



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 697 38 188 T2** 2008.01.17

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 0 823 085 B1**

(51) Int Cl.⁸: **G06F 9/38** (2006.01)

(21) Deutsches Aktenzeichen: **697 38 188.9**

(86) PCT-Aktenzeichen: **PCT/US97/02184**

(96) Europäisches Aktenzeichen: **97 906 557.0**

(87) PCT-Veröffentlichungs-Nr.: **WO 1997/030389**

(86) PCT-Anmeldetag: **10.02.1997**

(87) Veröffentlichungstag

der PCT-Anmeldung: **21.08.1997**

(97) Erstveröffentlichung durch das EPA: **11.02.1998**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **10.10.2007**

(47) Veröffentlichungstag im Patentblatt: **17.01.2008**

(30) Unionspriorität:

601744 **15.02.1996** **US**

(84) Benannte Vertragsstaaten:

DE, FR, GB

(73) Patentinhaber:

Fujitsu Ltd., Kawasaki, Kanagawa, JP

(72) Erfinder:

KULKARNI, Paritosh M., Campbell, CA 95008, US;
REEVE, Richard, Los Gatos, CA 95030, US;
SAXENA, Nirmal R., Los Altos Hills, CA 94024, US

(74) Vertreter:

W. Seeger und Kollegen, 81369 München

(54) Bezeichnung: **VERFAHREN UND APPARAT FÜR EINE ERHÖHTE GENAUIGKEIT BEI DER VERZWEIGUNGS-VORHERSAGE IN EINEM SUPERSKALAREN MIKROPROZESSOR**

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

Gebiet der Erfindung

[0001] Die vorliegende Erfindung betrifft Mikroprozessoren und insbesondere die Vorhersage von Verzweigungsanweisungen in superskalaren Mikroprozessoren.

Hintergrund der Erfindung

[0002] Gewöhnliche Mikroprozessoren, die keine superskalare Architektur oder Multipipline-Architektur verwenden, empfangen Anweisungen von einem seriellen Anweisungsstrom und verarbeiten diese Anweisungen sequentiell in einer logischen Reihenfolge, was Sprünge und Verzweigungen ermöglicht. Wenn auf eine bedingte Verzweigungsanweisung gestoßen wird, testet der Mikroprozessor bestimmte Merker, die von Anweisungen gesetzt wurden, die vorher von dem Mikroprozessor ausgeführt wurden und nimmt entweder die Ausführung bei der Anweisung auf, die auf die bedingte Verzweigungsanweisung in dem seriellen Anweisungsstrom folgte, oder nimmt die Ausführung bei einer Anweisung auf, die bei einem Ort gespeichert ist, der durch die bedingte Verzweigungsanweisung beschrieben ist.

[0003] Superskalare Mikroprozessoren können einen seriellen Anweisungsstrom empfangen und die gleichen Ergebnisse erzeugen wie nichtsuperskalare Mikroprozessoren. Jedoch können superskalare Mikroprozessoren intern mehrere Anweisungen gleichzeitig verarbeiten, was dazu führen kann, daß Anweisungen nicht in ihrer logischen Reihenfolge ausgeführt werden, wobei die Reihenfolge von dem ursprünglichen Erzeuger der Anweisungen beabsichtigt ist.

[0004] Mit Bezugnahme auf [Fig. 1](#) sind ein gewöhnlicher superskalarer Mikroprozessor **102** und ein Speicher **104** gezeigt. Die Abrufschaltung **106** weist den Speicher **104** an, Blöcke von Anweisungen **110**, **112** beginnend mit der Speicheradresse, die in dem Abrufprogrammzähler **108** enthalten ist, ein Block auf einmal in den Speicherbereich **114** zur gleichzeitigen Verarbeitung durch die Ausführungseinheiten **116**, **118**, **120** zu übermitteln. Obwohl die Größe der Blöcke **110**, **112** und des Speicherbereichs **114**, die in [Fig. 1](#) gezeigt sind, vier Wörter ist, und die Anzahl der Ausführungseinheiten **116**, **118**, **120**, die in [Fig. 1](#) gezeigt sind, drei ist, können gewöhnliche superskalare Mikroprozessoren Blöcke **110**, **112** und Speicherbereiche **114** von irgendeiner Größe und irgendeiner Anzahl von Ausführungseinheiten **116**, **118**, **120** haben.

[0005] Das Erzeugen von Ergebnissen in einem superskalaren Mikroprozessor, die mit den Ergebnissen identisch sind, die von einem gewöhnlichen nicht su-

perskalare Mikroprozessor erzeugt worden wären, stellt gewisse Probleme für einen superskalaren Mikroprozessor dar. Ein Problem, das sich beim Entwurf eines superskalaren Mikroprozessors stellt, beruht auf der Verarbeitung einer bedingten Verzweigungsanweisung. Weil die Anweisungen, welche die Merker in einem superskalaren Mikroprozessor setzen, zu der Zeit, zu der die Verzweigungsanweisung für die Ausführung durch den superskalaren Mikroprozessor bereit ist, nicht verarbeitet sein können, ist es unmöglich, mit Sicherheit zu bestimmen, welche Anweisung der nichtsuperskalare Mikroprozessor nach der Ausführung der bedingten Verzweigungsanweisung ausgeführt haben könnte, ohne auf alle Anweisungen zu warten, die der bedingten Verzweigungsanweisung logisch vorausgehen, um ausgeführt zu werden. Das Warten auf alle solche vorausgehenden Anweisungen, ausgeführt zu werden, würde unerwünschte Verzögerungen hervorrufen.

[0006] Ein Lösungsweg, diese Verzögerungen zu vermeiden, war es zu versuchen, das Ergebnis der bedingten Verzweigungsanweisung vorherzusagen, ohne darauf zu warten, daß die logisch vorhergehenden Anweisungen ausgeführt wurden, und mit der Verarbeitung der Anweisungen fortzufahren, als ob die Vorhersage genau wäre. Wenn die Anweisungen, die logisch der bedingten Verzweigung vorausgehen, alle vollständig ausgeführt wurden, kann die Genauigkeit der Vorhersage getestet werden. Wenn das Ergebnis der Verzweigungsvorhersage tatsächlich genau ist, geht die Verarbeitung weiter, und unerwünschte Verzögerungen werden vermieden. Wenn das Ergebnis der Verzweigungsvorhersage ungenau ist, hält die Verarbeitung an und wird wieder bei der Anweisung aufgenommen, die nach der bedingten Verzweigung ausgeführt worden sein sollte, wobei die Verzögerung nicht größer ist, als wenn die Verarbeitung beim Warten auf die Ausführung der Anweisungen ausgesetzt wurde, die logisch der bedingten Verzweigungsanweisung vorangehen.

[0007] Unterschiedliche gewöhnliche Ideen sind zur Vorhersage vorhanden, welche Verzweigungsrichtung genommen werden soll. Ein Lösungsweg ist es, immer vorherzusagen, daß die Verzweigung, die in der Verzweigungsanweisung beschrieben wird, genommen wird. Solch eine Vorhersage kann oftmals mehr als 50 % der Zeit richtig sein, da viele Programme Schleifenanweisungen enthalten, die zu der Verzweigung führen, die in der Verzweigungsanweisung beschrieben ist, die öfter genommen als nicht genommen wird. Zum Beispiel verursachen die Pascal-Anweisungen:

```
FOR i:= 1 to 100 do begin
```

```
...
```

```
END;
```

daß die Verzweigung, die in der Verzweigungsanwei-

sung beschrieben ist, mehr als 99 % der Zeit genommen wird. Selbstverständlich können andere Anweisungen wie `if ... when, while ... do, and repeat ... until` nicht zu der gleichen Vorhersagegenauigkeit führen, aber das Schema ist relativ einfach umzusetzen, was wertvollen Bereich in einem superskalaren Mikroprozessor **102** einspart.

[0008] Wenn eine Anweisung, die in der bedingten Verzweigungsanweisung beschrieben ist, nach der bedingten Verzweigungsanweisung ausgeführt wird, wird die Handlung als „die Verzweigung nehmen“ beschrieben und wird somit die Verzweigung oder „Richtung“ der Verzweigung „genommen“. Wenn die Anweisung, die der Verzweigungsanweisung physikalisch folgt, ausgeführt wird, weil die Bedingungen der bedingten Verzweigungsanweisung nicht wahr waren, wird die Handlung als „nicht die Verzweigung“ nehmen beschrieben, und die Verzweigung oder „Richtung“ der Verzweigung wird als „nicht genommen“ beschrieben.

[0009] Eine Idee, welche die Genauigkeit der Verzweigungsvorhersage verbessern kann, ist als „bimodale“ Verzweigungsvorhersage bekannt und bezieht die Verwendung von einem Sättigungszähler mit zwei Bits als Vorhersageanzeiger ein, um anzuzeigen, ob eine Verzweigung genommen werden sollte. Ein Sättigungsanzeiger mit zwei Bit verwendet die Annahme, daß Verzweigungen in einer Gruppe genommen werden sollten, und so kann, ob eine Verzweigung oder ein Gruppe von Verzweigungen genommen werden sollte, mit Bezugnahme darauf vorausgesagt werden, ob die letzte Verzweigung oder Verzweigungen genommen wurden. Nun mit Bezugnahme auf [Fig. 2A](#) und [Fig. 2B](#) ist eine Veranschaulichung einer Zustandstabelle eines Sättigungszählers mit zwei Bits gezeigt. Ein Zustand **210** stellt einen starken Hinweis dar, daß die Verzweigung genommen werden sollte. Der Zustand **212** stellt einen schwachen Hinweis dar, daß die Verzweigung nicht genommen werden sollte. Der Zustand **214** stellt einen schwachen Hinweis dar, daß die Verzweigung genommen werden sollte. Der Zustand **216** stellt einen starken Hinweis dar, daß die Verzweigung genommen werden sollte. Der Zustand der Vorhersage kann auf irgendeinen Zustand **210, 212, 214, 216** initialisiert werden. Die Verzweigung wird als genommen vorausgesagt, wenn das höchstwertige Bit des gegenwärtigen Vorhersagezustands einen Wert „1“ wie die Zustände **214, 216** hat, und die Verzweigung ist nicht genommen, wenn das höchstwertige Bit in den gegenwärtigen Vorhersagezustand einen Wert von „0“ hat, wie die Zustände **210, 212**. Wenn die Vorhersage getestet wird, nachdem die Anweisungen, die der Verzweigung logisch vorhergehen ausgeführt wurden, wird der Zustand der Vorhersage gemäß Tabelle **218** geändert. Spalte **220** stellt den gegenwärtigen Zustand dar, Spalte **222** stellt den neuen Zustand dar, und Spalte **224** stellt die tatsächliche

Verzweigungshandlung dar: genommen, was bedeutet, daß die Verzweigung tatsächlich genommen wurde, oder nicht genommen, was bedeutet, daß die Verzweigung tatsächlich nicht genommen wurde. Auf einen starken Hinweis sind zwei tatsächliche Verzweigungen, die dem Hinweis entgegengesetzt sind, erforderlich, bevor eine Änderung der Verzweigungsvorhersage durchgeführt wird. Andere Anordnungen von Zählern einschließlich denjenigen mit mehr als zwei Bits können verwendet werden, um die Anzahl der tatsächlichen Verzweigungen zu variieren, die dem starken Hinweis entgegengesetzt sind, der erforderlich ist, um die Vorhersage zu ändern.

[0010] Die Zustände der [Fig. 2A](#) können auch die Werte haben, die den gezeigten entgegengesetzt sind: stark eingenommen, schwach eingenommen, schwach nicht eingenommen und stark nicht genommen jeweils für die Zustände **210, 212, 214, 216**. In diesem Fall zeigt das höchstwertige Bit, das einen Wert von „1“ hat, an, daß die Verzweigung als nicht genommen vorhergesagt werden sollte, „0“ zeigt an, daß die Verzweigung als genommen vorhergesagt werden sollte. Tabelle **218** aus [Fig. 2B](#) wird wie oben beschrieben mit den entgegengesetzten tatsächlichen Handlungen in Spalte **224** verwendet.

[0011] Die Genauigkeit der bimodalen Verzweigungsvorhersage kann durch die Verwendung eines Verlaufsregisters erhöht werden, welches den Verlauf der tatsächlich genommenen Verzweigungshandlung speichert. Die Verwendung eines Verlaufsregisters setzt voraus, daß die bedingten Verzweigungen gemäß sich wiederholenden Mustern genommen werden. Zum Beispiel wird in dem folgenden Pascal-Programm

```
FOR i:=1 to 100 do
  FOR j:=1 to 3 do begin
...
end;
```

die innere Verzweigung zweimal genommen werden, aber nicht ein drittes Mal, worauf die äußere Verzweigung folgt, welche ihre Verzweigung nimmt, ein Verhalten das 98 mal auf Grund der äußeren Verzweigung wiederholt wird. Die Kenntnis des Verhaltens der letzten vier Verzweigungen, sowohl der inneren als der äußeren Verzweigung, kann das Verhalten der nächsten Verzweigung mit höherer Genauigkeit als bimodale Verzweigungsvorhersage vorhersagen. Ein Schieberegister kann als Verlaufsregister verwendet werden, um das Verhalten der Verzweigungen durch Verschieben von Bits um eine einzige Position in einer einzigen Richtung (rechts oder links) für jede angetroffene Verzweigung nachzuverfolgen, wobei eine „1“ für jede Verzweigung eingeschoben wird, die tatsächlich genommen wird, um eine „0“ für jede Verzweigung eingeschoben wird, die nicht tat-

sächlich genommen wird. Zum Beispiel würde ein Linksschieberegister 1101 lesen, nachdem die äußere Verzweigung genommen wurde, mit der Null in der zweiten niedrigstwertigen Position, was zeigt, daß das Ende der inneren Schleife erreicht wurde. Die nächste Verzweigung sollte als genommen vorhergesagt werden, da sie die erste Verzweigung in der nächsten Iteration der inneren Schleife sein wird.

[0012] Das Verlaufsregister wird mit einer Verlaufstabelle und Sättigungszählern mit zwei Bits der bimodalen Verzweigungsvorhersage verwendet, um die Vorhersage zu beenden. Nun mit Bezugnahme auf [Fig. 3](#) werden die Inhalte eines Verlaufsregisters **308**, wie oben beschrieben, als ein Index zu einer Verlaufstabelle **312** verwendet. Der Zeiger **316**, der den gleichen Index **314** wie derjenige des Verlaufsregisters **308** hat, zeigt zu einem Sättigungszähler **318**, **320**, **322**, **324**, **326** mit zwei Bits, der eine Zustandstabelle hat, wie sie oben mit Bezugnahme auf [Fig. 2A](#) beschrieben ist, die verwendet wird, um die Verzweigungsvorhersage wie oben beschrieben zu bestimmen. Das gesamte Verlaufsregister **308** kann als ein Index zu der Tabelle **312** verwendet werden, oder eine bestimmte Anzahl von Bits, die das zuletzt in das Verlaufsregister **308** eingeschobene Bit einschließen und mit diesem benachbart sind, kann als ein Index für die Tabelle **312** verwendet werden.

[0013] Ein weiteres Verfahren ist dem oben beschriebenen Verlaufstabellenverfahren abgesehen davon ähnlich, daß die Adresse von allen oder einer bestimmten Anzahl der niedrigstwertigen Bits der Adresse der bedingten Verzweigungsanweisung statt des Verlaufsregisters **308** als der Index zu der Tabelle **312** verwendet wurde.

[0014] Noch weitere Verfahren kombinieren die Bits niedriger Ordnung der Adresse der bedingten Verzweigungsanweisung und einen Teil von dem oder den gesamten Verzweigungsverlauf zum Beispiel durch Verketteten oder Verknüpfen durch exklusives Oder, um einen Index zu der Tabelle **312** statt des Verlaufsregisters **308** alleine zu erzeugen.

[0015] Wieder mit Bezugnahme auf [Fig. 1](#), wenn die Adresse der bedingten Verzweigungsanweisung verwendet wird, um den Index zu erzeugen, muß die Adresse aus dem Zählerregister **108** des Abrufprogramms und der Position der bedingten Verzweigungsanweisung in dem Speicher **114** berechnet werden, was zusätzliche Komplexität des Mikroprozessors **102** und eine Rechenverzögerung verursacht. Wenn der Verlauf verwendet wird, um den Index zu erzeugen, muß er für jede ausgeführte bedingte Verzweigungsanweisung aktualisiert werden, was zu einer zusätzlichen Komplexität beim Entwurf des Mikroprozessors **102** führt.

[0016] EP-A2-0 586 057 offenbart ein System zum

vorherigen Abfragen und Versenden von Anweisungen unter Verwendung von vorausgehendem Versenden von Vorhersageanmerkungen, in dem ein Satz von Anweisungen, der in einem Anweisungs-Cache vorgesehen ist, mit einem oder mehreren Verzweigungsvorhersagefeldern und einem oder mehreren nächsten Vorhersagefeldern zum Adressenabfragen versehen ist. Vorgehensweisen zum Aktualisieren der Vorhersagen für jeden Zweig werden diskutiert.

[0017] IEEE, Proceedings of MICRO-28, 1995, Seiten 252 bis 257, „Alternative Implementations of Hybrid Branch Predictors“ erörtert die Leistungsfähigkeit unterschiedlicher Hybridverfahren mit zwei Stufen der Verzweigungsvorhersage. In einigen erörterten Schemata wird ein Verzweigungsverlauf der ersten Stufe verwendet, um zwischen alternativen Vorhersagern der zweiten Stufe auszuwählen, die zum Beispiel einen Zählervorhersager mit zwei Bits einschließen können, der eine Anordnung von Zählern mit zwei Bits verwendet, in denen jede Verzweigung durch ihre Adresse zu einem Zähler erfaßt ist, der für seine Vorhersage sorgt.

Zusammenfassung der Erfindung

[0018] Die vorliegende Erfindung schafft ein Verfahren nach Anspruch 1 und eine Vorrichtung nach Anspruch 5.

[0019] Eine Ausführungsform (Verfahren und Vorrichtung) der Erfindung sagt vorher, ob jede bedingte Verzweigungsanweisung in einer Menge von Verzweigungsanweisungen, die von einem Block eines Speichers abgefragt wurde, unter Verwendung einer Tabelle von Zeigern zu einer Anordnung von Sättigungszählern mit zwei Bits genommen werden sollte. Für jede bedingte Verzweigungsanweisung in der Menge wird der Index zu der Tabelle aus den niedrigstwertigen Bits einer Adresse des gleichen der Bytes in dem Block, der an ein Verlaufsregister angehängt oder ansonsten mit diesem kombiniert wird, das nur einmal für die Menge durch Einschieben einer „1“, wenn irgendeine der Verzweigungen in der Menge tatsächlich genommen wurden, ansonsten einer „0“ aktualisiert wird. Weil der Index nicht die Berechnung des exakten Speicherorts von jeder bedingten Verzweigungsanweisung erfordert, wird die Zeitdauer und Komplexität, die erforderlich ist, um den Index zu bestimmen, verringert. Weil die Verlaufstabelle nur einmal für die Menge aktualisiert wird, statt einmal für jede bedingte Verzweigungsanweisung in der Menge, wird die Komplexität weiter verringert.

Kurze Beschreibung der Zeichnungen

[0020] [Fig. 1](#) ist ein schematisches Blockdiagramm eines gewöhnlichen superskalaren Mikroprozessors

und eines gewöhnlichen Speichers.

[0021] [Fig. 2A](#) ist ein Zustandsdiagramm eines gewöhnlichen Sättigungszählers mit zwei Bits.

[0022] [Fig. 2B](#) ist eine Zustandstabelle, welche die Betriebsweise eines gewöhnlichen Sättigungszählers mit zwei Bits veranschaulicht, der durch [Fig. 2A](#) beschrieben wird.

[0023] [Fig. 3](#) ist ein schematisches Blockdiagramm eines gewöhnlichen Verzweigungsvorhersagers, der eine Tabelle verwendet.

[0024] [Fig. 4A](#) ist ein Flußdiagramm, das ein Verfahren der Vorhersage einer bedingten Verzweigung gemäß einer Ausführungsform der vorliegenden Erfindung veranschaulicht.

[0025] [Fig. 4B](#) ist ein Flußdiagramm, das ein Verfahren zum Aktualisieren eines Vorhersageanzeigers gemäß einer Ausführungsform der vorliegenden Erfindung veranschaulicht.

[0026] [Fig. 4C](#) ist ein Flußdiagramm, das ein Verfahren zum Aktualisieren eines Verlaufsregisters gemäß einer Ausführungsform der vorliegenden Erfindung veranschaulicht.

[0027] [Fig. 5](#) ist ein schematisches Blockdiagramm einer Schaltung mit bedingter Verzweigungsanweisung in einem superskalaren Mikroprozessor gemäß einer Ausführungsform der vorliegenden Erfindung.

Detaillierte Beschreibung einer bevorzugten Ausführungsform

[0028] Nun mit Bezugnahme auf [Fig. 4A](#) ist ein Flußdiagramm gezeigt, das ein Verfahren zum Vorhersagen einer bedingten Verzweigungsanweisung in einer Menge von mehreren Anweisungen gemäß der vorliegenden Erfindung veranschaulicht. Ein Teil von der oder die gesamte Speicheradresse für eine der Anweisungen in dem Bündel werden als **410** identifiziert. In einer Ausführungsform ist diese Adresse die Speicheradresse des ersten Bytes der ersten Anweisung in der Menge von Anweisungen. Andere Ausführungsformen verwenden die Adresse von weiteren Bytes der ersten oder weiteren Anweisungen in der Menge oder weitere Identifizierer, die für die Menge einmalig sind.

[0029] Ein Teil von oder das gesamte Verlaufsregister wird zur Verwendung wie unten beschrieben abgefragt **412**. Das Verlaufsregister kann, nach, vor oder im Wesentlichen zum gleichen Zeitpunkt abgefragt werden **412**, zu dem die Speicheradresse abgefragt wird **410**. Das Verlaufsregister kann auf irgendeinen Wert, wie alle Nullen, initialisiert und wie in [Fig. 4B](#) unten beschrieben aktualisiert werden.

[0030] Eine Vorhersagetabellenadresse wird unter Verwendung des Verlaufsregisters und der Speicheradresse von einer der Anweisungen in der Menge **414** berechnet. In einer Ausführungsform ist die Tabellenadresse eine Verkettung, die durch Legen des gesamten Verlaufsregisters in die höchst- oder niedrigstwertigen Bitpositionen der Tabellenadresse und einer bestimmten Anzahl von niedrigstwertigen Bits der Speicheradresse von einer der Anweisungen in der Menge in die verbleibenden Bitpositionen der Tabellenadresse gebildet wird. In einer weiteren Ausführungsform ist die Tabellenadresse eine Verkettung einer bestimmten Anzahl von Bits des Verlaufsregisters in den höchst- oder niedrigstwertigen Bitpositionen der Tabellenadresse und einer bestimmten Anzahl von niedrigstwertigen Bits der Speicheradresse von einer der Anweisungen in der Menge in den verbleibenden Bitpositionen der Tabellenadresse. In einer Ausführungsform wird das Verlaufsregister durch Verschieben eines Bits in die niedrigstwertige Bitposition des Verlaufsregisters aktualisiert, wobei die vier niedrigstwertigen Bits des Verlaufsregisters in die vier höchstwertigen Bitpositionen der Tabellenadresse gelegt werden und die acht niedrigstwertigen Bits der Adresse der ersten Anweisung in der Menge von Anweisungen in die acht verbleibenden niedrigstwertigen Bits der Tabellenadresse gelegt werden, um eine Tabellenadresse mit zwölf Bits zu bilden.

[0031] In einer der oben beschriebenen Ausführungsformen sind die Bits der Adresse von einem der Bytes der Anweisungen in der Menge, die verkettet werden soll, die niedrigstwertigen Bits der Adresse und die Bits des Verlaufsregisters, die verkettet werden sollen, sind die Bits des Verlaufsregisters, das die zuletzt eingeschobenen und mit diesen benachbarten Bits einschließt.

[0032] In einer weiteren Ausführungsform wird das Verlaufsregister ohne die Adresse von einem der Bytes der Anweisungen in der Menge verwendet, um die Tabellenadresse zu berechnen. In einer weiteren Ausführungsform wird die Adresse von einem der Bytes der Anweisungen in der Menge ohne das Verlaufsregister verwendet, um die Tabellenadresse zu berechnen.

[0033] Die Tabellenadresse kann auch unter Verwendung des Verlaufsregisters und der Adresse von einem der Bytes einer Anweisung in der Menge unter Verwendung von anderen Verfahren als Verkettung berechnet werden. In einer weiteren Ausführungsform sind einige oder alle Bits des Verlaufsregisters und einige oder alle Bits der Adresse von einem der Bytes der Anweisungen in der Menge durch exklusives Oder verknüpft, um eine Verlaufstabelle zu erzeugen. In einer Ausführungsform sind die Bits der Adresse von einem der Bytes der Anweisungen in der Menge, die durch exklusives Oder verknüpft sein

soll, die niedrigstwertigen Bits der Adresse, und sind die Bits des Verlaufsregisters, das durch exklusives Oder verknüpft werden soll, die Bits des Verlaufsregisters, welche die zuletzt eingeschobenen und mit diesen benachbarten Bits einschließen.

[0034] Die Tabellenadresse kann dann verwendet werden, um ein Teil des oder den gesamten Vorhersageanzeiger **416** abzufragen. In einer Ausführungsform befindet sich der Vorhersageanzeiger bei der Tabellenadresse in einer Tabelle der Vorhersagenanzeiger. In einer weiteren Ausführungsform befindet sich der Vorhersageanzeiger über einem Zeiger bei der Tabellenadresse. Die Vorhersage wird gemäß dem abgefragten Vorhersageanzeiger durchgeführt. In einer Ausführungsform wirkt jeder Vorhersageanzeiger als ein Zähler mit zwei Bits wie der Sättigungszähler mit zwei Bits, der oben beschrieben ist und die Zustandstabelle hat, die in [Fig. 2A](#) veranschaulicht ist. Mit der bedingten Verzweigung, die als genommen vorhergesagt ist, wenn das höchstwertige Bit des Sättigungsanzeigers mit zwei Bits, das der Tabellenadresse entspricht, einen Wert wie eine „1“ **214, 216** hat, und als nicht genommen vorhergesagt ist, wenn das höchstwertige Bit des Sättigungsanzeigers mit zwei Bits, welcher der Tabellenadresse entspricht, den entgegengesetzten Wert hat, wie eine „0“ **210, 212**. Eine einzige Vorhersage, die unter Verwendung dieses Verfahrens abgeleitet wurde, kann einmal durchgeführt werden und für jede bedingte Verzweigung in der Menge verwendet werden.

[0035] Optional kann der Vorhersageanzeiger auf Grundlage davon, ob irgendeine Verzweigung in der Menge tatsächlich genommen wurde, unter Verwendung des Verfahrens wie des Verfahrens aktualisiert werden, das oben unter Verwendung von [Fig. 2B](#) beschrieben wurde. Nun mit Bezugnahme auf die [Fig. 4B](#) und [Fig. 2B](#), wenn irgendeine Verzweigung in der Menge tatsächlich genommen wird, wird der Status des Vorhersageanzeigers unter Verwendung des unteren Teils **228** der Tabelle **213** aktualisiert. Wenn keine Verzweigungen in der Menge tatsächlich genommen werden, wird der Vorhersageanzeiger unter Verwendung des oberen Teils **226** der Tabelle **218** aktualisiert.

[0036] In einer Ausführungsform wird das Verlaufsregister einmal für jede Menge von Anweisungen aktualisiert, nachdem alle Vorhersagen für Verzweigungsanweisungen in der Menge gemacht wurden. In einer Ausführungsform wird das Verlaufsregister, wie in [Fig. 4C](#) gezeigt, aktualisiert. Wenn irgendeine Verzweigung in der Menge tatsächlich genommen wurde **430**, wird ein Wert wie eine „1“ in das Verlaufsregister **432** verschoben. Ansonsten, wenn keine Verzweigungen in der Menge tatsächlich genommen wurden, wird der entgegengesetzte Wert wie eine „0“ in das Verlaufsregister **434** verschoben. Bits können in das Verlaufsregister von irgendeiner Richtung ein-

geschoben werden, solange die Richtung der Verschiebungen mit einer großen Anzahl von Mengen konsistent ist.

[0037] In einer Ausführungsform wird das Verlaufsregister nur aktualisiert, wenn es eine bedingte Verzweigungsanweisung in der Menge **436** gibt. Diese Ausführungsform ermöglicht es einem Verlaufsregister einer bestimmten Größe, einen längeren Verlauf als ein Verlaufsregister nachzuverfolgen, das selbst für Mengen aktualisiert wird, die keine bedingten Verzweigungsanweisungen enthalten, wie oben beschrieben.

[0038] Nun mit Bezugnahme auf [Fig. 5](#) ist eine Ausführungsform einer erfindungsgemäßen Vorrichtung, die verwendet wird, um bedingte Verzweigungsanweisungen vorherzusagen, gezeigt. Der Abfrageprogrammzähler **508** hält die Speicheradresse der ersten Anweisung einer Menge von Anweisungen, die in einer Speichereinrichtung wie einem Speicher gespeichert sind, der in [Fig. 5](#) nicht gezeigt ist, aber dem Speicher **104** aus [Fig. 1](#) ähnlich ist. Der Abfrager **504** adressiert die Speichereinrichtung so über den Adreßbus **506**, um eine bestimmte Anzahl von Anweisungsbytes von solch einer Speichervorrichtung über den Datenbus **509** in den Speicher **510** abzufragen. Ein Ausführungseinheitslader **502** lädt Anweisungen, die in dem Speicher **510** gespeichert sind, in die Ausführungseinheiten **512**. Als nächstes hängt der Anweisungsdecodierer **516** eine Marke in den Markenspeicher **518** an, wenn die Anweisung, die in einem Speicher **510** gespeichert ist und der Anweisung folgt, die in die Ausführungseinheit **512** geladen wird, eine bedingte Verzweigungsanweisung ist. Die Marke, die in dem Markenspeicher **518** gespeichert ist, besteht aus einer Anzahl von Bits, wobei jedes Bit der Bedingung oder den Bedingungen entspricht, die für die bedingte Verzweigungsanweisung erfüllt sein müssen, die in dem Speicher **510** gespeichert ist und der Anweisung in der entsprechenden Ausführungseinheit **512** folgt, die Verzweigung zu nehmen. Eine oder mehrere Ausführungseinheiten **512** enthalten jeweils ein Merkerregister **514**, das die Bedingungsmerkerbits in dem Merkerregister **514** auf Grundlage der Ergebnisse einstellt, die durch die Ausführungseinheit **512** erzeugt werden. Jedes Bit im Merkerregister **514** entspricht Bedingungen wie „Ergebnis=0“, „Ergebnis>0“ oder „Ergebnis<0“. Ein Ergebnisvergleich **520** vergleicht die Merkerbits im Merkerregister **514** mit den Markenbits, die in dem Markenspeicher **518** gespeichert sind. Wenn alle Markenbits des Markenspeichers **518** mit den entsprechend gesetzten Merkerbits im Merkerregister **514** übereinstimmen, gibt der Ergebnisvergleich **520** wahr an das VerlaufsLatch **521** aus, was anzeigt, daß die Bedingung für eine der bedingten Verzweigungen in der Menge, die Verzweigung zuzunehmen, tatsächlich auftrat. Das VerlaufsLatch **521** wird in das Verlaufsregister **522** verschoben, nachdem alle Anwei-

sungen, die bedingten Verzweigungsanweisungen in der Menge vorhergehen, durch die Ausführungseinheiten **512** ausgeführt wurden.

[0039] Das FPC-Latch **511** ist mit dem Abrufprogrammzähler **508** gekoppelt, um den Wert von einigen oder allen der Bits zu bewahren, die in dem Abrufprogrammzähler **508** enthalten sind. Das Verlaufsregister **522** wird mit dem FPC-Latch **511** in der Kombination oder Art wie oben beschrieben über den Verketter **524** verkettet, um den Tabellen-RAM **526** zu adressieren, der eine Tabelle von Sättigungszählern mit zwei Bits enthält wie oben beschrieben. In einer Ausführungsform verkettet der Verketter **524** Bits des Verlaufsregisters **522** und die Mengenadresse, die in den FPC-Latch **511** enthalten ist. In einer weiteren Ausführungsform verknüpft der Verketter **524** die Bits wie oben beschrieben als exklusives Oder. Das höchstwertige Bit der Sättigungszähler mit zwei Bits des Tabellen-RAM **526** wird an die Ausgabeleitung **528** ausgegeben, um die Ausführungseinheit **532** zu verzweigen, um der Verzweigungsausführungseinheit **532** anzuzeigen, ob irgendeine Verzweigung in der Menge, die in dem Speicher **510** geladen ist, wie oben beschrieben, genommen werden soll.

[0040] Die Aktualisierungseinheit **530** ist mit dem VerlaufsLatch **521** und dem Ausgang des Tabellen-RAM **529** gekoppelt, um beide Bits zu empfangen, die in dem Sättigungszähler mit zwei Bits gespeichert sind, der von dem Verketter **524** adressiert wird. Die Aktualisierungseinheit **530** aktualisiert den Sättigungszähler mit zwei Bits im Tabellen-RAM **526** unter Verwendung der Werte, die in [Fig. 2B](#) veranschaulicht sind. Die Aktualisierungseinheit **530** aktualisiert den Tabellen-RAM nach allen Nichtverzweigungsanweisungen, die im Speicher **510** gespeichert sind und den Anweisungen entsprechen, die von dem Abrufprogrammzähler **508** geladen werden und den Bits entsprechen, die in den FPC-Latch **511** gespeichert sind. Dies bedeutet, daß die Aktualisierungseinheit **530** eine einzige Adresse des Tabellen-RAM **526**, die dem Adreßausgang des Tabellen-RAM **526** entspricht, durch den Verketter **524** einmal für jedes Mal aktualisiert, für welches Anweisungen in den Speicher **510** geladen werden. Weitere Anordnungen zum Laden des Speicher **510** sind möglich wie eine Doppelpufferanordnung, wobei Zeiger verwendet werden, um dem Speicher **510** zu entsprechen, wobei ein Zeiger zu dem nächsten Ort im Speicher **510** zeigt, in den Anweisungen geladen werden sollen, und ein Zeiger zu dem nächsten Ort im Speicher **510** zeigt, von dem Anweisungen zu Ausführungseinheiten **512** übermittelt werden sollen. In solch einer Anordnung aktualisiert die Aktualisierungseinheit **530** den Zähler mit zwei Bits im Tabellen-RAM **526**, welcher der Adresse beim Ausgang des Verkettens **524** entspricht, einmal für jede Zeit, zu der alle Anweisungen, die im Speicher **510** zu einer Zeit waren, ausgeführt wurden.

Patentansprüche

1. Verfahren der Verzweigungsvorhersage, das umfaßt:

das Aktualisieren eines Verlaufsregisters mit mehreren Bits, wobei jedes einen ersten Wert und einen zweiten Wert hat, um den bedingte Verzweigungsverlauf von einer von mehreren Mengen von Anweisungen beizubehalten, die mehrere Anweisungen umfassen, wobei zumindest eine Menge von Anweisungen zumindest eine bedingte Verzweigungsanweisung einschließt, wobei jede bedingte Verzweigungsanweisung einen tatsächlich eingenommenen Zustand hat, der aus einem ersten Zustand und einem zweiten Zustand auswählbar ist, wobei der Verfahrensschritt des Aktualisierens des Verlaufsregisters umfaßt:

das Bestimmen des Vorhandenseins von zumindest einer der bedingten Verzweigungsanweisungen in der Menge, die einen tatsächlich eingenommenen Zustand des ersten Zustands hat;

ansprechend auf das Vorhandensein von zumindest einer bedingten Verzweigungsanweisung in der Menge, die einen tatsächlich eingenommenen Zustand des ersten Zustands hat, das Aktualisieren des Verlaufsregisters durch Verschieben eines ersten Werts in das Verlaufsregister;

ansprechend auf die Abwesenheit von zumindest einer bedingten Verzweigungsanweisung in der Menge, die einen tatsächlich eingenommenen Zustand des ersten Zustands hat, das Aktualisieren des Verlaufsregisters durch Verschieben eines zweiten Werts in das Verlaufsregister;

wobei ein solches Aktualisieren des Verlaufsregisters nur einmal für die Menge von Anweisungen bewirkt wird;

das Orten von einem Vorhersageanzeiger, der eine Richtung einer bedingten Verzweigungsanweisung in der Menge von Anweisungen anzeigt, in einer adressierbaren Tabelle mehrerer Vorhersageanzeiger, wobei der Schritt des Ortens des Vorhersageanzeigers außerdem umfaßt:

das Auswählen von zumindest einem Teil eines Identifizierers, der für die Menge einmalig ist;

das Anlegen einer Adresse für die Tabelle von Vorhersageanzeigern unter Verwendung des Teils des einmaligen ausgewählten Identifizierers, wobei der Anlageschritt das Anhängen einer Anzahl von Bits des Verlaufsregisters an das Teil des einmaligen ausgewählten Identifizierers umfaßt; und

das Adressieren der Tabelle der Vorhersageanzeiger unter Verwendung der angelegten Adresse, um daraus ein Vorhersageanzeiger zur Verwendung bei der Verzweigungsvorhersage mit Bezug auf die oder jede der bedingten Verzweigungsanweisungen in der betroffenen Menge von Anweisungen wiederzufinden.

2. Verfahren nach Anspruch 1, wobei der einmalige Identifizierer von der Menge eine Speicheradres-

se umfaßt, die der Menge entspricht.

3. Verfahren nach Anspruch 2, wobei jede Anweisung in der Menge zumindest ein Byte umfaßt, und die Speicheradresse, die der Menge entspricht, eine Speicheradresse eines Bytes von der jeden Anweisung ist.

4. Verfahren nach Anspruch 1, wobei:
 der einmalige Identifizierer eine Anzahl von Bits umfaßt, die ein niedrigstwertiges Bit umfassen und eine Reihenfolge haben;
 das Teil des einmaligen Identifizierers eine Gruppe von acht Bits des einmaligen Identifizierers neben dem und einschließlich des niedrigstwertigen Bits umfaßt;
 die Verlaufsregisterbits eine Reihenfolge haben und ein Bit umfassen, das zuletzt eingeschoben ist; und
 die Anzahl der Bits des Verlaufsregisters vier ist, wobei die Anzahl der Bits des Verlaufsregisters das Bit, das zuletzt eingeschoben ist, und drei Bits neben dem zuletzt eingeschobenen Bit umfaßt.

5. Verzweigungsvorhersagevorrichtung mit:
 einem Verlaufsregister (522), das ein Schieberegister umfaßt und eine Ausgabe hat, die ein zuletzt eingeschobenes Bit des Schieberegisters und zumindest ein zusätzliches Bit des Schieberegisters umfaßt, wobei jedes Bit einen ersten Wert und einen zweiten Wert hat, und betätigbar ist, um aktualisiert zu werden, um den bedingten Verzweigungsverlauf von einem von mehreren Mengen von Anweisungen beizubehalten, die mehrere Anweisungen umfassen, wobei zumindest eine Menge von Anweisungen zumindest eine bedingte Verzweigungsanweisung einschließt, wobei jede bedingte Verzweigungsanweisung einen tatsächlich eingenommenen Zustand hat, der aus einem ersten Zustand und einem zweiten Zustand auswählbar ist;
 einem Mittel (516, 518, 514, 520), um das Vorhandensein von zumindest einer der bedingten Verzweigungsanweisungen in der Menge zu bestimmen, die einen tatsächlich eingenommenen Zustand des ersten Zustands hat;
 einem Mittel (520, 521), das ansprechend auf das Vorhandensein von zumindest einer bedingten Verzweigungsanweisung in der Menge, die einen tatsächlich eingenommenen Zustand des ersten Zustands hat, um das Verlaufsregister durch Verschieben eines ersten Werts in das Verlaufsregister (522) zu aktualisieren, und ansprechend auf die Abwesenheit von zumindest einer bedingten Verzweigungsanweisung in der Menge, die einen tatsächlich eingenommenen Zustand des ersten Zustands hat, betätigbar ist, um das Verlaufsregister durch Verschieben eines zweiten Werts in das Verlaufsregister (522) zu aktualisieren;
 wobei das Mittel (520, 521) betätigbar ist, um das Verschieben in das Verlaufsregister (522) nur einmal für die Menge von Anweisungen zu bewirken,

wobei die Vorrichtung außerdem umfaßt:
 eine adressierbare Speichereinrichtung (526), die eine Dateneingabe, eine Adresseneingabe und eine Datenausgabe (528) hat, die mit der Vorrichtungsausgabe (532) gekoppelt ist, wobei die adressierbare Speichereinrichtung eine adressierbare Tabelle von mehreren Vorhersageanzeigern und zur Verwendung beim Orten der adressierbaren Tabelle einen Vorhersageanzeiger enthält, der eine Richtung einer bedingten Verzweigungsanweisung in der Menge von Anweisungen anzeigt,
 ein AbrufprogrammzählerLatch (511), das eine Ausgabe hat, wobei das AbrufprogrammzählerLatch zum Speichern von zumindest einem Teil eines einmaligen Identifizierers für die Menge von Anweisungen ist,
 einen Verketter (524), der eine erste Eingabe, die mit dem AbrufprogrammzählerLatch (511) gekoppelt ist, eine zweite Eingabe, die mit dem Verlaufsregister (522) gekoppelt ist, und eine Ausgabe hat, die mit der Adresseneingabe des adressierbaren Speichers (526) gekoppelt ist, um eine Adresse zum Adressieren der Tabelle von Vorhersageanzeigern vorzusehen, um daraus einen Vorhersageanzeiger zur Verwendung bei der Verzweigungsvorhersage mit Bezug auf die oder jede bedingte Verzweigungsanweisung in der Menge von Anweisungen wiederzufinden.

6. Vorrichtung nach Anspruch 5, wobei die Ausgabe des Verketters (524) eine Anzahl von Bits des AbrufprogrammzählerLatches (511), das zuletzt in die Ausgabe des Verlaufsregisters (522) eingeschobene Bit und zumindest eines der zusätzlichen Bits der Ausgabe des Verlaufsregisters umfaßt.

7. Vorrichtung nach Anspruch 6, wobei die Anzahl der Bits des AbrufprogrammzählerLatches (511) acht ist, und die Ausgabe des Verketters drei zusätzliche Bits der Ausgabe des Verlaufsregisters (522) umfaßt.

8. Vorrichtung nach Anspruch 5, wobei die Ausgabe des Verketters die erste Eingabe des Verketters als exklusives ODER mit der zweiten Eingabe des Verketters verknüpft umfaßt.

9. Vorrichtung nach Anspruch 5, wobei jede Anweisung in der Menge eine Speicheradresse und den einmaligen Identifizierer für die Menge von Anweisungen hat, der eine Speicheradresse der zumindest einen der Anweisungen in der Menge umfaßt, wobei die Anweisung dem Speicheridentifizierer für die Menge entspricht, der eine nicht bedingte Verzweigungsanweisung ist.

Es folgen 7 Blatt Zeichnungen

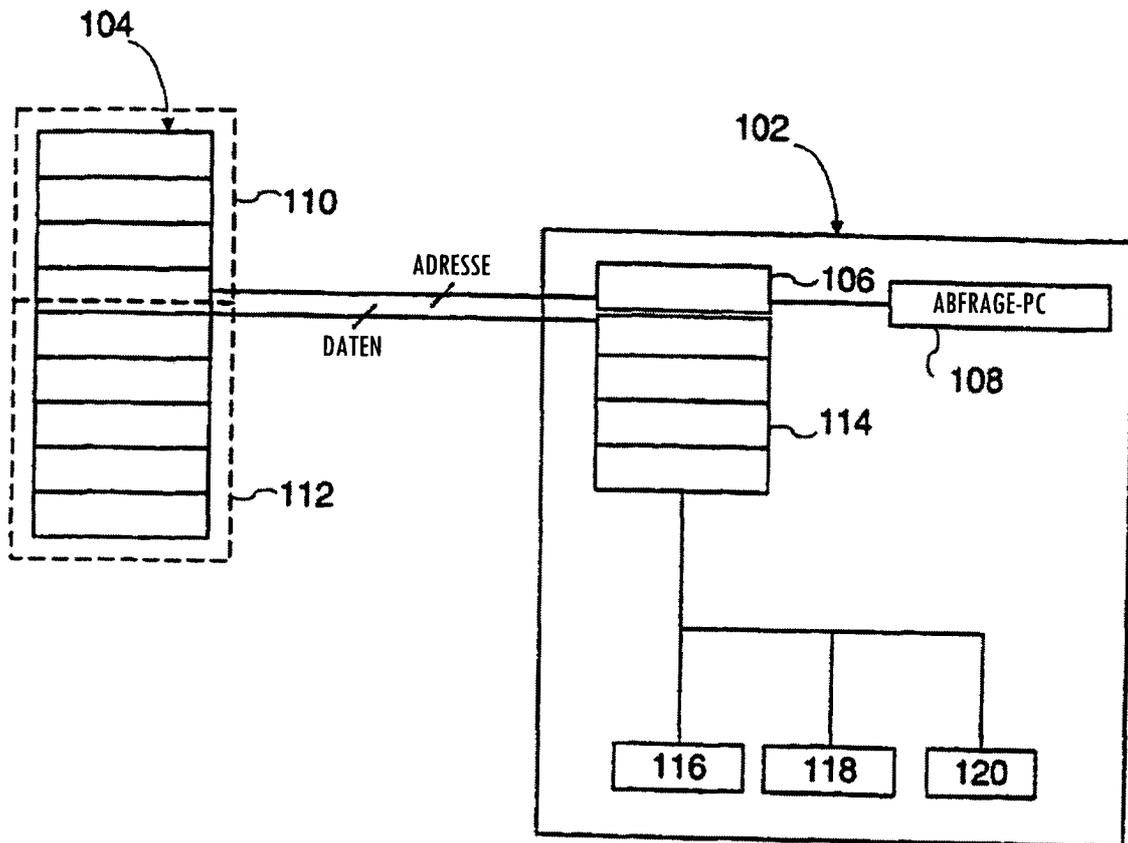


FIG. 1
(STAND DER TECHNIK)

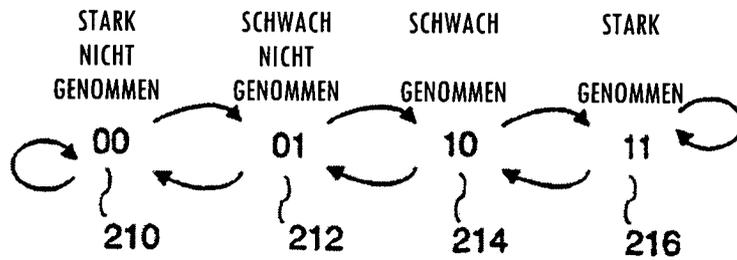


FIG. 2A
(STAND DER TECHNIK)

218

00	00	NICHT GENOMMEN
01	00	NICHT GENOMMEN
10	01	NICHT GENOMMEN
11	10	NICHT GENOMMEN
00	01	GENOMMEN
01	10	GENOMMEN
10	11	GENOMMEN
11	11	GENOMMEN

220 222 224

226

228

FIG. 2B
(STAND DER TECHNIK)

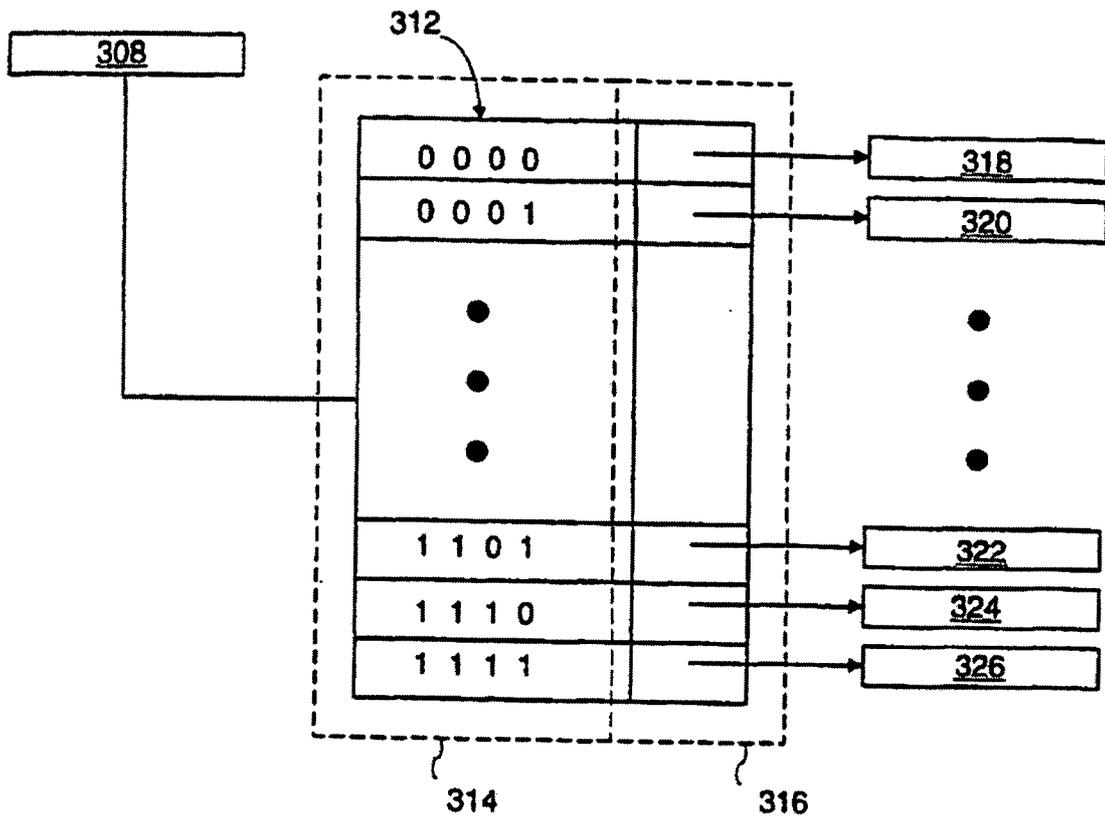


FIG. 3
(STAND DER TECHNIK)

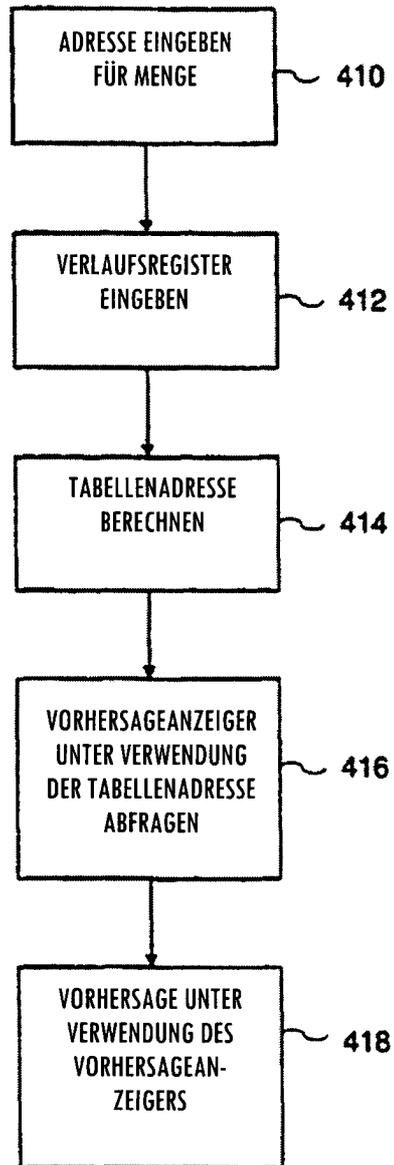


FIG. 4A

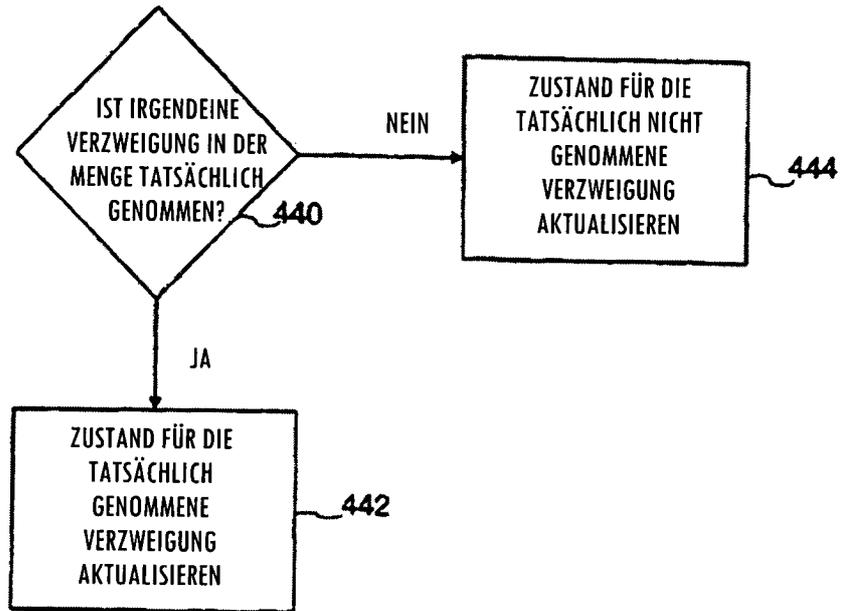


FIG. 4B

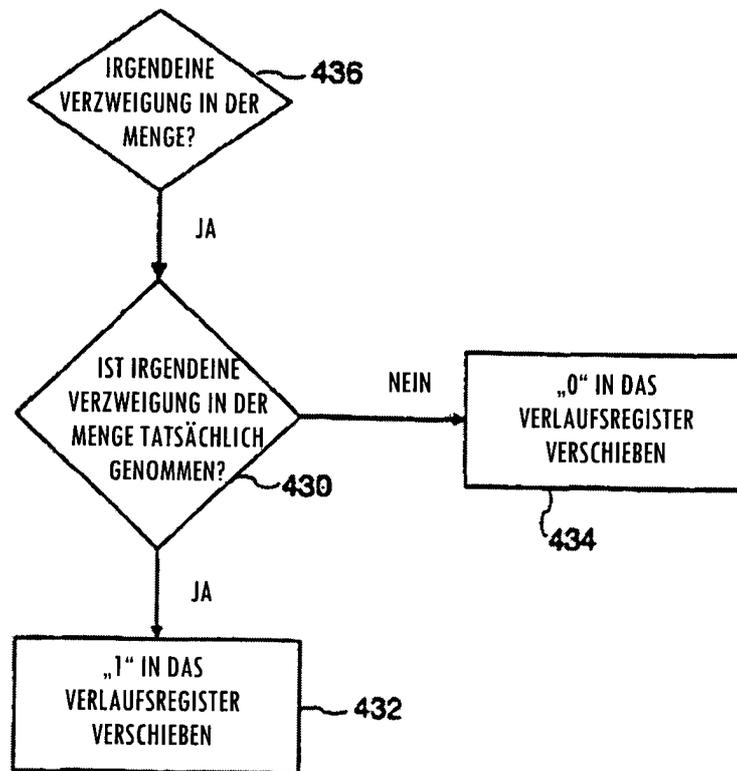


FIG. 4C

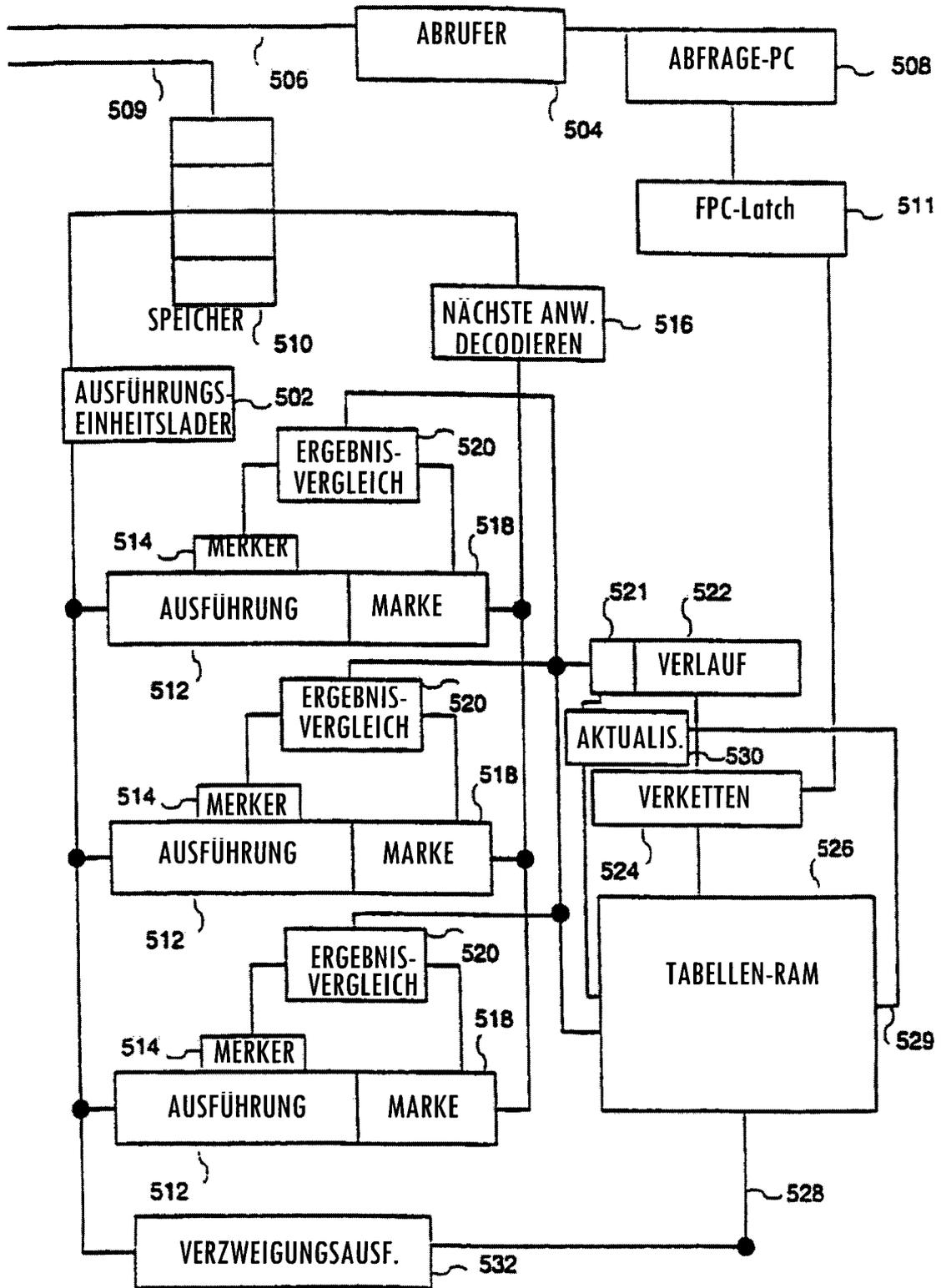


FIG. 5