufgebaut wird.

27. Vorrichtung nach Anspruch 22, weiterhin aufweisend:  
einen Rückordnungspuffer („ROB“) (**160, 260, 360**), welcher ROB-Einträge hat, und einen Retire-Zeiger, welcher auf einen variable der ROB-Einträge zeigt, und in welchem eine Gruppe der ROB-Einträge als ein ROB-Checkpoint-Fenster definiert ist, wobei das ROB-Checkpoint-Fenster gemäß einer dynamischen Beziehung (**187, 287**) in die CPT (**170, 270, 370**) mappt, wobei das ROB-Checkpoint-Fenster definiert ist basierend auf dem ROB-Eintrag, auf welchen der Retire-Zeiger zeigt, und wobei die Umbenennungstabelle (**136, 336**) unter Verwendung der Umbenennungsdaten, welche in der CPT (**170, 270, 370**) gespeichert sind, wiederaufgebaut wird, nur wenn es durch einen Aspekt der dynamischen Beziehung (**187, 287**) erlaubt ist.

Es folgen 11 Seiten Zeichnungen

## Anhängende Zeichnungen

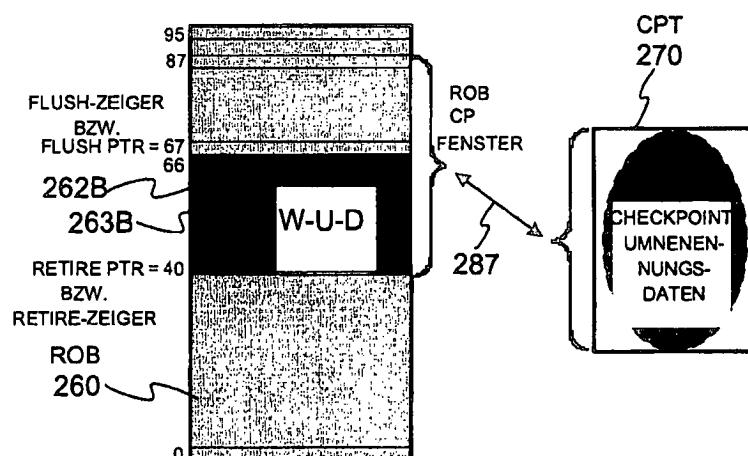
**FIG. 1** ALLGEMEINE AUSFÜHRUNGSFORM



W-U-D = WIEDERHERSTELLUNGS-UMBENENNUNGS-DATEN

BEISPIEL-SZENARIO BEI  
NORMALBETRIEB

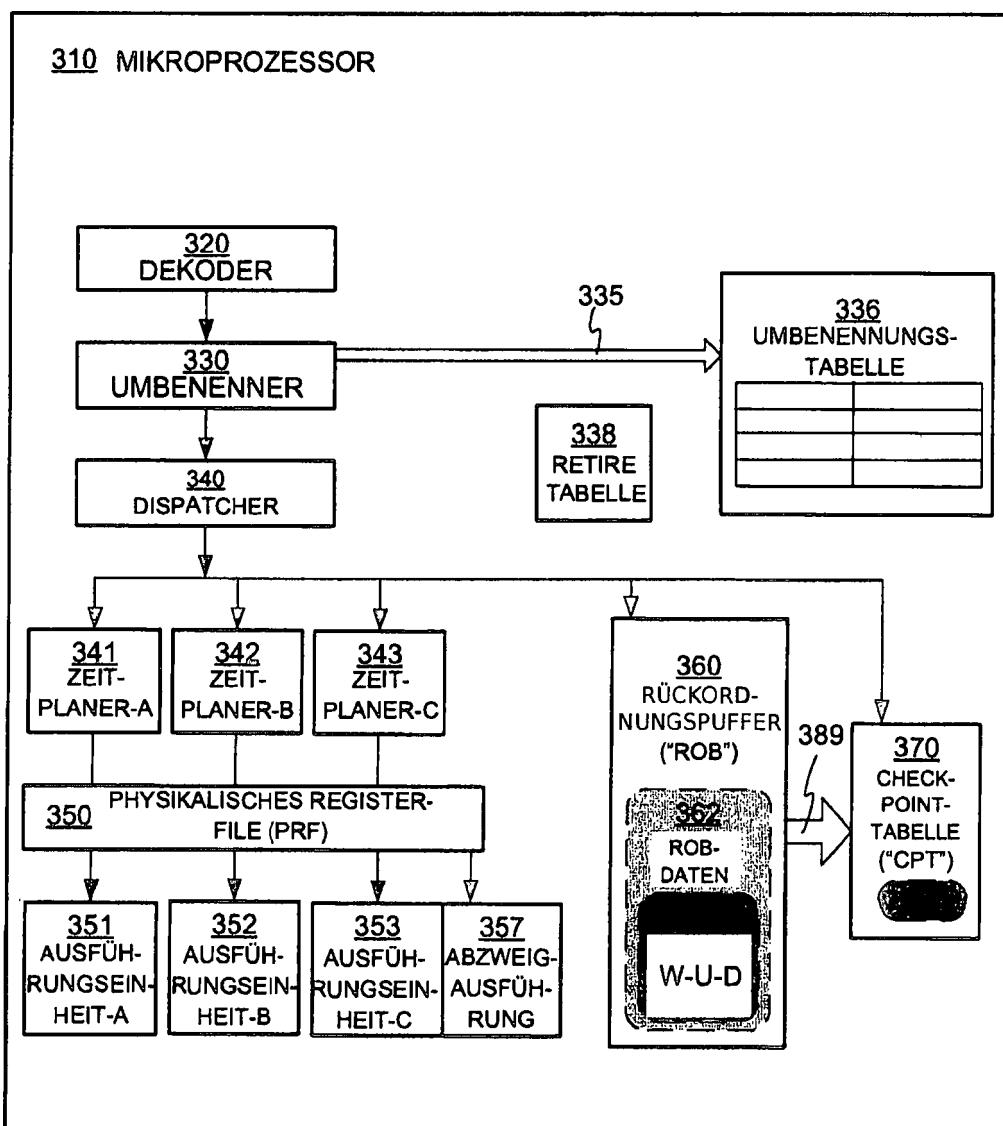
**FIG. 2A**



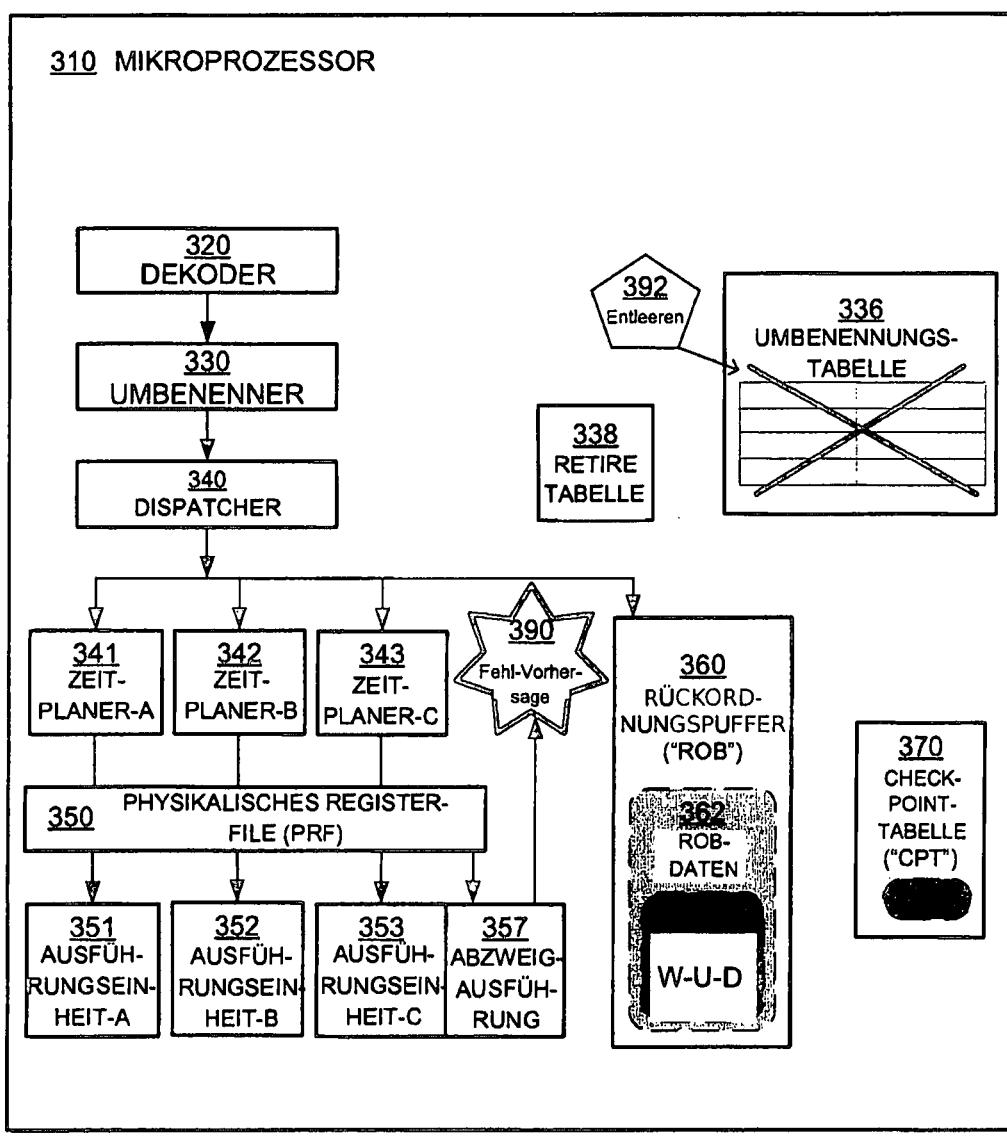
W-U-D = WIEDERHERSTELLUNGS-UMBENENNUNGS-DATEN

BEISPIEL-SZENARIO BEI  
NORMALBETRIEB

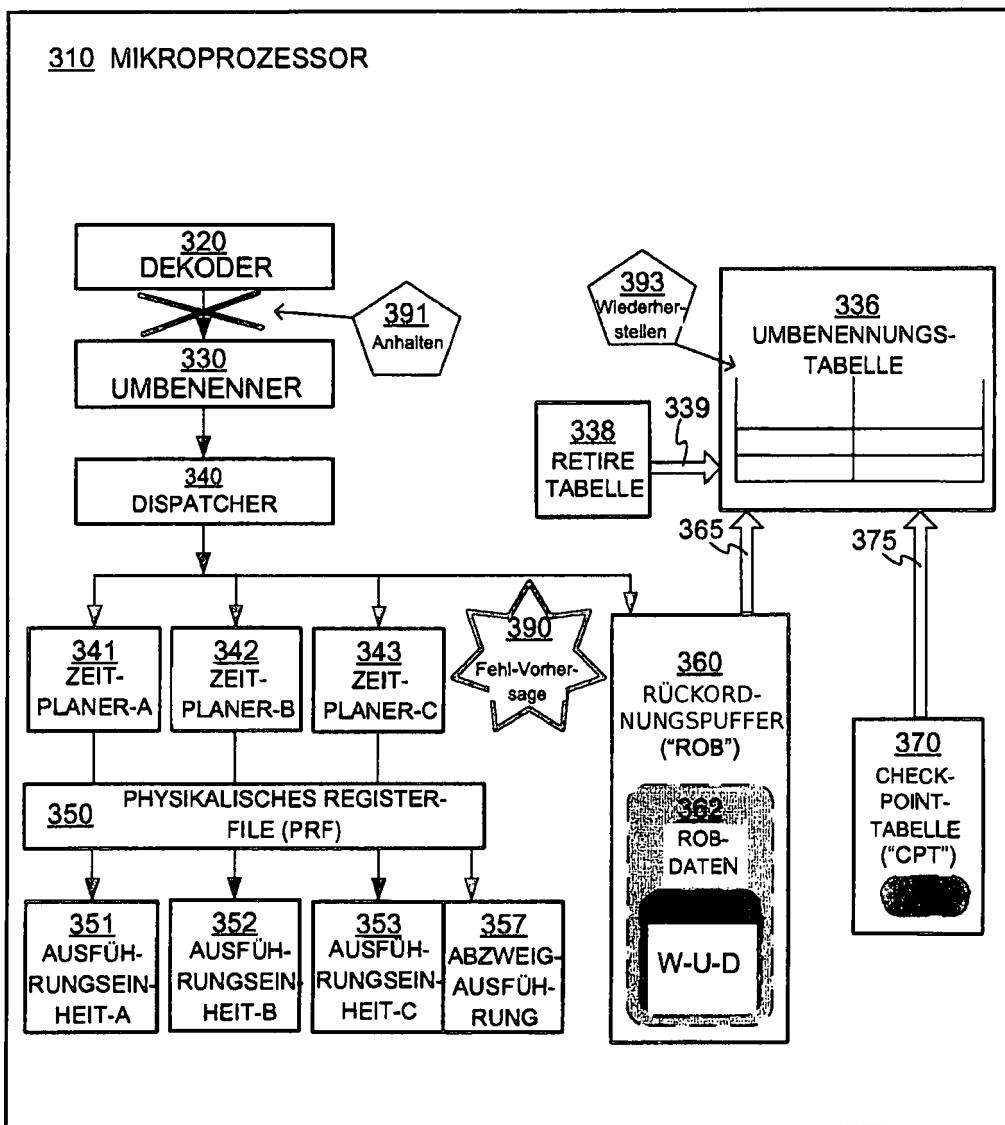
**FIG. 2B**

**FIG. 3A**

NORMALBETRIEB

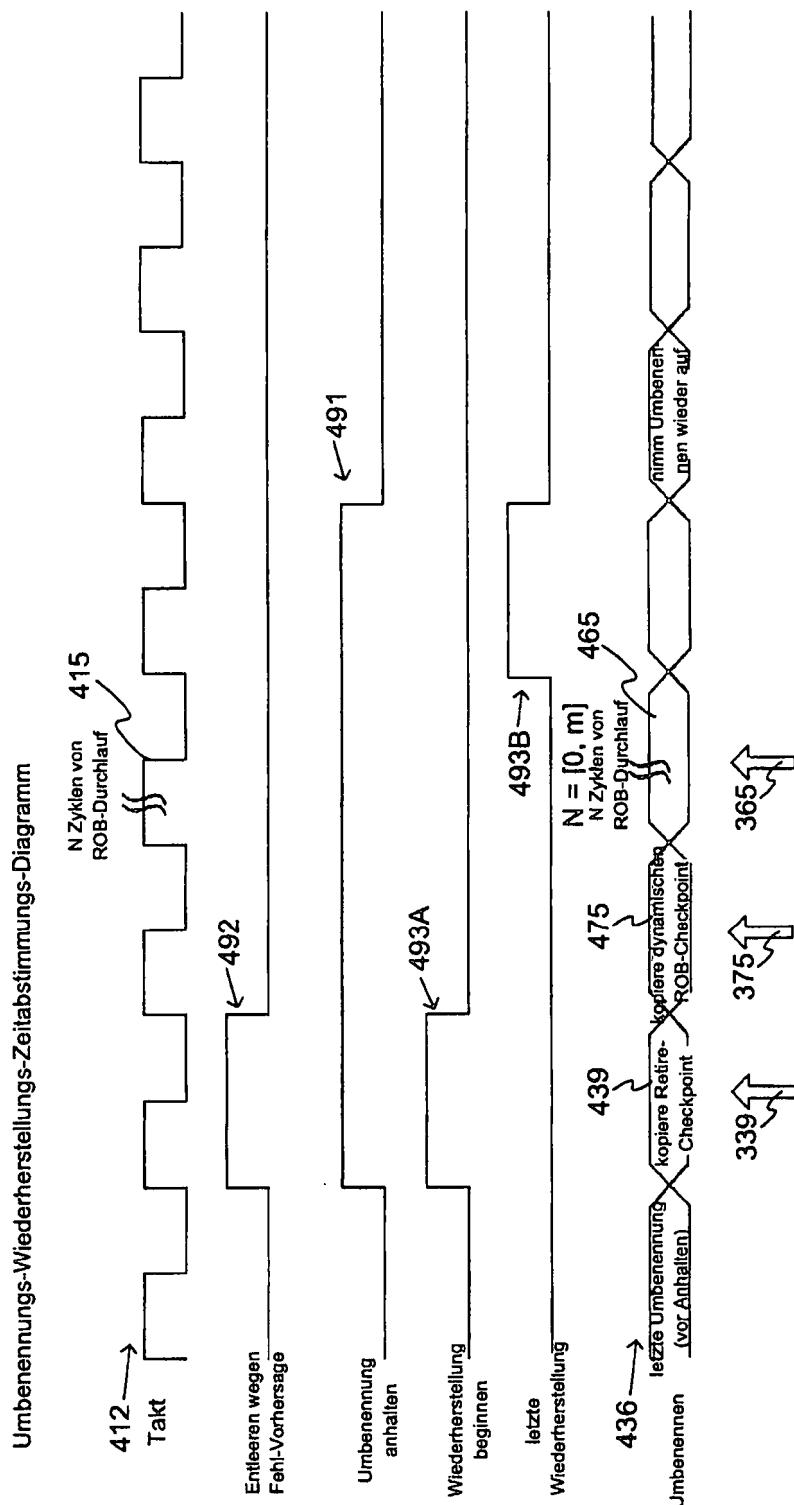
**FIG. 3B**

FEHL-VORHERSAGE ERFASST



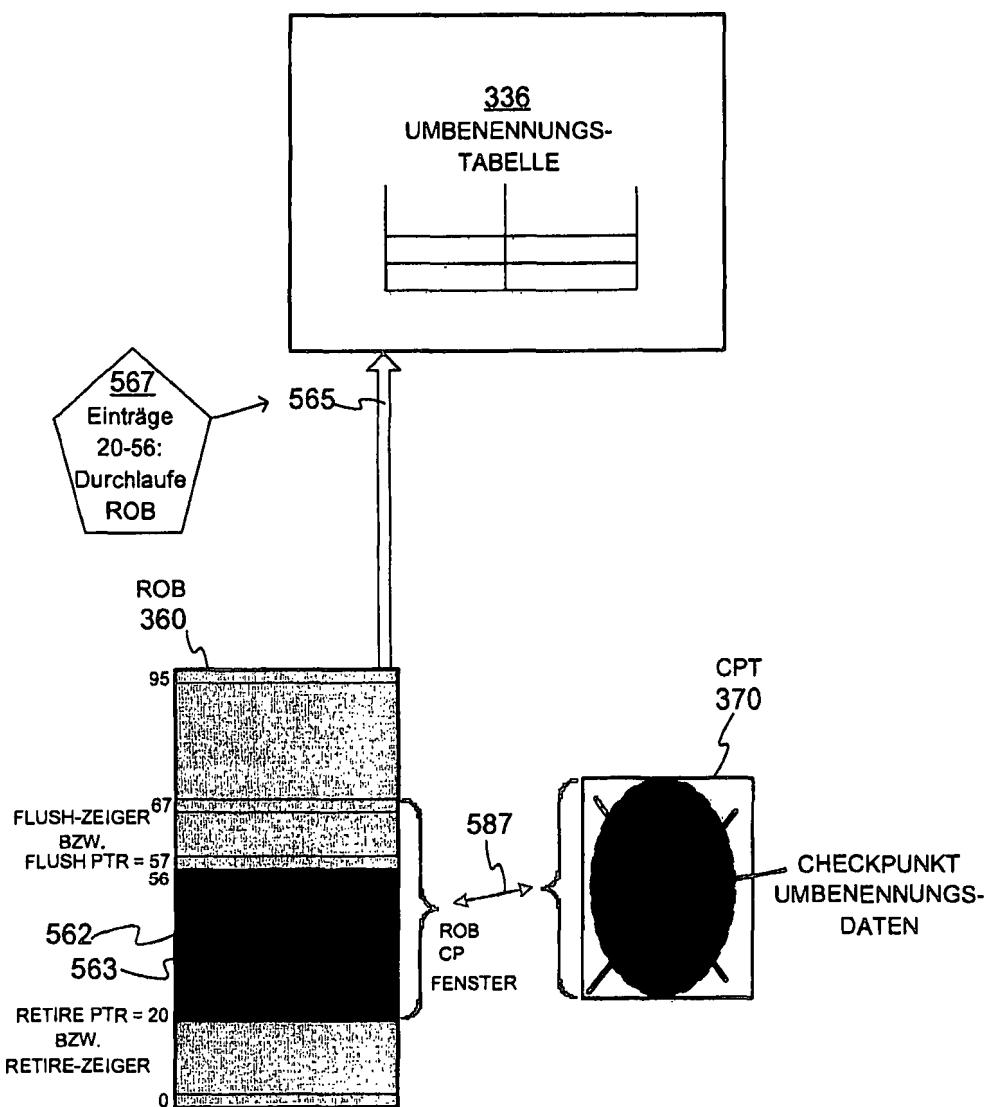
**FIG. 3C**

WIEDERHERSTELLEN DER  
UMBENENNUNGSTABELLE

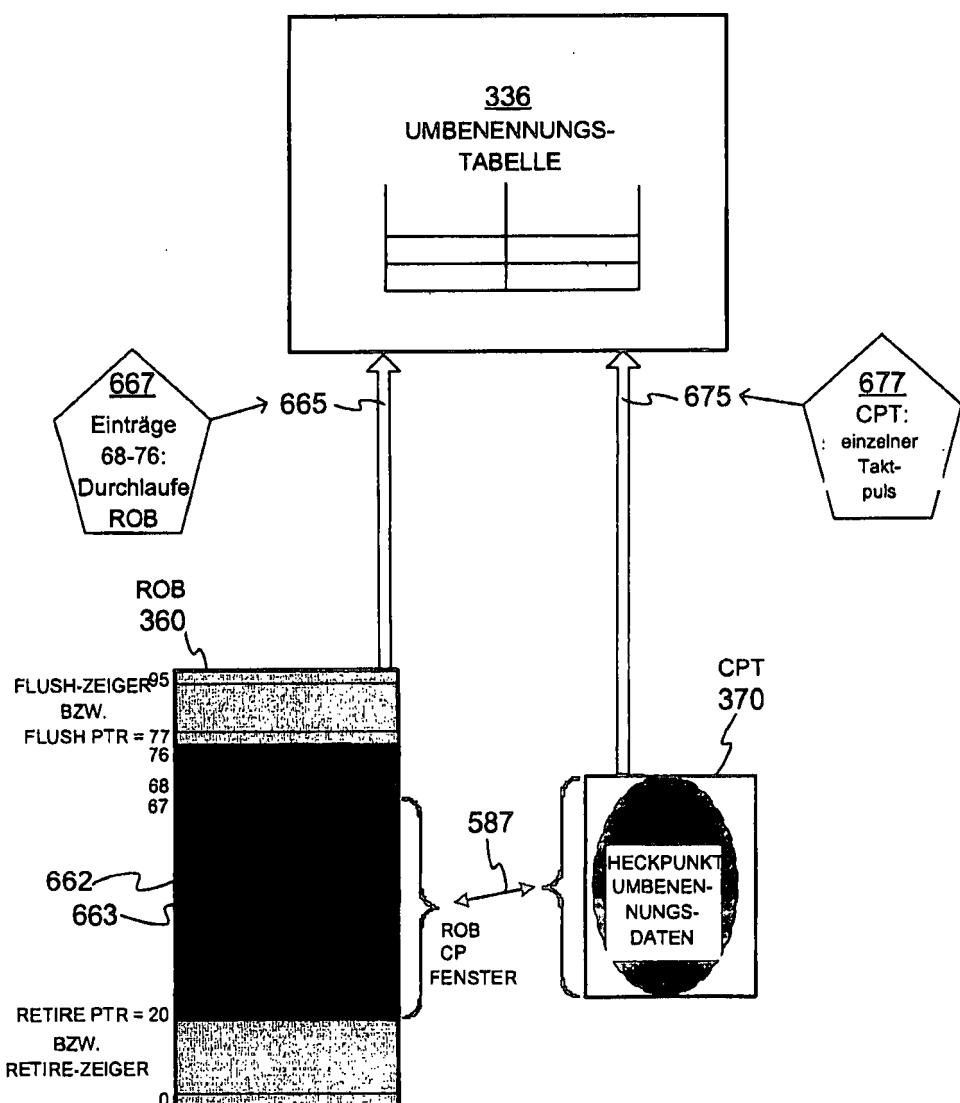


ZEITABSTIMMUNG

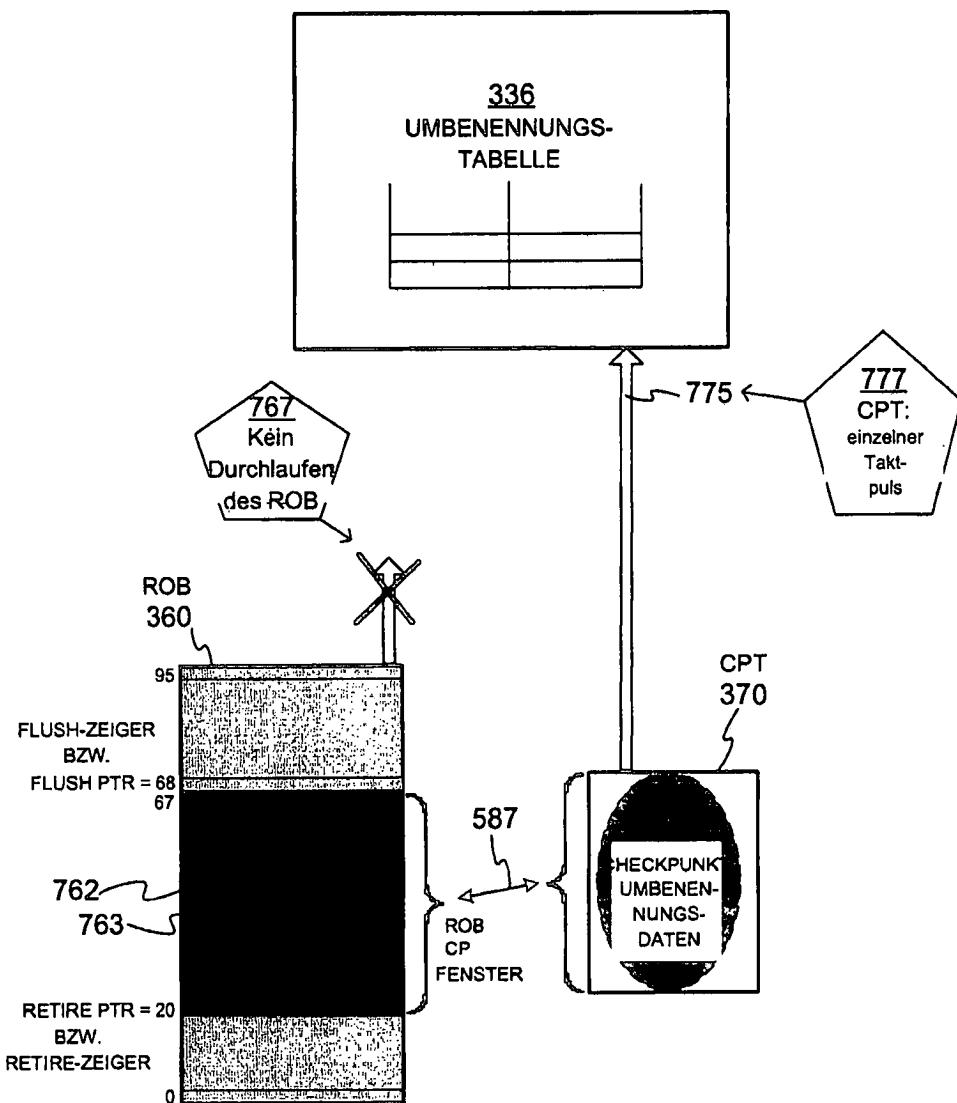
**FIG. 4**



BEISPIELS-  
WIEDERHERSTELLUNGS-SZENARIO  
**FIG. 5** (KEIN VORTEIL)



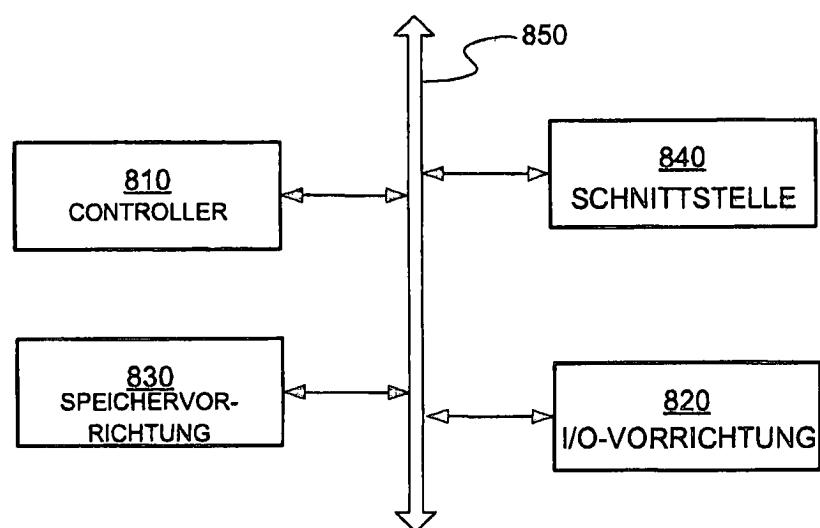
BEISPIELS-  
WIEDERHERSTELLUNGS-SZENARIO  
**FIG. 6** (VORTEIL)



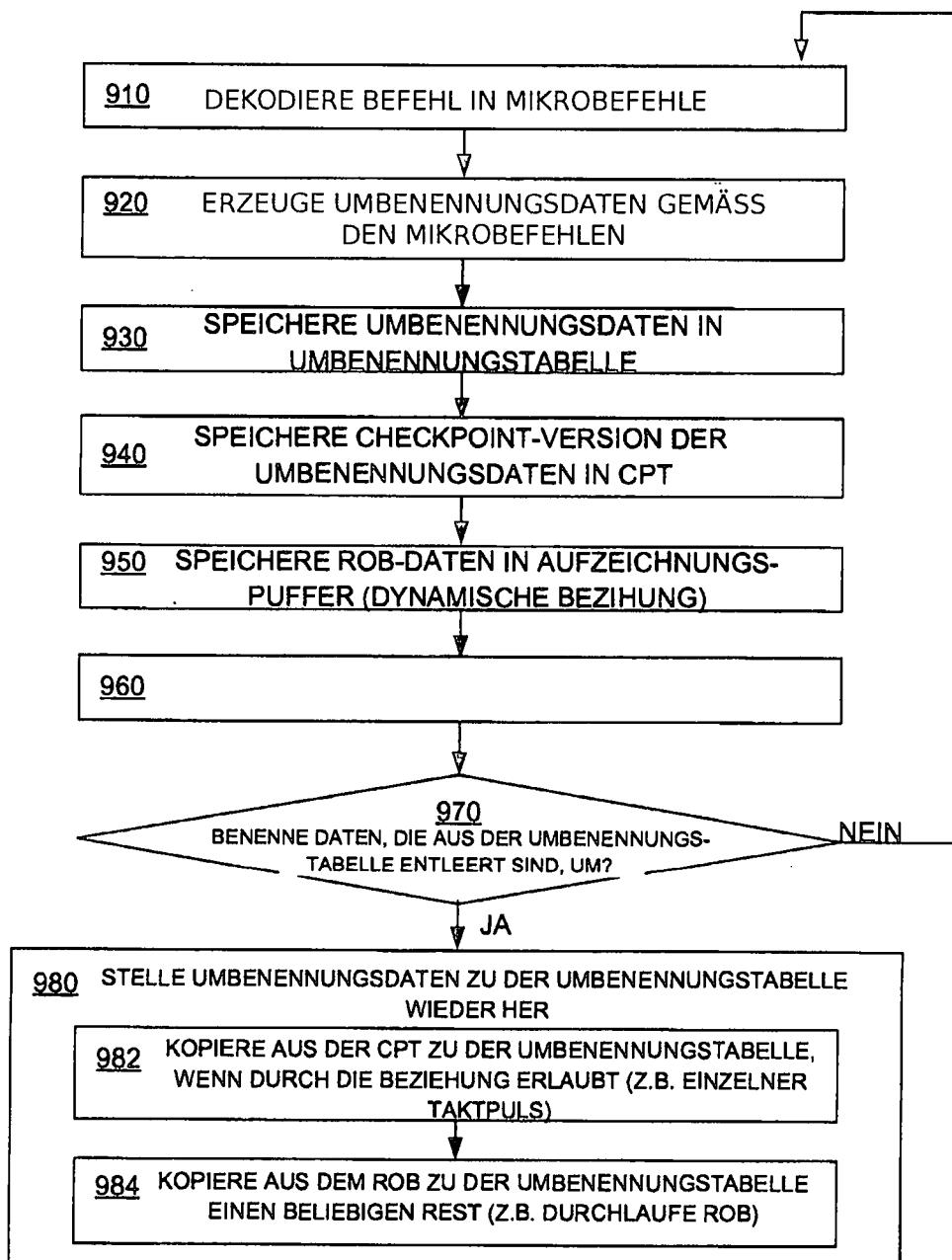
**FIG. 7**

BEISPIELS-  
WIEDERHERSTELLUNGS-SZENARIO  
(GRÖSSTER VORTEIL)

800



**FIG. 8**

900**FIG. 9**

VERFAHREN