



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 696 32 655 T2 2004.09.30**

(12)

Übersetzung der europäischen Patentschrift

(97) **EP 0 730 223 B1**

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

(21) Deutsches Aktenzeichen: **696 32 655.8**

(96) Europäisches Aktenzeichen: **96 103 187.9**

(96) Europäischer Anmeldetag: **01.03.1996**

(97) Erstveröffentlichung durch das EPA: **04.09.1996**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **09.06.2004**

(47) Veröffentlichungstag im Patentblatt: **30.09.2004**

(30) Unionspriorität:

4386095 03.03.1995 JP

(74) Vertreter:

**Grünecker, Kinkeldey, Stockmair &
Schwanhäusser, 80538 München**

(73) Patentinhaber:

**Matsushita Electric Industrial Co., Ltd., Kadoma,
Osaka, JP**

(84) Benannte Vertragsstaaten:

DE, FR, GB

(72) Erfinder:

Nakajima, Masaitsu, Osaka-shi, Osaka 536, JP

(54) Bezeichnung: **Pipeline-Datenverarbeitungsanordnung zur Durchführung einer Mehrzahl von untereinander datenabhängigen Datenprozessen**

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

[0001] Die vorliegende Erfindung bezieht sich auf eine verbesserte Pipeline-Datenverarbeitungsanordnung für ein gleichzeitiges Ausführen einer Mehrzahl von Datenprozessen durch eine Pipeline-Verarbeitung.

[0002] In den vergangenen Jahren haben merkbare Verbesserungen in der Leistung von Mikroprozessoren stattgefunden, wozu verbesserte Taktfrequenzen und eine interne, parallele Verarbeitung stark beigetragen haben. Die früheren, verbesserten Taktfrequenzen können dem Fortschritt der VLSI (Very-Large-Scale Integrated Circuit) Technologie, repräsentiert durch eine Verarbeitungstechnologie und eine Hochgeschwindigkeits-Schaltungstechnologie, zugeschrieben werden, während die letztere, interne, parallele Verarbeitung, hauptsächlich dem Fortschritt einer Architekturtechnologie, repräsentiert durch eine Pipeline-Technologie und eine Sparse-Farb-Technologie, zugeschrieben werden kann.

[0003] In der internen, parallelen Verarbeitung ist die vorstehend erwähnte Pipeline-Technologie eine temporäre, parallele Verarbeitungstechnologie, bei der eine Folge von Datenvorgängen in eine Mehrzahl von Stufen unterteilt wird (im Pipeline-Betrieb verarbeitet), so dass die Mehrzahl von Datenprozessen in einer überlappenden Art und Weise durchgeführt wird. Andererseits ist die verteilte Farb-Technologie (Sparse Color Technology) eine räumliche, parallele Verarbeitungstechnologie, mit der eine Mehrzahl von Datenprozessen gleichzeitig in einer parallelen Weise durchgeführt wird. Unter Verwendung dieser zwei Technologien in Kombination sind die Verbesserungen in der Leistung von Mikroprozessoren erreicht worden.

[0004] Unter Bezugnahme nun auf **Fig. 17** wird eine Beschreibung eines Beispiels der Struktur für ein sowohl räumliches als auch temporäres Durchführen einer Mehrzahl von Datenprozessen in einer parallelen Weise, was durch Anwenden der vorstehenden Sparse-Farb-Technologie auf ein herkömmliches Pipeline-Datenverarbeitungsgerät für ein temporäres Durchführen einer Mehrzahl von Datenprozessen in einer parallelen Weise erhalten wird, angegeben.

[0005] In der Zeichnung sind Pipeline-Datenverarbeitungsschaltungen **150**, **151** und **152**, parallel zueinander angeordnet, dargestellt. Die parallele Anordnung ermöglicht der Pipeline-Datenverarbeitungsschaltung **150**, **151** und **152**, räumlich drei Sätze von Daten zu verarbeiten. Da die Pipeline-Datenverarbeitungsschaltungen **150** bis **152** dieselbe Struktur haben, wird eine Beschreibung nur in Bezug auf die innere Struktur der Pipeline-Datenverarbeitungsschaltung **150** als repräsentativ für die drei Schaltungen angegeben. In der Pipeline-Datenverarbeitungsschaltung **150** werden Daten zu sowohl einem ersten als auch einem zweiten Eingangsanschluss **118** und **119** eingegeben, wobei ein erstes und ein zweites Register **120** und **121** jeweilige Da-

ten, eingegeben zu den vorstehenden Eingangsanschlüssen **118** und **119**, speichern, wobei ein Addierer/Subtrahierer **130** mit 2 Eingängen die jeweiligen Daten, gespeichert in den zwei Registern **120** und **121**, aufnimmt, um so eine Addition oder Subtraktion in Bezug auf die jeweiligen Daten durchzuführen, und wobei ein Register **132** ein Arbeitsergebnis **131** von dem Addierer/Subtrahierer **130** aufnimmt.

[0006] In jeder der vorstehenden Pipeline-Datenverarbeitungsschaltungen **150** bis **152** wird ein Datenprozess in drei Stufen in einem Pipeline-Betrieb durchgeführt, die sind: eine Datenlesestufe; eine Operationsausführungsstufe, bei der die gelesenen Daten einer Addition oder Subtraktion, die tatsächlich durchgeführt ist, unterworfen werden; und eine Datenschreibstufe, bei der ein Operationsergebnis gespeichert wird. In **Fig. 17** dienen Register **140a** bis **140f** für ein vorbereitendes Speichern jeweiliger Daten, um zu dem Addierer/Subtrahierer **130** mit 2 Eingängen jeder der Pipeline-Datenverarbeitungsschaltungen **150** bis **152** zugeführt zu werden und einer Operation, die dadurch durchgeführt wird, unterworfen zu werden, während die Register **140g** bis **140i** zum Speichern von Operationsergebnissen, ausgegeben von den jeweiligen Datenverarbeitungsschaltungen **150** bis **152**, dienen.

[0007] Nachfolgend wird eine Beschreibung in Bezug auf die Operation der vorstehenden, herkömmlichen Pipeline-Datenverarbeitungsvorrichtung, dargestellt in **Fig. 17**, angegeben.

[0008] Zuerst wird eine Betrachtung für den Fall vorgenommen, bei dem drei Typen von Addieroperationen C, D und G durch die erste, die zweite und die dritte Pipeline-Datenverarbeitungsschaltungen **150**, **151** und **152** jeweils ausgeführt werden:

$$C = A + B$$

$$D = E + F$$

$$G = H + I.$$

[0009] Da die vorstehenden drei Operationen keine datenabhängige Beziehung dazwischen haben, können sie vollständig unabhängig voneinander ausgeführt werden. Genauer gesagt kann, wie in dem Operationszeitdiagramm der **Fig. 18(a)** dargestellt ist, das Lesen der Daten A und B, das Lesen der Daten E und F und das Lesen der Daten H und I gleichzeitig in der Datenlesestufe durchgeführt werden. Weiterhin können eine Operation von A + B, eine Operation von E + F und eine Operation von H + I gleichzeitig in der Operationsausführungsstufe ausgeführt werden. Weiterhin kann das Schreiben von Daten C, das Schreiben von Daten D und das Schreiben von Daten G gleichzeitig in der Datenschreibstufe ausgeführt werden. Auf diese Art und Weise können drei Typen von Datenprozessen parallel zueinander durch die drei Pipeline-Datenverarbeitungsschaltungen **150** bis **152** ausgeführt werden.

[0010] Allerdings besitzt das vorstehende, herkömmliche Pipeline-Datenverarbeitungsgerät die folgenden Nachteile, die nachfolgend im Detail beschrieben werden. In dem Fall, bei dem die Ausführungsfolge der folgenden drei Additionsoperationen C, J und G in dieser Reihenfolge bestimmt worden ist, wird angenommen, dass die Addieroperation C durch die erste Pipeline-Datenverarbeitungsschaltung **150** durchgeführt wird, dass die Addieroperation J durch die zweite Pipeline-Datenverarbeitungsschaltung **151** durchgeführt wird und dass die Addieroperation G durch die dritte Pipeline-Datenverarbeitungsschaltung **152** durchgeführt wird:

$$C = A + B$$

$$J = E + C$$

$$G = H + I.$$

[0011] Unter den vorstehenden drei Typen von Operationen haben zwei Operationen C und J eine datenabhängige Beziehung dazwischen. Genauer gesagt können, da die zweite Operation $J = E + C$ unter Verwendung des Ergebnisses der ersten Operation $C = A + B$ durchgeführt werden sollte, die erste Operation $C = A + B$ und die zweite Operation $J = E + C$ nicht gleichzeitig durchgeführt werden. Demzufolge sollte, wie in dem Operationszeitdiagramm der **Fig. 18(b)** dargestellt ist, die zweite Operation $J = E + C$ einen Zyklus nach der Einleitung der ersten Operation $C = A + B$ initiiert werden.

[0012] Die Ausführungsfolge der zweiten Operation J und der dritten Operation G wird entweder durch ein Verfahren „Out-of-Order“ oder ein Verfahren „In-Order“ bestimmt. Das erstere Verfahren ermöglicht, dass die Ausführungssequenz einer Mehrzahl von Datenprozessen geändert wird, während das letztere Verfahren der vorbestimmten Ausführungssequenz einer Mehrzahl von Datenprozessen folgt, und unterbindet, dass die Datenprozesse transponiert werden. In dem Fall, bei dem die Verwendung des Verfahrens „Out-of-Order“ angenommen wird, kann deshalb die dritte Operation $G = H + I$ gleichzeitig mit der ersten Operation C vor der zweiten Operation J durchgeführt werden, wie dies in **Fig. 18(b)** dargestellt ist. In dem Fall, bei dem die Verwendung des Verfahrens „In-Order“ angenommen wird, sollte, andererseits, die dritte Operation G nicht vor der zweiten Operation J durchgeführt werden, sondern sollte gleichzeitig mit dieser durchgeführt werden, d. h. einen Zyklus nach der Ausführung der ersten Operation C, da die Transponierung der Operationen unterbunden wird.

[0013] Wie vorstehend beschrieben ist, kann, in dem herkömmlichen Pipeline-Datenverarbeitungsgerät, die Mehrzahl von Operationen C und J nicht gleichzeitig ausgeführt werden, unabhängig davon, ob das Verfahren „In-Order“ oder das Verfahren „Out-of-Order“ verwendet wird. Als Folge werden, falls eine datenabhängige Beziehung zwischen einer

Mehrzahl von Datenprozessen existiert, Befehle, um jeweilige Operationen C und J durchzuführen, nicht gleichzeitig zu den jeweiligen Pipeline-Datenverarbeitungsschaltungen **150** und **151** ausgegeben, und der Befehl, um die zweite Operation J durchzuführen, wird einen Zyklus nach dem ersten Befehl ausgegeben, um die erste Operation C durchzuführen. In dem Fall, bei dem die Verwendung des Verfahrens „In-Order“ angenommen wird, wird, andererseits, ein Befehl, um die darauffolgende dritte Operation G durchzuführen, die keine datenabhängige Beziehung besitzt, auch einen Zyklus nach dem Befehl, um die erste Operation C durchzuführen, ausgegeben. Demzufolge besitzt das herkömmliche Pipeline-Datenverarbeitungsgerät den Nachteil einer sehr geringen Verarbeitungsgeschwindigkeit, da eine Mehrzahl von Datenprozessen, die eine datenabhängige Beziehung dazwischen haben, nicht unter einer hohen Geschwindigkeit durchgeführt werden kann.

[0014] Die EP 0 312 764 A3 offenbart einen Datenprozessor, der die selektive, gleichzeitige oder asynchrone Ausführung von zueinander unabhängigen Befehlen von unterschiedlichen Klassen in parallel gekoppelten Ausführungseinheiten ermöglicht und der die sequenzielle Ausführung von zueinander abhängigen Befehlen von unterschiedlichen Klassen durch Verzögern der Ausführung eines abhängigen Befehls in einer zweiten Ausführungseinheit, bis zu dem Abschluss einer Ausführung eines Prekursor-Befehls in einer ersten Ausführungseinheit, ermöglicht.

[0015] Es ist deshalb eine Aufgabe der vorliegenden Erfindung, ein Pipeline-Datenverarbeitungsgerät zu schaffen, bei dem, gerade wenn eine datenabhängige Beziehung zwischen einer Mehrzahl von Datenprozessen existiert, eine Mehrzahl von Befehlen, um die Prozesse durchzuführen, gleichzeitig zu jeweiligen Pipeline-Datenverarbeitungsschaltungen ausgegeben werden kann, so dass die Mehrzahl von Datenprozessen entsprechend ausgeführt werden kann, unabhängig davon, ob das Verfahren „In-Order“ oder das Verfahren „Out-of-Order“ angewandt worden ist. Weiterhin ermöglicht, gerade wenn ein Datenprozess, der keine datenabhängige Beziehung besitzt, den Datenprozessen folgt, die eine datenabhängige Beziehung dazwischen haben, das Pipeline-Datenverarbeitungsgerät einen Befehl, um den darauffolgenden Datenprozess durchzuführen, um gleichzeitig mit den Befehlen ausgegeben zu werden, um die vorhergehenden Datenprozesse durchzuführen, die die datenabhängige Beziehung dazwischen haben, um dadurch die Leistung der Verarbeitungsgeschwindigkeit zu verbessern.

[0016] Um die vorstehende Aufgabe zu lösen, hat die vorliegende Erfindung, wie sie in den beigefügten Ansprüchen angegeben ist, eine Struktur, versehen mit einer Mehrzahl von Stufen, erhalten durch Hinzufügen einer spezifizierten Anzahl von Stufen zu den drei Stufen, die eine Datenlesestufe, eine Operationsausführungsstufe und eine Datenschreibstufe

sind, angewandt, so dass sich die vorstehende Datenlesestufe und die Operationsausführungsstufe, falls notwendig, zu einer wahlweisen Stufe in der Mehrzahl der Stufen verschieben können.

[0017] Genauer gesagt weist ein Pipeline-Datenverarbeitungsverfahren der vorliegenden Erfindung, mit dem eine Mehrzahl von Befehlen, um jeweilige Datenprozesse durchzuführen, durch eine Mehrzahl von Pipeline-Datenverarbeitungsschaltungen ausgeführt wird, die Schritte auf: gleichzeitiges Ausgeben, wenn mindestens zwei Datenprozesse eine datenabhängige Beziehung dazwischen haben, von Befehlen, um die jeweiligen Datenprozesse zu den jeweiligen Pipeline-Datenverarbeitungsschaltungen hin durchzuführen; dann Ausführen zuerst des vorangehenden einen der vorstehenden Datenprozesse, in Bezug auf den der darauffolgende eine der vorstehenden Datenprozesse datenabhängig ist, durch die Pipeline-Datenverarbeitungsschaltung für den vorstehenden, vorangehenden Datenprozess; und darauffolgendes Ausführen, zu einem Zeitpunkt einen Zyklus nach der Ausführung des vorstehenden, vorangehenden Datenprozesses, in Bezug auf den der vorstehende, darauffolgende Datenprozess datenabhängig ist, des darauffolgenden Datenprozesses durch die Pipeline-Verarbeitungsschaltung für den vorstehenden, darauffolgenden Datenprozess.

[0018] Ein Pipeline-Datenverarbeitungsgerät der vorliegenden Erfindung weist auf: eine Mehrzahl von Pipeline-Verarbeitungsschaltungen; und einen Befehl-Ausgabe-Steuerbereich zum gleichzeitigen Ausgeben einer Mehrzahl von Befehlen, um jeweilige Datenprozesse zu den vorstehenden, jeweiligen Pipeline-Verarbeitungsschaltungen durchzuführen, wobei jede der vorstehenden Pipeline-Verarbeitungsschaltungen aufweist: ein Eingangsdatenregister zum Aufnehmen und Speichern von Daten von der Außenseite; eine Datenoperationsschaltung zum Aufnehmen der Daten, gespeichert in dem vorstehenden Eingangsdatenregister und zum Durchführen einer spezifizierten Operation in Bezug auf die Eingangsdaten; ein Zwischen-Pipeline-Register, angeordnet in einer Stufe, die der vorstehenden Datenoperationsschaltung folgt; ein End-Pipeline-Register, angeordnet in einer Stufe, die dem vorstehenden Zwischen-Pipeline-Register folgt; und eine Pfadumschalterschaltung, um zu bewirken, dass ein Operationsergebnis von der vorstehenden Datenoperationsschaltung durch das vorstehende Zwischen-Pipeline-Register hindurchführt oder im Bypass vorbeiführt, und zum Eingeben des Operationsergebnisses zu dem vorstehenden End-Pipeline-Register, wobei der vorstehende Befehl-Ausgabe-Steuerbereich die Anzahl von Dummy-Stufen, die eingesetzt werden sollen, hinzufügt, wobei in keiner etwas ausgeführt wird, und zwar zu den vorstehenden Befehlen, die gleichzeitig ausgegeben sind, um dadurch eine Operationsausführungsstufe zu ermöglichen, in der die vorstehende Datenoperationsschaltung die vorstehende, spezifizierte Operation durchführt, um eine wahlweise Stufe

in eine Mehrzahl von Pipelinestufen zu verschieben. [0019] Mit der vorstehenden Struktur wird eine Mehrzahl von (z. B. drei) Pipeline-Datenverarbeitungsschaltungen in einer parallelen Weise gemäß der vorliegenden Erfindung angeordnet. In dem Fall, bei dem eine datenabhängige Beziehung zwischen der Mehrzahl von (drei) Datenprozessen existiert (z. B. wenn eine Operation $C = A + B$, eine Operation $J = E + C$ und eine Operation $K = H + I$ vorliegt), wird eine Mehrzahl von Befehlen, um die Mehrzahl von Datenprozessen durchzuführen, gleichzeitig zu den vorstehenden, jeweiligen Pipeline-Datenverarbeitungsschaltungen ausgegeben. In der Pipeline-Datenverarbeitungsschaltung zum Durchführen des ersten Datenprozesses C werden eine Datenlesestufe und eine Operationsausführungsstufe zu der ersten und der zweiten Stufe jeweils zugeordnet. In der Pipeline-Datenverarbeitungsschaltung zum Durchführen des zweiten Datenprozesses J werden die Datenlesestufe und die Operationsausführungsstufe der zweiten und der dritten Stufe zugeordnet. In der Pipeline-Datenverarbeitungsschaltung zum Durchführen des dritten Datenprozesses K werden die Datenlesestufe und die Operationsausführungsstufe der dritten und der vierten Stufe zugeordnet. Demzufolge wird die Mehrzahl von Datenprozessen C, J und K sequenziell ausgeführt, so dass, gerade dann, wenn eine datenabhängige Beziehung zwischen diesen Datenprozessen besteht, die Datenprozesse entsprechend ausgeführt werden.

[0020] Hierbei werden die jeweiligen Befehle, um die vorstehende Mehrzahl von Datenprozessen C, J und K durchzuführen, gleichzeitig ausgegeben, und der darauffolgende Befehl (d. h. der Befehl, um den vierten Datenprozess durchzuführen) wird in dem zweiten Zyklus ausgegeben. In der herkömmlichen Ausführungsform wird der vierte Befehl in dem vierten Zyklus ausgegeben. Demzufolge kann, gemäß der vorliegenden Erfindung, eine große Anzahl von aufeinanderfolgenden Befehlen ohne eine Stagnation ausgegeben werden, so dass die Mehrzahl von Datenprozessen, die eine datenabhängige Beziehung dazwischen haben, unter einer hohen Geschwindigkeit ausgeführt werden kann.

[0021] Die vorstehenden Aufgaben und neuen Merkmale der vorliegenden Erfindung werden unter Lesen der nachfolgenden Beschreibung in Verbindung mit den beigefügten Zeichnungen ersichtlicher werden.

[0022] Die beigefügten Zeichnungen stellen die bevorzugten Ausführungsformen der vorliegenden Erfindung dar.

[0023] In den Zeichnungen:

[0024] **Fig. 1** stellt die gesamte Struktur eines Prozessors dar;

[0025] **Fig. 2** stellt die Operation einer Pipeline-Operation dar;

[0026] **Fig. 3** stellt die Gesamtstruktur eines Pipeline-Datenverarbeitungsgeräts gemäß einer ersten Ausführungsform der vorliegenden Erfindung dar;

[0027] **Fig. 4** stellt die innere Struktur einer Pipeline-Datenverarbeitungsschaltung dar, die das Pipeline-Datenverarbeitungsgerät bildet;

[0028] **Fig. 5** stellt eine erste Variation der Pipeline-Datenverarbeitungsschaltung dar;

[0029] **Fig. 6** stellt eine zweite Variation der Pipeline-Datenverarbeitungsschaltung dar;

[0030] **Fig. 7** stellt eine dritte Variation der Pipeline-Datenverarbeitungsschaltung dar;

[0031] **Fig. 8** zeigt ein Zeitdiagramm, das eine Basisoperation der Pipeline-Datenverarbeitungsschaltung darstellt;

[0032] **Fig. 9** zeigt ein Zeitdiagramm, das eine andere Basisoperation der Pipeline-Datenverarbeitungsschaltung darstellt;

[0033] **Fig. 10** zeigt ein Zeitdiagramm, das eine noch andere Basisoperation der Pipeline-Datenverarbeitungsschaltung darstellt;

[0034] **Fig. 11** zeigt ein operationsmäßiges Zeitdiagramm in dem Fall, bei dem drei Datenprozesse durch ein Pipeline-Datenverarbeitungsgerät gemäß der vorliegenden Erfindung durchgeführt wird;

[0035] **Fig. 12** zeigt ein operationsmäßiges Zeitdiagramm in dem Fall, bei dem eine große Anzahl von Datenprozessen durch ein herkömmliches Pipeline-Datenverarbeitungsgerät durchgeführt wird;

[0036] **Fig. 13** zeigt ein operationsmäßiges Zeitdiagramm in dem Fall, bei dem eine Mehrzahl von Datenprozessen durch das herkömmliche Pipeline-Datenverarbeitungsgerät durchgeführt wird;

[0037] **Fig. 14** zeigt die Gesamtstruktur eines Pipeline-Datenverarbeitungsgeräts gemäß einer zweiten Ausführungsform der vorliegenden Erfindung;

[0038] **Fig. 15** stellt die Struktur eines grundsätzlichen Bereichs eines Prozessors, umfassend das Pipeline-Datenverarbeitungsgerät, gemäß der zweiten Ausführungsform der vorliegenden Erfindung, dar;

[0039] **Fig. 16** zeigt ein operationsmäßiges Zeitdiagramm in dem Fall, bei dem sechs Datenprozesse durch das Pipeline-Datenverarbeitungsgerät gemäß der zweiten Ausführungsform der vorliegenden Erfindung durchgeführt werden;

[0040] **Fig. 17** stellt die Gesamtstruktur eines herkömmlichen Pipeline-Datenverarbeitungsgeräts dar;

[0041] **Fig. 18(a)** zeigt ein operationsmäßiges Zeitdiagramm, das eine Basisoperation des herkömmlichen Pipeline-Datenverarbeitungsgeräts darstellt;

[0042] **Fig. 18(b)** zeigt ein operationsmäßiges Zeitdiagramm, das ein anderes Basisbeispiel des herkömmlichen Pipeline-Datenverarbeitungsgeräts darstellt; und

[0043] **Fig. 18(c)** zeigt ein operationsmäßiges Zeitdiagramm, das eine noch andere Basisoperation des herkömmlichen Pipeline-Datenverarbeitungsgeräts darstellt.

[0044] Nachfolgend werden die bevorzugten Ausführungsformen der vorliegenden Erfindung unter Bezugnahme auf die Zeichnungen beschrieben.

(Erste Ausführungsform eines Pipeline-Datenverarbeitungsgeräts)

[0045] **Fig. 1** stellt eine erste Ausführungsform der vorliegenden Erfindung dar, in der dargestellt sind: ein Prozessor **1**; ein Befehl-Speicher **2a** zum vorbereitenden Speichern eines Befehl-Programms; und einen Datenspeicher **2b** zum vorbereitenden Speichern einer großen Anzahl von Datensätzen. Es ist nicht notwendigerweise erforderlich, den vorstehenden Befehl-Speicher **2a** und den Datenspeicher **2b** als separate Komponenten vorzusehen. Eine große Anzahl von Befehlen und Datensätzen kann zusammen in einem einzelnen Speicher gespeichert werden.

[0046] Der vorstehende Prozessor **1** besteht aus einem Befehl-Steuerabschnitt **3** und einem Befehl-Ausführungsabschnitt **4** zum Ausführen eines solchen Befehls, um die Verarbeitung von Daten durchzuführen. Der vorstehende Befehl-Steuerabschnitt **3** besteht aus einem Befehl-Abrufbereich **5**, einem Befehl-Register **6** und einem Befehl-Decodierabschnitt **7**, und einem Befehl-Ausgabesteuerabschnitt **8**. Der vorstehende Befehl-Abrufbereich **5** erzeugt eine oder eine Mehrzahl von Befehl-Adresse(n) und überträgt sie zu dem vorstehenden Befehl-Speicher **2a**, während ein Abrufen eines Befehls oder einer Mehrzahl von Befehlen notwendig ist. Die vorstehenden, abgerufenen Befehle werden durch den vorstehenden Befehl-Abrufabschnitt **5** in dem Befehl-Register **6** gespeichert. Der vorstehende Befehl-Decodierabschnitt **7** decodiert jede spezifizierte Anzahl (z. B. drei) von Befehlen, gespeichert in dem Befehl-Register **6**. Der vorstehende Befehl-Ausgabesteuerabschnitt **8** gibt einen Befehl zu der Ressource entsprechend zu dem Befehl basierend auf dem Ergebnis eines Decodierens aus. Zum Beispiel gibt, falls der Befehl die Verarbeitung von Daten anweist, der Befehl-Ausgabesteuerabschnitt **8** ihn zu dem vorstehenden Befehl-Ausführungsabschnitt **4** aus. Falls der Befehl ein Verzweigungsbefehl ist, um die Datenfolge des Befehl-Programms zu steuern, gibt der Befehl-Ausgabesteuerabschnitt **8** ihn zu dem Befehl-Abrufabschnitt **5** aus. Weiterhin prüft, falls die Operation des vorstehenden Befehl-Ausgabesteuerabschnitts **8** grob beschrieben werden soll, er, wenn der Befehl zu dem Befehl-Ausführungsabschnitt **4** ausgegeben werden soll, den Zustand der Ressource, die zum Ausführen des Befehls in dem Befehl-Ausführungsabschnitt **4** erforderlich ist, und das Vorhandensein oder das Nichtvorhandensein einer datenabhängigen Beziehung zwischen einer Mehrzahl von Befehl-Datensätzen, und gibt nur einen ausführbaren Befehl zu dem Befehl-Ausführungsabschnitt **4** aus. Die innere Struktur und die Betriebsweise des vorstehenden Befehl-Ausgabesteuerabschnitts **8** werden später im Detail beschrieben.

[0047] Der vorstehende Befehl-Ausführungsabschnitt **4** besteht aus einer Registerdatei **9** für ein temporäres Speichern von Daten, einem Ausfüh-

rungsverarbeitungsabschnitt **10** zum Durchführen einer Operation in Bezug auf Daten und einem Ausführungssteuerabschnitt **11**. Der vorstehende Ausführungssteuerabschnitt **11** lädt Daten ein, die durch den Ausführungsverarbeitungsabschnitt **10** erforderlich sind, um eine Operation in Bezug auf Daten von dem vorstehenden Datenspeicher **2b** durchzuführen, während die Daten in der Registerdatei **9** gespeichert sind. In der tatsächlichen Ausführung der Operation in dem Ausführungsverarbeitungsabschnitt **10** liest der vorstehende Ausführungssteuerabschnitt **11** Daten, gespeichert in der vorstehenden Registerdatei **9**, und steuert den Ausführungsverarbeitungsabschnitt **10** so, dass die Operation entsprechend dem Befehl dadurch ausgeführt wird, und speichert das Operationsergebnis in der Registerdatei **9**.

[0048] In dem vorstehenden Prozessor **1** wird die Operation in Bezug auf Daten, durchgeführt durch die Pipeline-Verarbeitung, durchgeführt. Wie in **Fig. 2** dargestellt ist, besteht die Pipeline-Verarbeitung aus fünf Stufen. In der ersten Stufe wird ein Befehl abgerufen. In der zweiten Stufe wird der Befehl decodiert und ausgegeben. In der dritten Stufe werden Daten von der Registerdatei **9** gelesen. In der vierten Stufe wird eine Operation in Bezug auf Daten ausgeführt. In der fünften Stufe wird das Operationsergebnis in die Registerdatei **9** hineingeschrieben. Die vorstehende zweite und dritte Stufe können durch eine einzelne Stufe ausgeführt werden.

[0049] **Fig. 3** stellt eine spezifische Struktur des vorstehenden Ausführungsverarbeitungsabschnitts **10** als ein Pipeline-Datenverarbeitungsgerät der vorliegenden Erfindung dar. In der Zeichnung bezeichnen die Bezugszeichen **100**, **101** und **102** Pipeline-Datenverarbeitungs-schaltungen, angeordnet parallel zueinander. Da die Pipeline-Datenverarbeitungs-schaltungen **100** bis **102** dieselbe Struktur haben, wird eine Beschreibung in Bezug auf die innere Struktur der Pipeline-Datenverarbeitungs-schaltung **100** als repräsentativ für die drei Schaltungen vorgenommen. In der Zeichnung sind dargestellt: erste und zweite Dateneingangsanschlüsse **18** und **19** zum Aufnehmen von Daten von der vorstehenden Registerdatei **9**; erste und zweite Dateneingangsregister **20** und **21** zum Speichern jeweiliger Sätze von Daten, empfangen von den vorstehenden, jeweiligen Dateneingangsanschlüssen **18** und **19**; und ein Kalkulator mit 2 Eingängen (Datenberechnungsschaltung) zum Aufnehmen der jeweiligen Sätze von Daten, gespeichert in den vorstehenden zwei Dateneingangsregistern **20** und **21** und zum Ausführen einer Addition oder Subtraktion in Bezug auf die jeweiligen Sätze von Daten.

[0050] In der Zeichnung sind auch dargestellt: ein erstes (Zwischen-) Pipeline-Register **40**, angeordnet in einer Stufe, die dem vorstehenden Kalkulator **30** folgt; ein zweites (Zwischen-) Pipeline-Register **60**, angeordnet in einer Stufe, die dem vorstehenden ersten Pipeline-Register **40** folgt; und ein Selektor bzw. eine Auswahleinrichtung (Pfadumschalt-schaltung)

70, angeordnet in einer Stufe, die dem zweiten Pipeline-Register **60** folgt. Das vorstehende erste Pipeline-Register **40** nimmt ein Datenoperationsergebnis **31**, ausgegeben von dem vorstehenden Kalkulator **30**, auf und speichert es. Das vorstehende, zweite Pipeline-Register **60** nimmt die Ausgangsdaten von dem vorstehenden ersten Pipeline-Register **40** auf und speichert sie. Der vorstehende Selektor **70** nimmt das Datenoperationsergebnis **31**, ausgegeben von dem vorstehenden Kalkulator **30**, und jeweilige Sätze von Daten, gespeichert in dem ersten und dem zweiten Pipeline-Register **40** und **60**, auf, wählt einen der drei aus und gibt den ausgewählten einen aus. In der Zeichnung bezeichnet ein Bezugszeichen **81** ein End-Pipeline-Register zum Aufnehmen der Ausgangsdaten **80** von dem vorstehenden Selektor **70** und speichert sie.

[0051] In jeder der vorstehenden Pipeline-Datenverarbeitungs-schaltungen **100** bis **102** werden die vorstehenden Register **20**, **21**, **40** und **60** und der Selektor **70**, der darin vorgesehen ist, durch den vorstehenden Ausführungssteuerabschnitt **11** der **Fig. 1** gesteuert (**Fig. 4** stellt nur die Pipeline-Datenverarbeitungs-schaltung **100** dar). Die Bezugszeichen **9a** bis **9i** bezeichnen Register in der vorstehenden Registerdatei **9**.

[0052] Die vorstehende Pipeline-Datenverarbeitungs-schaltung **100** kann auch aus irgendeiner der Pipeline-Datenverarbeitungs-schaltungen, dargestellt in den **Fig. 5** bis **7**, aufgebaut sein. Die Pipeline-Datenverarbeitungs-schaltung **120**, dargestellt in **Fig. 5**, wird durch Erhöhen der Anzahl der Pipeline-Stufen der Pipeline-Datenverarbeitungs-schaltung **100** der **Fig. 4** um eins erhalten. Genauer gesagt werden, in **Fig. 4**, ein zusätzliches Pipeline-Register **90** und ein zusätzlicher Selektor (Pfadumschalt-schaltung) **91** zwischen dem Selektor **70** und dem End-Pipeline-Register **81** vorgesehen. Das vorstehende, zusätzliche Pipeline-Register **90** nimmt das Ergebnis einer Auswahl von dem Selektor **70** auf und speichert es. Der vorstehende, zusätzliche Selektor **91** nimmt das Operationsergebnis **31** von dem Kalkulator **30** und die jeweiligen Sätze von Daten, gespeichert in dem Zwischenersten und zweiten) Pipeline-Register **40** und **60** und in dem zusätzlichen Pipeline-Register **90**, auf, wählt irgendeines der drei aus und gibt den ausgewählten einen aus. Der Ausgang **92** von dem zweiten Selektor **90** wird zu dem End-Pipeline-Register **81** ausgegeben und darin gespeichert. Die Pipeline-Datenverarbeitungs-schaltung **130** der **Fig. 6** wird durch Vorsehen eines Pipeline-Registers **30a** in dem Kalkulator **30'** zum Ausführen einer Addition oder Subtraktion in einer Pipeline mit 2 Stufen erhalten. In der Pipeline-Datenverarbeitungs-schaltung **140** der **Fig. 7** ist ein erster Selektor **52** zwischen dem ersten und dem zweiten Pipeline-Register **40** und **60** zwischengefügt und ein zweiter Selektor **72** ist zwischen dem zweiten und dem End-Pipeline-Register **60** und **81** zwischengefügt. Der vorstehende erste Selektor **52** nimmt das Opera-

tionsergebnis von dem Kalkulator **30** und die Daten, gespeichert in dem Pipeline-Register **40** der vorherigen Stufe (erste) auf, wählt irgendeines der zwei aus und gibt das ausgewählte eine aus. Der vorstehende zweite Selektor **72** nimmt das Ergebnis einer Auswahl von dem vorstehenden, ersten Selektor **52** und die Daten, gespeichert in dem Pipeline-Register **60** der vorherigen Stufe (zweite), auf, wählt irgendeinen der zwei aus und gibt es zu dem End-Pipeline-Register **81** aus. Der Kalkulator **30** in jeder der vorstehenden Pipeline-Datenverarbeitungsschaltungen **120** und **140** der **Fig. 5** und **7** kann aus dem Kalkulator **30'** in der Pipeline-Datenverarbeitungsschaltung **130** der **Fig. 6** aufgebaut sein. Um die drei Pipeline-Datenverarbeitungsschaltungen **100** bis **102**, dargestellt in **Fig. 1**, aufzubauen, können irgendeine oder einige der Verarbeitungsschaltungen **120** bis **140**, die vorstehend beschrieben sind, einzeln oder in Kombination verwendet werden. Allerdings sollten, in dem Fall, bei dem einige der Verarbeitungsschaltungen **120** bis **140** in Kombination verwendet werden, um die Pipeline-Datenverarbeitungsschaltungen **100** bis **102** aufzubauen, deren Pipelines dieselbe Zahl von Stufen haben.

[0053] Als nächstes wird eine Beschreibung der Ausführung einer Operation der einzelnen Pipeline-Datenverarbeitungsschaltung (100 zum Beispiel) vor einer Beschreibung angegeben, die in Bezug auf das Pipeline-Datenverarbeitungsgerät, dargestellt in **Fig. 3**, angegeben wird, wo eine Mehrzahl von Operationen parallel durchgeführt wird. Da die Pipeline-Datenverarbeitungsschaltungen, dargestellt in **Fig. 4** bis **Fig. 7**, dieselbe Basisoperation zeigen, wird eine Beschreibung nachfolgend in Bezug auf die Pipeline-Datenverarbeitungsschaltung **100** der **Fig. 4** als repräsentativ angegeben.

[0054] Die Pipeline-Datenverarbeitungsschaltung **100** ermöglicht die Ausführung einer Datenverarbeitung mit drei Zeitabstimmungen, dargestellt in den **Fig. 8, 9** und **10**. In den **Fig. 8** bis **10** werden jeweilige Zyklen zum Darstellen der Zeitabstimmungen für die Pipeline-Verarbeitung als erster bis fünfter Zyklus bezeichnet, während jeweilige Stufen in einer Hardware zum Darstellen eines Datenflusses in der Pipeline-Datenverarbeitungsschaltung **100** als eine Datenlesestufe, eine Operationsausführungsstufe, eine Durchgangs-(1)-Stufe (Stufe für den ersten Durchgang), eine Durchgangs-(2)-Stufe (Stufe für den zweiten Durchgang) und eine Datenschreibstufe für die Vereinfachung der Darstellung bezeichnet. Ein Datenprozess wird grundsätzlich in fünf Pipeline-Stufen ausgeführt. In den **Fig. 8** bis **10** ist die Zeitabstimmung für die Pipeline-Verarbeitung einem tatsächlichen Datenfluss in der Pipeline-Datenverarbeitungsschaltung zugeordnet.

[0055] **Fig. 8** stellt die Basisoperation, die am üblichsten ist, der Pipeline-Datenverarbeitungsschaltung **100** dar. Die Datenverarbeitung (z. B. eine Operation $C = A + B$) wird wie folgt ausgeführt: der Selektor **70** wählt den Ausgang von dem zweiten Pipe-

line-Register **60** aus. In dem ersten Zyklus werden Daten A und B gelesen und in den jeweiligen Eingangsregistern **20** und **21** in der Datenlesestufe gespeichert. In dem zweiten Zyklus wird die Operation $A + B$ in der Operationsausführungsstufe ausgeführt und das Operationsergebnis C wird in dem ersten Pipeline-Register **40** gespeichert. In dem dritten Zyklus führen Daten (Operationsergebnis C) durch die Durchgangs-(1)-Stufe hindurch und werden, so wie sie sind, in dem zweiten Pipeline-Register **60** gespeichert. In dem vierten Zyklus führen Daten (Operationsergebnis C) durch die Durchgangs-(2)-Stufe hindurch und werden, so wie sie sind, in dem dritten Pipeline-Register **80** gespeichert. Demzufolge wird, in den zwei Durchgangsstufen, keine Verarbeitung in Bezug auf die Daten (Operationsergebnis C) durchgeführt, und die Daten verschieben sich einfach zu den darauffolgenden Stufen in Folge. In dem fünften Zyklus wird das Operationsergebnis C in die Datenschreibstufe hinein geschrieben. In **Fig. 8** fließen die Daten in der Pipeline-Datenverarbeitungsschaltung **100** ohne Stagnation.

[0056] In **Fig. 9** wird der erste Zyklus der Fünf-Stufen-Pipeline als ein Dummy-Zyklus verwendet. Der Selektor **70** wählt den Ausgang von dem ersten Pipeline-Register **40** aus. In dem ersten Zyklus (Dummy-Zyklus) wird nichts ausgeführt, so dass sich die Pipeline in einem Haltezustand befindet. Als Folge verschieben sich, gerade dann, wenn die Zeitabstimmung zu dem darauffolgenden, zweiten Zyklus fortschreitet, die Daten tatsächlich nicht von der Datenlesestufe zu der darauffolgenden Operationsausführungsstufe. Es tritt nicht auf, dass, bis zu dem zweiten Zyklus, die Daten A und B in der Datenlesestufe gelesen werden und in den jeweiligen Registern **20** und **21** gespeichert werden. In dem dritten Zyklus wird die Operation $A + B$ in der Operationsausführungsstufe ausgeführt und das Operationsergebnis C wird in dem ersten Pipeline-Register **40** gespeichert. In dem vierten Zyklus führt das Operationsergebnis C durch die Durchgangs-(1)-Stufe hindurch und wird in dem dritten Pipeline-Register **80** über den Selektor **70** gespeichert. In dem fünften Zyklus wird das Operationsergebnis C in die Datenschreibstufe hineingeschrieben. Kurz gesagt werden die Daten von der Durchgangs-(1)-Stufe (d. h. das Operationsergebnis C, gespeichert in dem ersten Pipeline-Register **40**) im Bypass an der Durchgangs-(2)-Stufe vorbeigeführt und verschieben sich zu der Datenschreibstufe.

[0057] Hier sollte angemerkt werden, dass, wenn der erste Zyklus ein Dummy-Zyklus ist, die Daten J und K für den darauffolgenden Datenprozess (z. B. $L = J + K$) nicht in den zweiten Zyklus eingegeben werden können. In diesem Fall können, da die Datenlesestufe nur in dem zweiten Zyklus erreicht wird, die darauffolgenden Daten J und K nur dann eingegeben werden, wenn die Daten A und B gleichzeitig Verschiebungen zu der Operationsausführungsstufe, d. h. der dritte Zyklus, ausführten.

[0058] **Fig. 10** stellt die Operation dar, wenn Dum-

my-Zyklen dem ersten und dem zweiten Zyklus zugeordnet werden. Der Selektor **70** wählt den Ausgang von dem Kalkulator **30** aus. In dem ersten und dem zweiten Zyklus (Dummy-Zyklen) wird nichts ausgeführt, so dass sich die Pipeline in einem Haltezustand befindet. Als Folge werden, gerade wenn die Zeitabstimmung zu dem darauffolgenden, dritten Zyklus fortschreitet, die Daten A und B nicht tatsächlich von der Datenlesestufe zu der Operationsausführungsstufe verschoben. Es erfolgt nichts, bis in dem dritten Zyklus die Daten A und B gelesen werden und in den jeweiligen Registern **20** und **21** in der Datenlesestufe gespeichert werden. In dem vierten Zyklus wird die Operation $A + B$ in der Operationsausführungsstufe ausgeführt und das Operationsergebnis C wird in dem dritten Pipeline-Register **80** über den Selektor **70** gespeichert. In dem fünften Zyklus wird das vorstehende Operationsergebnis C in die Datenschreibstufe geschrieben. Kurz gesagt führen die Daten von der Operationsausführungsstufe (Operationsergebnis C) im Bypass an der Durchgangs-(1)- und -(2)-Stufe vorbei und verschieben sich zu der Datenschreibstufe.

[0059] Hier sollte angemerkt werden, dass, wenn der erste und der zweite Zyklus Dummy-Zyklen sind, die Daten für den darauffolgenden Datenprozess nicht eingegeben werden können, bis der dritte Zyklus eingeleitet wird. In diesem Fall können, da der Datenlesezyklus zum ersten Mal in dem dritten Zyklus erreicht wird, die darauffolgenden Daten nur dann eingegeben werden, wenn sich die Daten A und B, die gleichzeitig ausgeführt sind, zu der Operationsausführungsstufe hin verschieben, d. h. dem vierten Zyklus.

[0060] Als nächstes wird eine Beschreibung der inneren Struktur des vorstehenden Befehl-Ausgabesteuerabschnitts **8** der **Fig. 1** beschrieben. Wie in der Zeichnung dargestellt ist, weist der Befehl-Ausgabesteuerabschnitt **8** auf: einen Daten-Abhängigkeits-Erfassungsabschnitt **8a**; einen Ressource-Hazard-(Gefahr)-Steuerabschnitt **8b**; und einen Ausgabesteuerabschnitt **8c**. Der vorstehende Daten-Abhängigkeits-Erfassungsabschnitt **8a** nimmt Informationen, erhalten über die Decodierung einer Mehrzahl von (drei) Befehlen von dem vorstehenden Befehl-Decodierabschnitt **7**, auf und beurteilt, basierend auf den Informationen, das Vorhandensein oder Nichtvorhandensein einer datenabhängigen Beziehung zwischen den Datenprozessen (z. B. $C = A + B$, $E = C + D$ und $G = E + F$), angewiesen durch die drei Befehle. Der vorstehende Ressource-Hazard-Steuerabschnitt **8b** nimmt ein Signal auf, das darstellt, dass irgendeine der vorstehenden Pipeline-Datenverarbeitungsschaltungen **100** bis **102** den Dummy-Zyklus, dargestellt in den **Fig. 9** und **10** (Dummy-Zyklus-Ausführungs-Signal), ausführt, beurteilt, dass dort ein Ressource-Hazard (eine Ressource-Gefahr) in Bezug auf einen Empfang des Signals vorhanden ist, und gibt das Ressource-Hazard-Signal zu dem Ausgabesteuerabschnitt **8c** aus. Der vorstehende Ausgabesteuerab-

schnitt **8c** beurteilt, ob sich jede der Pipeline-Datenverarbeitungsschaltungen **100** bis **102** in dem ausführbaren Zustand befindet oder nicht, in dem der Befehl ausführbar ist, und beurteilt, ob weiter beurteilt worden ist, dass der Befehl ausführbar ist, ob der vorstehende Dummy-Zyklus, dargestellt in den **Fig. 9** und **10**, eingesetzt werden sollte oder nicht. Falls der Dummy-Zyklus eingesetzt werden sollte, bestimmt der Ausgabesteuerabschnitt **8c** die Zahl von Dummy-Zyklen, die eingefügt werden sollen, und addiert die Zahl zu dem Befehl hinzu und gibt einen Befehl oder eine Mehrzahl von Befehlen zu dem vorstehenden Ausführungssteuerabschnitt **11** aus, so dass der Befehl oder die Befehle in der Pipeline-Datenverarbeitungsschaltung oder den -schaltungen in der ausführbaren Stufe ausgeführt werden.

[0061] Genauer gesagt beurteilt der vorstehende Ausgabesteuerabschnitt **8c**, ob sich die Pipeline-Datenverarbeitungsschaltungen **100** bis **102** in dem ausführbaren Zustand befinden oder nicht, und zwar durch Prüfen, ob jede der Pipeline-Datenverarbeitungsschaltungen **100** bis **102** einen Dummy-Zyklus ausführt oder nicht. Falls jede der Pipeline-Datenverarbeitungsschaltungen **100** bis **102** keinen Dummy-Zyklus ausführt, beurteilt der Ausgabesteuerabschnitt **8c**, dass sie sich in dem Ausführungszustand befindet.

[0062] Das Einsetzen oder Nichteinsetzen eines Dummy-Zyklus und die Zahl von Dummy-Zyklen, die eingesetzt werden sollen, werden in der folgenden Art und Weise bestimmt. Zuerst wird beurteilt, ob eine datenabhängige Beziehung zwischen dem Datenprozess, der zuvor ausgeführt ist, und dem Datenprozess, der darauffolgend ausgeführt werden soll, oder zwischen einer Mehrzahl von Datenprozessen, die gleichzeitig ausgeführt werden sollen, besteht. Falls keine datenabhängige Beziehung besteht, wird bestimmt, dass die Zahl von Dummy-Zyklen 0 ist, was anzeigt, dass kein Dummy-Zyklus eingesetzt werden sollte. Falls irgendeine datenabhängige Beziehung, andererseits, vorhanden ist, wird die Zahl von Dummy-Zyklen, die eingesetzt werden soll, wie folgt bestimmt: zum besseren Verständnis wird eine Betrachtung für den Fall angegeben, bei dem drei Datenprozesse $C = A + B$, $D = E + C$ und $G = F + D$ in einer geordneten Folge ausgeführt werden, wie dies in **Fig. 11** dargestellt ist. Die Zahl von Dummy-Zyklen, die eingesetzt werden sollen, wird zuerst für den Datenprozess bestimmt, der mit der höchsten Priorität ausgeführt werden soll, und wird dann sequenziell für die anderen Prozesse mit niedrigeren Prioritäten bestimmt. Falls die Zahl von Pipeline-Stufen des Kalkulators **30** in der Pipeline-Datenverarbeitungsschaltung (die „1“ in der Pipeline-Datenverarbeitungsschaltung der **Fig. 4** ist und die „2“ in der Pipeline-Datenverarbeitungsschaltung der **Fig. 6** ist) zum Ausführen des Datenprozesses D (oder G), der eine datenabhängige Beziehung mit dem vorhergehenden Datenprozess C (oder D) besitzt, als a bezeichnet wird (anhand eines Beispiels wird eine Beschreibung

nachfolgend in Bezug auf die Pipeline-Datenverarbeitungsschaltung der **Fig. 4** angegeben, in der $a = 1$ ist), die Zahl der Dummy-Zyklen, die in Bezug auf den vorhergehenden Datenprozess C (oder D) eingesetzt werden soll, der durch die Pipeline-Datenverarbeitungsschaltung ausgeführt wird und in Bezug auf den darauffolgenden Datenprozess D (oder G) datenabhängig ist, mit b bezeichnet wird, und eine Differenz in den einleitenden Zyklen der vorstehenden Datenprozesse, die eine datenabhängige Beziehung haben (C und D oder D und G), als c bezeichnet wird, kann die Zahl x von Dummy-Zyklen durch die folgende Gleichung berechnet werden:

$$x = a + b - c.$$

[0063] In dem vorstehenden Beispiel, dargestellt in **Fig. 11**, beträgt, da der erste Datenprozess $C = A + B$ keinen vorhergehenden Datenprozess hat, von dem er datenabhängig ist, die Zahl von Dummy-Zyklen „0“. Wie für den zweiten Datenprozess $D = E + C$, der eine datenabhängige Beziehung zu dem ersten Datenprozess C besitzt, wird die Zahl b von Dummy-Zyklen, die in den Datenprozess C eingesetzt werden soll, „0“, und die Differenz c in den einleitenden Zyklen der zwei Datenprozesse ist „0“ (gleichzeitige Ausführung), so dass die Zahl x von Dummy-Zyklen, die berechnet werden soll, $1 + 0 - 0 = 1$ wird. Wie für den dritten Datenprozess $G = F + D$, der eine datenabhängige Beziehung in Bezug auf den vorhergehenden Datenprozess D besitzt, ist die Zahl b von Dummy-Zyklen, die in den Datenprozess D, von dem der Datenprozess G abhängig ist, eingesetzt werden sollen, „1“. und eine Differenz c in den einleitenden Zyklen der zwei Prozesse ist „0“ (gleichzeitige Ausführung), so dass die Zahl x von Dummy-Zyklen, die ausgeführt werden soll, $1 + 1 - 0 = 2$ wird.

[0064] Wie anhand von **Fig. 11** ersichtlich werden wird, wird, von den drei Datenprozessen C, D und G, der erste Datenprozess C mit der Zeitabstimmung der **Fig. 8** ausgeführt, der zweite Datenprozess D, der eine datenabhängige Beziehung zu dem vorhergehenden Prozess C besitzt, wird mit der Zeitabstimmung der **Fig. 9** ausgeführt, und der dritte Datenprozess G, der eine datenabhängige Beziehung zu dem vorhergehenden Prozess D besitzt, wird mit der Zeitabstimmung der **Fig. 10** ausgeführt. Demzufolge können, gerade wenn jeweilige Befehle, um die drei Datenprozesse C, D und G durchzuführen, die eine datenabhängige Beziehung dazwischen haben, gleichzeitig ausgegeben werden, sie ohne Stagnation ausgeführt werden. Die Datenprozesse C, D und G werden unter derselben Zeitabstimmung abgeschlossen, so dass das Verfahren „In-Order“ sicher ausgeführt wird.

[0065] In **Fig. 1** nimmt der Ausführungssteuerabschnitt **11** einen Befehl, ausgegeben von dem Befehl-Steuerabschnitt **8c**, auf und bewirkt, dass eine der Pipeline-Datenverarbeitungsschaltungen **100** bis **102**, die durch den Befehl bezeichnet ist, den Daten-

prozess, angewiesen durch den Befehl, ausführt. Der vorstehende Ausführungssteuerabschnitt **11** steuert den Selektor **70** der vorstehend bezeichneten Pipeline-Datenverarbeitungsschaltung so, dass der Ausgang von dem zweiten Pipeline-Register **60** ausgewählt wird, wenn die Zahl x von Dummy-Zyklen, die zu dem entsprechenden Befehl hinzugefügt worden sind, 0 ist, dass der Ausgang von dem ersten Pipeline-Register **40** ausgewählt wird, wenn die Zahl x von Dummy-Zyklen 1 ist, und dass der Ausgang von dem Kalkulator **30** (Operationsergebnis) ausgewählt wird, wenn die Zahl x von Dummy-Zyklen 2 ist. Falls irgendeine der Pipeline-Datenverarbeitungsschaltungen **100** bis **102** einen Dummy-Zyklus ausführt, überträgt der vorstehende Ausführungssteuerabschnitt **11** ein Dummy-Zyklus-Ausführungs-Signal zu dem vorstehenden Ressource-Hazard-Steuerabschnitt **8b**.

[0066] Als nächstes wird, unter Bezugnahme auf **Fig. 12**, die Operation des Pipeline-Datenverarbeitungsgeräts der vorliegenden Erfindung unter Verwendung einer komplizierteren Datenverarbeitungssequenz beschrieben:

$$C = A + B \quad (1)$$

$$E = C + D \quad (2)$$

$$G = E + F \quad (3)$$

$$J = H + I \quad (4)$$

$$L = J + K \quad (5)$$

$$N = L + M \quad (6)$$

$$P = N + O \quad (7)$$

$$S = Q + R \quad (8)$$

$$U = S + T \quad (9)$$

$$X = U + V \quad (10).$$

[0067] Von diesen zehn Datenprozessen besitzt jeder der Datenprozesse, mit Ausnahme der Datenprozesse (1), (4) und (8), eine datenabhängige Beziehung in Bezug auf den vorhergehenden Datenprozess.

[0068] In **Fig. 12** werden drei Befehle, um die Datenprozesse C, E und G durchzuführen, zuerst decodiert und gleichzeitig ausgegeben. Der ersten Datenprozess C, der zweite Datenprozess E und der dritte Datenprozess G schreiten in den Pipeline-Datenverarbeitungsschaltungen **100**, **101** und **102** jeweils fort. Unter diesen existiert eine datenabhängige Beziehung zwischen dem ersten und dem zweiten Datenprozess C und E, während eine datenabhängige Beziehung auch zwischen dem zweiten und dem dritten Datenprozess E und G besteht. Die Zahl x von Dummy-Zyklen, um in die Pipeline-Datenverarbeitungs-

schaltung **101** eingesetzt zu werden, wird 1, da $a = 1$, $b = 0$ und $c = 0$ in der vorstehenden Berechnungsgleichung gilt. Die Zahl x von Dummy-Zyklen, die in die Pipeline-Datenverarbeitungsschaltung **102** eingesetzt werden soll, wird 2, da $a = 1$, $b = 1$ und $c = 0$ in der vorstehenden Berechnungsgleichung gilt.

[0069] In dem ersten Zyklus werden drei Befehle, um die darauffolgenden Datenprozesse J, L und N durchzuführen, decodiert. Da die zwei Pipeline-Datenverarbeitungsschaltungen **101** und **102** Dummy-Zyklen ausführen, werden Ressource-Hazard-Signale für diese Verarbeitungsschaltungen **101** und **102** erzeugt. Hieraufhin wird nur ein Befehl, um den vierten Datenprozess J auszuführen, ausgegeben, so dass der Datenprozess J in der Pipeline-Datenverarbeitungsschaltung **100** fortschreitet. Da der vierte Datenprozess J keine datenabhängige Beziehung zu irgendeinem der existierenden Datenprozesse C, E und G hat, ist die Zahl x von Dummy-Zyklen 0 in der Pipeline-Datenverarbeitungsschaltung **100** zum Durchführen des Datenprozesses J.

[0070] In dem zweiten Zyklus werden die anderen zwei Befehle, die nicht in dem vorhergehenden Zyklus ausgegeben worden sind, und ein Befehl, um den siebten Datenprozess P durchzuführen, decodiert. Da nur die Pipeline-Datenverarbeitungsschaltung **102** einen Dummy-Zyklus ausführt, wird ein Ressource-Hazard-Signal für die Verarbeitungsschaltung **102** erzeugt. Hieraufhin werden zwei Befehle, um den fünften und den sechsten Datenprozess L und N durchzuführen, ausgegeben, und die Datenprozesse L und N schreiten in den Pipeline-Datenverarbeitungsschaltungen **100** und **101** jeweils fort. Hier existiert eine datenabhängige Beziehung zwischen dem vierten und dem fünften Datenprozess L und N, während eine datenabhängige Beziehung auch zwischen dem fünften und dem sechsten Datenprozess L und N existiert. Demzufolge wird die Zahl x von Dummy-Zyklen, die in die Pipeline-Datenverarbeitungsschaltung **100** eingesetzt werden soll, 0, da $a = 1$, $b = 0$ und $c = 1$ in der vorstehenden Berechnungsgleichung gilt. Die Zahl x von Dummy-Zyklen, die in die Pipeline-Datenverarbeitungsschaltung **101** eingesetzt werden soll, wird 1, da $a = 1$, $b = 0$ und $c = 0$ in der vorstehenden Berechnungsgleichung gilt.

[0071] In dem dritten Zyklus werden der andere Befehl, der nicht in dem vorherigen Zyklus ausgegeben worden ist, und Befehle, um den achten und den neunten Datenprozess S und U durchzuführen, decodiert. Da nur die Pipeline-Datenverarbeitungsschaltung **101** einen Dummy-Zyklus ausführt, wird ein Ressource-Hazard-Signal für die Verarbeitungsschaltung **101** erzeugt. Hieraufhin werden zwei Befehle, um den siebten und den achten Datenprozess P und S durchzuführen, ausgegeben, so dass die Datenprozesse P und S in den Pipeline-Datenverarbeitungsschaltungen **100** und **102** jeweils fortschreiten. Hier existiert eine datenabhängige Beziehung zwischen dem siebten und dem achten Datenprozess N

und P. Demzufolge wird die Zahl x von Dummy-Zyklen, die in die Pipeline-Datenverarbeitungsschaltung **100** eingesetzt werden soll, 1, da $a = 1$, $b = 1$ und $c = 1$ gilt. Andererseits ist die Zahl x von Dummy-Zyklen, die in die Pipeline-Datenverarbeitungsschaltung **102** eingesetzt werden soll, 0, da keine datenabhängige Beziehung zwischen dem siebten und dem achten Datenprozess P und S existiert.

[0072] In dem vierten Zyklus werden der andere Befehl, der nicht in dem vorherigen Zyklus ausgegeben worden ist, und ein Befehl, um den zehnten Datenprozess X durchzuführen, decodiert. Da nur die Pipeline-Datenverarbeitungsschaltung **100** einen Dummy-Zyklus ausführt, wird ein Ressource-Hazard-Signal für die Verarbeitungsschaltung **100** erzeugt. Hieraufhin werden zwei Befehle, um den neunten und den zehnten Datenprozess U und X durchzuführen, ausgegeben, so dass die Datenprozesse U und X in den Pipeline-Datenverarbeitungsschaltungen **101** und **102**, jeweils, fortschreiten. Hierbei existiert eine datenabhängige Beziehung zwischen dem achten und dem neunten Datenprozess S und U, während eine datenabhängige Beziehung auch zwischen dem neunten und dem zehnten Datenprozess U und X existiert. Demzufolge wird die Zahl x von Dummy-Zyklen, die in die Pipeline-Datenverarbeitungsschaltung **101** eingesetzt werden soll, 0, da $a = 1$, $b = 0$ und $c = 1$ in der vorstehenden Berechnungsgleichung gilt. Andererseits wird die Zahl x von Dummy-Zyklen, die in die Pipeline-Datenverarbeitungsschaltung **102** eingesetzt werden soll, 1, da $a = 1$, $b = 0$ und $c = 0$ in der vorstehenden Berechnungsgleichung gilt.

[0073] In der vorliegenden Ausführungsform fällt der Zeitpunkt, zu dem alle Befehle, um zehn Datenprozesse durchzuführen, ausgegeben worden sind, in den fünften Zyklus, während der Zeitpunkt, zu dem die Ausführung aller Datenprozesse abgeschlossen worden ist, in den neunten Zyklus fällt, wie anhand der **Fig. 12** ersichtlich werden wird. Andererseits fällt, falls die vorstehende Datenverarbeitungssequenz unter Verwendung des herkömmlichen Pipeline-Datenverarbeitungsgeräts der **Fig. 17** ausgeführt wird, der Zeitpunkt, zu dem alle zehn Datenprozesse ausgegeben worden sind, in den achten Zyklus, während der Zeitpunkt, zu dem die Ausführung aller Datenprozesse abgeschlossen worden ist, in den zehnten Zyklus fällt, wie in dem Zeitabstimmungsdiagramm der **Fig. 13** dargestellt ist.

[0074] Aus einem Vergleich der Ausführungsform der vorliegenden Erfindung, dargestellt in **Fig. 12**, mit der herkömmlichen Ausführungsform, dargestellt in **Fig. 13**, wird ersichtlich werden, dass das Pipeline-Datenverarbeitungsgerät in der Ausführungsform der vorliegenden Erfindung einen Vorteil gegenüber der herkömmlichen Ausführungsform im Hinblick auf die Leistung der Betriebsgeschwindigkeit besitzt.

(Zweite Ausführungsform eines Pipeline-Datenverarbeitungsgeräts)

[0075] **Fig. 14** stellt ein Pipeline-Datenverarbeitungsgerät, das eine zweite Ausführungsform der vorliegenden Erfindung darstellt, dar.

[0076] In der vorliegenden Ausführungsform sind drei zusätzliche Pipeline-Datenverarbeitungsschaltungen **103** bis **105**, die dieselbe Struktur haben, parallel in der Struktur der **Fig. 1** vorgesehen. Jede der Pipeline-Datenverarbeitungsschaltungen **103** bis **105** besitzt einen Mehrfach-Eingang-Kalkulator **30''**, der drei oder mehr Eingänge (drei Eingänge in der Zeichnung) besitzt. In Übereinstimmung mit dem Kalkulator **30''** mit 3 Eingängen sind drei Dateneingangsanschlüsse **17** bis **19** und drei Dateneingangsregister **20** bis **22**, jeweils zum Speichern von Daten, eingegeben zu dem entsprechenden Dateneingangsanschluss, vorgesehen worden. Die Daten, gespeichert in den Dateneingangsregistern **20** bis **22**, werden zu dem vorstehenden Kalkulator **30''** mit 3 Eingängen eingegeben und einer Addition oder Subtraktion unterworfen. Die anderen Komponenten sind dieselben wie solche der Pipeline-Datenverarbeitungsschaltung der ersten Ausführungsform, dargestellt in **Fig. 4**, so dass die Beschreibung hiervon weggelassen wird, indem dieselben Komponenten mit denselben Bezugszeichen bezeichnet werden. In **Fig. 14** bezeichnen die Bezugszeichen **9j** bis **9r** Register in der Registerdatei **9**.

[0077] **Fig. 15** stellt die Struktur eines grundsätzlichen Bereichs eines Prozessors, umfassend das vorstehende Pipeline-Datenverarbeitungsgerät der **Fig. 14**, dar. Da die gesamte Struktur des Prozessors ähnlich zu der Struktur des Prozessors der vorstehenden ersten Ausführungsform, dargestellt in **Fig. 1**, ist, werden nur die unterschiedlichen Komponenten nachfolgend beschrieben. Der Prozessor der **Fig. 15** ist gegenüber dem Prozessor der **Fig. 1** nur dahingehend unterschiedlich, dass der Befehl-Ausgabesteuerabschnitt **8'** mit dem Ausgabesteuerabschnitt **8c'** versehen ist, anstelle des Befehl-Ausgabesteuerabschnitts **8** mit dem Befehl-Steuerabschnitt **8c**. Der Befehl-Steuerabschnitt **8c'** ändert eine Mehrzahl von Datenprozessen, die ausgeführt werden sollen, um so eine datenabhängige Beziehung zwischen den Datenprozessen zu beseitigen. Zum Beispiel wird, in einer Mehrzahl von (zwei) Datenprozessen, die die folgende, datenabhängige Beziehung

$$C = A + B$$

$$D = E + C,$$

dazwischen haben, der letztere Datenprozess durch

$$D = E + C = E + (A + B) = E + A + B$$

ersetzt.

[0078] Danach wird ein Befehl, um den Datenpro-

zess **D** durchzuführen, ausgegeben, so dass der Datenprozess **D** durch die vorstehende Pipeline-Datenverarbeitungsschaltung **103**, die den Kalkulator **30''** mit 3 Eingängen besitzt, ausgeführt. Dementsprechend kann eine Mehrzahl von Datenprozessen, die ursprünglich eine datenabhängige Beziehung haben, gleichzeitig parallel wie eine Mehrzahl von Datenprozessen, die keine datenabhängigen Beziehung haben, ausgeführt werden.

[0079] Dementsprechend können, in dem Fall, bei dem die folgenden sechs Datenprozesse, die eine datenabhängige Beziehung haben, sie in dem Pipeline-Datenverarbeitungsgerät der vorliegenden Ausführungsform ausgeführt werden:

$$C = A + B$$

$$D = E + C$$

$$G = F + D$$

$$H = I + G$$

$$J = K + H$$

$$L = M + J,$$

Befehle, um die sechs Datenprozesse auszuführen, die eine datenabhängige Beziehung haben, können ausgegeben werden und können gleichzeitig unter den zwei Effekten einer Beseitigung der datenabhängigen Beziehung durch das Pipeline-Datenverarbeitungsgerät, das den Kalkulator **30''** mit 3 Eingängen besitzt, und eines Einsetzens des Dummy-Zyklus, der in der ersten Ausführungsform beschrieben ist, ausgeführt werden.

[0080] Obwohl die vorliegende Ausführungsform den Kalkulator **30''** mit 3 Eingängen für den Kalkulator **30** der Pipeline-Datenverarbeitungsschaltung, dargestellt in **Fig. 4**, substituiert hat, wird ersichtlich werden, dass der Kalkulator **30** jeder der Pipeline-Datenverarbeitungsschaltungen, dargestellt in den **Fig. 5, 6** und **7**, durch den Kalkulator **30''** mit 3 Eingängen ersetzt werden kann.

Patentansprüche

1. Pipeline-Datenverarbeitungsgerät, das mindestens eine Pipeline-Verarbeitungsschaltung (**100**) aufweist, wobei die Pipeline-Verarbeitungsschaltung (**100**) aufweist: ein Eingangsdatenregister (**20, 21**) zum Aufnehmen und Speichern von Daten von der Außenseite; eine Datenoperationsschaltung (**30**) zum Aufnehmen der Daten, gespeichert in dem Eingangsdatenregister (**20, 21**), und zum Durchführen einer spezifizierten Operation in Bezug auf die Eingangsdaten; ein Zwischen-Pipeline-Register (**40, 60**), angeordnet in einer Stufe, auf die Datenoperationsschaltung (**30**) folgend;

ein End-Pipeline-Register (**81**), angeordnet in einer Stufe, dem Zwischen-Pipeline-Register (**40, 60**) folgend;
 eine Pfadumschalterschaltung (**70**) zum Bewirken, dass ein Operationsergebnis von der Datenoperationsschaltung (**30**) durch das Zwischen-Pipeline-Register (**40, 60**) hindurchgeführt oder im Bypass daran vorbeigeführt wird, und zum Eingeben des Operationsergebnisses zu dem End-Pipeline-Register (**81**);
 eine Einrichtung, angepasst so, um, falls erforderlich, da die spezifizierte Operation Daten von einer anderen Operation erfordert, mindestens einen Dummy-Zyklus vor einem Ausführen des Zyklus, in dem die Daten empfangen und gespeichert sind, einzuführen; und
 eine Einrichtung, angepasst so, um mindestens ein Zwischen-Pipeline-Register (**40, 60**) im Bypass zu umgehen, um so den mindestens einen Dummy-Zyklus zu kompensieren,
 um dadurch einen Operationsausführungszeitzyklus zu ermöglichen, in dem die Datenoperationsschaltung (**30**) die spezifizierte Operation durchführt, um zu einem wahlweisen Zeitzyklus in einer Mehrzahl von Pipeline-Zeitzyklen zu verschieben.

2. Pipeline-Datenverarbeitungsgerät nach Anspruch 1, wobei das Pipeline-Datenverarbeitungsgerät weiterhin aufweist:

mindestens eine weitere Pipeline-Verarbeitungsschaltung (**101, 102**), die dieselbe wie die Pipeline-Verarbeitungsschaltung (**100**) ist; und
 einen Befehl-Ausgabesteuerabschnitt (**8c**) zum gleichzeitigen Ausgeben einer Mehrzahl von Befehlen, um jeweilige Datenprozesse durchzuführen, zu den jeweiligen Pipeline-Datenverarbeitungsschaltungen (**100, 101, 102**), und zum Hinzufügen der Zahl von Dummy-Zyklen, die eingesetzt werden sollen, wobei in jedem davon nichts ausgeführt wird, zu jedem der Befehle, die gleichzeitig ausgegeben sind.

3. Pipeline-Datenverarbeitungsgerät nach Anspruch 2, wobei die Mehrzahl von Befehlen, gleichzeitig ausgegeben von dem Befehl-Ausgabesteuerabschnitt (**8c**), jeweilige Befehle umfassen, um mindestens zwei Datenprozesse durchzuführen, die eine datenabhängige Beziehung dazwischen haben.

4. Pipeline-Datenverarbeitungsgerät nach Anspruch 1, wobei
 das Pipeline-Datenverarbeitungsgerät weiterhin mindestens eine weitere Pipeline-Verarbeitungsschaltung (**101, 102**) aufweist, die dieselbe wie die Pipeline-Verarbeitungsschaltung (**100**) ist, und
 wobei die Mehrzahl von Pipeline-Verarbeitungsschaltungen (**100, 101, 102**) parallel zueinander angeordnet sind.

5. Pipeline-Datenverarbeitungsgerät nach Anspruch 2, wobei die Mehrzahl von Pipeline-Verarbeitungsschaltungen (**100, 101, 102**) parallel zueinander

der angeordnet sind.

6. Pipeline-Datenverarbeitungsgerät nach einem der Ansprüche 1 bis 5, wobei das Zwischen-Pipeline-Register aus einem ersten (**40**) und einem zweiten (**60**) Pipeline-Register aufgebaut ist, wobei die Pfadumschalterschaltung aus einem Selektor (**70**), zwischengefügt zwischen dem zweiten (**60**) und dem End-(**81**)-Pipeline-Register, aufgebaut ist, und
 wobei der Selektor (**70**) das Operationsergebnis von der Datenoperationsschaltung (**30**) und jeweilige Operationsergebnisse, gespeichert in dem ersten (**40**) und dem zweiten (**60**) Pipeline-Register, aufnimmt, eines der Operationsergebnisse auswählt und das ausgewählte eine zu dem End-Pipeline-Register (**81**) ausgibt.

7. Pipeline-Datenverarbeitungsgerät nach einem der Ansprüche 1 bis 5, wobei das Zwischen-Pipeline-Register aus einem ersten (**40**) und einem zweiten (**60**) Pipeline-Register aufgebaut ist, wobei die Pfadumschalterschaltung aus einem ersten Selektor (**52**), zwischengefügt zwischen dem ersten (**40**) und dem zweiten (**60**) Pipeline-Register, und einem zweiten Selektor (**72**), zwischengefügt zwischen dem zweiten (**60**) und dem End-(**81**)-Pipeline-Register, aufgebaut ist, wobei der erste Selektor (**52**) das Operationsergebnis von der Datenoperationsschaltung (**30**) und ein Operationsergebnis, gespeichert in dem ersten (**40**) Pipeline-Register, aufnimmt, irgendeines der Operationsergebnisse auswählt und das ausgewählte eine des zweiten Pipeline-Registers (**60**) ausgibt, und
 der zweite Selektor (**72**) das Operationsergebnis, ausgewählt durch den ersten Selektor (**52**), und ein Operationsergebnis, gespeichert in dem zweiten Pipeline-Register (**60**), aufnimmt, irgendeines der Operationsergebnisse auswählt und das ausgewählte eine zu dem End-Pipeline-Register (**81**) ausgibt.

8. Pipeline-Datenverarbeitungsgerät nach Anspruch 6 oder 7, wobei eine Stufe, in der die Daten von der Außenseite in dem Eingangs-Datenregister (**20, 21**) gespeichert sind, eine Datenlesestufe ist, die Stufe, in der die Datenoperationsschaltung (**30**) die spezifizierte Operation durchführt und das Operationsergebnis in dem ersten Pipeline-Register (**40**) speichert, die Operationsausführungsstufe ist, eine Stufe, in der das Operationsergebnis, gespeichert in dem ersten Pipeline-Register (**40**), in dem zweiten Pipeline-Register (**60**) gespeichert wird, eine erste Durchgangsstufe ist, und
 eine Stufe, in der das Operationsergebnis, gespeichert in dem zweiten Pipeline-Register (**60**), in dem End-Pipeline-Register (**81**) gespeichert wird, eine zweite Durchgangsstufe ist.

9. Pipeline-Datenverarbeitungsgerät nach einem der Ansprüche 1 bis 8, wobei die Datenoperations-

schaltung (**30**) ein internes Pipeline-Register besitzt.

10. Pipeline-Datenverarbeitungsgerät nach einem der Ansprüche 1 bis 9, wobei die Datenoperationsschaltung (**30**) aus einem Addierer oder einem Subtrahierer, wobei jeder davon zwei oder mehr Dateneingänge besitzt, aufgebaut ist.

11. Pipeline-Datenverarbeitungsgerät nach Anspruch 6 oder 7, wobei in der Pipeline-Datenverarbeitungsschaltung (**100**), die den Befehl aufgenommen hat, wenn die Zahl von Dummy-Zyklen, die eingesetzt werden sollen, die zu dem empfangenen Befehl hinzugefügt worden sind, „0“ ist, das Dateneingangsregister (**20, 21**) die Daten von der Außenseite in einem ersten Zyklus aufnimmt und speichert, während der Selektor (**70**) das Operationsergebnis, gespeichert in dem zweiten Pipeline-Register (**60**), aufnimmt.

12. Pipeline-Datenverarbeitungsgerät nach Anspruch 6 oder 7, wobei in der Pipeline-Datenverarbeitungsschaltung (**100**), die den Befehl aufgenommen hat, wenn die Zahl von Dummy-Zyklen, die eingesetzt werden sollen, die zu dem empfangenen Befehl hinzugefügt worden sind, „1“ ist, das Dateneingangsregister (**20, 21**) die Daten von der Außenseite in einem zweiten Zyklus aufnimmt und speichert, während der Selektor (**70**) das Operationsergebnis, gespeichert in dem ersten Pipeline-Register, auswählt.

13. Pipeline-Datenverarbeitungsgerät nach Anspruch 6 oder 7, wobei in der Pipeline-Datenverarbeitungsschaltung (**100**), die den Befehl aufgenommen hat, wenn die Zahl von Dummy-Zyklen, die eingesetzt werden sollen, die zu dem empfangenen Befehl hinzugefügt worden sind, „2“ ist, das Dateneingangsregister (**20, 21**) die Daten von der Außenseite in einem dritten Zyklus aufnimmt und speichert, während der Selektor (**70**) das Operationsergebnis von der Datenoperationsschaltung (**30**) auswählt.

Es folgen 18 Blatt Zeichnungen

Fig.1

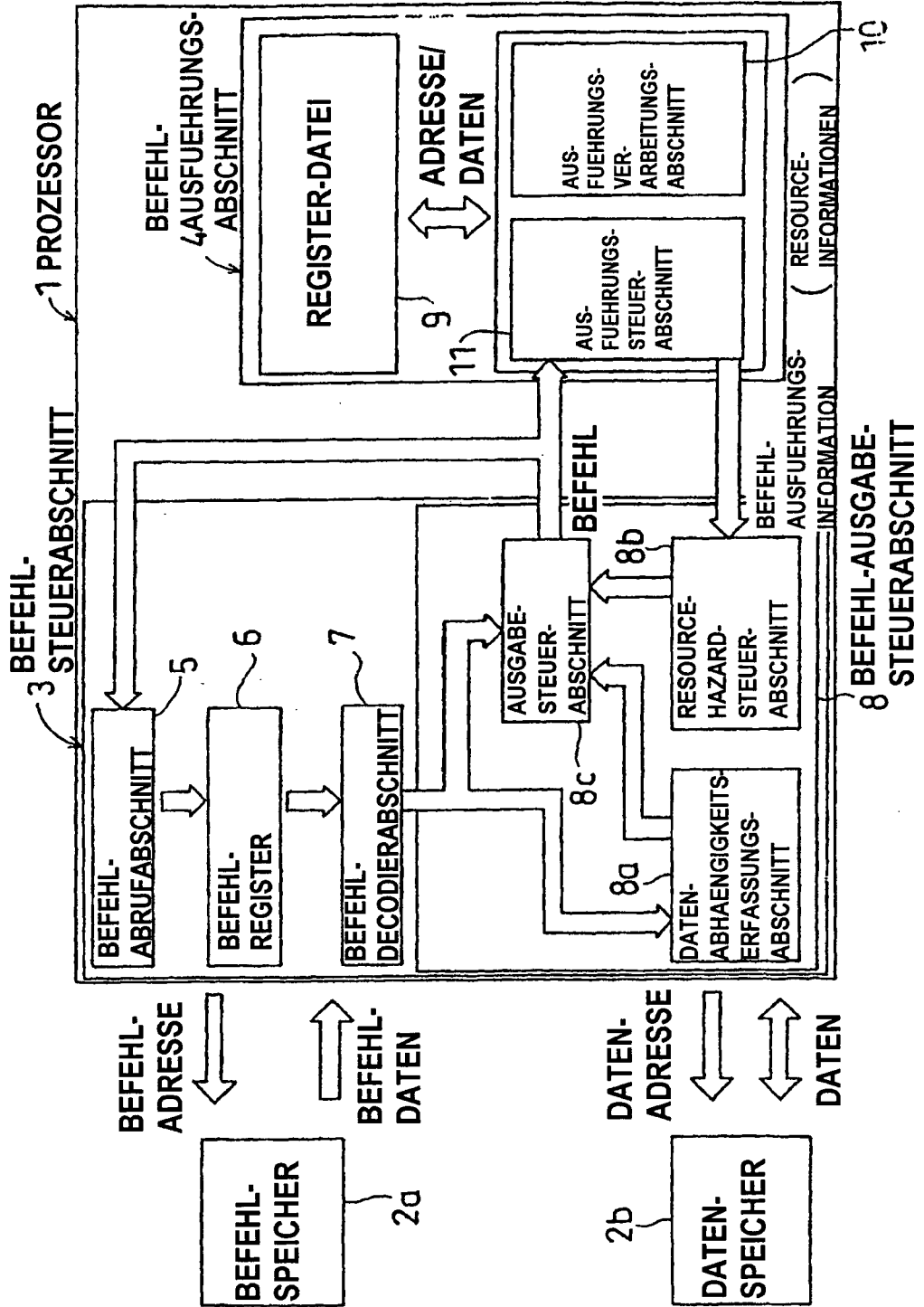


Fig.2

RUFE BEFEHL AB	DECODIERE BEFEHL/ GEBE BEFEHL AUS	LESE DATEN	FUEHRE BE- RECHNUNG AUS	SCHREIBE DATEN
----------------	--------------------------------------	------------	----------------------------	-------------------

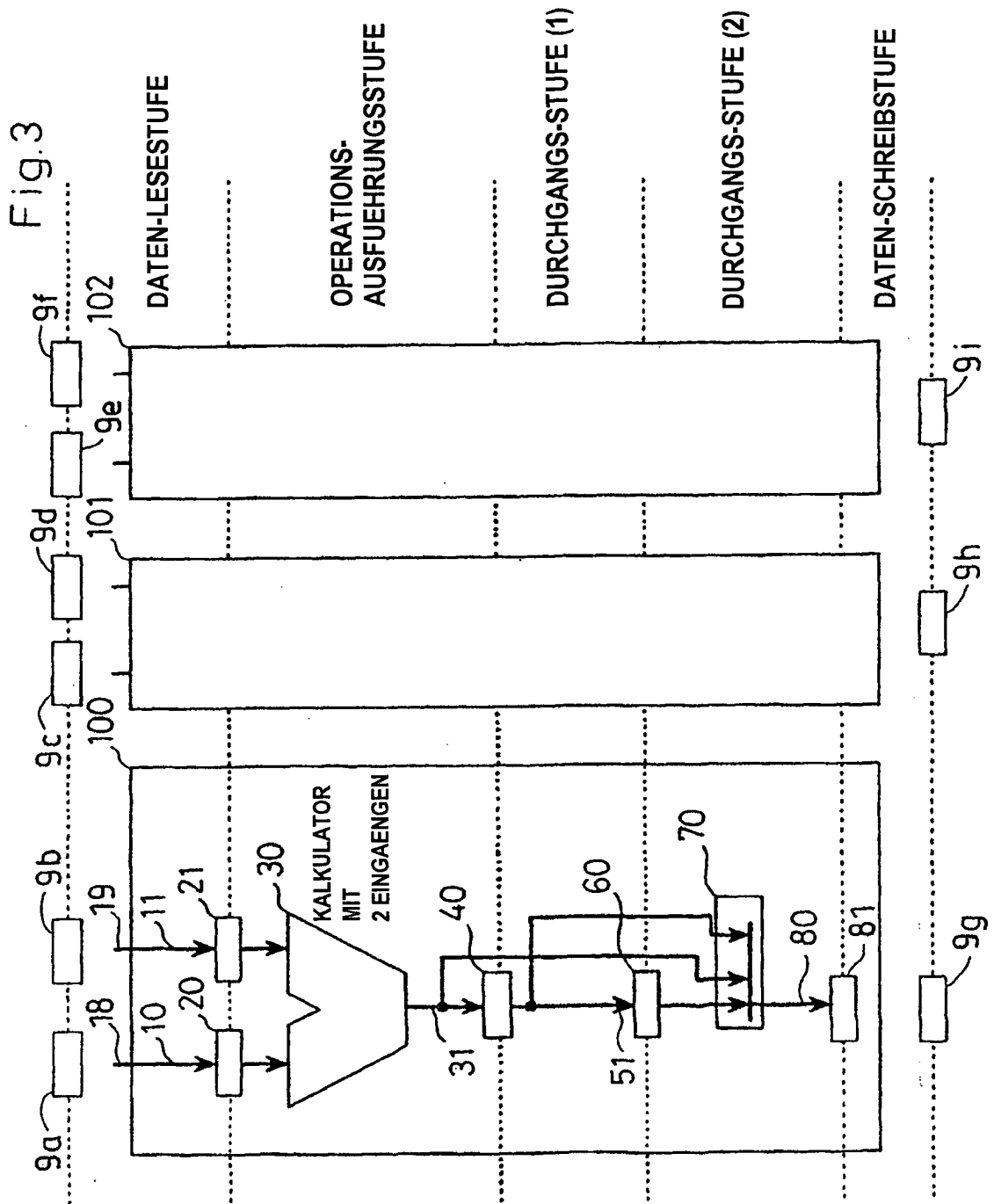


Fig.4

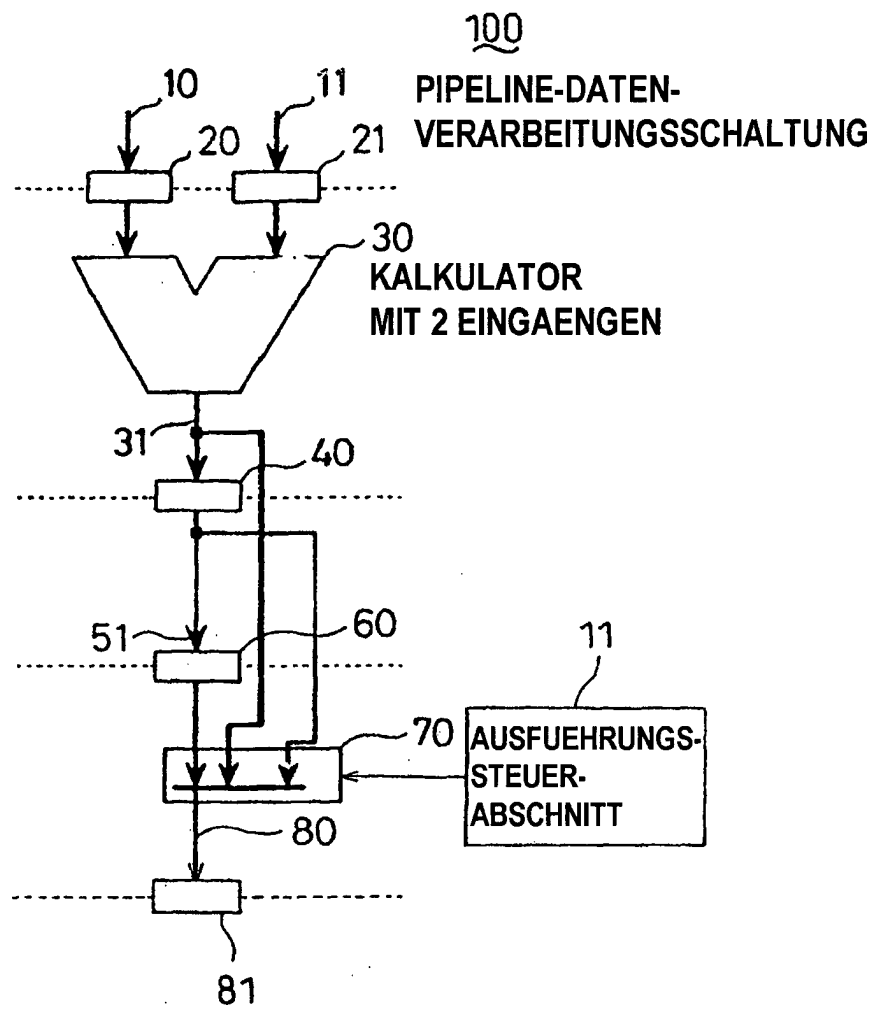


Fig.5

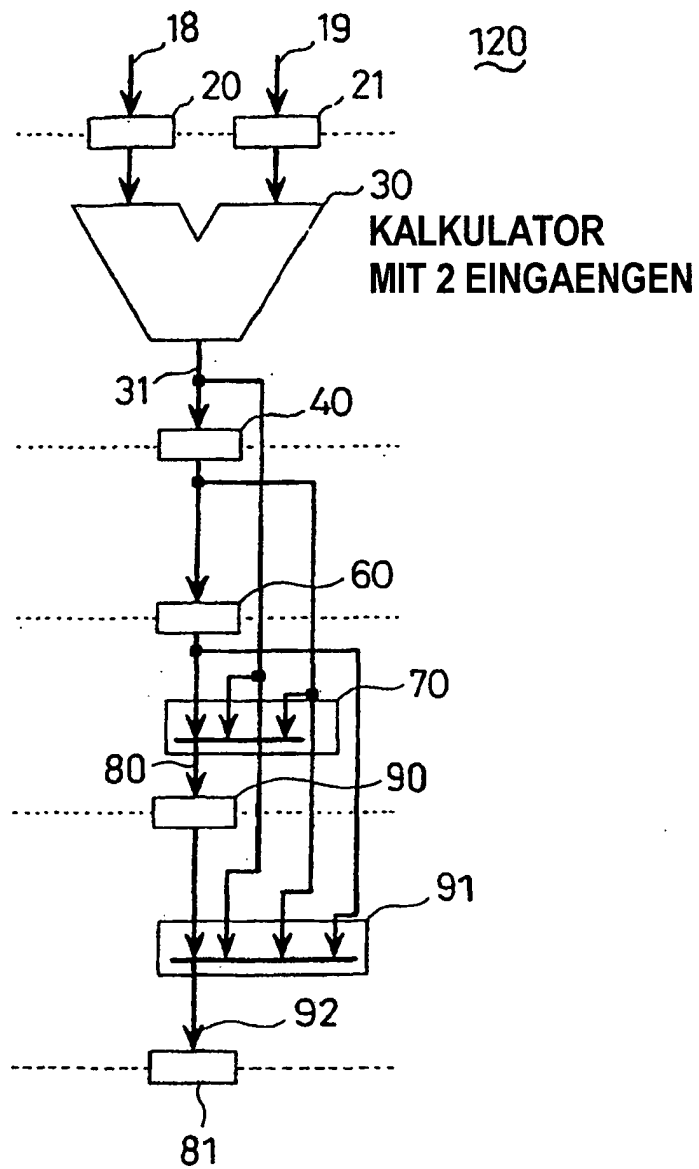


Fig.6

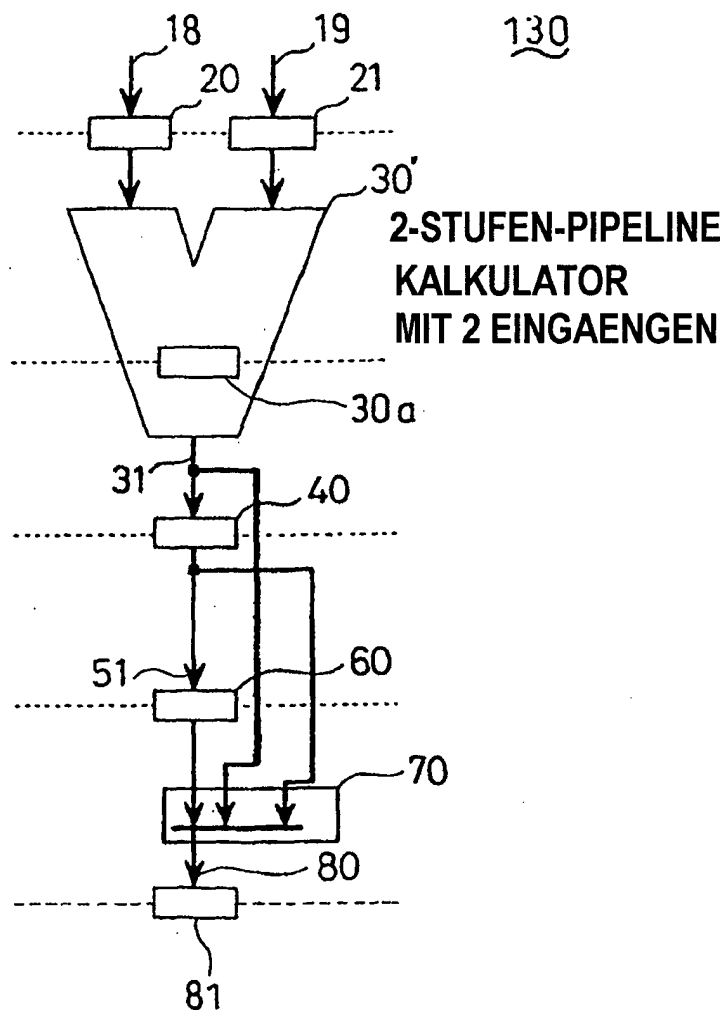


Fig.7

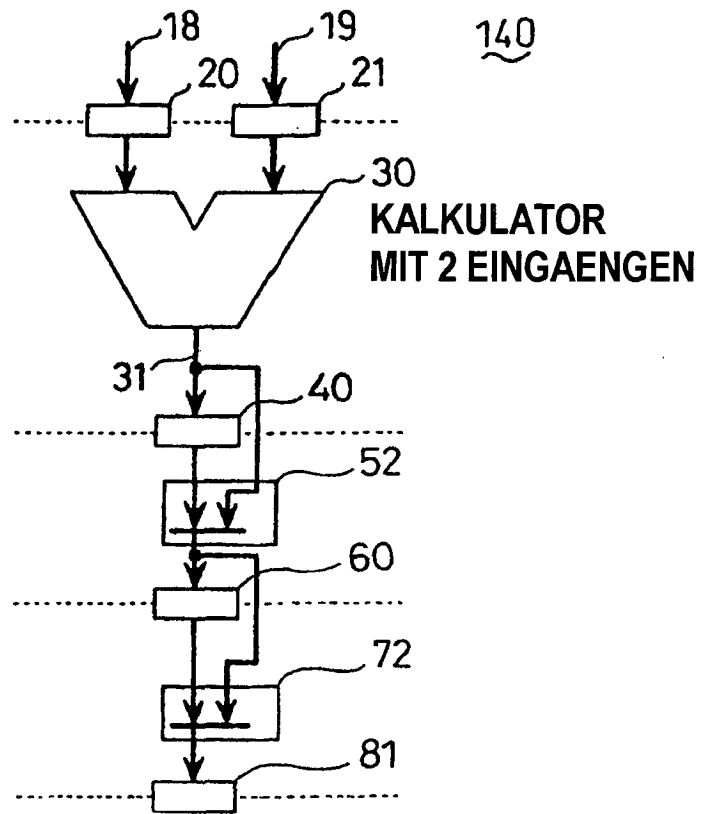


Fig.8

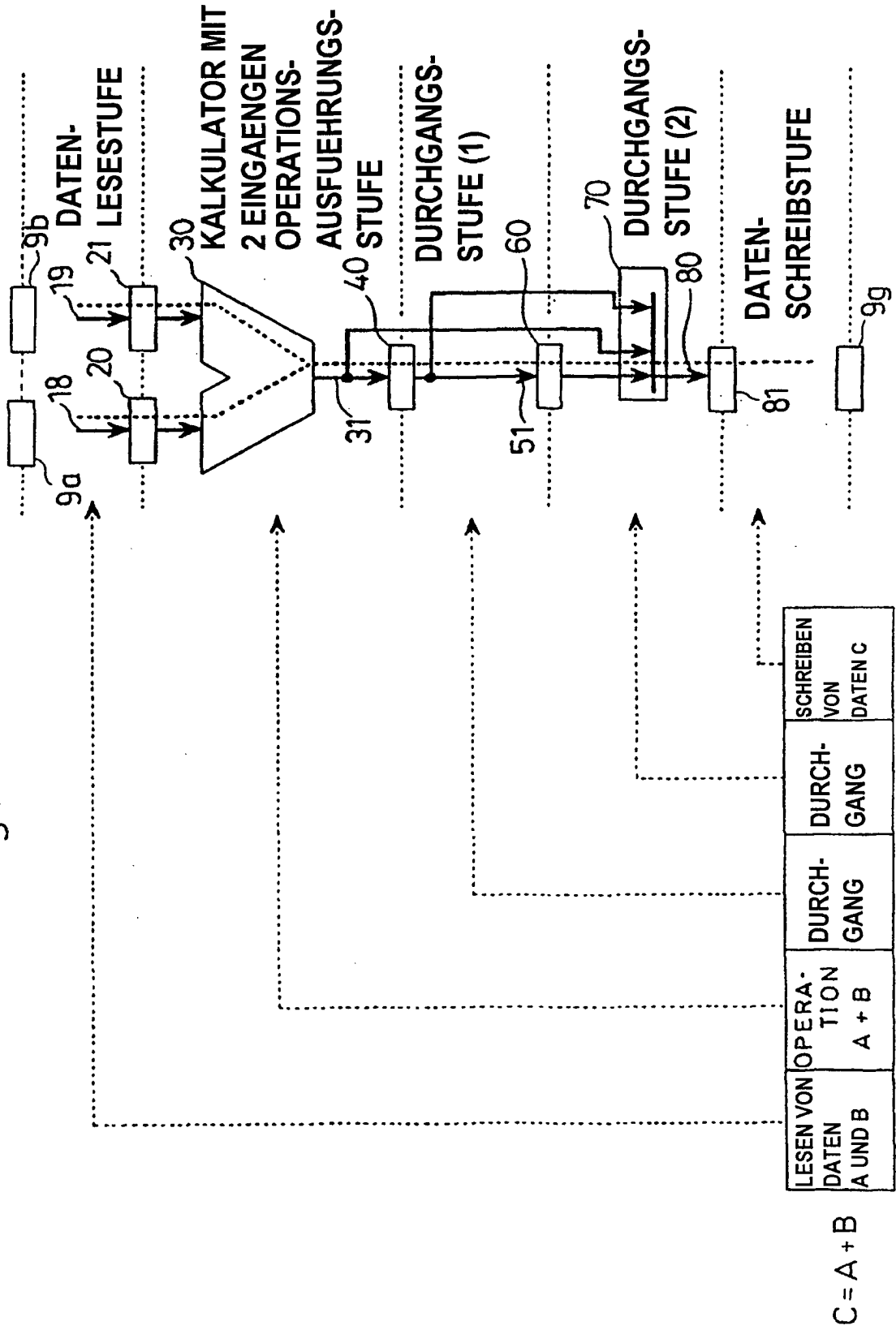


Fig.9

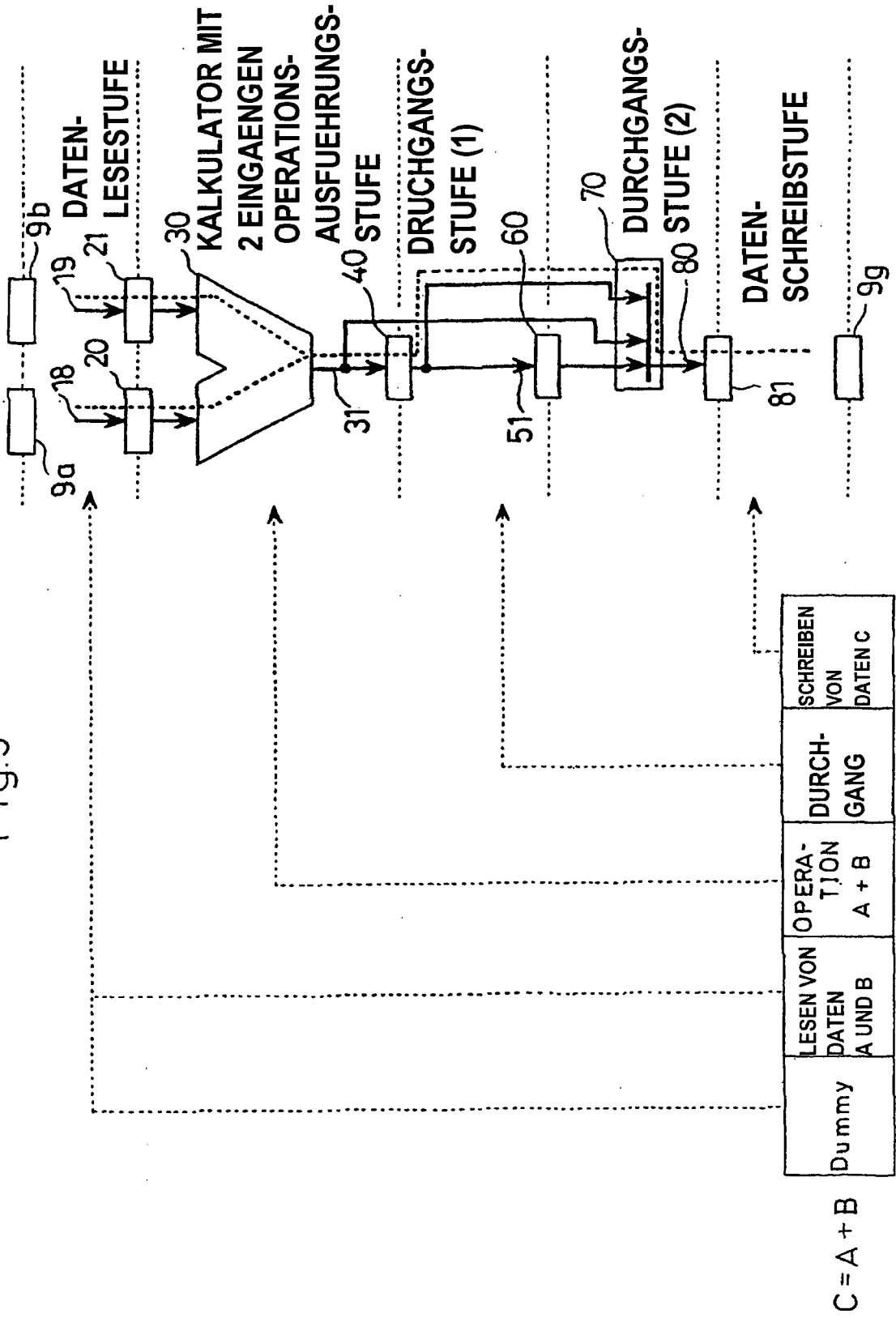


Fig.10

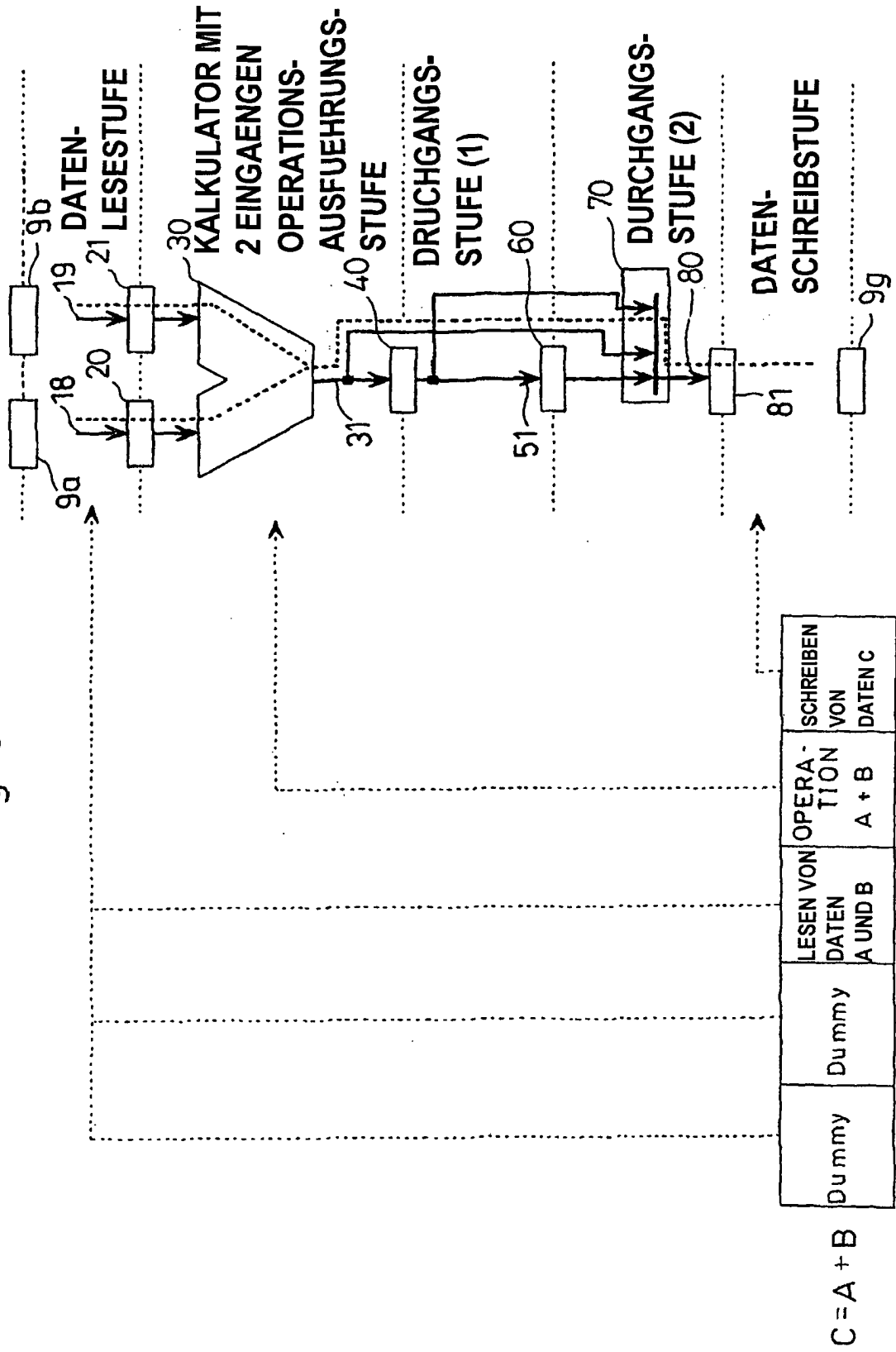


Fig.11

$C = A + B$	LESEN VON DATEN A UND B	AUSFUEHRUNG VON OPERATION A+B	DURCHGANG	DURCHGANG	SCHREIBEN VON DATEN C
$D = E + C$	Dummy	LESEN VON DATEN E UND C	AUSFUEHRUNG VON OPERATION E+C	DURCHGANG	SCHREIBEN VON DATEN D
$G = F + D$	Dummy	Dummy	LESEN VON DATEN F UND D	AUSFUEHRUNG VON OPERATION F+D	SCHREIBEN VON DATEN G

Fig.13

STAND DER TECHNIK

	1	2	3	4	5	6	7	8	9	10
C = A+B	LESEN VON A UND B	OPERATION A+B	SCHREIBEN VON DATEN C	100						
E = C+D		LESEN VON C UND D	OPERATION C+D	SCHREIBEN VON DATEN E	100					
G = E+F			LESEN VON C UND F	OPERATION E+F	SCHREIBEN VON DATEN G	100				
J = H+I			LESEN VON H UND I	OPERATION H+I	SCHREIBEN VON DATEN J	101				
L = J+K				LESEN VON J UND K	OPERATION J+K	SCHREIBEN VON DATEN L	100			
N = L+M					LESEN VON L UND M	OPERATION L+M	SCHREIBEN VON DATEN N	100		
P = N+O						LESEN VON N UND O	OPERATION N+O	SCHREIBEN VON DATEN P	100	
S = Q+R						LESEN VON Q UND R	OPERATION Q+R	SCHREIBEN VON DATEN S	101	
U = S+T							LESEN VON S UND T	OPERATION S+T	SCHREIBEN VON DATEN U	100
X = U+V							100	LESEN VON U UND V	OPERATION U+V	SCHREIBEN VON DATEN X

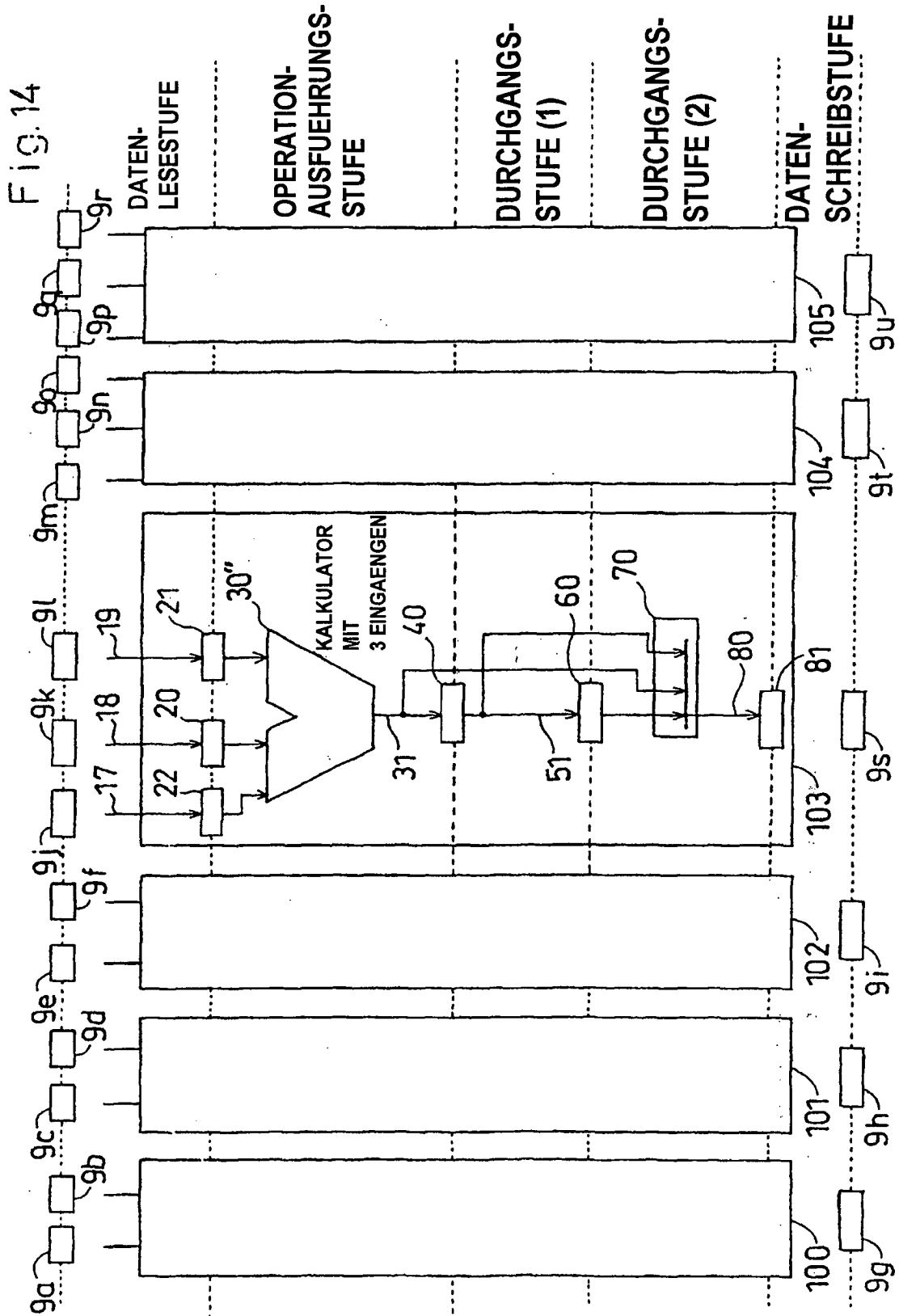


Fig.15

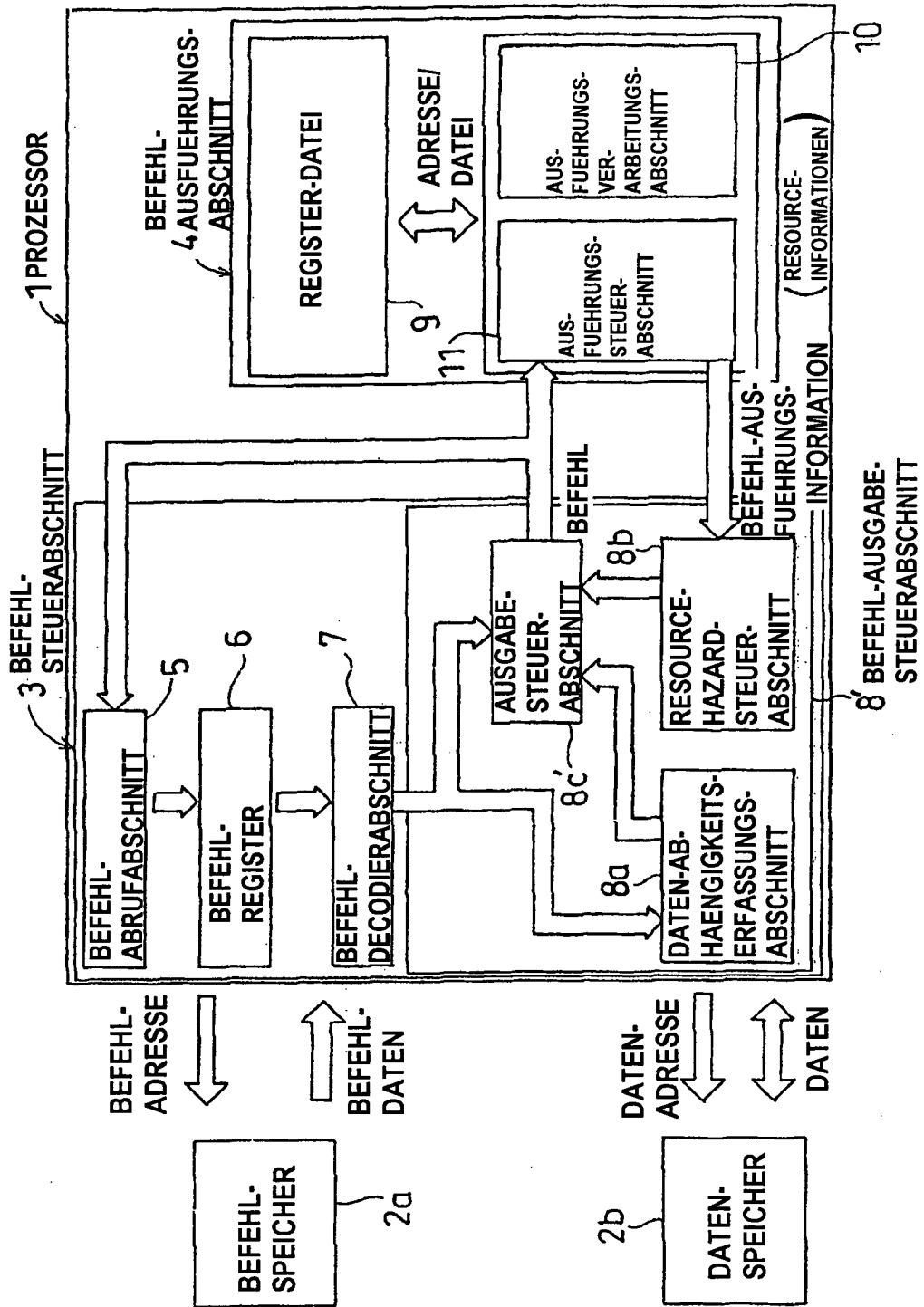


Fig.16

$C = A + B$	LESEN VON A UND B	OPERATION A+B	DURCH- GANG	DURCH- GANG	SCHREIBEN VON DATEN C
$D = E + C$ ($= E + A + B$)	LESEN VON E, A, UND B	OPERATION E+A+B	DURCH- GANG	DURCH- GANG	SCHREIBEN VON DATEN D
$G = F + D$	Dummy	LESEN VON F UND D	OPERATION F+D	DURCH- GANG	SCHREIBEN VON DATEN G
$H = I + G$ ($= I + F + D$)	Dummy	LESEN VON I, F, UND D	OPERATION I+F+D	DURCH- GANG	SCHREIBEN VON DATEN H
$J = K + H$	Dummy	Dummy	LESEN VON K UND H	OPERATION K+H	SCHREIBEN VON DATEN J
$L = M + J$ ($= M + K + H$)	Dummy	Dummy	LESEN VON M, K, UND H	OPERATION M+K+H	SCHREIBEN VON DATEN L

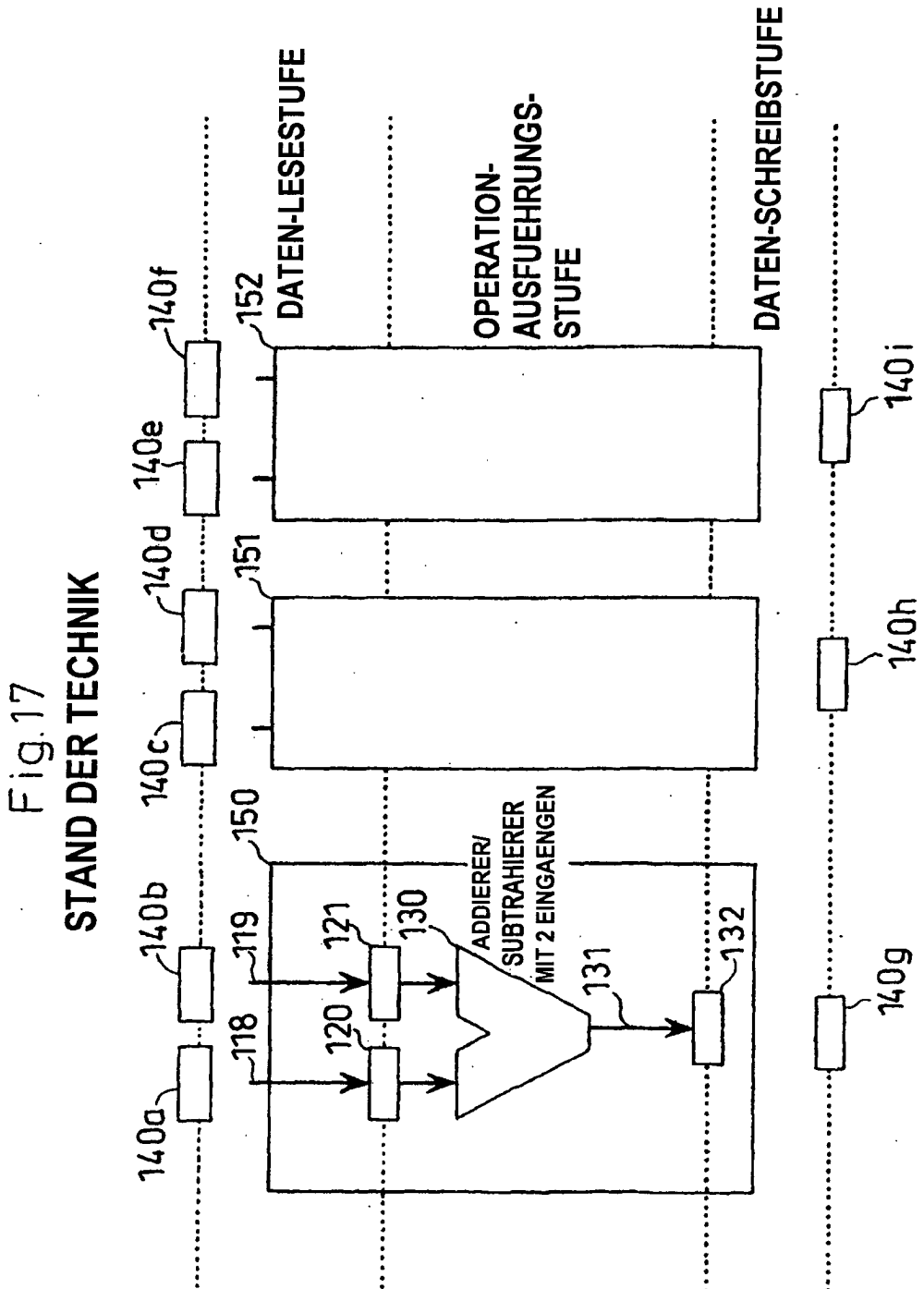


Fig.18(a)
STAND DER TECHNIK

$C = A + B$	LESEN VON DATEN A UND B	AUSFUEHRUNG VON OPERATION A+B	SCHREIBEN VON DATEN C
$D = E + F$	LESEN VON DATEN E UND F	AUSFUEHRUNG VON OPERATION E+F	SCHREIBEN VON DATEN D
$G = H + I$	LESEN VON DATEN H UND I	AUSFUEHRUNG VON OPERATION H+I	SCHREIBEN VON DATEN G

Fig.18(b)
STAND DER TECHNIK
 $C = A + B$

$C = A + B$	LESEN VON DATEN A UND B	AUSFUEHRUNG VON OPERATION A+B	SCHREIBEN VON DATEN C
$D = E + C$	← AUSFUEHRUNG VON WARTE-ZYKLUS →	LESEN VON DATEN E UND C	AUSFUEHRUNG VON OPERATION E+C
			SCHREIBEN VON DATEN D
$G = H + I$	LESEN VON DATEN H UND I	AUSFUEHRUNG VON OPERATION H+I	SCHREIBEN VON DATEN G

Fig.18 (c)
STAND DER TECHNIK
 $C = A + B$

$C = A + B$	LESEN VON DATEN A UND B	AUSFUEHRUNG VON OPERATION A+B	SCHREIBEN VON DATEN C
$D = E + C$	← AUSFUEHRUNG VON WARTE-ZYKLUS →	LESEN VON DATEN E UND C	AUSFUEHRUNG VON OPERATION E+C
			SCHREIBEN VON DATEN D
$G = H + I$	← AUSFUEHRUNG VON WARTE-ZYKLUS →	LESEN VON DATEN H UND I	AUSFUEHRUNG VON OPERATION H+I
			SCHREIBEN VON DATEN G