



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 697 25 278 T2** 2004.08.26

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 0 855 058 B1**

(21) Deutsches Aktenzeichen: **697 25 278.7**

(86) PCT-Aktenzeichen: **PCT/US97/13307**

(96) Europäisches Aktenzeichen: **97 935 174.9**

(87) PCT-Veröffentlichungs-Nr.: **WO 98/006032**

(86) PCT-Anmeldetag: **25.07.1997**

(87) Veröffentlichungstag
der PCT-Anmeldung: **12.02.1998**

(97) Erstveröffentlichung durch das EPA: **29.07.1998**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **01.10.2003**

(47) Veröffentlichungstag im Patentblatt: **26.08.2004**

(51) Int Cl.7: **G06F 9/30**
G06F 9/38

(30) Unionspriorität:
691785 02.08.1996 US

(73) Patentinhaber:
Freeman, Jackie A., San Jose, Calif., US

(74) Vertreter:
Betten & Resch, 80333 München

(84) Benannte Vertragsstaaten:
**AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LI,
LU, MC, NL, PT, SE**

(72) Erfinder:
gleich Anmelder

(54) Bezeichnung: **PROGRAMMIERBARES SYSTEM UND VERFAHREN ZUM VORHERSAGEN VON VERZWEIGUN-
GEN FÜR COMPUTERSYSTEME**

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

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

Beschreibung

Hintergrund der Erfindung

[0001] Diese Erfindung bezieht sich im Allgemeinen auf Computer-Systeme und insbesondere auf ein programmierbares Verzweigungsvorhersagesystem und auf ein Verzweigungsvorhersageverfahren für Computer-Systeme.

[0002] In Hochleistungsprozessoren ist es üblich, einen Befehl in mehrere Schritte zu zerlegen, wie z. B. einen Holschritt, einen Decodierungsschritt und einen Ausführungsschritt, und jeden Schritt durch eine andere Befehlsverarbeitungs-Untereinheit auszuführen. Diese Befehlsverarbeitungseinheiten können asynchron arbeiten, wobei sie nicht den gleichen Befehl verarbeiten müssen. Falls jede Befehlsverarbeitungseinheit normalerweise nicht den gleichen Befehl verarbeitet, vergrößert dies die Geschwindigkeit des Prozessors. Folglich ist es eine häufige Praxis, aufeinander folgende Befehle um einen Taktzyklus zu überlappen, sodass, wenn eine Holeinheit die Verarbeitung eines zweiten Befehls beginnt, eine Decodierungseinheit den ersten Befehl verarbeiten kann. Im nächsten Taktzyklus kann die Holeinheit einen dritten Befehl verarbeiten, während die Decodierungseinheit den zweiten Befehl verarbeitet, wobei die Ausführungseinheit den ersten Befehl verarbeiten kann. In einem normalen System arbeitet jede Befehlsverarbeitungseinheit synchron, sodass jeder Befehl für die Ausführung drei Taktzyklen erfordert. Folglich sind zwölf Taktzyklen erforderlich, um in einem nicht überlappenden System vier Befehle auszuführen. Mit der Befehlsüberlappung können im Gegensatz diese vier Befehle in nur sechs Taktzyklen ausgeführt werden. Diese Überlappung vergrößert die Verarbeitungsgeschwindigkeit des Prozessors signifikant. Ähnlich können die Befehle, die ausgeführt werden, in einer derartigen Weise überlappt werden, sodass der Decodierungsschritt für zwei oder mehr Befehle gleichzeitig ausgeführt werden kann. In diesem überlappenden System können mehrere Befehle gleichzeitig ausgeführt werden.

[0003] Diese Befehlsüberlappung kann jedoch nicht verfügbar oder ineffizient sein, hauptsächlich infolge des häufigen Auftretens verschiedener Typen von Verzweigungsbefehlen in den meisten Programmen. Ein Verzweigungsbefehl kann jeden Nutzen der Befehlsüberlappung vollständig beseitigen, insbesondere wenn die Verzweigung auftritt und ein neuer Code in die Pipeline geladen und dann ausgeführt werden muss.

[0004] Die Verzweigungsbefehle besitzen das signifikante Merkmal, dass sie den Wirkungsgrad der Befehlsüberlappung verringern oder beseitigen können. Die Verzweigung kann genommen oder nicht genommen werden, wobei dies eine zeitliche Unbestimmtheit einführt, was den nächsten Befehl angeht, wobei sie jede Befehlsüberlappung verhindert, weil der nächste Befehl nicht bekannt ist.

[0005] Die durch die Verzweigungsbefehle erzeugten Probleme für Befehlspipelines können verringert werden, indem ein Verzweigungsbefehls-Vorhersagesystem geschaffen wird, das vor der tatsächlichen Ausführung der Verzweigung vorhersagt, ob die Verzweigung genommen wird oder nicht, und die nächsten Befehle, Adressen oder eine andere Referenz auf das Ziel der Verzweigung, die ausgeführt werden, falls die Verzweigung genommen wird, und die nächsten Befehle oder die nächste Referenz auf das Ziel der Verzweigung, wie z. B. die Adresse, die ausgeführt werden, falls die Verzweigung nicht genommen wird, vorhersagt. Eine erfolgreiche Verzweigungsvorhersage erlaubt dem Prozessor, ohne die Verzögerung in der Verarbeitungszeit zu arbeiten, die durch eine Verzweigung verursacht wird. Es kann jedoch eine große Zeitstrafe geben, falls die Vorhersage falsch ist; wobei die Strafe für die falsche Vorhersage größer als die Verzögerung sein kann, die auf die Unbestimmtheit der Verzweigung zurückzuführen ist. Deshalb ist eine hohe Vorhersagegenauigkeit in einem Verzweigungsbefehls-Vorhersagesystem erwünscht.

[0006] Die meisten herkömmlichen Verzweigungsbefehls-Vorhersagesysteme sind automatische Verzweigungsbefehls-Vorhersageeinrichtungen, die das Ergebnis der Verzweigung vorhersagen, indem sie entweder auf die vorhergesagte Befehlssequenz, auf das letzte Verzweigungsbefehlsverhalten oder auf das Speichern der Programmoperanden oder -befehle reagieren.

[0007] Herkömmliche automatische Verzweigungsbefehls-Vorhersagesysteme können mit einer herkömmlichen Speicherhierarchie integriert sein oder sie können separat von der Speicherhierarchie aufrechterhalten werden. Ein herkömmliches System integriert z. B. ein automatisches Verzweigungsbefehls-Vorhersagesystem in einen Befehls-Cache. Diese Integration erlaubt dem System, sowohl als ein Befehls-Cache als auch als eine Verzweigungsvorhersageeinrichtung zu dienen, was vorteilhaft ist, weil beide Systeme auf die gleichen Befehle wirken müssen, wobei sie den Speicher oder die anderen Betriebsmittel gemeinsam verwenden können. Andere herkömmliche Systeme verbesserten das grundlegende Integrationskonzept, indem sie Systeme besitzen, die automatisch das ausgeführte Programm in Spuren neu organisieren, die die Befehle von nicht sequentiellen Adressbereichen enthalten, und diese Spuren zusammen mit den Vorhersage- und Wiederherstellungsinformationen im Cache speichern.

[0008] Um Vorhersagen des Verzweigungsverhaltens mit höherer Genauigkeit zu erhalten, die sein können, ob eine Verzweigung genommen wird (das vorhergesagte Ergebnis) und/oder die Adresse oder die Referenz auf das Ziel der Verzweigung, zu der die Verzweigung nach einer erfolgreichen Vorhersage des Ergebnisses (des vorhergesagten Resultats) geht, gibt es eine Anzahl von herkömmlichen Verzweigungsvorhersagesyste-

men, die versuchen, das Ergebnis und das Resultat einer Verzweigung genau vorherzusagen. Diese herkömmlichen Systeme besitzen sich verändernde Grade der Genauigkeit, wie im Folgenden beschrieben ist. Einige sagen 60% der Verzweigungsergebnisse und -resultate genau vorher, während die genauesten System eine Genauigkeit von 90% besitzen können. Diese herkömmlichen Verzweigungsvorhersagesysteme können in mehrere verschiedene Kategorien gruppiert werden, die statische Verzweigungsvorhersageeinrichtungen, dynamische Verzweigungsvorhersageeinrichtungen, implizit programmierbare Verzweigungsvorhersageeinrichtungen und explizit programmierbare Verzweigungsvorhersageeinrichtungen enthalten. Jede dieser Kategorien besitzt verschiedene Vorteile und Nachteile und verschiedene Vorhersagegenauigkeiten.

[0009] In einem statischen Verzweigungsbefehls-Vorhersagesystem werden die Verzweigungen auf der Grundlage statischer sich nicht ändernder Informationen vorhergesagt. Ein herkömmliches statisches automatisches Verzweigungsbefehls-Vorhersagesystem identifiziert z. B. die Verzweigungen in einer vorhergesagten Befehlssequenz und sagt dann immer vorher, dass jede Verzweigung genommen wird. Dieses Verzweigungsvorhersagesystem besitzt eine Genauigkeit von etwa 60% (d. h., es schätzt etwa 60% der Zeit richtig). Ein genaueres statisches Verzweigungsvorhersagesystem sagt vorher, dass eine Verzweigung genommen wird, falls die Zieladresse der Verzweigung eine numerisch niedrigere Speicherstelle als der Verzweigungsbefehl selbst ist. Ein weiteres herkömmliches statisches Verzweigungsvorhersagesystem verwendet die Vorhersageinformationen, die im Verzweigungsbefehl selbst codiert sind, zu dem Zeitpunkt, zu dem der Verzweigungsbefehl kompiliert wird. Alle diese Systeme sagen vorher, dass sich eine gegebene Verzweigung entsprechend den statischen Vorhersageinformationen verhalten wird. Wenn jedoch sich das Verzweigungsverhalten zu irgendeinem Zeitpunkt ändert, nachdem die statischen Verzweigungsvorhersageinformationen erzeugt worden sind, dann kann das statische Verzweigungsvorhersagesystem das Verzweigungsverhalten falsch vorhersagen. Wenn z. B. das Verhalten eines Verzweigungsbefehls von einer anderen Variable abhängig ist, die zum Zeitpunkt der Kompilierung unbestimmt ist, dann kann sich das Verhalten der Verzweigung ändern, nachdem die statischen Vorhersageinformationen erzeugt worden sind. Die Vorhersageentscheidung für jede gegebene Verzweigung ist fest, deshalb ist die Genauigkeit dieser statischen Verzweigungsvorhersagesysteme eingeschränkt. Um die Genauigkeit zu vergrößern, können dynamische Verzweigungsvorhersagesysteme verwendet werden.

[0010] In einem dynamischen Verzweigungsvorhersagesystem wird das Ergebnis einer Verzweigung auf der Grundlage von dynamischen Informationen, wie z. B. dem früheren Verzweigungsverhalten, vorhergesagt, die sich während der Ausführung des Programms ändern können oder modifiziert werden können. Ein herkömmliches dynamisches Verzweigungsvorhersagesystem verwendet z. B. einen Sättigungszähler, der vor den Entscheidungen über eine genomme/nicht genomme Verzweigung aktualisiert wird, um vorherzusagen, dass eine Verzweigung genommen wird, falls frühere Entscheidungen über die Verzweigungen anzeigen, dass die Verzweigung vor kurzem häufiger genommen als nicht genommen worden ist. Ansonsten sagt das dynamische System vorher, dass die Verzweigung nicht genommen wird. Nachdem die Verzweigung ausgeführt worden ist, können die Verzweigungsvorhersageinformationen aktualisiert werden.

[0011] Es ist eine Vielzahl dynamischer Verzweigungsvorhersagesysteme vorgeschlagen worden, die das frühere Verzweigungsbefehlsverhalten verwenden, um das künftige Verhalten des Verzweigungsbefehls vorherzusagen. Diese dynamischen Verzweigungsvorhersagesysteme besitzen jedoch trotzdem eine eingeschränkte Genauigkeit, weil diese Systeme frühere Programmdate und das frühere Verzweigungsverhalten verwenden, um das künftige Verzweigungsverhalten vorherzusagen. Das frühere Programmverhalten kann jedoch das künftige Verzweigungsverhalten nicht genau vorhersagen.

[0012] Ein weiteres herkömmliches dynamisches Verzweigungsvorhersagesystem verwendet spezielle Verzweigungsbefehle für bestimmte Verzweigungen mit vorhersagbarem Verhalten, wie z. B. Prozedurrücksprünge und Schleifen. Diese speziellen Verzweigungsbefehle besitzen eine vereinbarte Verwendung der Operanden, wobei diese Verwendung in genaue Vorhersagen dieser bestimmten Verzweigungen umgesetzt werden kann. Diese Systeme sagen jedoch andere Verzweigungen nicht genau vorher. Einige dieser obenbeschriebenen dynamischen Systeme sagen das Ergebnis der Verzweigung vorher (d. h., ob die Verzweigung genommen wird oder nicht), aber sie sagen nicht die Zieladresse der Verzweigung voraus, falls die Verzweigung genommen wird. Die Genauigkeit und die Wirksamkeit einer Verzweigungsvorhersageeinrichtung kann vergrößert werden, indem sowohl das Ergebnis einer Verzweigung als auch das Resultat einer Verzweigung genauer vorhergesagt werden.

[0013] Wenn eine Vorhersage, ob eine Verzweigung genommen werden wird, ausgeführt wird, ist der nächste Schritt, zu bestimmen, was die Referenz zum Ziel der Verzweigung sein wird, wie z. B. die Adresse. Wenn die Verzweigung nicht genommen wird, dann wird das nächste Ziel der Verzweigung die nächste Adresse nach der Verzweigung sein, sodass das Ziel der Verzweigung leicht vorhergesagt werden kann. Falls vorhergesagt wird, dass die Verzweigung genommen wird, kann es jedoch schwieriger sein, zu bestimmen, was die nächste Adresse sein wird. Ein herkömmliches Zieladressen-Vorhersagesystem sagt das Ziel der prozeduralen Rücksprünge unter Verwendung eines Stapelmechanismus voraus. In diesem System kann, wenn z. B. das Programm zu einem Unterprogramm verzweigt, die Adresse des letzten vor dem Unterprogramm ausgeführten

Befehls im Stapel gespeichert werden. Wenn das Unterprogramm abgeschlossen ist, kann die im Stapel gespeicherte Adresse verwendet werden, um die nächste Adresse zu bestimmen. Dieser Stapel kann jedoch nicht computerbestimmte Ziele vorhersagen, wie z. B. eine Aufrufadresse für virtuelle C++-Funktionen, weil die Aufrufadresse der virtuellen Funktion nicht statisch fest ist oder durch eine vorhergehende Verzweigung erzeugt wird. Folglich arbeiten diese Systeme nicht für alle Typen von Verzweigungen effektiv.

[0014] Ein weiteres herkömmliches Zieladressen-Vorhersagesystem verwendet eine Verzweigungshistorientabelle, die die letzten Zieladressen für verschiedene Verzweigungen enthält. Abermals kann dieses System prozedurale Rücksprünge vorhersagen, es kann aber nicht computererzeugte Ziele vorhersagen, weil die letzten Ziele keine genauen künftigen Ziele sein können, weil sich das Ziel einer Verzweigung ändern kann. Ein drittes Verzweigungsadressen-Vorhersagesystem verwendet dedizierte Rücksprungsadressenregister, um Verzweigungszieladressen durch das Beobachten der gespeicherten Werte in den dedizierten Registern vorherzusagen. Diese Systeme mit dedizierten Registern erfordern spezielle Hardware und Vorkehrungen für diese Register im Befehlsvorrat, die nicht immer verfügbar sind, deshalb besitzt dieses System einen eingeschränkten Nutzen. Ein viertes Verzweigungsadressen-Vorhersagesystem überwacht alle gespeicherten Operanden, um künftige Verzweigungsadressen zu bestimmen, dieses System ist jedoch nur so gut wie die Informationen, die verwendet werden, die nur das frühere Programmverhalten enthalten, und wie die eingeschränkte Menge der Analyse, die zur Laufzeit ausgeführt werden kann. Diese Verzweigungsadressen-Vorhersagesysteme besitzen etwa die gleiche Genauigkeit wie einige der Verzweigungsergebnis-Vorhersageeinrichtungen. Um die Genauigkeit jedes dieser obigen Systeme weiter zu vergrößern, kann ein programmierbares Verzweigungsvorhersagesystem verwendet werden.

[0015] Ein programmierbares Verzweigungsvorhersagesystem schafft irgendeine Form der programmierbaren Steuerung über den Betrieb von jedem der obenbeschriebenen automatischen Verzweigungsvorhersagesysteme. In diesen programmierbaren Systemen führt das automatische Verzweigungsvorhersagesystem die tatsächlichen Verzweigungsvorhersagen aus, wobei das programmierbare System nur die Vorhersageeinrichtung beeinflusst, um die Vorhersageinformationen einzustellen, die falsch sein können, oder um die aus der Untersuchung des Programmzustandes abgeleiteten Informationen aufzunehmen. Diese programmierbaren Systeme sind im Ergebnis transparent für alle automatischen Verzweigungsvorhersagesysteme und kompatibel mit allen automatischen Verzweigungsvorhersagesystemen, weil diese Systeme lediglich die Steuerung über das Vorhersagesystem ausüben. Diese programmierbaren Verzweigungsvorhersagesysteme können die implizite Steuerung oder die explizite Steuerung verwenden, wie im Folgenden beschrieben ist.

[0016] Die implizit programmierbaren Verzweigungsvorhersagesysteme beeinflussen die Verzweigungsvorhersageinformationen nicht direkt. Ein programmierbares System, das z. B. die implizite Steuerung verwendet, kann zwischen abwechselnden Codefragmenten wählen, die die gleiche Funktion in verschiedenen Arten implementieren. Das implizite programmierbare System kann das Codefragment wählen, das durch das Verzweigungsvorhersagesystem in einem speziellen Ausführungskontext am wahrscheinlichsten genau vorhergesagt wird. Diese impliziten Systeme erfordern jedoch mehr Speicherraum, weil mehrere Codefragmente gespeichert werden müssen, wobei sie für komplexere Verzweigungen nicht nützlich sein können. Weil außerdem einige Programme aus einem Festwertspeicher (ROM) ausgeführt werden, kann dieses System für diese Programme nicht verwendet werden, weil alternative Codefragmente nicht in einen Festwertspeicher geschrieben werden können. Es gibt außerdem implizite programmierbare Systeme, die Vorhersageinformationen verwenden, die innerhalb eines Verzweigungsbefehls codiert sind. Diese Systeme leiden jedoch, wenn es notwendig wird, die innerhalb des Befehls codierten Informationen zu ändern. Alle diese Systeme sind implizit, weil der Programmierer das Verzweigungsvorhersagesystem nicht tatsächlich direkt steuert, sondern die Verzweigungen steuert, die dem Verzweigungsvorhersagesystem gegeben werden. Die explizite programmierbare Verzweigungsvorhersage kann genau sein und weniger Speicherraum erfordern.

[0017] Die explizit programmierbaren Systeme steuern die Verzweigungsvorhersagesysteme unter Verwendung von Vorhersageoperationen, Befehlen oder Befehlssequenzen direkt. Diese Befehle werden zum Programmcode hinzugefügt. Diese expliziten Systeme sind flexibler, weil die zusätzlichen Befehle zu jedem Befehlsvorrat hinzugefügt werden können und die `op_codes` der Vorhersage so gewählt werden können, dass die älteren Prozessorkonstruktionen sie als `no_op`-Befehle behandeln, sodass diese älteren Systeme trotzdem den Programmcode, selbst mit den zusätzlichen Verzweigungsvorhersageoperationen, verwenden können. Die im Stand der Technik bekannten expliziten Systeme, wie z. B. US-Patent Nr. 5.454.117, steuern, ob das zugrunde liegende Verzweigungsvorhersagesystem zu einem speziellen Zeitpunkt verwendet wird, ob die Vorhersageinformationen aktualisiert werden und die Masseninitialisierung der Vorhersageinformationen. Diese expliziten programmierbaren Systeme des Standes der Technik beeinflussen jedoch alle durch eine Verzweigungsvorhersageeinrichtung ausgeführten Vorhersagen, wenn nicht die Verzweigungsvorhersageeinrichtung aktiviert und wiederum deaktiviert wird.

[0018] Keines dieser obenbeschriebenen Systeme schafft ein programmierbares Verzweigungsvorhersagesystem, das die Verzweigungsergebnisse und die Zieladressen für jede Verzweigung genau und zuverlässig vorhersagen kann. Außerdem schafft keines dieser herkömmlichen Systeme eine explizite Steuerung über die

Vorhersagen jeder einzelnen Verzweigung.

[0019] Es gibt einen Bedarf an einem programmierbaren Verzweigungsvorhersagesystem und einem Verzweigungsvorhersageverfahren, die diese und andere Probleme der bekannten Vorrichtungen vermeiden, wobei auf dieses Ziel die vorliegende Erfindung gerichtet ist.

Zusammenfassung der Erfindung

[0020] Gemäß einem Aspekt der vorliegenden Erfindung wird ein System und ein Verfahren geschaffen, wie sie in den unabhängigen Ansprüchen 1 und 14 definiert sind.

[0021] Die Erfindung schafft ein programmierbares Verzweigungsvorhersagesystem, das flexibler als einige herkömmliche Systeme ist, weil die Erfindung die Speicherung der zugrunde liegenden Vorhersageinformationen innerhalb des Verzweigungsvorhersagesystems steuern kann. Außerdem schafft die Erfindung ein programmierbares Verzweigungsvorhersagesystem, das den Programmcode nicht modifiziert oder zwischen Codefragmenten wählt, sodass die Nachteile dieser herkömmlichen Systeme nicht vorhanden sind. Die Erfindung schafft außerdem ein programmierbares Verzweigungsvorhersagesystem, das die explizite Steuerung der Vorhersagedaten für eine ausgewählte Verzweigung besitzen kann, bevor die Verzweigung tatsächlich ausgeführt wird, und ohne die Vorhersagen für andere Verzweigungen zu beeinflussen.

[0022] Die explizite Manipulation der Vorhersageinformationen der zugrunde liegenden Verzweigungsvorhersageinformationen schafft Vorhersagegenauigkeiten, die wenigstens so hoch wie, wenn nicht höher als, jedes herkömmliche Verzweigungsvorhersagesystem sind. Weil außerdem die zugrunde liegenden Verzweigungsvorhersageinformationen gesteuert werden, kann ein programmierbares Verzweigungsvorhersagesystem gemäß der Erfindung mit jedem Typ von Verzweigungsbefehl verwendet werden und kann zu jedem Befehlsvorrat hinzugefügt werden. Das programmierbare Verzweigungsvorhersagesystem kann außerdem jeden Typ des automatischen Verzweigungsvorhersagesystems enthalten.

[0023] Die Erfindung kann außerdem ein programmierbares Verzweigungsvorhersagesystem schaffen, das die Vorhersageinformationen sichert und speichert, sodass verschiedene Typen von Verzweigungen vorhergesagt werden können. Für Prozessschalter, in denen die Ausführung eines Stücks des Programmcodes durch ein weiteres anderes Stück des Programmcodes unterbrochen wird, können z. B. die Vorhersageinformationen für den alten Prozess gespeichert werden. Dann können, falls das Programm jemals zu diesem Prozess zurückkehrt, die Vorhersageinformationen für diesen Prozess zurück in das Verzweigungsvorhersagesystem geladen werden. Ähnlich können die Vorhersagedaten für eine Verzweigung später unabhängig von den Prozessschaltern gesichert und wiederhergestellt werden. Folglich wird die Vorhersagegenauigkeit der Erfindung für Prozessschalter und Programme, die in Phasen arbeiten, dramatisch vergrößert.

Kurzbeschreibung der Zeichnung

[0024] **Fig. 1** ist ein Blockschaltplan einer automatischen Verzweigungsvorhersage, die ein programmierbares Verzweigungsvorhersagesystem gemäß der Erfindung enthalten kann;

[0025] **Fig. 2** ist ein Blockschaltplan einer ersten Ausführungsform des programmierbaren Verzweigungsvorhersagesystems gemäß der Erfindung;

[0026] **Fig. 3** ist eine schematische Darstellung der Vorhersagedaten, die im Vorhersageinformationen-Speichersystem nach **Fig. 2** gespeichert sein können;

[0027] **Fig. 4** ist ein Blockschaltplan einer zweiten Ausführungsform eines programmierbaren Verzweigungsvorhersagesystems gemäß der Erfindung;

[0028] **Fig. 5** ist ein Blockschaltplan einer dritten Ausführungsform des programmierbaren Verzweigungsvorhersagesystems gemäß der Erfindung;

[0029] **Fig. 6** ist eine graphische Darstellung eines Beispiels eines Vorhersagedaten-Speichersystems, das einen 2-Bit-Zähler verwenden kann;

[0030] **Fig. 7** ist eine Tabelle, die die Manipulation des 2-Bit-Zählers nach **Fig. 6** durch ein herkömmliches Verzweigungsvorhersagesystem zeigt;

[0031] **Fig. 8** ist eine Tabelle, die ein Beispiel der Manipulation des 2-Bit-Zählers nach **Fig. 6** durch ein programmierbares Verzweigungsvorhersagesystem gemäß der Erfindung zeigt;

[0032] **Fig. 9** ist ein Ablaufplan eines ersten Beispiels eines Verfahrens gemäß der Erfindung zum Einfügen von Vorhersageoperationen in einen Computer-Code;

[0033] **Fig. 10** ist ein Ablaufplan eines zweiten Beispiels eines Verfahrens gemäß der Erfindung zum Einfügen von Vorhersageoperationen in einen Computer-Code; und

[0034] **Fig. 11** ist eine Tabelle von Beispiel-Vorhersagedatenwerten für ein Fragment des Programmcodes, in dem die Verzweigungen durch das programmierbare Verzweigungsvorhersagesystem vorhergesagt werden.

[0035] Die Erfindung ist auf ein programmierbares Verzweigungsvorhersagesystem besonders anwendbar. In diesem Kontext wird die Endung beschrieben. Es ist jedoch klar, dass das System und das Verfahren gemäß der Erfindung einen größeren Nutzen besitzen.

[0036] **Fig. 1** ist ein Blockschaltplan eines Computer-Systems **20** mit einem Verzweigungsvorhersagesystem. Das Computer-System kann eine Verzweigungsvorhersageeinrichtung **22**, einen Prozessor **24** und einen Speicher **26** enthalten. Der Prozessor steuert wie in den meisten Computer-Systemen den Betrieb des Computers durch das Ausführen von Befehlen, die im Speicher **26** gespeichert sein können. Die Erfindung ist nicht auf irgendeinen speziellen Prozessortyp eingeschränkt, wobei sie sowohl Prozessoren mit komplexem Befehlsvorrat (CISC-Prozessoren) als auch Prozessoren mit eingeschränktem Befehlsvorrat (RISC-Prozessoren) enthalten kann.

[0037] Innerhalb des Prozessors kann es eine oder mehrere Befehlspipelines geben, sodass mehrere Befehle gleichzeitig verarbeitet werden, wie in der Technik bekannt ist. Eine Gruppe von Befehlen kann typischerweise aus dem Speicher abgerufen werden und in eine oder mehrere Befehlspipelines gelegt werden. Wenn die Gruppe der Befehle in die eine oder die mehreren Befehlspipelines innerhalb des Prozessors geladen wird, können die Verzweigungen durch bekannte Systeme erfasst und markiert werden. Das Ergebnis der markierten Verzweigungen kann durch die Verzweigungsvorhersageeinrichtung vorhergesagt werden. Es kann z. B. auf der Grundlage einiger Verzweigungsvorhersagedaten bestimmt werden, ob die Verzweigung genommen wird, wie im Folgenden beschrieben ist. Wenn vorhergesagt wird, dass die Verzweigung genommen wird, dann kann die Verzweigungsvorhersageeinrichtung außerdem das Ziel der Verzweigung vorhersagen; das Ziel kann als eine Adresse dargestellt werden, zu der die Verzweigung gehen wird. Die Verzweigungsvorhersageeinrichtung kann außerdem gleichzeitig vorhersagen, ob die Verzweigung genommen wird und was das Ziel der Verzweigung ist, wie z. B. die Adresse. Wie oben beschrieben worden ist, kann die Vorhersage durch eine Vielzahl verschiedener automatischer Verzweigungsvorhersageeinrichtungen ausgeführt werden, einschließlich statischer Verzweigungsvorhersageeinrichtungen und dynamischer Verzweigungsvorhersageeinrichtungen. Wenn vorhergesagt wird, dass die Verzweigung nicht genommen wird, dann führt der Prozessor nichts spezielles aus, weil der Befehl direkt nach der Verzweigung ausgeführt werden sollte. Wenn jedoch vorhergesagt wird, dass die Verzweigung genommen wird, dann werden der Prozessor oder die automatische Verzweigungsvorhersageeinrichtung auf der Grundlage der Verzweigungsvorhersage die geeigneten Schritte unternehmen, so dass die in die Befehlspipeline geladenen nächsten Befehle die vorhergesagten Befehle sind. In dieser Weise führt der Prozessor die Befehle im Speicher in Übereinstimmung mit der durch die Verzweigungsvorhersageeinrichtung getroffenen Vorhersage aus. Nun wird ein programmierbares Verzweigungsvorhersagesystem gemäß der Erfindung beschrieben.

[0038] **Fig. 2** ist ein Blockschaltplan einer ersten Ausführungsform eines programmierbaren Verzweigungsvorhersagesystems gemäß der Erfindung. Ein Computer-System **30** kann ein programmierbares Verzweigungsvorhersagesystem **32** enthalten, wie gezeigt ist. Das Computer-System kann außerdem einen Prozessor **34** enthalten. Der Prozessor kann, wie oben beschrieben worden ist, die Befehle ausführen, die in einem Speicher gespeichert sein können, der nicht gezeigt ist. Wie oben beschrieben worden ist, ist die Erfindung nicht auf irgendeinen speziellen Prozessortyp eingeschränkt, wobei er CISC- oder RISC-Prozessoren enthalten kann.

[0039] Das programmierbare Verzweigungsvorhersagesystem **32** kann eine automatische Verzweigungsvorhersageeinrichtung (ABP) **36** und ein Vorhersagedaten-Speichersystem (P-STORE) **38** enthalten. Die automatische Verzweigungsvorhersageeinrichtung **36** kann irgendeinen Typ eines herkömmlichen Verzweigungsvorhersagesystems sein, weil das programmierbare Verzweigungsvorhersagesystem **32** jedes herkömmliche Verzweigungsvorhersagesystem aufnehmen kann, wie im Folgenden beschrieben ist. Ein Verzweigungsvorhersagesystem, das die Verzweigungen auf der Grundlage der früheren Verzweigungshistorie vorhergesagt, kann z. B. verwendet werden. Für die beschriebenen Ausführungsformen wird ein Vorhersagesystem anhand der früheren Verzweigungshistorie mit einem Ein-Bit- oder Zweit-Bit-Zähler für die Klarheit beschrieben, es kann aber außerdem jeder andere Typ eines automatischen Verzweigungsvorhersagesystems verwendet werden, weil das programmierbare Verzweigungsvorhersagesystem die Vorhersagedaten im P-STORE modifiziert, wobei es modifiziert werden kann, um mit jedem Typ von Vorhersagedaten zu arbeiten. Obwohl eine automatische Verzweigungsvorhersageeinrichtung, die nur das Ergebnis der Verzweigung vorhersagt, gezeigt ist, kann eine Verzweigungsvorhersageeinrichtung, die sowohl das Verzweigungsergebnis als auch die Zieladresse vorhersagt, außerdem verwendet werden.

[0040] Das automatische Verzweigungsvorhersagesystem **36** kann z. B. die Vorhersagedaten, die sich auf die frühere Verzweigungshistorie beziehen, im P-STORE **38** speichern. In **Fig. 3** ist ein Beispiel der im P-STORE **38** gespeicherten Vorhersagedaten für eine automatische Vorhersageeinrichtung anhand der früheren Verzweigungshistorie mit einem Ein-Bit-Zähler gezeigt. Der Vorhersagespeicher **38** kann für dieses spezielle Beispiel eine Verzweigungsreferenzspalte **40** und eine entsprechende Vorhersagedatenspalte **42** für jede

Verzweigung besitzen. Wenn ein anderer Typ von Verzweigungsvorhersagesystem verwendet worden wäre, dann würden die Daten im P-STORE außerdem verschieden sein. Die Verzweigungsreferenzspalte **40** kann die Adresse einer Verzweigung **44** enthalten. Die entsprechende Vorhersagespalte **42** kann das vorhergesagte Ergebnis **46** der speziellen Verzweigung speichern. In dem gezeigten Beispiel wird vorhergesagt, dass die erste Verzweigung genommen wird, während vorhergesagt wird, dass die zweite Verzweigungsadresse nicht genommen wird.

[0041] Die Erfindung ist nicht auf irgendein spezielles Verfahren, um auf die Vorhersagedaten Bezug zu nehmen, eingeschränkt, es werden jedoch zwei bevorzugte Verfahren der Bezugnahme beschrieben. Zuerst kann eine Verzweigungsreferenz, wie z. B. eine Verzweigungsadresse, auf eine P-STORE-Referenz, wie z. B. eine P-STORE-Adresse, abgebildet werden. Die Abbildung kann durch eine Anzahl bekannter Verfahren erfolgen, wie z. B. die direkte Abbildung, die bereichsassoziative oder die assoziative Abbildung. Auf die Vorhersagedaten kann außerdem insofern direkt Bezug genommen werden, als jede der Vorhersageoperationen die P-STORE-Referenz spezifiziert. Wie oben beschrieben worden ist, kann die Referenz für die Vorhersagedaten eine Adresse sein, es kann aber jedes anderer Stück der Daten sein, die der speziellen Verzweigung zugeordnet sind. Die Vorhersagedaten können die Abbildungsinformationen enthalten, die anzeigen, wie eine Verzweigungsreferenz in eine Referenz im P-STORE umgesetzt wird.

[0042] In **Fig. 2** bestimmt die ABP, wenn die ABP einem Verzweigungsbefehl oder einer Referenz auf den Verzweigungsbefehl begegnet, die Referenz für die spezielle Verzweigung, wie oben beschrieben worden ist, und greift dann durch eine A_Addr-Leitung auf die richtigen Vorhersagedaten im P-STORE **38** zu und liest die Vorhersagedaten über eine A_Read-Leitung aus dem P-STORE aus. Dann sendet die ABP die Vorhersagedaten über eine A_Result-Leitung zurück zum Prozessor. Die ABP kann in der Lage sein, den P-STORE zu aktualisieren, oder nicht. Für die erste Verzweigung wird z. B. der Prozessor informiert, dass vorhergesagt wird, dass die Verzweigung genommen wird, deshalb lädt der Prozessor die Befehle, auf die durch die Verzweigung gezeigt wird, in die Befehlspipeline.

[0043] Die Erfindung schafft ein System und ein Verfahren zum Ändern der im P-STORE **38** gespeicherten Vorhersagedaten unabhängig von der automatischen Verzweigungsvorhersageeinrichtung. Die Erfindung kann die Vorhersagedaten jederzeit modifizieren. Wohingegen die obenbeschriebene automatische Verzweigungsvorhersageeinrichtung die Vorhersagedaten verwendet, die im P-STORE gespeichert sind, wobei die Erfindung diese Vorhersagedaten während der Ausführung eines Programms in Reaktion auf die Vorhersageoperationen erzeugt und/oder modifiziert. Diese Vorhersageoperationen können vor der Ausführung des Programms durch ein separates Computer-System erzeugt werden, das ein Kompilierer-Programm ausführt, wie im Folgenden beschrieben ist. Die Vorhersagedaten im P-STORE für einen Prozeduraufruf-Befehl werden z. B. erzeugt, wenn das Programm kompiliert wird. Vor der Ausführung des Programms kann jedoch die tatsächliche Zieladresse für den Prozeduraufruf nicht bekannt sein. Folglich würde eine automatische Verzweigungsvorhersageeinrichtung das Ziel des Prozeduraufrufs falsch vorhersagen. Sobald das Programm jedoch ausgeführt wird, kann das Ziel des Prozeduraufrufs unter Verwendung einer Vorhersageoperation bestimmt werden, wie im Folgenden beschrieben ist, wobei dann vor der Ausführung des Prozeduraufruf-Befehls das richtige Ziel in den P-STORE geladen werden kann. Folglich wird, wenn der tatsächliche Prozessumschaltbefehl ausgeführt wird, die automatische Verzweigungsvorhersageeinrichtung auf der Grundlage der durch die Erfindung bereitgestellten aktualisierten Vorhersagedaten das Ziel des Prozeduraufrufs richtig vorhersagen.

[0044] Die Erfindung arbeitet im Zusammenhang mit jedem automatischen Verzweigungsvorhersagesystem, wobei sie die Vorhersagedaten auf der Grundlage von Vorhersageoperationen modifiziert, wie im Folgenden ausführlicher beschrieben ist. Die Erfindung vergrößert die Genauigkeit jedes herkömmlichen automatischen Verzweigungsvorhersagesystems, sie beeinflusst jedoch nicht den Betrieb der automatischen Verzweigungsvorhersageeinrichtung, weil die Erfindung nur die Vorhersagedaten aktualisiert, modifiziert oder auf sie zugreift. In der gezeigten Ausführungsform kann die Fähigkeit zum Modifizieren der Daten innerhalb des P-STORE unabhängig von der automatischen Verzweigungsvorhersageeinrichtung durch eine P_Write-Leitung **48** und eine P_Addr-Leitung **50** bereitgestellt sein. Diese Leitungen können mit einer Aktualisierungseinrichtung **52** verbunden sein, die die im P-STORE **38** gespeicherten Vorhersagedaten aktualisieren kann. Die Aktualisierungseinrichtung **52** kann außerdem die Vorhersagedaten aktualisieren, indem sie in bestimmten Situationen die Vorhersagedaten aus dem P-STORE liest, wie im Folgenden beschrieben ist.

[0045] Die Aktualisierungseinrichtung kann ein zweiter Prozessor sein, der die gleichen Befehle wie der Prozessor **34** ausführt und auf die Befehlsoperationen reagiert, die in den Programmcode, der ausgeführt wird, eingefügt worden sind, wie im Folgenden beschrieben ist. In Reaktion auf diese Vorhersageoperationen können die Vorhersagedaten innerhalb des P-STORE modifiziert werden, aktualisiert werden oder es kann auf sie zugegriffen werden. Ein Beispiel der Vorhersageoperation, die die Aktualisierungseinrichtung veranlasst, einen Wert im P-STORE zu modifizieren, wird im Folgenden beschrieben.

[0046] **Fig. 4** ist ein Blockschaltplan einer zweiten Ausführungsform des programmierbaren Verzweigungsvorhersagesystems **80** für einen Computer. Das programmierbare Verzweigungsvorhersagesystem kann einen Prozessor **82**, einen Speicher **84** zum Speichern der Befehle, eine automatische Verzweigungsvorhersage-

geeinrichtung **86** und einen P-STORE **88** enthalten. Diese Ausführungsform ist zur vorherigen Ausführungsform ähnlich, der P-STORE ist jedoch mit dem Prozessor verbunden, weil die Aktualisierungseinrichtung, wie sie oben beschrieben worden ist, in dieser Ausführungsform ein Programm sein kann, das im Prozessor ausgeführt wird. In der Tat, kann die Aktualisierung in dieser Ausführungsform Teil des Programms sein, das durch den Prozessor ausgeführt wird, da die Vorhersageoperationen innerhalb des Programmcodes eingebettet sind. Für diese Ausführungsform kann der Befehlsvorrat des Prozessors modifiziert sein, sodass der Prozessor versteht, wie die Vorhersageoperationen zu interpretieren sind, wobei er die Inhalte des P-STORE aktualisiert kann.

[0047] Der Prozessor **82** besitzt die gleichen A_Result- und C_Result-Leitungen zum Steuern der automatischen Verzweigungsvorhersageeinrichtung, er kann aber außerdem eine G_Control-Leitung besitzen, um die automatische Verzweigungsvorhersageeinrichtung zu veranlassen, die Inhalte des P-STORE zurückzusetzen. Der Prozessor kann außerdem Informationen von der automatischen Verzweigungsvorhersageeinrichtung über die G_Control-Leitung anfordern. Die frühere Vorhersagegenauigkeit kann z. B. über eine G_Result-Leitung zum Prozessor geliefert werden.

[0048] Wie gezeigt ist, ruft die automatische Verzweigungsvorhersageeinrichtung **86** die Vorhersagedaten über eine A_Addr-Leitung vom Speichersystem ab und empfängt diese Vorhersagen über die A_Read-Leitung. In dieser Ausführungsform kann die automatische Verzweigungsvorhersageeinrichtung außerdem über eine A_Write-Leitung Daten in den P-STORE **88** schreiben, nachdem die Verzweigung ausgeführt worden ist. Wenn die automatische Verzweigungsvorhersageeinrichtung eine Verzweigung falsch vorhersagt, dann kann die automatische Verzweigungsvorhersageeinrichtung die Vorhersagedaten für diese Verzweigung im Speichersystem revidieren. Die automatische Verzweigungsvorhersageeinrichtung schreibt unter Verwendung einer A_WRITE-Leitung die Vorhersagedaten in den P-STORE, nachdem eine Verzweigung ausgeführt worden ist.

[0049] Wie oben beschrieben worden ist, kann in dieser Ausführungsform das System zum Aktualisieren der Daten innerhalb des Vorhersagedatenspeichers ein Programm sein, das im Prozessor ausgeführt wird. Sowohl für diese Ausführungsform als auch für die erste Ausführungsform aktualisiert die Aktualisierungseinrichtung die Daten im P-STORE in Reaktion auf eine oder mehrere Vorhersageoperationen, die innerhalb des Programmcodes eingebettet sein können. Wenn der Programmcode kompiliert wird, kann z. B. ein Kompilierer, der vollständig vom programmierbaren Verzweigungsvorhersageprozessor separat sein kann, den Code, der kompiliert wird, nach bestimmten Bedingungen und Befehlstypen überprüfen, sodass in den kompilierten Befehlscode Vorhersageoperationen eingefügt werden können oder Vorhersageoperationen eingefügt werden können, wie dieser Code erzeugt wird. Die Erzeugung und die Einfügung dieser Vorhersageoperationen in den Programmcode wird im Folgenden beschrieben. Wenn die eingefügten Vorhersageoperationen ausgeführt werden, bewirken sie, dass die Vorhersagedaten im P-STORE aktualisiert werden. Wie oben beschrieben worden ist, besitzen zum Zeitpunkt der Kompilierung einige Verzweigungen ein unbekanntes Ziel, sie werden jedoch ein bestimmtes Ziel besitzen, sobald die Ausführung des Programms beginnt, oder das Ziel kann sich während der Programmausführung ändern. Folglich kann es im kompilierten Code einer Vorhersageoperationen geben, die die Aktualisierungseinrichtung veranlasst, die spezielle Verzweigung zu betrachten, um die Adresse zu bestimmen, und dann die gespeicherten Vorhersagedaten zu aktualisieren.

[0050] Um die Kompatibilität der Erfindung mit den früheren Befehlsvorräten zu sichern, können die Vorhersageoperationen für das programmierbare Verzweigungsvorhersagesystem in Form von nicht verwendeten no_operation-Codes in älteren Systemen ausgedrückt werden. Folglich kann die programmierbare Verzweigungsvorhersageeinrichtung gemäß der Erfindung mit modernen Computer-Architekturen verwendet werden, die für Verzweigungsvorhersageeinrichtungen konstruiert sind, sie kann aber außerdem mit älteren Systemen verwendet werden, weil die Vorhersageoperationen irgendeinen richtigen Betrieb in einem älteren System nicht stören.

[0051] In dieser zweiten Ausführungsform kann die Aktualisierung der Vorhersagedaten im Speichersystem durch die programmierbare Verzweigungsvorhersageeinrichtung ausgeführt werden, die im Prozessor ausgeführt wird. Falls eine Aktualisierung der Vorhersagedaten erforderlich ist, kann dann die Aktualisierung unter Verwendung der obenbeschriebenen P_Addr- und P_Write-Leitungen ausgeführt werden. Die Vorhersagedaten im P-STORE können außerdem durch das Lesen der Vorhersagedaten aus dem P-STORE durch eine P_READ-Leitung aktualisiert werden. Nun wird eine dritte Ausführungsform der Erfindung beschrieben.

[0052] **Fig. 5** ist ein Blockschaltplan der dritten Ausführungsform **100** des programmierbaren Verzweigungsvorhersagesystems gemäß der Erfindung. Diese Ausführungsform der programmierbaren Verzweigungsvorhersageeinrichtung kann einen Prozessor **102**, einen Speicher **104**, eine automatische Verzweigungsvorhersageeinrichtung **106** und einen P-STORE **108** enthalten. Diese Systeme sind oben ausführlich beschrieben worden, wobei sie hier nicht beschrieben werden. Das programmierbare Verzweigungsvorhersagesystem **100** kann außerdem eine Synchronisierereinrichtung **110** enthalten. Die Synchronisierereinrichtung kann den Prozessor durch die P_Sync-Leitung und die automatische Verzweigungsvorhersageeinrichtung durch die A_Sync-Leitung steuern. Die Synchronisierereinrichtung kann verwendet werden, um zu sichern, dass die Vorhersagedaten vom P-STORE durch die automatische Verzweigungsvorhersageeinrichtung zum richtigen Zeit-

punkt empfangen werden (d. h. das Empfangen der Verzweigungsvorhersagedaten, nachdem die Verzweigung ausgeführt worden ist, ist zu spät). Das System der Synchronisierereinrichtungen ist in der Technik wohl bekannt. Diese programmierbare Verzweigungsvorhersageeinrichtung kann die innerhalb des P-STORE gespeicherten Daten durch die P_Access-Leitung aktualisieren, wie oben beschrieben worden ist. In dieser Ausführungsform kann jedoch die P_Access-Leitung als die P_Write-, P_read- und P_Addr-Leitungen arbeiten. Diese Ausführungsform arbeitet in der gleichen Weise wie die vorherigen Ausführungsformen, wobei sie hier nicht beschrieben ist. Nun wird ein Beispiel des Betriebs des programmierbaren Verzweigungsvorhersagesystems beschrieben.

[0053] **Fig. 6** ist eine schematische Darstellung der Inhalte des P-STORE für ein 2-Bit-Zähler-Vorhersagesystem. In einem automatischen Zwei-Bit-Zähler-Verzweigungsvorhersagesystem werden zwei Bits verwendet, um die früheren Verzweigungsvorhersagen zu verfolgen. Wie gezeigt ist, kann ein Datenspeichersystem **140** ein Adressenfeld **142**, ein bit_1-Feld **144** und ein bit_0-Feld **146** besitzen. Falls eine Verzweigung früher niemals genommen worden ist oder sie nicht wenigstens zweimal genommen worden ist, dann sind, wie für die Adresse 3 gezeigt ist, beide Bits 0. Falls die Verzweigung einmal genommen worden ist, ist das bit_0 "1", wie für die Adresse 0 gezeigt ist. Wenn eine Verzweigung zweimal genommen worden ist, dann ist das bit_1 "1" und das bit_0 "0", wie bei jedem binären Zähler. Für diese automatische Zwei-Bit-Zähler-Verzweigungsvorhersageeinrichtung wird vorhergesagt, dass künftige Verzweigungen genommen werden, falls das bit_1 "1" ist.

[0054] In einem herkömmlichen System, wie in **Fig. 7** gezeigt ist, sind die vier möglichen Zustände des Zählers (00, 01, 10, 11) zusammen mit der Wirkung einer genommenen Verzweigung und einer nicht genommenen Verzweigung auf den Zähler gezeigt. Wie gezeigt ist, inkrementiert eine genomme Verzweigung, die vorhergesagt wird, den Zähler um eins. Folglich wird "10" zu "11". Eine nicht genomme Verzweigung, die vorhergesagt wird, dekrementiert den Zähler um eins. In dieser Weise werden nach der Ausführung jeder Verzweigung die Zähler entweder um eins vergrößert oder um eins verkleinert.

[0055] **Fig. 8** ist eine Tabelle, die die Wirkungen der Erfindung auf den 2-Bit-Zähler zeigt. Die Erfindung inkrementiert oder dekrementiert den Zähler nicht notwendigerweise, kann aber den Zähler so einstellen, dass vorhergesagt wird, dass eine Verzweigung genommen wird, wenn sie das nächste Mal ausgeführt wird. Um dies zu erreichen, kann z. B. das bit_1 des Zählers auf "1" gesetzt werden, um zu erzwingen, dass vorhergesagt wird, dass die Verzweigung genommen wird, oder es kann auf "0" gesetzt werden, um zu erzwingen, dass vorhergesagt wird, dass die Verzweigung nicht genommen wird. Ein Beispiel der Typen der Vorhersageoperationen, die in einen kompilierten Code eingefügt werden können, wird im Folgenden beschrieben. Im Allgemeinen enthält das Verfahren des Einfügens der Vorhersageoperationen das Analysieren des Programms, um zu bestimmen, ob es irgendwelche Gelegenheiten für das Berechnen von Daten gibt, die verwendet werden können, um das künftige Verzweigungsverhalten vorherzusagen, und das Einfügen von Vorhersageoperationen in den Computer-Code, um den P-STORE zu aktualisieren oder zu modifizieren, sodass die Verzweigungsvorhersageeinrichtung entsprechend reagiert. Nun werden zwei Beispiele eines Verfahrens zum Einfügen dieser Vorhersageoperationen in einen Programmcode beschrieben.

[0056] Die Vorhersageoperationen können einen Vorhersageausdruck oder ein Vorhersagemuster verwenden, um die Inhalte des P-STORE zu modifizieren. Ein Beispiel eines Vorhersageausdrucks kann ein verbreiteter Wert sein, wie im Folgenden beschrieben ist. Ein Beispiel eines Vorhersagemusters kann eine AnzeigevARIABLE sein, wie im Folgenden beschrieben ist. Der Vorhersageausdruck kann irgendein Ausdruck sein, der verwendet werden kann, um den Ausdruck vorherzusagen oder zu berechnen, von dem das Verhalten einer Verzweigung abhängt. Ähnlich ist ein Vorhersagemuster ein Muster der Vorhersagen, das verwendet werden kann, um den Ausdruck vorherzusagen oder zu berechnen, von dem das Verhalten einer Verzweigung abhängt. Es gibt viele Arten von Prozeduren oder Verfahren, die verwendet werden können, um Vorhersageausdrücke oder -muster zu identifizieren, wobei alle diese Ausdrücke oder Muster gemäß der Erfindung verwendet werden können, um Gelegenheiten zu identifizieren, um Vorhersageoperationen einzufügen. Nun werden zwei Beispiele dieser verschiedenen Typen von Ausdrücken beschrieben.

[0057] **Fig. 9** ist ein Ablaufplan eines ersten Beispiels eines Verfahrens **160** gemäß der Erfindung zum Einfügen von Vorhersageoperationen in einen Computer-Code. Für die in den **Fig. 9** und **10** gezeigten Beispiele kann die Einfügung der Vorhersageoperationen außerdem durch den Prozessor des programmierbaren Vorhersagesystems ausgeführt werden. Sie kann außerdem durch ein vollständig separates Computer-System ausgeführt werden, das einen Kompilierer ausführt, der diese spezialisierten Vorhersageoperationen erzeugt und in den Programmcode einfügen kann. Sie kann außerdem durch denselben Prozessor, der den Kompilierer ausführt, zu einem früheren Zeitpunkt als die Ausführung des Programms ausgeführt werden. Folglich kann das programmierbare Vorhersagesystem den Programmcode ausführen, der die Vorhersageoperationen enthält, die durch ein anderes Computer-System in den Programmcode eingefügt worden sind. Das Verfahren beginnt im Schritt **162**, wobei im Schritt **164** der Programmcode, der kompiliert wird, analysiert wird, um nach verschiedenen Eigenschaften zu suchen, wie z. B. Verzweigungsadressen und Variable. Im Schritt **166** werden die Induktionsvariable, die Beispiele der Vorhersagemuster sind, innerhalb des Programmcodes identifiziert, wie in der Technik bekannt ist. Dann werden im Schritt **168** die Ausdrücke für die Verzweigungen berechnet,

die von der Induktionsvariable abhängen. Eine Variable, die die Anzahl steuert, in der eine Schleife ausgeführt werden kann, kann z. B. eine Induktionsvariable sein, wobei die tatsächliche Anzahl, in der die Schleife wahrscheinlich ausgeführt wird, berechnet werden kann. Dann wird im Schritt **170** eine Vorhersageoperation in den Programmcode eingefügt, sodass die Vorhersagedaten für den Verzweigungsbefehl, der z. B. die Schleife steuert, aktualisiert werden, um die berechnete Anzahl widerzuspiegeln, mit der vorhergesagt wird, wie oft die Schleife ausgeführt wird. Im Schritt **172** bestimmt der Kompilierer, ob es irgendwelche weiteren Verzweigungen im Computer-Code gibt, die analysiert werden müssen. Wenn es weitere Verzweigungen gibt, dann kann die Steuerung zurück zum Schritt **164** gehen. Die Steuerung kann außerdem zum Schritt **168** zurückkehren, weil eine einzelne Induktionsvariable oder ein Satz von Induktionsvariable mehrere Verzweigungen steuern kann, oder es können alle Induktionsvariable oder Verzweigungen anfangs identifiziert werden. Wenn der Computer-Code vollständig analysiert worden ist, dann endet das Verfahren im Schritt **174**.

[0058] **Fig. 10** ist ein Ablaufplan eines zweiten Beispiels eines Verfahrens **200** gemäß der Erfindung zum Einfügen von Vorhersageoperationen in einen Computer-Code.

[0059] Das Verfahren beginnt im Schritt **202**, wobei im Schritt **204** der Programmcode, der kompiliert wird, analysiert wird, um nach verschiedenen Eigenschaften zu suchen, wie z. B. Verzweigungsadressen und Variable. Im Schritt **206** werden alle verbreiteten Werte, die Beispiele eines Vorhersageausdrucks sind, innerhalb des Programmcodes identifiziert, wie in der Technik bekannt ist. Dann werden im Schritt **208** die Ausdrücke für die Verzweigungen berechnet, die von dem verbreiteten Wert abhängen. Der Merker, der z. B. eine Verzweigung steuert, die der if-Anweisung entspricht, kann z. B. ein verbreiteter Wert sein, wobei dieser verbreitete Wert berechnet werden kann. Dann wird im Schritt **210** eine Vorhersageoperation in den Programmcode eingefügt, sodass z. B. die Vorhersagedaten für den Verzweigungsbefehl, die den Verzweigungsbefehl steuern, der der if-Anweisung entspricht, aktualisiert werden, um den berechneten Wert des verbreiteten Wertes widerzuspiegeln. Im Schritt **212** bestimmt der Kompilierer, ob es irgendwelche weiteren Verzweigungen im Computer-Code gibt, die analysiert werden müssen. Wenn es weitere Verzweigungen gibt, dann kann die Steuerung zurück zum Schritt **204** gehen. Außerdem kann die Steuerung zum Schritt **208** zurückkehren, weil ein einzelner verbreiteter Wert oder ein Satz verbreiteter Werte mehrere Verzweigungen steuern kann, oder es können alle verbreiteten Werte oder Verzweigungen anfangs identifiziert werden. Wenn der Computer-Code vollständig analysiert worden ist, dann endet das Verfahren im Schritt **214**. Nun wird ein Beispiel des Einfügens einer Vorhersageoperation für eine Induktionsvariable und einen verbreiteten Wert in ein Stück des Computer-Codes beschrieben.

[0060] Für dieses Beispiel wird ein Fragment einer einzelnen C-Quellcodeschleife ohne die Vorhersageoperationen kompiliert und dann mit den Vorhersageoperationen am Platz kompiliert und der Wirkung, die die Vorhersageoperationen auf die Vorhersagegenauigkeit des Systems besitzen, gezeigt.

[0061] Das Fragment des C-Quellcodes kann:

```
i=0; flag = old; do{
    @1 flag= new;
    @2
    if(!flag){
        @3}
    else @4
} while (++i<n);
```

sein, wobei **@1**, **@2**, **@3** und **@4** beliebige Fragmente des Codes sind. In der Tat kann die Variable **flag** an anderen Punkten im Code geändert werden. Dieses Fragment eines C-ähnlichen Quellcodes kann in Assemblercode kompiliert werden, der wie:

```

mov          #0,1
mov          old, flag
TOP c@1
mov          new, flag
c@2
branch flag, FLGSKP
c@3
FLGSKP c@4
add         #1,i
branch i<n, TOP

```

aussieht, wobei c@1, c@2, c@3 und c@4 der kompilierte Code für @1, @2, @3 bzw. @4 ist, wobei die Marken innerhalb des kompilierten Codes, wie in der Technik bekannt ist, in Großbuchstaben vorliegen. Die Variable "i" ist eine Induktionsvariable, wie in der Technik bekannt ist. Die Werte "old" und "new" sind verbreitete Werte, wie in der Technik bekannt ist.

[0062] Wie durch den kompilierten Code gezeigt wird, kann, wenn zum ersten Mal in die Schleife eingetreten wird, das Verzweigungsvorhersagesystem falsch vorhersagen, dass die Verzweigung nicht genommen wird, weil es diese Verzweigung vorher niemals gesehen hat. Dann kann, sobald die Schleife das letzte Mal ausgeführt wird, das herkömmliche Verzweigungsvorhersagesystem falsch vorhersagen, dass die Verzweigung ein weiteres Mal genommen wird. Im Gegensatz sagt die Erfindung alle Verzweigungen richtig vorher, wie im Folgenden beschrieben ist, wobei dies eine Genauigkeit von 100% ergibt.

[0063] Für die Erfindung können einige Vorhersageoperationen in den kompilierten Code hinzugefügt werden, sodass der revidierte kompilierte Code:

```

mov          #0,1
repeat      n-1, BACK
    mov      old, flag
    predict  old, TSTFLG
TOP c@1
    predict  new, TSTFLG
    mov      new, flag
    c@2
TSTFLG branch flag, FLGSKP
    c@3
FLGSKP c@4
    add     #1,i
BACK branch i<n, TOP

```

sein kann, wobei die zwei Beispiel-Vorhersageoperationen "predict" und "repeat" gemäß der Erfindung in den kompilierten Code eingefügt worden sind. Die Vorhersageoperation kann ein erstes Argument, das der vorhergesagte Wert der Bedingung ist, und ein zweites Argument, das die Adresse der Verzweigung ist, für die die Vorhersage gilt, besitzen. Die Vorhersageoperation ändert auf der Grundlage des Programms diesen Wert der Vorhersagedaten, sodass das Verzweigungsvorhersagesystem die Schleife das erste Mal nicht falsch vorhersagt.

[0064] Die Vorhersageoperation repeat kann ein erstes Argument besitzen, das die vorhergesagte Anzahl sein kann, in der die Verzweigung genommen wird, bevor sie nicht genommen wird. In diesem Beispiel verursacht die Verzweigung, dass das Programm die Schleife n – 1-mal durchläuft. Das zweite Argument der Vorhersageoperation repeat kann die Adresse der Verzweigung sein, für die die Vorhersage gilt. Die Vorhersageoperation repeat modifiziert den Wert der Vorhersagedaten innerhalb des P-STORE, sodass das Verzweigungsvorhersagesystem das Ende der Schleife nicht falsch vorhersagt. Folglich erlaubt die Kombination der

Vorhersageoperationen `repeat` und `predict` dem programmierbaren Verzweigungsvorhersagesystem, für dieses einfache Beispiel zu 100% genau zu sein. Weil die Werte von `FLAG` und `n` bekannt sind, bevor die geeigneten Verzweigungen ausgeführt werden, kann das programmierbare Verzweigungsvorhersagesystem Vorhersageoperationen hinzufügen, um dies auszunutzen. Das gezeigte Beispiel ist ein einfaches Beispiel, wobei viele andere verschiedene Vorhersageoperationen in den Programmcode eingefügt werden können, wie z. B. eine Vorhersageoperation "destination", die das Ziel einer Verzweigung vorhersagen kann. Nun werden die Änderungen im P-STORE für das gezeigte Beispiel, die durch die Erfindung verursacht werden, beschrieben. [0065] Für dieses Beispiel kann der P-STORE ein 2-Bit-Sättigungszähler, wie oben beschrieben worden ist, und ein größerer Zähler für jede Verzweigungsadresse sein. Ein Beispiel des Speichers für die vorhergesagten Daten ist in **Fig. 11** gezeigt. Wie gezeigt ist, besitzt in diesem Beispiel die `if`-Anweisung im ursprünglichen C-ähnlichen Quellcode den 2-Bit-Zähler, der der `TST_FLG`-Marke zugeordnet ist, wobei sie in der Tabelle mit 2 Bits markiert ist. Der größere Countdown-Zähler ist der `BACK`-Marke (für die `do`-Schleife) zugeordnet, wobei er in der Tabelle mit `C` markiert ist. Die Werte dieser zwei Stücke der Vorhersagedaten (2 Bits und `C`) innerhalb des Speichersystems sind an verschiedenen Punkten bei der Ausführung des Programmcodes gezeigt, so dass die durch die Vorhersageoperationen vorgenommenen Änderungen deutlicher werden.

[0066] Am Anfang der Ausführung des kompilierten Programmcodes können der 2-Bit-Zähler für die `if`-Anweisung und die `do/while`-Schleife beide auf "00" gesetzt sein, wobei der Countdown-Zähler für beide Verzweigungen außerdem auf "0" gesetzt ist. Für ein komplexeres Programm würde es mehr Einträge in der Tabelle geben. Wenn die Vorhersageoperation "`repeat n - 1, BACK`" ausgeführt wird, bleibt der 2-Bit-Zähler für die `if`-Anweisung unbeeinflusst, aber der `C`-Zähler für die `do/while`-Schleife wird auf "`n - 1`" gesetzt, dies stellt die vorhergesagte Anzahl dar, in der die Verzweigung genommen wird, um die Schleife auszuführen. Außerdem kann der 2-Bit-Zähler für die `do/while`-Schleife auf "10" oder irgendeinen anderen Wert gesetzt sein, der anzeigt, dass die Verzweigung das nächste Mal genommen werden sollte. Wie oben beschrieben worden ist, verhindert das Setzen des `C`-Zählers und des 2-Bit-Zählers auf "`n - 1`" bzw. "10" vor der tatsächlichen Ausführung der `do`-Schleife, dass die Verzweigungsvorhersageeinrichtung falsch vorhersagt, dass die erste Verzweigung der `do/while`-Schleife nicht genommen wird. Das Verhalten der gezeigten Schleife kann außerdem vorhergesagt werden, indem der 2-Bit-Zähler ignoriert wird und auf den `C`-Zähler vertraut wird. Wenn der `C`-Zähler null erreicht, sollte vorhergesagt werden, dass die nächste Ausführung der Verzweigung nicht genommen wird. Folglich kann, selbst wenn die Verzweigungsvorhersageeinrichtung die `do/while`-Schleife vorher niemals gesehen hat, die programmierbare Verzweigungsvorhersageeinrichtung sicherstellen, dass die erste Verzweigung der `do/while`-Schleife richtig vorhergesagt wird.

[0067] Wenn der Vorhersageoperation "`predict old, TSTFLAG`" begegnet wird, kann der 2-Bit-Zähler für den Verzweigungsbefehl, der der `if`-Anweisung entspricht, auf "0X" zurückgesetzt werden, falls z. B. angenommen wird, dass "old" falsch ist, wobei die Zähler für die `do/while`-Schleife unbeeinflusst bleiben. Das "X" zeigt an, dass der Wert dieses Bits nicht wichtig ist, wobei es irgendeinen Wert sein kann. Wenn die Vorhersageoperation "`predict new, TSTFLAG`" ausgeführt wird, wird der 2-Bit-Zähler, der dem Verzweigungsbefehl entspricht, der der `if`-Anweisung entspricht, auf "1X" gesetzt, falls z. B. angenommen wird, dass "new" wahr ist, was erzwingt, dass vorhergesagt wird, dass die Verzweigung genommen wird. Diese Einstellung des 2-Bit-Zählers für die Verzweigung, die der `if`-Anweisung entspricht, verhindert abermals, dass die Verzweigungsvorhersageeinrichtung die erste Verzweigung der `if`-Anweisung falsch vorhersagt. Die Zähler für die Verzweigung, die der `do/while`-Schleife entspricht, bleiben unbeeinflusst.

[0068] Wenn die Verzweigungsoperation "`branch i < n, TOP`" ausgeführt wird, wird die Verzweigung zurück zum Anfang des Codes richtig vorhergesagt, weil der `C`-Zähler und der 2-Bit-Zähler des Verzweigungsbefehls, der die `do/while`-Schleife implementiert, eingestellt worden sind, um vorherzusagen, dass die Verzweigung genommen wird, wobei der `C`-Zähler um eins dekrementiert wird, sodass er gleich "`n - 2`" sein kann. Der 2-Bit-Zähler für die Verzweigung, die der `do/while`-Schleife entspricht, hält fortgesetzt einen Wert von "11", weil die `do/while`-Verzweigung mehrmals genommen worden ist. Die Zähler für die Verzweigung, die der `if`-Anweisung entspricht, bleiben unbeeinflusst. Die Ausführung des Programmcodes wird in dieser Weise bis zum Ende der `do/while`-Schleife fortgesetzt.

[0069] Am Ende der `do/while`-Schleife ist der `C`-Zähler gleich "0", während der 2-Bit-Zähler für die Verzweigung, die der `do/while`-Schleife entspricht, durch das programmierbare Vorhersagesystem auf "00" zurückgesetzt wird, sodass das nächste Mal, wenn die `do/while`-Schleife ausgeführt wird, die Verzweigungsvorhersageeinrichtung richtig vorhersagen wird, dass die Verzweigung nicht genommen wird. Eine herkömmliche Verzweigungsvorhersageeinrichtung kann falsch vorhersagen, dass die Verzweigung genommen wird, weil diese herkömmlichen Systeme das frühere Verzweigungsverhalten verwenden, um die Werte der Vorhersagedaten zu aktualisieren und um folglich das künftige Verhalten vorherzusagen. Wie in diesem Beispiel gezeigt ist, verringert die programmierbare Verzweigungsvorhersageeinrichtung gemäß der Erfindung die falschen Vorhersagen in einer herkömmlichen automatischen Verzweigungsvorhersageeinrichtung, wie z. B. Ein-Bit- oder Zwei-Bit-Zählern. Die Vorhersagegenauigkeit der programmierbaren Verzweigungsvorhersageeinrichtung beträgt für dieses Beispiel 100%. Weil das Vorhersagen des Verhaltens einer Verzweigung am Anfang und am

Ende von Schleifen, wie z. B. Verzweigungen, die do/while implementieren, verursacht, dass die meisten herkömmlichen Verzweigungsvorhersageeinrichtungen wenigstens einmal falsch vorhersagen, kann die programmierbare Verzweigungsvorhersageeinrichtung gemäß der Erfindung diesen Anfang und dieses Ende von Schleifenverzweigungen richtig vorhersagen.

[0070] Das System und das Verfahren zum Einfügen von einer oder mehreren Vorhersageoperationen in einen Programmcode, um die Leistung einer automatischen Verzweigungsvorhersageeinrichtung zu verbessern, kann innerhalb des Prozessors des Computer-Systems ausgeführt werden, wie oben beschrieben worden ist, oder es kann durch einen separaten Prozessor ausgeführt werden, der ein Kompilierungsprogramm ausführt. Die Erzeugung und das Einfügen von einer oder mehreren Vorhersageoperationen können in einem vom System mit der programmierbaren Verzweigungsvorhersageeinrichtung vollständig verschiedenen Computer-System ausgeführt werden. Außerdem können die Erzeugung und die Einfügung der einen oder der mehreren Vorhersageoperationen unmittelbar vor der Verzweigungsvorhersage oder irgendwann vor irgendeiner Verzweigungsvorhersage ausgeführt werden. Das obenbeschriebene programmierbare Verzweigungsvorhersagesystem kann folglich, muss aber nicht, die eine oder die mehreren Vorhersageoperationen erzeugen oder einfügen. Deshalb kann das programmierbare Verzweigungsvorhersagesystem Programmcode ausführen, in dem ein oder mehrere Verzweigungsbefehle vorher eingefügt worden sind. Ähnlich können das System und das Verfahren zum Erzeugen und Einfügen der einen oder der mehreren Vorhersageoperationen unabhängig vom obenbeschriebenen programmierbaren Verzweigungsvorhersagesystem verwendet werden. Es werden nun zwei spezifische Beispiele der Vorhersageoperationen beschrieben.

[0071] Die Vorhersageoperationen, die erzeugt und in einen Programmcode eingefügt werden, können entweder die Vorhersagedaten aktualisieren, indem sie neue Daten in den P-STORE schreiben, oder die Vorhersagedaten aktualisieren, indem sie die neuen Daten spezifizieren, die von einer bestimmten Stelle in den P-STORE gelesen werden sollten. Diese Vorhersageoperationen, die Daten schreiben, können z. B. die obenbeschriebenen Vorhersageoperationen predict und repeat sein. Diese Vorhersageoperationen für das Schreiben der Vorhersagedaten können aber außerdem irgendein Typ der Vorhersageoperation sein, die eine Referenz auf die Verzweigung, wie z. B. Verzweigungsadresse, und die zu aktualisierenden Vorhersagedaten besitzt.

[0072] Diese Vorhersageoperationen für das Schreiben der Vorhersagedaten können aus einer oder mehreren Vorhersageoperationen bestehen oder sie können als verschiedene fundamentale Befehle ausgedrückt sein. Die Vorhersageoperation "predict old, TSTFLG" und die Vorhersageoperation "predict new, TSTFLG" können z. B. tatsächlich fundamentale Befehle sein, sodass die Vorhersageoperationen den P-STORE beeinflussen, aber keine Wirkung in älteren Systemen besitzen, die das programmierbare Verzweigungsvorhersagesystem nicht besitzen. Beide Befehle verursachen, dass die Vorhersagedaten für die Verzweigung, auf die durch TSTFLG Bezug genommen wird, durch das Schreiben der Vorhersagedaten in den P-STORE aktualisiert werden. Die aktualisierten Vorhersagedaten können verursachen, dass vorhergesagt wird, dass die Verzweigung entweder genommen oder nicht genommen wird.

[0073] Die Vorhersageoperation kann außerdem als mehr als ein fundamentaler Befehl ausgedrückt sein. Die Vorhersageoperation "predict old, TSTFLG" kann z. B. als:

```
mov<data>, E
mov<address>, F
oder
output<data>, G
output<address>, H
```

ausgedrückt sein, wobei E–H so gewählt sind, dass der Befehl in einem älteren System nichts tut, er aber einem neueren System den P-STORE beeinflusst. Bei diesem Beispiel aus zwei Befehlen aktualisiert der erste Befehl in jedem Paar die Daten, während der zweite Befehl in jedem Paar die Adresse aktualisiert. Ähnlich kann eine Vorhersageoperation mit drei fundamentalen Befehlen implementiert sein, wobei der erste Befehl die Vorhersagedaten spezifiziert, der zweite Befehl die Verzweigungsreferenz spezifiziert und der dritte Befehl die Operation an den Vorhersagedaten ausführt. Folglich kann die Vorhersageoperation aus irgendeiner Anzahl fundamentaler Befehle bestehen. Nun wird ein Beispiel einer Lesevorhersageoperation unter Verwendung des Befehls gemäß der Erfindung beschrieben.

[0074] Eine Lesevorhersageoperation ist zu den obenbeschriebenen Schreibvorhersageoperationen ähnlich, mit Ausnahme, dass die Lesevorhersageoperation spezifiziert, wo die Vorhersagedaten gespeichert sind, aber nicht die geschriebenen Daten spezifiziert, wie im Folgenden beschrieben ist. Wie die Schreibvorhersageoperationen kann die Lesevorhersageoperation in einen oder mehrere fundamentale Befehle zerlegt sein. Ein Beispiel der Lesevorhersageoperation wird nun beschrieben.

[0075] Wie oben beschrieben worden ist, kann ein Prozessschalter durch das programmierbare Verzweigungsvorhersagesystem gemäß der Erfindung gehandhabt werden, weil die Inhalte des P-STORE vor dem Prozessschalter gespeichert werden können, wobei dann, wenn der Prozess abermals gestartet wird, die Vorhersagedaten für den speziellen Prozess zurück im P-STORE gespeichert werden können. Wenn durch den

Kompilierer ein Prozessschalter lokalisiert wird, werden die Vorhersagedaten aus dem P-STORE gelesen und unter Verwendung z. B. einer Vorhersageoperation "save <prediction data>" gespeichert, die z. B. durch einen fundamentalen Befehl "mov PSTORE[0..n], SAVE[0..n]", eine Folge von Befehlen oder ein Unterprogramm implementiert sein kann. Folglich werden einige oder alle Daten aus dem P-STORE in einer SAVE-Stelle gespeichert. Dann können, wenn das Programm zurück zum ursprünglichen Prozess schaltet, die in einer SAVE-Stelle gespeicherten Vorhersagedaten zurück in den P-STORE wiederhergestellt werden, z. B. durch eine Vorhersageoperation "restore <prediction data>", die z. B. durch einen fundamentalen Befehl "mov SAVE[0..n], PSTORE[0..n]", eine Folge von Befehlen oder ein Unterprogramm implementiert sein kann. Diese Lesevorhersageoperationen würden so eingefügt werden, dass sie etwa zu dem Zeitpunkt ausgeführt werden, zu dem ein Prozessschalter auftritt.

Patentansprüche

1. System zum Vorhersagen von Verzweigungen in einem Computersystem, das umfaßt: einen Speicher (**26**), der einen Programmcode enthält, der mehrere Befehle umfaßt, wobei die Befehle einen Verzweigungsbefehl und eine Vorhersageoperation enthalten, wobei die Vorhersageoperation in den Programmcode an einer vorgegebenen Stelle in dem Programmcode in Bezug auf den Verzweigungsbefehl eingefügt wird und in den Programmcode vor der Ausführung des Programmcodes eingefügt wird; einen Vorhersagespeicher (**38**), der verzweigungsspezifische Vorhersagedaten enthält; eine Einrichtung zum Schreiben von Vorhersagedaten für die Verzweigung in den Vorhersagespeicher (**38**) in Reaktion auf die Ausführung der in dem Programmcode enthaltenen Vorhersageoperation; und eine Einrichtung (**36**) zum Vorhersagen des Verhaltens des Verzweigungsbefehls auf der Grundlage der Vorhersagedaten in dem Vorhersagespeicher (**38**).
2. System nach Anspruch 1, bei dem die Vorhersageeinrichtung (**36**) eine Einrichtung zum Vorhersagen einer Zielreferenz einer Verzweigung, deren Wahl vorhergesagt wird, umfaßt.
3. System nach Anspruch 1, bei dem die Vorhersageeinrichtung (**36**) eine Einrichtung zum Vorhersagen einer Entscheidung über eine Verzweigung umfaßt.
4. System nach Anspruch 1, bei dem die Vorhersageeinrichtung (**36**) ein Computerprogramm umfaßt, das durch den Prozessor ausgeführt wird.
5. System nach Anspruch 1, bei dem die Vorhersageoperation eine Entscheidung über eine vorhergesagte Verzweigung vornimmt.
6. System nach Anspruch 1, bei dem der Verzweigungsbefehl den Programmcode zu einer Schleife veranlaßt und die Vorhersageoperation eine Wiederholungsvorhersageoperation umfaßt, so daß die vorhergesagte Anzahl, in der die Schleife ausgeführt wird, in den Vorhersagespeicher (**38**) geschrieben wird, so daß sie von der Vorhersageeinrichtung (**36**) verwendet werden kann, um das Schleifenverhalten der Verzweigung vorherzusagen.
7. System nach Anspruch 1, bei dem die Vorhersageoperation ein Ziel einer vorhergesagten Verzweigung schafft.
8. System nach Anspruch 1, bei dem die Vorhersageoperation ein Argument enthält, das die Adresse des Verzweigungsbefehls angibt.
9. System nach Anspruch 1, bei dem die Vorhersageoperation ein Argument enthält, das eine Stelle in dem Vorhersagespeicher (**38**) angibt.
10. System nach Anspruch 1, bei dem die Vorhersageoperation Vorhersagedaten modifiziert, die der Verzweigung entsprechen und bereits in dem Vorhersagespeicher (**38**) vorhanden sind.
11. System nach Anspruch 1, bei dem der Programmcode außerdem eine zweite Vorhersageoperation enthält und wobei das System ferner eine Einrichtung zum Lesen von Vorhersagedaten aus dem Vorhersagespeicher (**38**) in Reaktion auf die Ausführung der zweiten Vorhersageoperation umfaßt.
12. System nach Anspruch 11, bei dem die zweite Vorhersageoperation die aus dem Vorhersagespeicher ausgelesenen Vorhersagedaten unter Verwendung eines Arguments, das die Adresse der Verzweigung an-

gibt, identifiziert.

13. System nach Anspruch 11, bei dem die zweite Vorhersageoperation die aus dem Vorhersagespeicher (38) ausgelesenen Vorhersagedaten unter Verwendung eines Arguments, das ihren Ort in dem Vorhersagespeicher (38) angibt, identifiziert.

14. Verfahren (160) zum Einfügen von Vorhersageoperationen in einen Programmcode vor der Ausführung des Programmcodes, wobei der Programmcode mehrere Befehle einschließlich Verzweigungsbefehlen umfaßt, um ein Verhalten einer Verzweigung vorherzusagen, wobei das Verfahren umfaßt:

Analysieren (164) des Programmcodes vor der Ausführung des Programmcodes, um eine Verzweigung zu lokalisieren, deren Verhalten von einer vorgegebenen Eigenschaft abhängt;

Berechnen (168) der Werte der vorgegebenen Eigenschaft vor der Ausführung der Verzweigung;

Erzeugen einer Vorhersageoperation, um die Werte der vorgegebenen Eigenschaft in einem Vorhersagespeicher zu speichern; und

Einfügen (170) der Vorhersageoperation in den Programmcode vor der Ausführung des Programmcodes, so daß die Vorhersageoperation vor dem Verzweigungsbefehl ausgeführt wird, um das Verhalten des Verzweigungsbefehls auf der Grundlage der in dem Vorhersagespeicher gespeicherten berechneten Werte vorherzusagen.

15. Verfahren nach Anspruch 14, bei dem die vorgegebene Eigenschaft eine Induktionsvariable enthält.

16. Verfahren nach Anspruch 14, bei dem die vorgegebene Eigenschaft ein Vorhersagemuster enthält.

17. Verfahren nach Anspruch 14, bei dem die vorgegebene Eigenschaft einen Vorhersageausdruck enthält.

18. Verfahren nach Anspruch 14, bei dem die vorgegebene Eigenschaft einen verbreiteten Wert enthält.

19. Verfahren nach Anspruch 14, bei dem das Einfügen das Einfügen der Vorhersageoperation in den Programmcode während der Kompilierung des Programmcodes umfaßt.

20. Verfahren nach Anspruch 14, bei dem die Berechnung ferner die Bestimmung, welche Argumente der Vorhersageoperation Werte besitzen, die bekannt sind, wenn die Vorhersageoperation eingefügt wird, die Berechnung der Werte der Argumente und die Einfügung eines oder mehrerer Befehle für die Berechnung der Werte irgendwelcher verbleibender Argumente der Vorhersageoperation in den Programmcode umfaßt.

21. Verfahren nach Anspruch 20, bei dem einer oder mehrere der eingefügten Befehle eine Vorhersageoperation umfassen, die Vorhersagedaten aus dem Vorhersagespeicher lesen.

Es folgen 7 Blatt Zeichnungen

Fig. 1

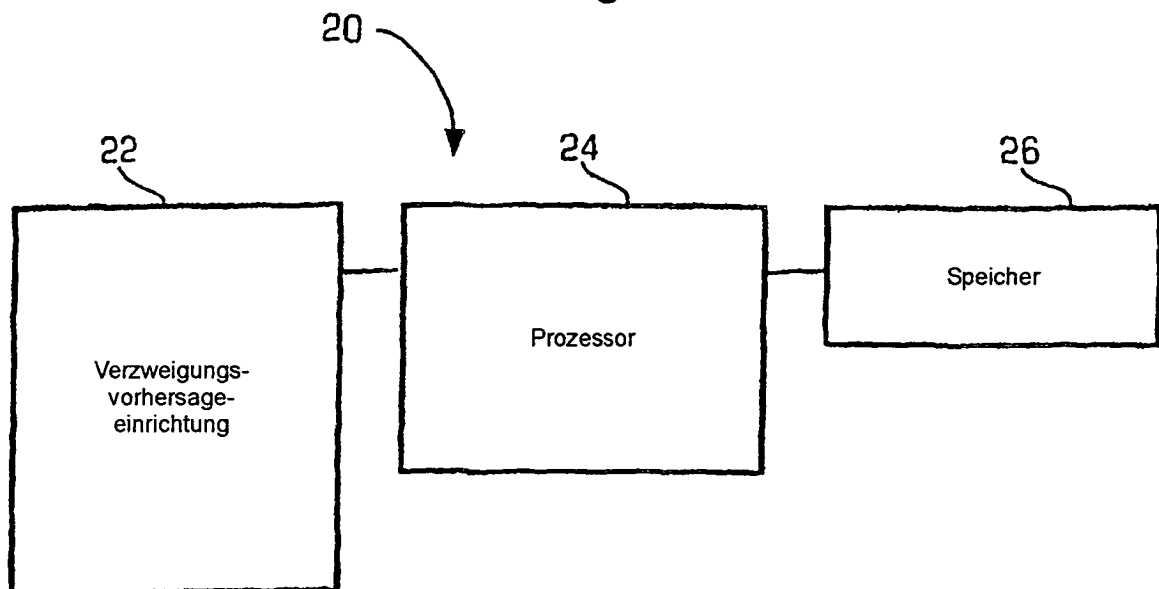


Fig. 2

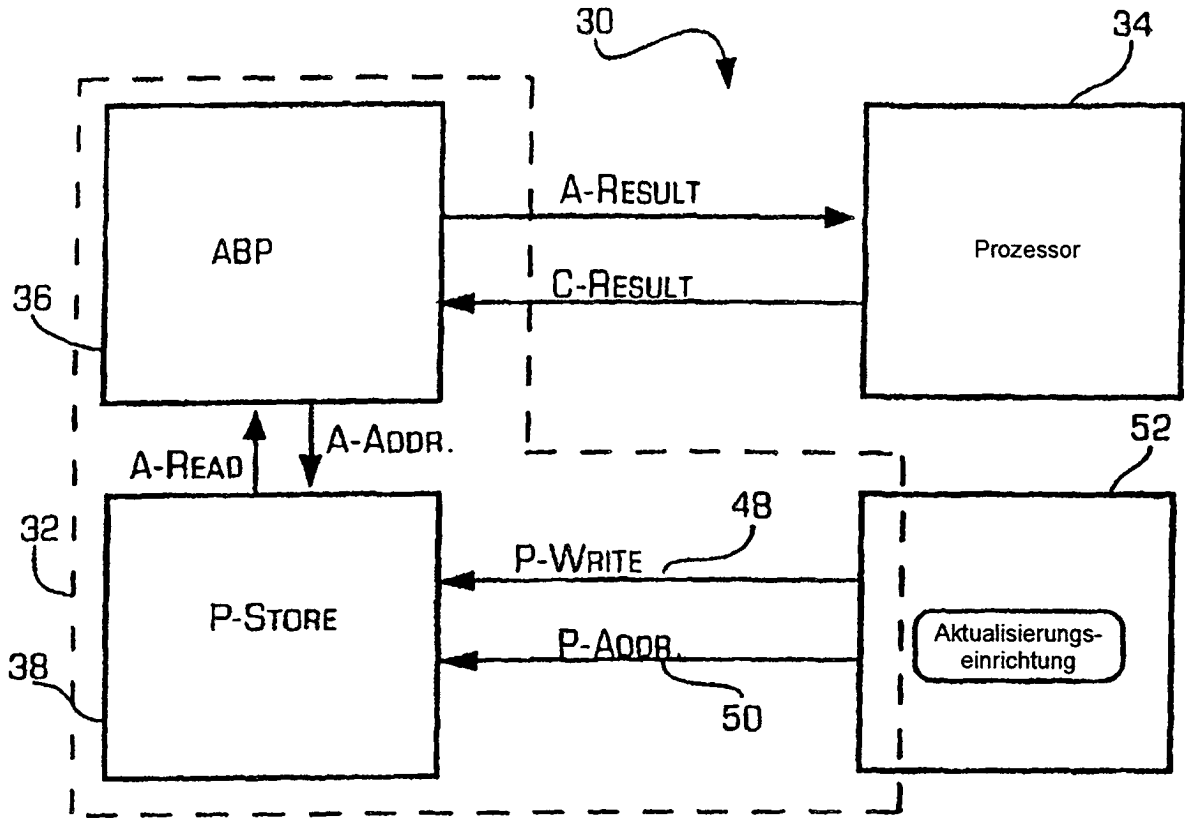


Fig. 3

Verzweigung Nr.	Vorhersage
1	ja
2	nein
3	ja
	•
	•
	•
20	ja

Fig. 4

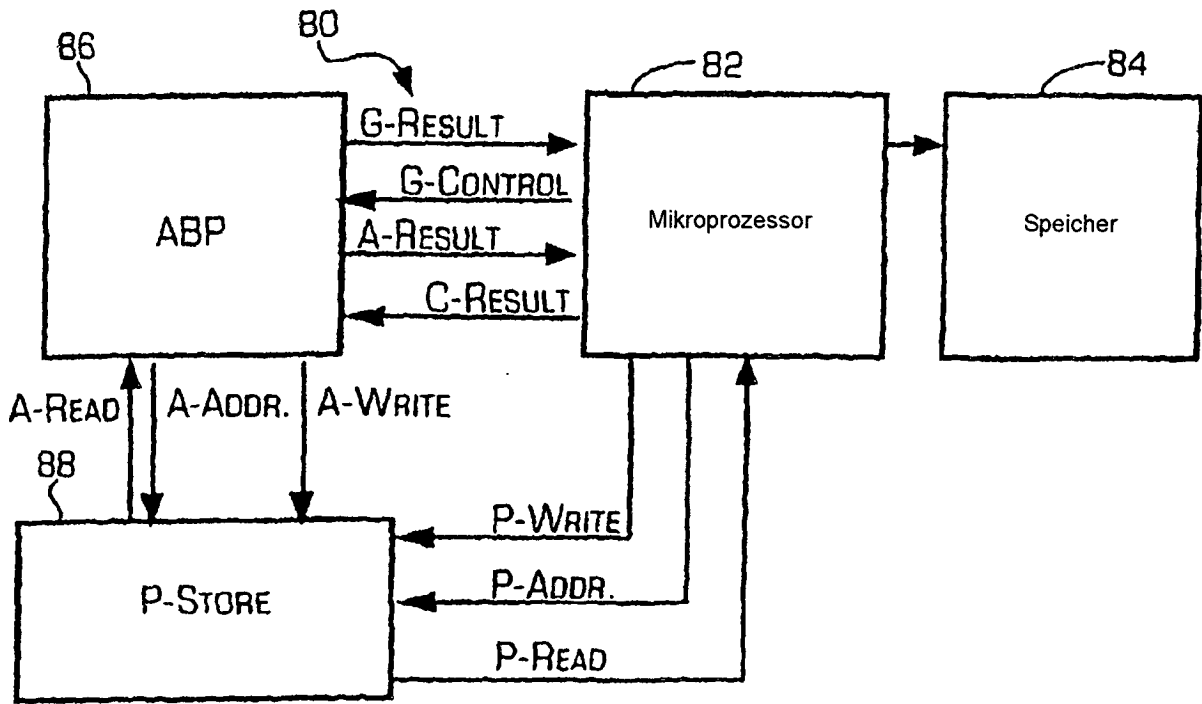


Fig. 5

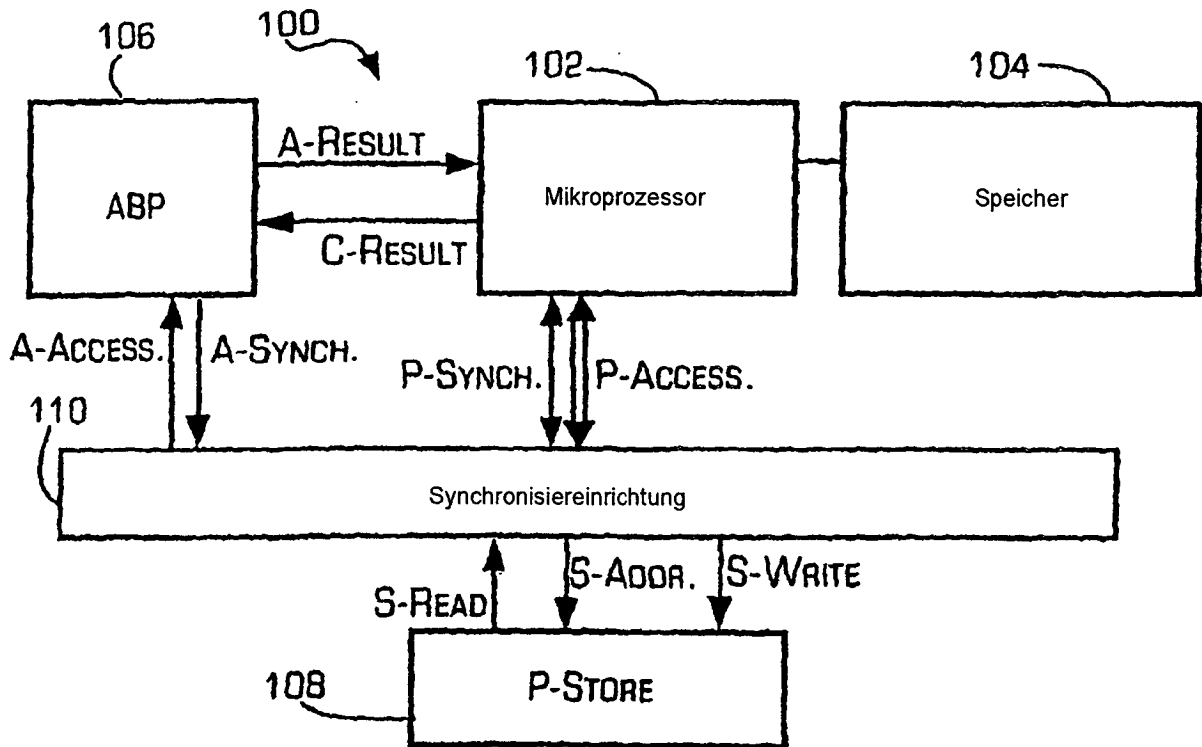


Fig. 6

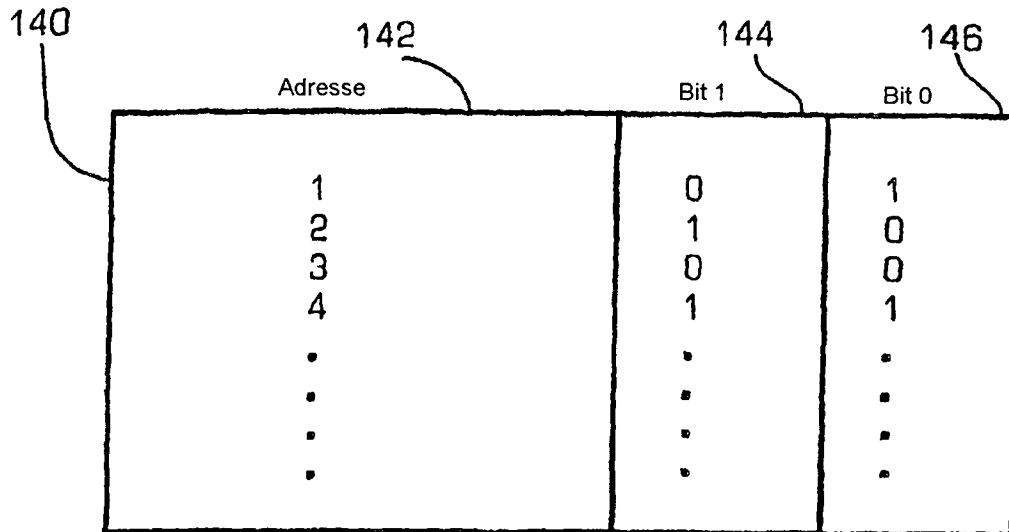


Fig. 7

	<u>genommen</u>	<u>nicht genommen</u>
11	11	10
10	11	01
01	10	00
00	→ 01	00

Fig. 8

	<u>genommen</u>	<u>nicht genommen</u>
11	11	01
10	10	00
01	11	01
00	10	00

Fig. 9

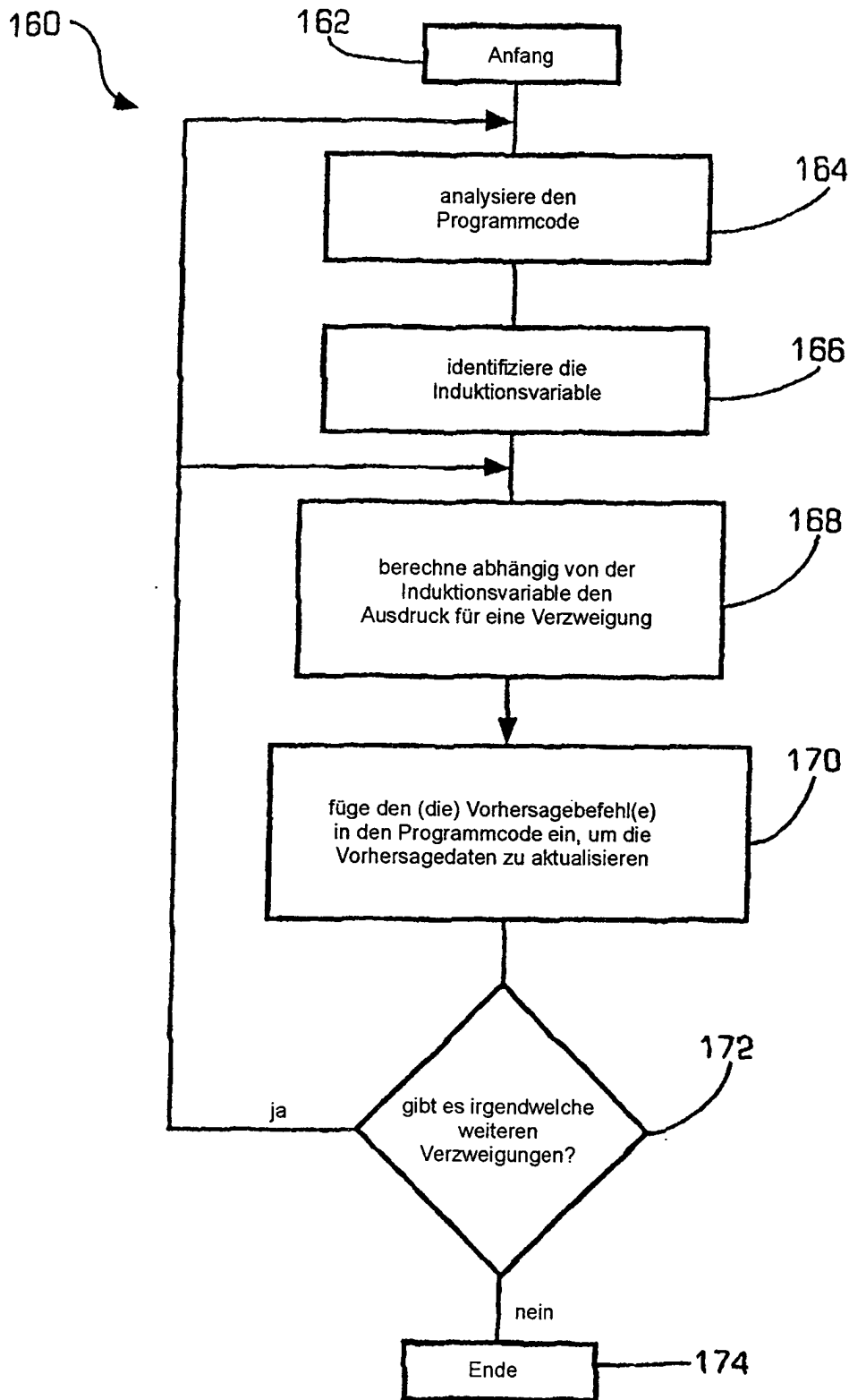


Fig. 10

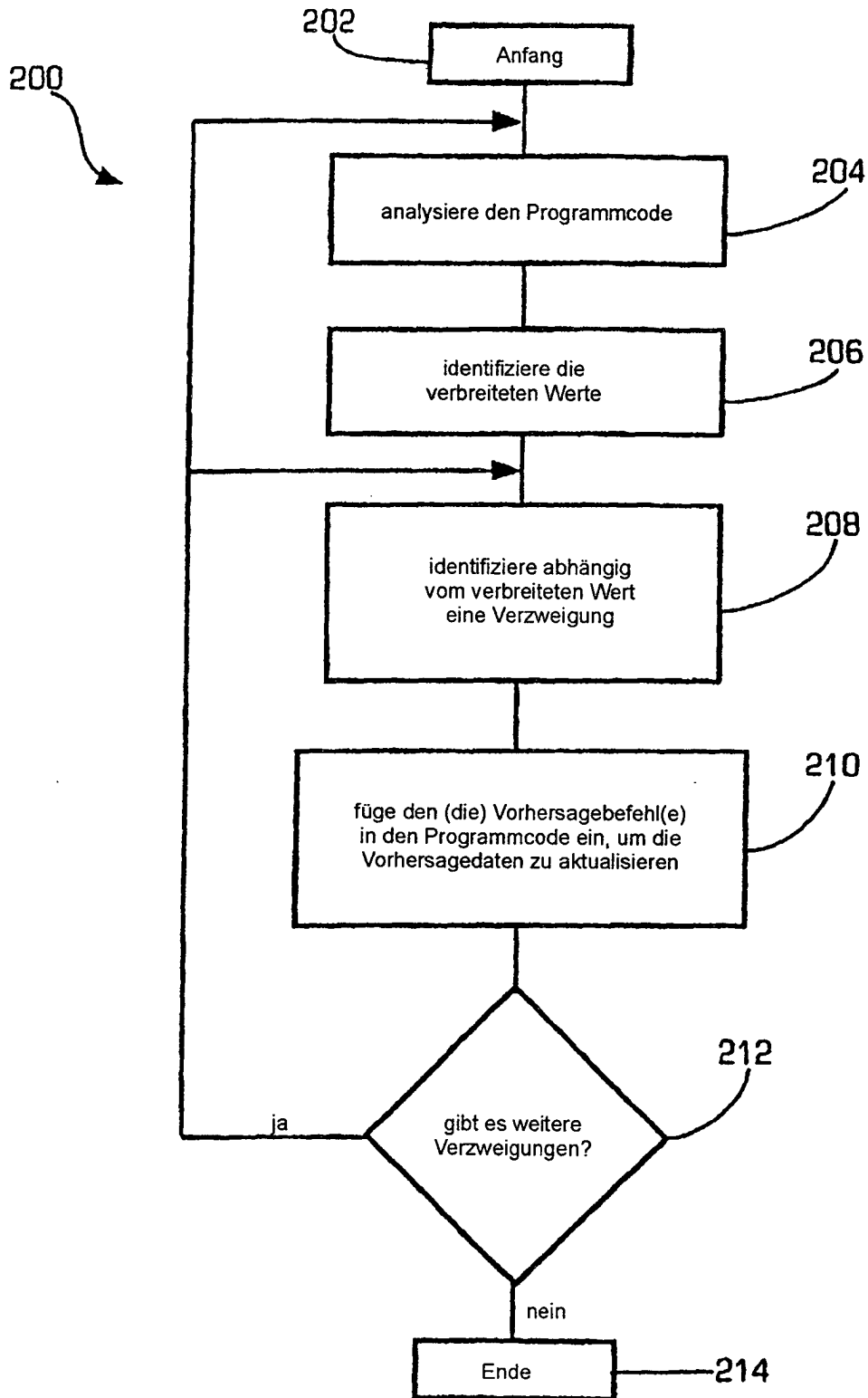


Fig. 11

