(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 698 25 344 T2** 2005.07.21

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 864 960 B1**
(21) Deutsches Aktenzeichen: **698 25 344.2**
(96) Europäisches Aktenzeichen: **98 301 933.2**
(96) Europäischer Anmeldetag: **13.03.1998**
(97) Erstveröffentlichung durch das EPA: **16.09.1998**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **04.08.2004**
(47) Veröffentlichungstag im Patentblatt: **21.07.2005**

(51) Int Cl.[7]: **G06F 1/26**

(30) Unionspriorität:
9705314 **14.03.1997** **GB**

(84) Benannte Vertragsstaaten:
DE, FR, GB, IT

(73) Patentinhaber:
Nokia Corp., Espoo, FI

(74) Vertreter:
TER MEER STEINMEISTER & Partner GbR Patentanwälte, 81679 München

(72) Erfinder:
Gabzdyl, Rebecca, Frimley, Surrey, GU16 5TR, GB; Lansdowne, Richard, Chandlers Ford, Hampshire, SO53 5NQ, GB; McGovern, Brian Patrick, Camberley, Surrey, GU15 2SE, GB

(54) Bezeichnung: **Energieeinsparung in einer Datenverarbeitungsschaltung**

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 Datenverarbeitungsschaltung, die mehrere Elemente besitzt, die in Reaktion auf codierte Befehle arbeiten, während Taktsignale empfangen werden.

[0002] Es sind Datenverarbeitungsumgebungen bekannt, in denen Elemente in Reaktion auf decodierte Befehle arbeiten, während Taktsignale empfangen werden. Ein Taktsignal wird oft durch einen Quarzoszillator oder in ähnlicher Weise erzeugt, wobei die Signale über die ganze Schaltung verteilt werden, um einen gemeinsamen Bezug für Manipulationen, die in den Elementen ausgeführt werden, bereitzustellen. Die Gesamtverarbeitungsgeschwindigkeit einer Vorrichtung kann durch die Erhöhung der Frequenz des Taktgebers erhöht werden, auch wenn eine Obergrenze erreicht wird, über der ein zufrieden stellender Betrieb nicht möglich ist. Abgesehen davon führt das Takten von Elementen in einer Schaltung auch zu einer Verlustleistung, die ein wichtiger Faktor ist, wenn die betrachtete Vorrichtung mit einer begrenzten Leistungsversorgung wie etwa der, die durch eine Batterie bereitgestellt wird, betrieben wird.

[0003] Ein bekannter Zugang zur Verringerung der Leistungsaufnahme besteht darin, den Betrieb des Taktgebers wirksam zu unterbrechen, wenn keine Verarbeitung in der Schaltung erforderlich ist. Es heißt, dass der Chip in seine Leerlaufbetriebsart versetzt wird, wobei zusätzliche Komponenten erforderlich sind, um die Schaltung wahlweise in eine ihrer Betriebsarten zu versetzen; d. h. in ihre aktive Leistung verbrauchende Betriebsart oder in ihre Leerlaufbetriebsart mit verringerter Verlustleistung. Ein Problem bei diesem Zugang besteht darin, dass die Unterbrechung des Taktgebers ein vollständiges Abschalten der Schaltung zur Folge hat, so dass Zeiträume, während denen der Taktgeber tatsächlich unterbrochen werden kann, begrenzt sind, wobei die Leistungseinsparung durch zusätzliche Maßnahmen, die erfolgen müssen, wenn Abschaltzeiträume ausgewählt werden, aufgehoben wird.

[0004] Das Dokument US 5.392437 beschreibt einen Zugang, bei dem die Takt-Torsteuerung gemäß der Steuerung der jeweiligen Funktionseinheit erfolgt. Jede Funktionseinheit besitzt ihren eigenen unabhängigen Taktgeber.

[0005] Das Dokument US 5.586.332 beschreibt einen Haupttaktgeber, der sekundäre Nebentaktgeber ansteuert. Der Haupttaktgeber bleibt jederzeit eingeschaltet, wobei er den Start und die Unterbrechung eines der sekundären Taktgeber ansteuert. Dieser sekundäre Taktgeber wird gemäß Befehlen, die von dem Prozessor ausgeführt werden, ein- und ausgeschaltet. Nichts vom Stand der Technik lehrt einen einzelnen zentralen Taktgeber, der Taktsignale an entsprechende Elemente einer Datenverarbeitungseinheit bereitstellt, sowie Mittel, die in Reaktion auf Befehle von einem Befehlsdecodierer Taktsignale von dem zentralen Taktgeber wahlweise freigeben oder sperren, wobei der zentrale Taktgeber in Abhängigkeit davon, ob ein Entsprechendes der Elemente für die Ausführung eines Befehlstyps erforderlich ist, aktiv bleibt.

[0006] Gemäß einem ersten Aspekt der Erfindung wird eine Datenverarbeitungsschaltung geschaffen, die mehrere Elemente umfasst, die in Reaktion auf decodierte Befehle arbeiten, während Taktsignale empfangen werden, gekennzeichnet durch einen zentralen Taktgeber, der nur Taktsignale an entsprechende Elemente der Datenverarbeitungsschaltung bereitstellt; Mittel zum Identifizieren von Befehlstypen, wobei die Mittel zum Identifizieren von Befehlstypen eine Befehlsdecodierungseinheit sind, die einen Befehlstyp-Decodierer besitzt, der auf einen in die Datenverarbeitungsschaltung eingegebenen Befehl anspricht; und Mittel, die in Reaktion auf Befehle von dem Befehlsdecodierer Taktsignale von dem zentralen Taktgeber wenigstens für eines von Entsprechenden der Elemente in Abhängigkeit davon, ob das Entsprechende der Elemente für die Ausführung des Befehlstyps erforderlich ist, wahlweise freigeben oder sperren, wobei der zentrale Taktgeber aktiv bleibt.

[0007] In einer bevorzugten Ausführungsform umfassen die Elemente eine Datenverarbeitungseinheit und eine Datenadressen-Erzeugungseinheit, wobei die Datenverarbeitungseinheit mehr als einen Multiplizierer und mehr als eine Arithmetikeinheit mit zusammenarbeitenden multiplexierten Schaltungen umfassen kann.

[0008] Der Befehlstyp-Decodierer kann vorzugsweise so konfiguriert sein, dass er eine Torsteuerung für Taktsignale ausführt. Torgesteuerte Taktsignale und eingegebene Befehle werden vorzugsweise an mehrere Befehlsdecodierer geliefert. Optional wird ein Befehlstyp-Decodierer so konfiguriert, dass er Befehlsdecodierer für entsprechende Befehlstypen freigibt oder sperrt, wobei die Befehlsdecodierer so konfiguriert werden, dass sie eine Torsteuerung für Taktsignale ausführen.

[0009] Gemäß einem zweiten Aspekt der vorliegenden Erfindung wird ein Verfahren zum Verarbeiten von Daten bereitgestellt, bei dem mehrere Elemente in einer Datenverarbeitungsschaltung in Reaktion auf decodierte Befehle arbeiten, während Taktsignale empfangen werden, gekennzeichnet durch die folgenden Schritte: Vorsehen eines zentralen Taktgebers, der nur Taktsignale an entsprechende Elemente der Datenverarbeitungsschaltung ausgibt; Identifizieren von Befehlstypen, wobei Befehlstypen mittels einer Befehlsdecodierungseinheit identifiziert wer-

den, die einen Befehlstyp-Decodierer besitzt und in Reaktion auf einen Befehl, der in die Datenverarbeitungsschaltung eingegeben wird, arbeitet; und in Reaktion auf Befehle von dem Befehlsdecodierer wahlweises Freigeben oder Sperren von Taktsignalen von dem zentralen Taktgeber an wenigstens eines von Entsprechenden der Elemente in Abhängigkeit davon, ob ein Entsprechendes der Elemente für die Ausführung des Befehlstyps erforderlich ist, wobei der zentrale Taktgeber aktiv bleibt.

[0010] Die Erfindung wird nun lediglich beispielhaft in Bezug auf die beigefügte Zeichnung beschrieben, in der:

[0011] Fig. 1 ein Mobiltelephon zeigt, das eine digitale Signalverarbeitungs-Schaltungsanordnung besitzt;

[0012] Fig. 2 die Funktionsweise der digitalen Signalverarbeitungs-Schaltungsanordnung in dem in Fig. 1 gezeigten Mobiltelephon, die einen digitalen Signalprozessor umfasst, zeigt;

[0013] Fig. 3 einen digitalen Signalprozessor des in Fig. 2 angegebenen Typs, der eine Befehlsdecodierungseinheit und eine Datenverarbeitungseinheit umfasst, ausführlich darstellt;

[0014] Fig. 4 die in Fig. 3 gezeigte Befehlsdecodierungseinheit, die einen Befehlstyp-Decodierer umfasst, ausführlich darstellt;

[0015] Fig. 5 den in Fig. 4 angegebenen Befehlstyp-Decodierer ausführlich darstellt;

[0016] Fig. 6 die in Fig. 3 angegebene Datenverarbeitungseinheit, die zwei Multiplizierer, zwei Arithmetik-Logik-Einheiten, eine Vergleichs- und Auswahleinheit und eine Normierungseinheit zum Erleichtern der Verarbeitung von Gleitkommazahlen besitzt, ausführlich darstellt;

[0017] Fig. 7 einen Block von Gleitkommazahlen zeigt;

[0018] Fig. 8 einen dem in Fig. 7 gezeigten Block ähnlichen Block von Gleitkommazahlen, die normiert sind, so dass sie zueinander passende Exponentenwerte besitzen, zeigt;

[0019] Fig. 9 einen der in Fig. 6 gezeigten Multiplizierer ausführlich darstellt;

[0020] Fig. 10 eine der in Fig. 6 gezeigten Arithmetik-Logik-Einheiten ausführlich darstellt;

[0021] Fig. 11 die in Fig. 6 gezeigte Vergleichs- und Auswahleinheit ausführlich darstellt; und

[0022] Fig. 12 die in Fig. 6 gezeigte Normierungseinheit ausführlich darstellt.

[0023] Die vorliegende Erfindung besitzt viele Anwendungen insbesondere in Systemen, in denen die Leistungsaufnahme zu minimieren ist. Als ein Beispiel einer solchen Umgebung wird die Erfindung anhand eines Mobiltelephons beschrieben, obgleich klar ist, dass viele andere Anwendungen ebenso relevant sind.

[0024] In Fig. 1 ist ein Mobiltelephon gezeigt, das konfiguriert ist, um in Übereinstimmung mit den GSM-Empfehlungen digital zu arbeiten. Das Telephon umfasst ein Mikrophon **101**, das ausgelegt ist, um Sprachäußerungen zu empfangen, in Verbindung mit einem Lautsprecher **102**, der ausgelegt ist, um empfangene Stimmsignale an ein Ohr des Bedieners zu liefern. Das Telephon umfasst manuell bedienbare Tasten **103** und eine Sichtanzeige **104**.

[0025] Das in Fig. 1 gezeigte Mobiltelephon umfasst eine digitale Datenverarbeitungsschaltung **201**, wie sie in Fig. 2 gezeigt ist. Die Verarbeitungsschaltung empfängt digitale Eingangssignale von einem Analog/Digital-Umsetzer **202**, der seinerseits analoge Sprachsignale vom Mikrophon **101** empfängt. In der Datenverarbeitungsschaltung **201** werden digitale Signale von dem Analog/Digital-Umsetzer **202** an das Sprachcodierungs-Untersystem **203** geliefert, das seinerseits codierte Sprache an ein Kanalcodierungs- und Redundanz-Untersystem **204** liefert. Ein Lautsprecher **102** empfängt über einen Digital/Analog-Umsetzer **205** ein Ausgangssignal von der Datenverarbeitungsschaltung **201**. Eine Funkantenne **206** empfängt Übertragungssignale von einem Sender **207**, der seinerseits so beschaffen ist, dass er Ausgangssignale von dem Kanalcodierungs-Untersystem **204** empfängt. Die Antenne **206** wird außerdem gemeinsam mit einer Empfangsschaltung **208** genutzt, die ihrerseits Eingangssignale an ein Kanalrekonstruktions- und Kanaldecodierungs-Untersystem **209** in der Datenverarbeitungsschaltung **201** liefert. Das Ausgangssignal vom Untersystem **209** wird an ein Sprachdecodierungs-Untersystem **210** geliefert, das seinerseits ein digitales Sprachsignal an den Digital/Analog-Umsetzer **205** bereitstellt.

[0026] Die Datenverarbeitungsschaltung **201** ist als eine programmierbare Vorrichtung implementiert, die die Prozesse **203**, **204**, **209** und **210** multiplexieren kann. Eine Hardware-Implementierung der Verarbeitungsschaltung **201** ist in Fig. 3 gezeigt und umfasst einen Programmspeicher **301** und einen Datenspeicher **302**. Die Verarbeitungsschaltung **201** ist ereignisgesteuert, was bedeutet, dass die Schaltung in eine Leerlaufbetriebsart versetzt wird, wenn keine Unterbrechungen anstehen. Unterbrechungssignale werden über die Unterbrechungsleitungen **304** an einen Unterbrechungsbehandler **303** geliefert. Beim

Empfang einer Unterbrechung liefert der Unterbrechungsbehandler ein Freigabesignal über eine Leitung **305** an eine Phasenregelkreis-Schaltung **306**. Die Phasenregelkreis-Schaltung empfängt über eine Leitung **307** Taktsignale von einem Systemtaktgeber und ist so beschaffen, dass sie über eine Leitung **309** Taktsignale höherer Frequenz an einen Taktgeber **308** liefert.

[0027] In herkömmlichen Systemen ist der Taktgeber **308** so beschaffen, dass er Taktsignale an praktisch alle aktiven Elemente in der Schaltung liefert. Allerdings stellt in Übereinstimmung mit der vorliegenden Erfindung eine Befehlsdecodierungseinheit **320** über eine Leitung **319** Taktfreigabesignale an einem Taktgeber **308** bereit. Diessteuert die Lieferung der Taktsignale an die aktiven Elemente in dem System.

[0028] Die Unterbrechungsbehandlungsschaltung **303** liefert über die Leitung **309** außerdem Unterbrechungssignale an eine Befehlsabrufeinheit **312**, die ihrerseits eine Adresse über einen Adressenbus **313** an den Programmspeicher **301** erzeugt. Der resultierende Befehl, der aus dem Programmspeicher gelesen wird, wird über einen Befehlsbus **314** an die Befehlsdecodierungseinheit **320** zurückgegeben. Die Befehlsdecodierungseinheit **320** decodiert den aus dem Programmspeicher **301** gelesenen Befehl und liefert geeignete Steuersignale sowohl an eine Datenverarbeitungseinheit **315** und an eine Datenadressen-Erzeugungseinheit **316** als auch an andere Einheiten. Die Datenadressen-Erzeugungseinheit **316** adressiert den Datenspeicher über einen Adressenbus **317**, wobei die Datenübertragung zwischen dem Datenspeicher **302** und der Datenverarbeitungseinheit **315** oder der Datenadressen-Erzeugungseinheit **316** über einen Datenbus **318** erfolgt.

[0029] In Fig. 4 ist ausführlich die Befehlsdecodierungs- und Taktgebereinheit **310** dargestellt, in der Befehle an das Befehlsregister **401** und an den Befehlstyp-Decodierer **402** geliefert werden, die über den Befehlsbus **314** von dem Programmspeicher **301** empfangen werden. Das Befehlsregister **401** löst den Beginn der Befehlsdecodierung aus, so dass ein codierter Befehl zu entsprechenden Steuersignalen führt, die an die Datenverarbeitungseinheit **315** zusammen mit irgendwelchen anderen erforderlichen Komponenten geliefert werden.

[0030] Die Befehlsdecodierungs- und Taktgebereinheit **310** umfasst außerdem mehrere Decodierer für einen Befehlstyp **403**, **404**, **405** und **406**. Eine Ausgabe von dem Befehlstyp-Decodierer **402** wird an ein Befehlstyp-Register **407** geliefert, wobei die durch das Befehlsregister **401** eingebrachte Verzögerung ausgeglichen wird. Das Befehlstyp-Register **403** empfängt außerdem das von dem Taktgeber **308** auf der Leitung **311** erzeugte Taktsignal, das auch an das Befehlsregister **401** und an jeden Decodierer **403** bis **406** für einen Befehlstyp geliefert wird. Auf ähnliche Weise werden die Ausgabe von dem Befehlstyp-Register **407** und die Ausgabe von dem Befehlsregister **401** an jeden Decodierer **403** bis **406** für einen Befehlstyp geliefert.

[0031] Die Decodierer **403** bis **406** für einen Befehlstyp sind alle so beschaffen, dass sie Taktsignale und Steuersignale an entsprechende Teile der Datenverarbeitungsschaltung übermitteln. Folglich ist die Befehlsdecodierungs- und Taktgebereinheit **310** zusätzlich zur Verteilung der Steuersignale auch für die Verteilung der Taktsignale verantwortlich. Abgesehen davon werden diese Taktsignale derart torgesteuert, dass für einen bestimmten Befehl Taktsignale lediglich an die relevanten Teile der Schaltung geliefert werden, so dass die Teile der Schaltung, die nicht erforderlich sind, um einen bestimmten Befehl zu implementieren, nicht aktiv sind und deshalb nicht unnötig Leistung verbrauchen.

[0032] Der Befehlstyp-Decodierer **403** ist in Fig. 5 ausführlich dargestellt. Der Decodierer umfasst einen Multiplexer **501**, einen Decodierer **502** und einen Taktschalter **503**. Der Ausgang von dem Befehlstyp-Register **407** umfasst einen Mehrbit-Bus, wobei von dem Bus zu jedem der Decodierer **403** bis **406** eine Leitung geführt wird, die anzeigt, dass ein bestimmter Decodierer freizugeben ist. Dieses Freigabesignal wird folglich an den Multiplexer **501** geliefert, der das Übermitteln von Daten von dem Befehlsregister **401** an den Decodierer **502** zulässt, wenn er freigegeben ist. Falls der Multiplexer **501** kein Freigabesignal von dem Befehlstyp-Register **407** empfängt, wird alternativ ein "Nulloperations"-Signal an den Decodierer **502** übermittelt, das angibt, dass dieser Teil der Schaltung während des betrachteten Taktzyklus keine Operation auszuführen braucht.

[0033] Falls ein Befehl über den Multiplexer **501** an den Decodierer **502** übermittelt worden ist, wird der Befehl in dem Decodierer **502** decodiert, was Steuersignale zu Folge hat, die über die Steuerleitungen **504** geliefert werden. Außerdem wird ein Freigabesignal über die Leitung **505** an den Taktschalter **503** geliefert, was zur Folge hat, dass das Prozessortaktsignal auf der Leitung **311** zu den torgesteuerten Taktsignalleitungen **506** torgesteuert wird. Wenn aktive Steuersignale auf den Steuersignalleitungen **504** geliefert werden, werden folglich passende Taktsignale über die Leitungen **506** durch die ganze Schaltung verteilt. Wenn diese Elemente in Reaktion auf die Steuersignale auf der Leitung **504** nicht arbeiten müssen, wird der Taktschalter jedoch ausgeschaltet, die Taktsignale werden gesperrt und die relevanten Schaltungselemente werden in einem Ruhezustand gelassen. Auch wenn die Komponenten Leistung aufnehmen, werden sie somit nicht tatsächlich getaktet. Die Schaltungselemente sind so konstruiert, dass

die Leistungsaufnahme fast null ist, wenn sie in einen nicht getakteten Zustand versetzt werden, was durch eine Implementierung unter Verwendung der CMOS-Technologie möglich ist.

[0034] In Fig. 6 ist die Datenverarbeitungseinheit **315** ausführlich dargestellt. Die Datenverarbeitungseinheit ist so beschaffen, dass sie durch Bereitstellen eines ersten Multiplizierers **601** und eines zweiten Multiplizierers **602** Multiplikationsoperationen ausführt. Jeder Multiplizierer stellt eine 34-Bit-Ausgabe über entsprechende Busse **603** und **604** an entsprechende Arithmetik-Logik-Einheiten bereit, die aus einer ersten Arithmetik-Logik-Einheit **605**, die so ausgelegt ist, dass sie Daten von dem Bus **603** empfängt, und aus einer zweiten Arithmetik-Logik-Einheit **606**, die so ausgelegt ist, dass sie Daten von dem Bus **604** empfängt, bestehen.

[0035] Der Datenbus **407** ist für die Datenübertragung zwischen der Datenverarbeitungseinheit **405** und dem Datenspeicher **402**, die in Fig. 4 gezeigt sind, vorgesehen. Nunmehr in Fig. 6 ist dieser Datenbus als zwei 16-Bit-Busse implementiert, von denen ein Erster, der erste Datenbus **607**, das Lesen aus dem Datenspeicher **402** und das Schreiben in diesen Datenspeicher ermöglicht. Ein zweiter Datenbus **608** ist ähnlich zum Datenbus **607**, wobei er jedoch lediglich zum Lesen aus dem Datenspeicher **402** konfiguriert ist.

[0036] Die Datenverarbeitungseinheit umfasst einen Ausgangsmultiplexer **609**, der konfiguriert ist, um zu ermöglichen, dass mehrere Ausgangssignale wahlweise an den ersten Datenbus **607** oder an die Datenadressen-Erzeugungseinheit **406** zurückgegeben werden.

[0037] Um die Datenübertragung von dem Datenspeicher zu ermöglichen, kann ein ganzer Taktzyklus verwendet werden, vorausgesetzt, dass jede Übertragung an die Datenverarbeitungseinheit **405** durch eines von acht Registern **610** bis **617** gepuffert wird. Die Verarbeitungseinheit umfasst ferner eine Vergleichs- und Auswahleinheit **618** sowie eine Normierungseinheit **619**, wobei die ganze Einheit von einer gemeinsamen Taktquelle getaktet wird. Um die Operationsgeschwindigkeit effektiv zu verdoppeln, werden allerdings die Vorrichtungen links von der Vergleichs- und Auswahleinheit **618** bei der ansteigenden Flanke der Taktquelle getaktet, während die Vorrichtungen rechts von der Vergleichs- und Auswahleinheit **618** bei der fallenden Flanke der Taktquelle getaktet werden. Die Vergleichs- und Auswahleinheit **618** muss beide Hälften der Einheit bedienen und wird deshalb sowohl bei der ansteigenden Flanke als auch bei der fallenden Flanke der Taktquelle getaktet. Das Eingangsregister **610** empfängt Daten von dem ersten Datenbus **607** und taktet beim Empfangen einer ansteigenden Taktflanke 16-Bit-Daten über einen Bus **620** zu der Normierungseinheit **619**. Das Eingangsregister **611** taktet bei einer ansteigenden Flanke Daten von dem zweiten Datenbus **608** über den Bus **621** zu dem ersten Multiplizierer **601** und zu der ersten Arithmetik-Logik-Einheit **605**. Das Eingangsregister **612** taktet bei einer ansteigenden Flanke Daten von dem ersten Datenbus **607** über den Bus **622** zu dem ersten Multiplizierer **601**. Das Eingangsregister **613** taktet bei einer ansteigenden Flanke Daten von dem zweiten Datenbus **608** über den Bus **623** zu der ersten Arithmetik-Logik-Einheit **605**.

[0038] Das Eingangsregister **614** taktet bei einer fallenden Flanke Daten von dem ersten Datenbus **607** über einen Bus **624** zu der zweiten Arithmetik-Logik-Einheit **606**. Das Eingangsregister **615** taktet bei einer fallenden Flanke Daten von dem zweiten Datenbus **608** über den Bus **625** zu dem zweiten Multiplizierer **602** sowie zu der zweiten Arithmetik-Logik-Einheit **606**. Das Eingangsregister **616** taktet bei einer fallenden Flanke Daten von dem ersten Datenbus **607** über einen Bus **625** zu dem zweiten Multiplizierer **602** und zu der zweiten Arithmetik-Logik-Einheit **606**. Das Eingangsregister **617** taktet bei einer fallenden Flanke Daten von dem zweiten Datenbus **608** über den Bus **626** zu der zweiten Arithmetik-Logik-Einheit **606**.

[0039] Die über den Bus **603** an die erste Arithmetik-Logik-Einheit **605** gelieferte Ausgabe von dem ersten Multiplizierer **601** wird außerdem über einen ähnlichen Bus **626** an den Ausgangsmultiplexer **609** geliefert. Die Ausgabe von der Arithmetik-Logik-Einheit **605** wird über einen Bus **627** an einen Eingang des ersten Multiplizierers **601** zurückgegeben, wobei der Bus außerdem eine Ausgabe an den Ausgangsmultiplexer **609** liefert. In ähnlicher Weise wird diese Ausgabe auch an einen Eingang der zweiten Arithmetik-Logik-Einheit, effektiv als eine Verlängerung desselben Busses, zurückgegeben.

[0040] Der Zehnbit-Bus **628** und die Einbit-Leitungen **629** und **630** liefern Daten an die Ablaufsteuerung **403**, um bedingte Sprünge zu erleichtern. Die zweite Arithmetik-Logik-Einheit **606** kann keinen bedingten Sprung aufrufen, es können jedoch Überlaufbedingungen über zwei Steuerleitungen **631** angegeben werden. Die Ausgabe von der zweiten Arithmetik-Logik-Einheit **606** wird über einen Bus **632** an einen Eingang der ersten Arithmetik-Logik-Einheit **605** zurückgegeben, wobei der Bus die Ausgabe auch an den Ausgangsmultiplexer **609** und an einen Eingang zu dem zweiten Multiplizierer **602** liefert. Eine Ausgabe von der ersten Arithmetik-Logik-Einheit **605** wird über einen 32-Bit-Bus **633** an die Vergleichs- und Auswahleinheit **618** geliefert, während eine ähnliche Ausgabe von der zweiten Arithmetik-Logik-Einheit **606** über einen 32-Bit-Bus **634** an die Vergleichs- und Auswahleinheit **618** geliefert wird.

[0041] Die Normierungseinheit **619** erzeugt einen Ausgabeexponenten, der über einen Bus **635** an den Ausgangsmultiplexer **609** und an die Vergleichs- und Auswahleinheit **618** geliefert wird. Die Normierungseinheit **619** erzeugt Schiebesignale, die über die Schiebebusse **636** und **637** an die erste Arithmetik-Logik-Einheit **605** geliefert werden. Eine 42-Bit-Ausgabe wird von der ersten Arithmetik-Logik-Einheit **605** über einen Bus **638** an die Normierungseinheit **619** geliefert, während ein Bus **639** eine Ausgabe von der Vergleichs- und Auswahleinheit **618** an die Normierungseinheit **619** und an den Ausgangsmultiplexer **609** liefert. Die Daten auf dem Bus **639** beziehen sich auf den Unterzyklus der steigenden Flanke für die linke Seite, wobei ein ähnliches Ausgangssignal von der Vergleichs- und Auswahleinheit, das den Unterzyklus der fallenden Flanke für die rechte Seite betrifft, über einen Bus **640** an den Ausgangsmultiplexer **609** geliefert wird.

[0042] Für die Architektur, die in Fig. 6 gezeigt und in Bezug auf eine spezifische Ausführungsform der Erfindung beschrieben ist, stellen die Arithmetik-Logik-Einheiten **605**, **606** entsprechende Exponentenwerte aus den Registern **611**, **612**, **615**, **616** an der Vergleichs- und Auswahleinheit **618** bereit. Optional kann eine Architektur so konfiguriert werden, dass diese Register **611**, **612**, **615**, **616** Exponentenwerte direkt bereitstellen, um eine Auswahleinheit **618** zu vergleichen.

[0043] Eine Vergleichs- und Auswahleinheit **618** arbeitet sowohl mit der ersten Arithmetik-Logik-Einheit als auch mit der zweiten Arithmetik-Logik-Einheit zusammen, wobei sie auf diese Weise ein Paar von Exponentenwerten vergleichen kann, woraufhin je nach dem Ergebnis des Vergleichs einer von diesem Paar gespeichert werden kann. Auf diese Weise wird ein aus mehreren Exponentenwerten herausgenommenes äußerstes Beispiel, normalerweise der größte Exponentenwert, von der Vergleichs- und Auswahleinheit gespeichert, um eine nachfolgende Verarbeitung zu erleichtern und um den Operationsaufwand der Arithmetik-Logik-Einheiten deutlich zu verringern. Dieser gespeicherte Wert kann insbesondere über den Bus **639** an die Normierungseinheit **619** geliefert werden. Auf diese Weise kann die Normierungseinheit in Reaktion auf die von der Vergleichs- und Auswahleinheit durchgeführten Vergleiche Schiebesignale an die Arithmetik-Logik-Einheit liefern, um die entsprechende Mantisse zu normieren, die mit den von der Vergleichs- und Auswahleinheit betrachteten Exponenten der Gleitkommazahlen verknüpft ist.

[0044] In Fig. 7 ist ein Block von Gleitkommazahlen gezeigt, der eine Gleitkommazahl **701**, eine Gleitkommazahl **702**, eine Gleitkommazahl **703** und eine Gleitkommazahl **704** umfasst. Selbstverständlich wird ein großer Block von Gleitkommazahlen in der Datenverarbeitung oft angetroffen. Jede Gleitkommazahl besteht aus einer Mantisse und einem Exponententeil. Ein Block von Gleitkommazahlen dieses Typs resultiert typisch aus einem Verarbeitungsalgorithmus, der in einem der in Fig. 2 gezeigten Prozessoren **203**, **204**, **210** oder **209**, die in dem Mobiltelephon arbeiten, verwendet wird. Der Exponent der Gleitkommazahl **701** und der Exponent der Gleitkommazahl **702** sind ähnlich, aber nicht die gleichen. Abgesehen davon sind die anderen Exponenten in dem Block von Gleitkommazahlen wahrscheinlich ähnlich, jedoch nicht völlig gleich. Um eine effiziente Datenverarbeitung an einem Block von Gleitkommazahlen auszuführen, ist es bevorzugt, zusammenpassende Exponenten in dem ganzen Zahlenblock zu haben.

[0045] Die in Fig. 6 gezeigte Anordnung erleichtert eine effiziente Übertragung eines Blocks von Gleitkommazahlen des in Fig. 7 gezeigten Typs in einen Block von Gleitkommazahlen des in Fig. 8 gezeigten Typs, der dadurch gekennzeichnet ist, dass er gleiche Exponenten aufweist. Es gibt zwei Stufen für den Umsetzungsprozess von dem in Fig. 7 dargestellten Block von Gleitkommazahlen in den in Fig. 8 gezeigten Block von Gleitkommazahlen. Der erste Prozess kann während der Erzeugung des Blocks von Gleitkommazahlen oder durch einen nachfolgenden Durchlauf durch den Block von Gleitkommazahlen, während dessen die Exponenten verglichen werden und der größte Exponent beibehalten wird, ausgeführt werden. Die Exponenten werden unter Programmsteuerung an die Vergleichs- und Auswahleinheit **618** geliefert, die Vergleiche zwischen einem ankommenden Exponenten und einem aus vorhergehenden Vergleichen beibehaltenen Exponenten mit hohem Wert ausführen kann. Dadurch können die Exponenten in einem Block von Gleitkommazahlen verglichen werden, wobei ein größter Exponent beibehalten wird.

[0046] Die zweite Stufe des Prozesses der Umsetzung von dem in Fig. 7 gezeigten Block von Gleitkommazahlen in den in Fig. 8 gezeigten Block von Gleitkommazahlen wird von der Normierungseinheit **619** ausgeführt. Der Exponent mit dem größten Wert, der in dem ersten Prozess durch die Vergleichs- und Auswahleinheit bestimmt worden ist, wird als eine Zahl beibehalten, die über den Bus **639** an die Normierungseinheit **619** übertragen wird. Die Normierungseinheit empfängt Exponenten für jede Zahl in dem umzusetzenden Block von Gleitkommazahlen, wobei sie eine Subtraktion zwischen dem Exponenten mit dem größten Wert und den Exponenten für jede Zahl in dem Block ausführt, so das die Anzahl von Verschiebungen der Mantisse, die für jede auf den größten Exponentenwert zu normierende Gleitkommazahl erforderlich sind, bestimmt und an die erste Arithmetik-Logik-Einheit **605** geliefert wird. Die erste Arithmetik-Logik-Einheit **605** umfasst eine Schiebeschaltung, die die Mantisse der Gleitkommazahl in einem einzelnen Zyklus um die gewünschte

Anzahl von Binärstellen verschieben kann. Auf diese Weise wird der in Fig. 7 gezeigte Block von Gleitkommazahlen in einem Zweistufenprozess in den in Fig. 8 gezeigten Block von Gleitkommazahlen, dargestellt als Gleitkommazahl **801**, **802**, **803** und **804**, umgesetzt. Dadurch wird die Manipulation eines Blocks normierter Gleitkommazahlen in der Festkomma-Arithmetik-Logik-Einheit **605** erleichtert.

[0047] Der in Fig. 6 gezeigte Multiplizierer **601** ist in Fig. 9 ausführlich dargestellt. Ein erster vorzeichenbehafteter/vorzeichenloser Multiplexer **901** empfängt Steuersignale **911** von der Ablaufsteuerung **403**. Von dem Multiplexer **901** werden von dem Datenbus **621** Datensignale empfangen, die hierauf unter Steuerung der Ablaufsteuerung **403** als vorzeichenbehaftet oder vorzeichenlos betrachtet werden können. Eine ähnliche Anordnung kommt bei einem vorzeichenbehafteten/vorzeichenlosen Multiplexer **902** vor, der Signale von dem Datenbus **622** empfängt. Ein Multiplexer **903** liefert die Signale A und B an einen 17-Bit-Zweierkomplement-Multiplizierer **904**. Das Produkt dieser Multiplikation wird an die Multiplizierer-Schiebe/Rundungs-Einheit **905** geliefert, die durch die Steuersignale **912** von der Ablaufsteuerung **403** gesteuert wird. Die Ausgabe der Multiplizierer-Schiebe/Rundungs-Einheit **905** wird an ein Multipliziererregister **906** geliefert, das von der Ablaufsteuerung **403** über die Steuerleitungen **913** gesteuert wird. Die Ausgabe des Multipliziererregisters **906** bildet den Bus **603**, der als eine Eingabe an den Multiplexer **903** geliefert werden kann, was folglich die Rückkopplung von Multiplikationsprodukten in der Multiplizierereinheit **601** selbst erleichtert. Der Multiplexer **903** kann außerdem vorgegebene feste Nullwerte als einen oder als beide der Multiplikationsoperanden empfangen, so dass die Schaltungsanordnung in dem Multiplizierer selbst auf einen statischen Wert gesetzt werden kann, wodurch die Leistungsaufnahme verringert wird.

[0048] Die in Fig. 6 gezeigte erste Arithmetik-Logik-Einheit **605** ist in Fig. 10 ausführlich dargestellt. Ein Multiplexer **1001** wählt Signale vom Bus **621**, vom Bus **603**, vom Bus **632**, vom Bus **622** oder den numerischen Wert null aus. Der Multiplexer **1001** wird durch Steuersignale gesteuert, die von der Ablaufsteuerung **403** geliefert werden. Eine Arithmetik-Logik-Schaltung **1002** wird durch die Steuerleitungen **1102**, die von der Ablaufsteuerung **403** empfangen werden, gesteuert. Die Arithmetik-Logik-Schaltung **1002** besitzt einen numerischen Ausgang **1203**, der an eine Arithmetik-Logik-Schiebeeinheit **1003** geliefert wird. Die Arithmetik-Logik-Schaltung **1002** liefert ferner Bedingungssignale an eine Bedingungslogik **1008**, **1009**, die bedingte Befehle und die Erfassung einer Ablaufunterbrechung erleichtert.

[0049] Die Arithmetik-Logik-Schiebeeinheit **1003** wird über die Steuerlogik **1004** durch Steuerleitungen **1103**, die von der Ablaufsteuerung **403** zugeführt werden, gesteuert. Die Steuerlogik **1004** wird außerdem durch die Busse **636** und **637** beliefert, die verwendet werden können, um die Anzahl von Binärstellen zu bestimmen, die die Schiebeeinheit einer über den Bus **1203** empfangenen Binärzahl auferlegt. Mit der Ausgabe der Arithmetik-Logik-Schiebeeinheit **1003** wird eine Rundungseinheit **1005** beliefert, die über eine Steuerleitung **1104** von der Ablaufsteuerung **403** gesteuert wird. Die Rundungseinheit **1005** wird verwendet, um hoch genaue, aber lange Binärzahlen mit 42 Bits in kürzere Darstellungen dieser Zahlen umzusetzen, so dass sie effizient im Speicher gespeichert und von anderen Teilen der Datenverarbeitungseinheit, die Eingabewortlängen mit weniger als 42 Bits aufweisen, verarbeitet werden können. Die Rundungseinheit liefert außerdem ein Bedingungssignal an die Bedingungserfassungslogik **1010** und an eine Sättigungsschaltung **1006**.

[0050] Die Sättigungsschaltung **1006** empfängt die numerische Ausgabe von der Rundungsschaltung **1005**. Damit die großen Binärzahlen effizient auf eine zur Speicherung oder zur nachfolgenden Verarbeitung geeignete Wortlänge verringert werden können, kann die Sättigungsschaltung **1006** wiederum eine Begrenzungsfunktion an numerischen Werten ausführen, die einen vorgegebenen Grenzwert überschreiten. Die Sättigungsschaltung **1006** wird durch die Steuerleitung **1105** von der Ablaufsteuerung **403** gesteuert. Die Sättigungsschaltung kann ferner durch ein Bedingungssignal von der Arithmetik-Logik-Schaltung **1002** gesteuert werden, das angeben kann, wenn eine bestimmte Arithmetik-Logik-Operation zu einem numerischen Wert geführt hat, der gesättigt werden muss.

[0051] Die Ausgabe der Sättigungseinheit **1006** wird an das Akkumulatorregister **1007** geliefert, das durch ein Taktsignal **1106** getaktet wird. Das Akkumulatorregister **1007** liefert Signale an die Busse **627** und **628**. Dadurch ist es möglich, dass das Akkumulatorregister **1007** einen numerischen Wert für weitere Arithmetik-Logik-Operationen zurück an den Multiplexer **1001** liefert.

[0052] Der Bedingungserzeugungsblock **1011** empfängt numerische Werte von der Arithmetik-Logik-Schiebeeinheit **1003** und eine Bedingungssignalisierung von der Signalisierungslogik **1008**, **1009**, **1010** und **1012**. Die Signale **1012** werden von der Vergleichs- und Auswahleinheit **618** geliefert. Der Bedingungserzeugungsblock **1011** empfängt außerdem Signale von dem Bus **631**. Die Bedingungserzeugungseinheit liefert die Signale **629** und **630**.

[0053] Die in Fig. 6 gezeigte Vergleichs- und Auswahleinheit **618** ist in Fig. 11 ausführlich dargestellt. Ein Multiplexer **1151** für das niederwertige empfängt Eingangswerte vom Bus **633** oder vom Bus **634**. Al-

ternativ kann ein Nullwert als eine numerische Eingabe in diesen Multiplexer verwendet werden. Ein Multiplexer **1152** für das höherwertige Eingangswort kann ebenso Signale vom Bus **633** oder vom Bus **634** oder einen numerischen Wert null empfangen. Beide Eingabewort-Multiplexer **1151** und **1152** werden durch Steuersignale **1161** und **1162**, die von der Ablaufsteuerung **403** geliefert werden, gesteuert. Der Multiplexer **1151** für das niederwertige Eingabewort kann numerische Werte an eine Vergleichseinheit **1153** oder einen zusätzlichen Multiplexer **1154** liefern. Der Multiplexer **1152** für das höherwertige Eingabewort liefert seinen numerischen Wert an einen zweiten Eingang der Vergleichseinheit **1153** und an einen zweiten Eingang des zusätzlichen Multiplexers **1154**. Eine Vergleichseinheit **1153** erzeugt einen Hinweis darauf, ob das niederwertige Eingabewort vom Multiplexer **1151** größer oder gleich dem höherwertigen Eingabewort, das vom Multiplexer **1152** geliefert wird, ist. Dieser Hinweis wird als ein Signal an eine Maximum- oder Minimum-Mehrfachumkehrschaltung **1155** geliefert, die ein Vergleichssignal erzeugt, das seinerseits an den zusätzlichen Multiplexer **1154** oder an ein Vergleichs-Auswahl-Zustandsregister **1156** geliefert wird.

[0054] Als eine alternative Eingabe kann der Multiplexer **1152** für das höherwertige Eingabewort ferner den Ausgabeexponenten von der Normierungseinheit **619**, der als ein Bus **635** dargestellt ist, empfangen. Die Ausgabe des zusätzlichen Multiplexers **1154** wird als eine Eingabe an ein Vergleichs- und Auswahlregister **1157** geliefert, wobei sein Inhalt typisch den größten einer Folge von in einer Block-Gleitkommaexponenten-Vergleichsprozedur verglichenen Exponenten darstellt. Um eine solche Prozedur zu erleichtern, muss die Ausgabe des Vergleichs- und Auswahlregisters **1157** als ein numerischer Wert über einen internen Bus **1199** und über den Multiplexer **1151** für das niederwertige Eingabewort an den ersten Eingang der Vergleichseinheit **1153** geliefert werden. Wenn die Exponenten an einen Eingang des Multiplexers **1152** für das höherwertige Eingabewort geliefert werden, können daraufhin Vergleiche so ausgeführt werden, dass der größte einer Folge verglichener Exponenten schließlich in dem Vergleichs- und Auswahlregister **1157** gespeichert wird.

[0055] Das Vergleichs- und Auswahlzustandsregister **1156** liefert numerische Werte an die Busse **639** und **640**. Dieses Register empfängt außerdem ein Taktsignal **1165** von der Steuerschaltungsanordnung. Das Vergleichs- und Auswahlregister **1157** wird durch ein Signal **1166** von der Steuerschaltungsanordnung getaktet.

[0056] Die in Fig. 6 gezeigte Normierungseinheit **619** ist in Fig. 12 ausführlich dargestellt. Der Bus **638** liefert numerische 42-Bit-Werte an eine Einheit **1271**, die das höchstwertige Bit des numerischen Werts bestimmt, und codiert diese als einen 6-Bit-Wert, der über den Bus **637** geliefert wird. Dieser 6-Bit-Wert **637** wird ferner an einen Multiplexer **1272** geliefert, der alternativ einen numerischen 16-Bit-Wert von dem Bus **639**, der mit der Vergleichs- und Auswahleinheit **618** verbundenen ist, empfangen kann. Die 16-Bit-Ausgabe des Multiplexers **1272** wird als eine Subtrahiereingabe an einen Subtrahierer **1273** geliefert. Der Multiplexer **1272** wird durch ein Steuersignal **1281** gesteuert. Eine zusätzliche Eingabe an den Subtrahierer **1273** wird vom Bus **620** geliefert. Die Ausgabe des Subtrahierers **1273** wird an ein Ausgabeexponentenregister **1274** geliefert, das durch ein von der Steuerschaltungsanordnung geliefertes Taktsignal **1282** gesteuert wird. Sechs Bits der Ausgabe von dem Subtrahierer **1273** werden als Bus **636** geliefert. Die Ausgabe des Ausgabeexponentenregisters **1274** wird als ein numerischer 16-Bit-Wert an den Bus **635** geliefert.

[0057] Im Hinblick auf die vorhergehende Beschreibung ist für den Fachmann auf dem Gebiet offensichtlich, dass verschiedene Änderungen innerhalb des Umfangs der Erfindung vorgenommen werden können.

[0058] Beispielsweise sind in einer alternativen Ausführungsform der Befehlstyp-Decodierer **402** und das Befehlstyp-Register **407** aus der Befehlsdecodierungseinheit **310** entfernt. Die neue Befehlsdecodierungseinheit **310** umfasst das Befehlsregister **401** und einen Befehlsdecodierer. Wie in der vorherigen Ausführungsform werden Befehle, die von dem Programmspeicher **301** empfangen werden, über den Befehlsbus **314** an das Befehlsregister **401** geliefert. Die Ausgabe von dem Befehlsregister **401** wird an den Befehlsdecodierer geliefert. Der Befehlsdecodierer umfasst die gleichen Komponenten wie der Decodierer für einen Befehlstyp **403** mit der Ausnahme, dass der Multiplexer **501** entfernt ist und dass der Befehl von dem Befehlsregister **401** direkt in den Decodierer **502** übertragen wird. Der Decodierer decodiert den Befehl wie zuvor beschrieben. Somit empfangen lediglich die Elemente des Prozessors, die zur Ausführung des Befehls erforderlich sind, aktive Steuersignale sowie passende Taktsignale. Die Stromaufnahme in den übrigen Schaltungselementen ist fast null, wenn sie unter Verwendung einer Technik wie etwa CMOS implementiert sind. Diese Ausführungsform ist kleiner und daher preiswerter als die vorherige Ausführungsform, wobei sie jedoch mehr Leistung verbraucht, da die ganze Befehlsdecodierungseinheit während einer Befehlsdecodierung aktiv ist. Diese Ausführungsform ist ein äußerstes Beispiel für die Abwägung zwischen der Komplexität des Befehlstyp-Decodierers und der Höhe der Leistungsaufnahme in der Befehlsdecodierungseinheit. Der Fachmann auf dem Gebiet kann bestimmen, welche Höhe der Komplexität in dem Befehlstyp-Decodierer für

eine bestimmte Anwendung geeignet ist.

Patentansprüche

1. Datenverarbeitungsschaltung, die mehrere Elemente umfasst, die in Reaktion auf decodierte Befehle arbeiten, während Taktsignale empfangen werden, gekennzeichnet durch
einen zentralen Taktgeber (**308**), der nur Taktsignale an entsprechende Elemente der Datenverarbeitungsschaltung bereitstellt;
Mittel zum Identifizieren von Befehlstypen, wobei die Mittel zum Identifizieren von Befehlstypen eine Befehlsdecodierungseinheit (**403**, **404**, **405**, **406**) sind, die einen Befehlstyp-Decodierer besitzt, der auf einen in die Datenverarbeitungsschaltung eingegebenen Befehl anspricht; und
Mittel, die in Reaktion auf Befehle von dem Befehlsdecodierer (**403**, **404**, **405**, **406**) Taktsignale von dem zentralen Taktgeber (**308**) wenigstens für eines von Entsprechenden der Elemente in Abhängigkeit davon, ob das Entsprechende der Elemente für die Ausführung des Befehlstyps erforderlich ist, wahlweise freigeben oder sperren, wobei der zentrale Taktgeber aktiv bleibt.

2. Datenverarbeitungsschaltung (**201**) nach Anspruch 1, bei der die Elemente eine Datenverarbeitungseinheit (**315**) und eine Datenadressen-Erzeugungseinheit (**316**) umfassen.

3. Datenverarbeitungsschaltung (**201**) nach Anspruch 2, bei der die Datenverarbeitungseinheit mehr als einen Multiplizierer und mehr als eine Arithmetikeinheit mit zusammenarbeitenden multiplexierten Schaltungen umfasst.

4. Datenverarbeitungsschaltung (**201**) nach Anspruch 3, bei der der Befehlstyp-Decodierer so konfiguriert ist, dass er eine Torsteuerung von Taktsignalen freigibt.

5. Datenverarbeitungsschaltung (**201**) nach Anspruch 4, bei der die torgesteuerten Taktsignale und die eingegebenen Befehle an mehrere Befehlsdecodierer geliefert werden.

6. Datenverarbeitungsschaltung (**201**) nach Anspruch 5, bei der der Befehlstyp-Decodierer so konfiguriert ist, dass er Befehlsdecodierer für entsprechende Befehlstypen freigibt oder sperrt.

7. Datenverarbeitungsschaltung (**201**) nach Anspruch 6, bei der die Befehlsdecodierer so konfiguriert sind, dass sie eine Torsteuerung für Taktsignale ausführen.

8. Verfahren zum Verarbeiten von Daten, bei dem mehrere Elemente (**315**, **316**) in einer Datenverarbeitungsschaltung (**201**) in Reaktion auf decodierte Befehle arbeiten, während Taktsignale empfangen werden, gekennzeichnet durch die folgenden Schritte:
Vorsehen eines zentralen Taktgebers (**308**), der nur Taktsignale an entsprechende Elemente (**315**, **316**) der Datenverarbeitungsschaltung (**201**) ausgibt;
Identifizieren von Befehlstypen, wobei Befehlstypen mittels einer Befehlsdecodierungseinheit identifiziert werden, die einen Befehlstyp-Decodierer (**403**, **404**, **405**, **406**) besitzt und in Reaktion auf einen Befehl, der in die Datenverarbeitungsschaltung eingegeben wird, arbeitet; und
in Reaktion auf Befehle von dem Befehlsdecodierer (**403**, **404**, **405**, **406**) wahlweises Freigeben oder Sperren von Taktsignalen von dem zentralen Taktgeber (**308**) an wenigstens eines von Entsprechenden der Elemente in Abhängigkeit davon, ob ein Entsprechendes der Elemente für die Ausführung des Befehlstyps erforderlich ist, wobei der zentrale Taktgeber aktiv bleibt.

9. Verfahren zum Verarbeiten von Daten nach Anspruch 8, bei dem das Freigeben oder Sperren von Taktsignalen in Bezug auf Taktsignale erfolgt, die an eine Datenverarbeitungseinheit und an eine Datenadressen-Erzeugungseinheit geliefert werden.

10. Verfahren nach Anspruch 9, bei dem die Datenverarbeitungseinheit so konfiguriert ist, dass sie Multiplikationsfunktionen und Arithmetikfunktionen ausführt, die durch mehr als einen Multiplizierer und mehr als eine Arithmetikeinheit mit zusammenarbeitenden multiplexierten Schaltungen implementiert sind.

11. Verfahren nach Anspruch 9, bei dem Taktsignale mittels des Befehlstyp-Decodierers torgesteuert werden.

12. Verfahren nach Anspruch 11, bei dem torgesteuerte Taktsignale und eingegebene Befehle an mehrere Befehlsdecodierer geliefert werden.

13. Verfahren nach Anspruch 1, bei dem der Befehlstyp-Decodierer so konfiguriert ist, dass er Befehlsdecodierer für entsprechende Befehlstypen freigibt oder sperrt.

14. Verfahren nach Anspruch 13, bei dem die Befehlsdecodierer so konfiguriert sind, dass sie eine Torsteuerung für Taktsignale ausführen.

Es folgen 11 Blatt Zeichnungen

Anhängende Zeichnungen

Fig. 1

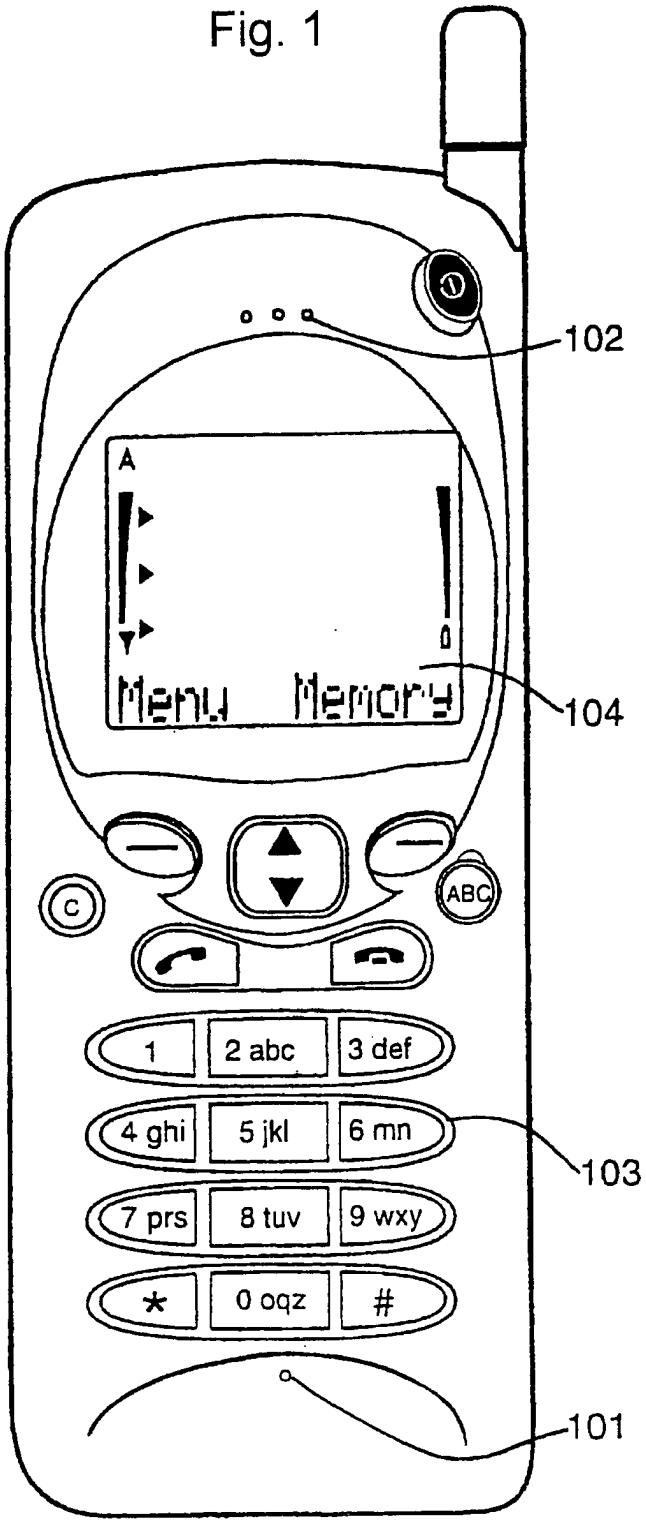


Fig. 2

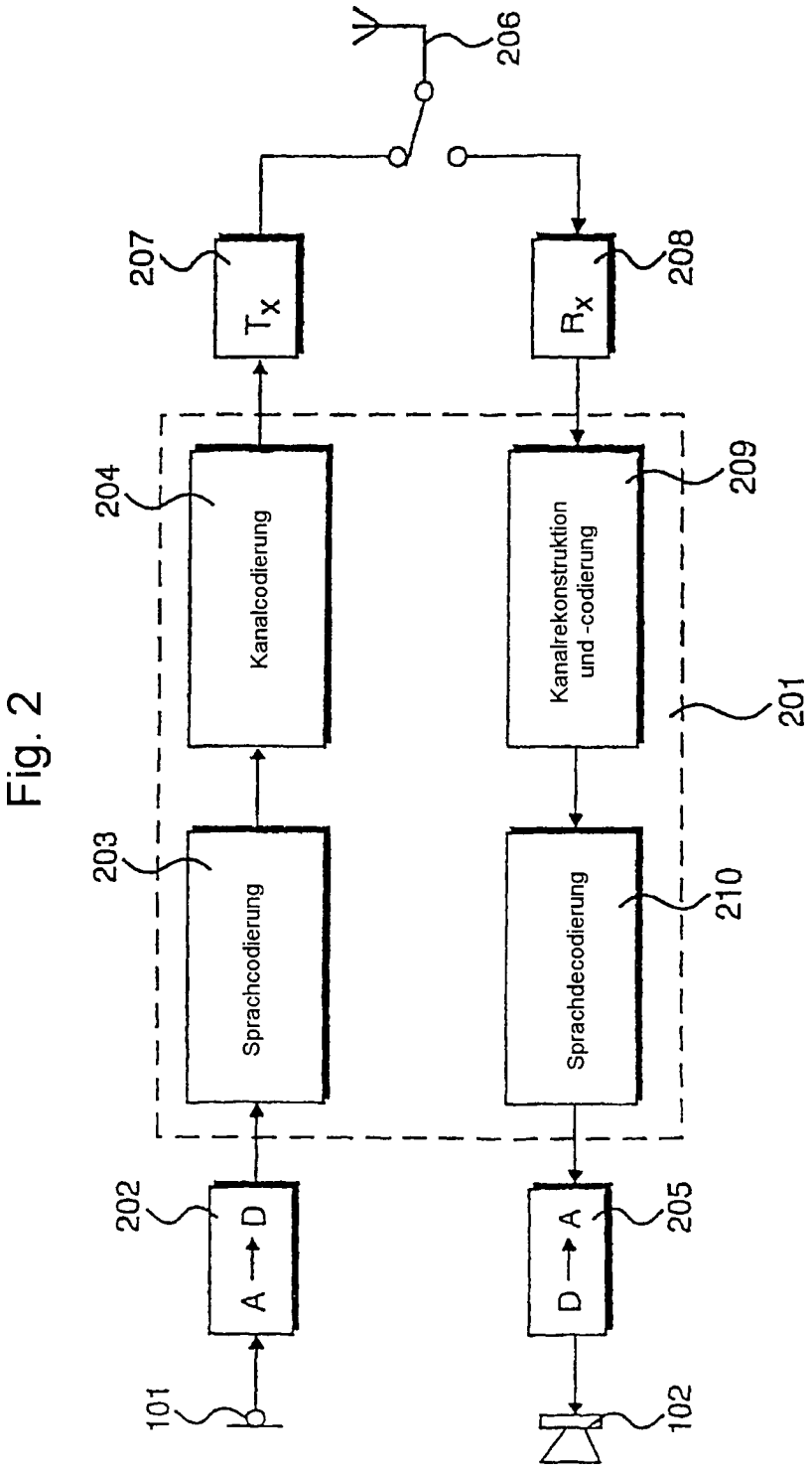
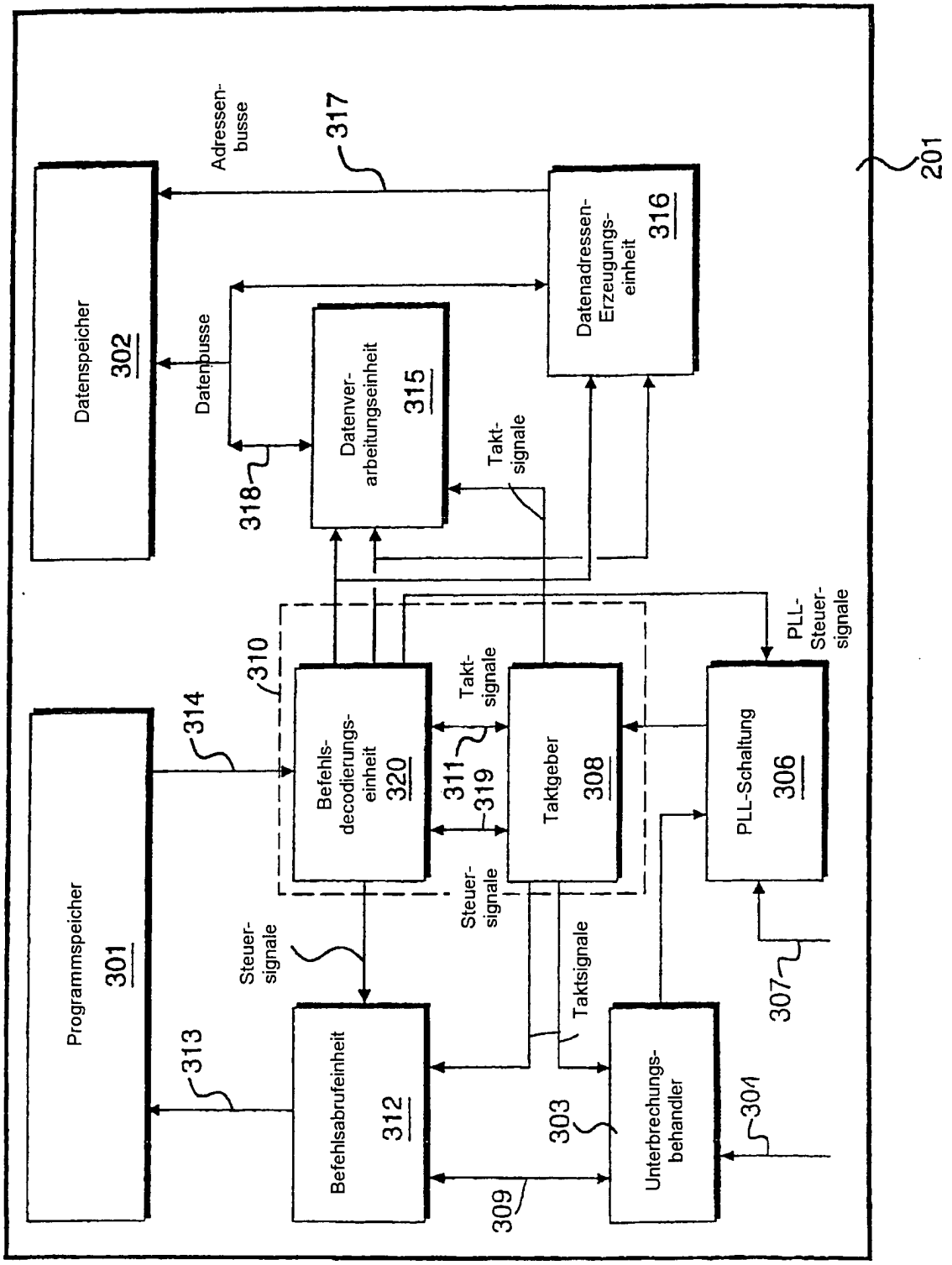

206
207
Tx
208
Rx
204
Kanalcodierung
209
Kanalrekonstruktion und -codierung
201
203
Sprachcodierung
210
Sprachdecodierung
202
A → D
205
D → A
101
102

Fig. 3


201
Adressen-busse
317
Datenspeicher
302
Datenbusse
318
Datenver-arbeitungseinheit
315
Datenadressen-Erzeugungs-einheit
316
Takt-signale
PLL-Steuer-signale
310
Befehls-decodierungs-einheit
320
311
319
Taktgeber
308
PLL-Schaltung
306
314
Programmspeicher
301
Steuer-signale
Steuer-signale
Taktsignale
307
313
Befehlsabrufeinheit
312
303
Unterbrechungs-behandler
304
309

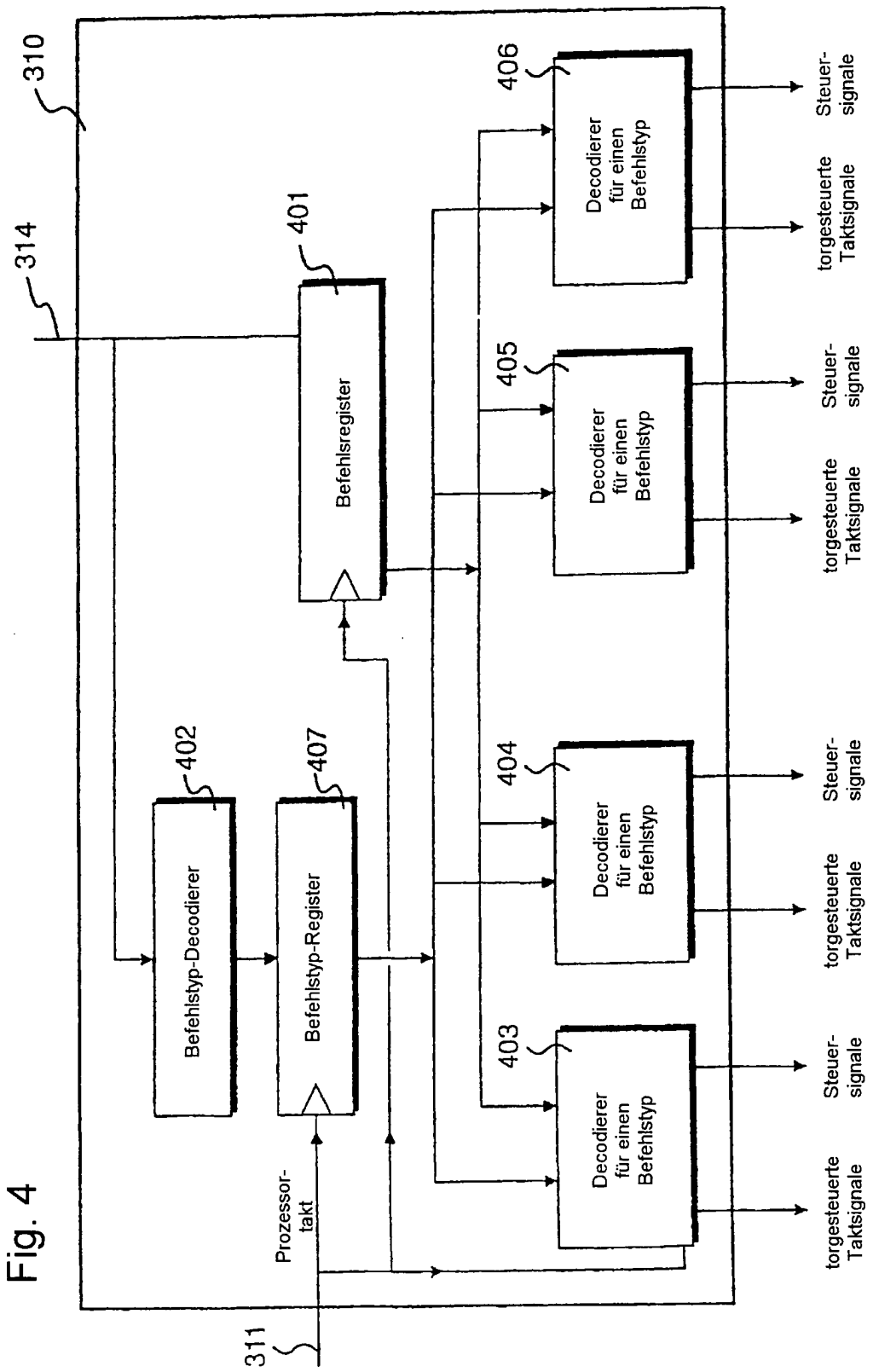

310
314
311
401
402
403
404
405
406
407
Befehlsregister
Befehlstyp-Decodierer
Befehlstyp-Register
Decodierer für einen Befehlstyp
Decodierer für einen Befehlstyp
Decodierer für einen Befehlstyp
Decodierer für einen Befehlstyp
Prozessor-takt
Steuer-signale
torgesteuerte Taktsignale
Steuer-signale
torgesteuerte Taktsignale
Steuer-signale
torgesteuerte Taktsignale
Steuer-signale
torgesteuerte Taktsignale

Fig. 4

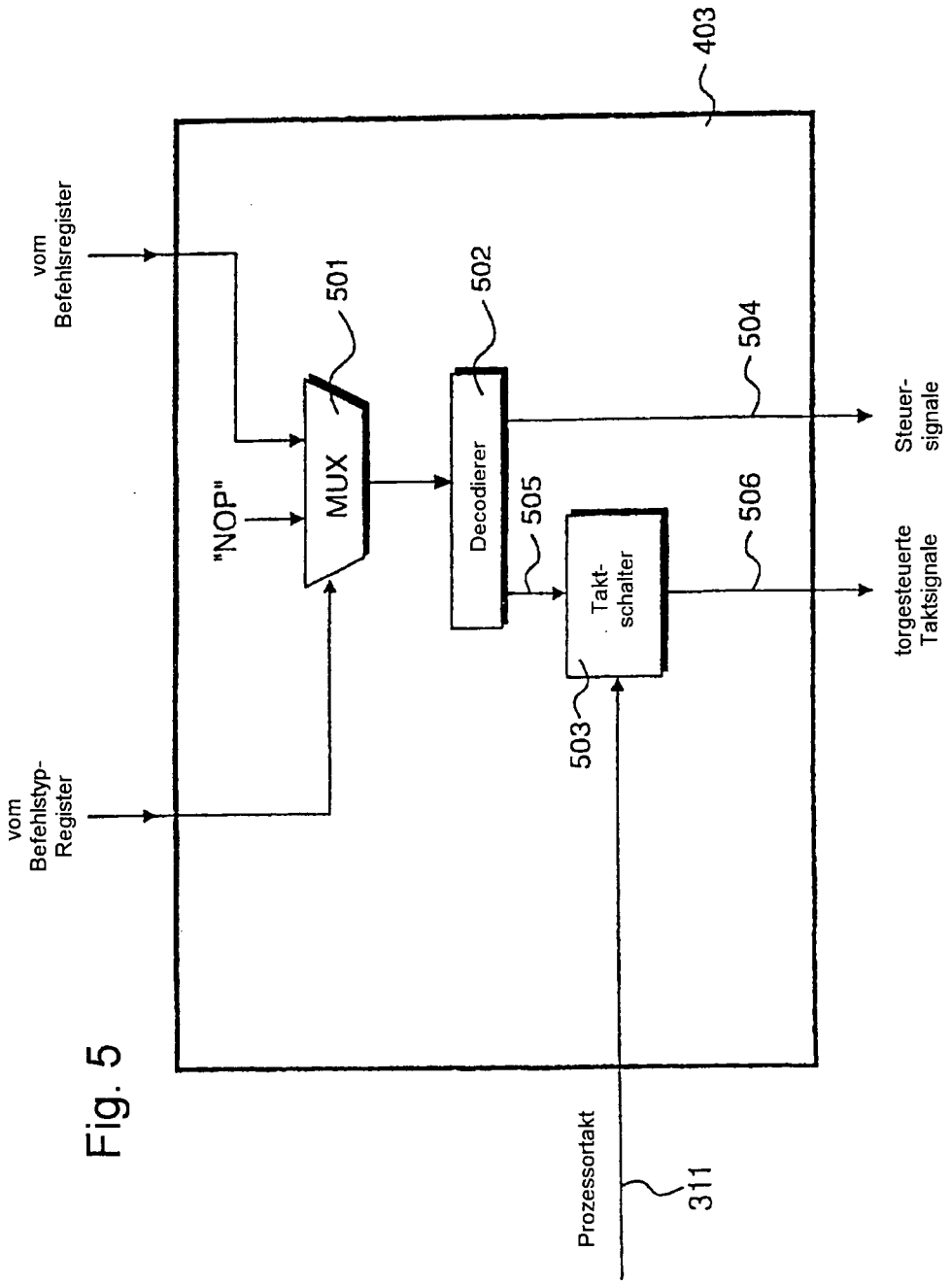

403
vom Befehlsregister
501
502
504
Steuer-signale
"NOP"
MUX
Decodierer
505
506
torgesteuerte Taktsignale
Takt-schalter
503
vom Befehlstyp-Register
Prozessortakt
311

Fig. 5

Fig. 6

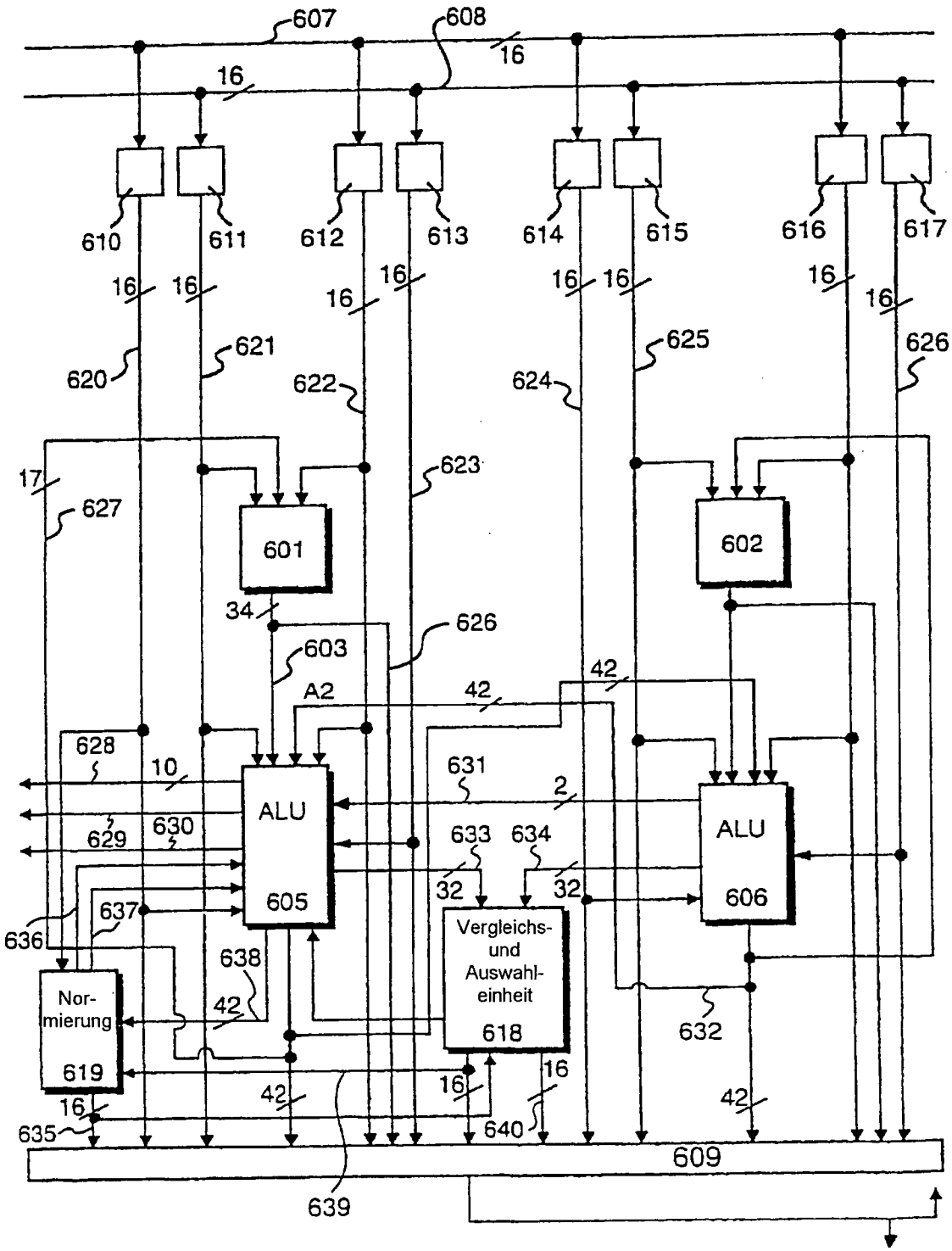

607
608
16
16
610
611
612
613
614
615
616
617
620
621
622
624
625
626
17
627
601
602
623
34
603
626
A2
42
42
628
10
631
2
ALU
605
ALU
606
629
630
633
634
32
32
637
636
638
Nor-
mierung
619
42
Vergleichs-
und
Auswahl-
einheit
618
632
42
16
16
635
640
609
639

Fig. 7

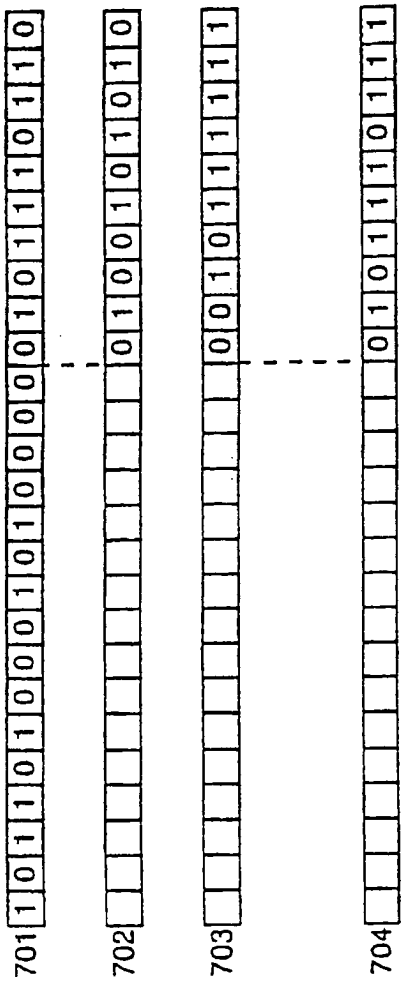

Block von Gleitkomma-zahlen
(größte)
701
702
703
704

Fig. 8

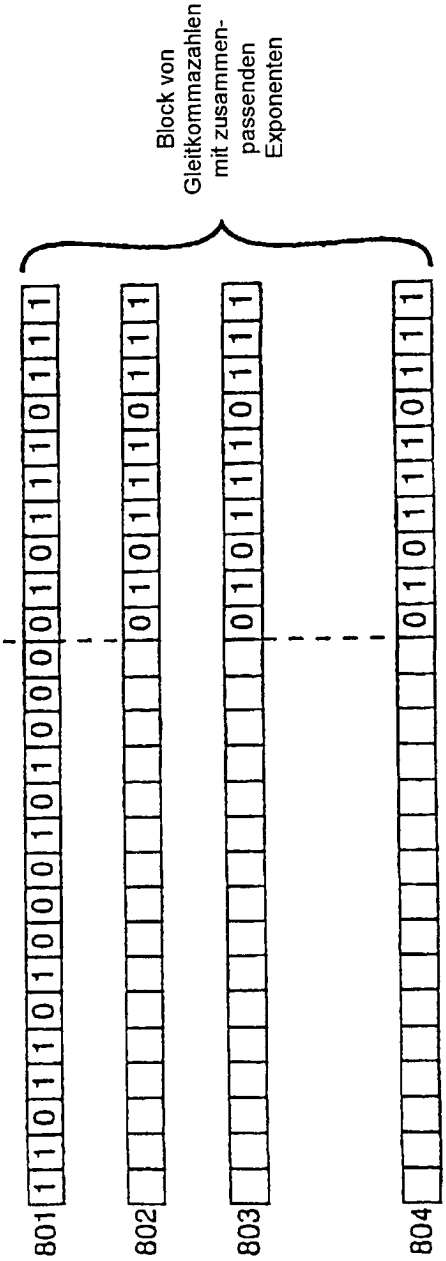

Block von Gleitkommazahlen mit zusammen-passenden Exponenten
801
802
803
804

Fig. 9

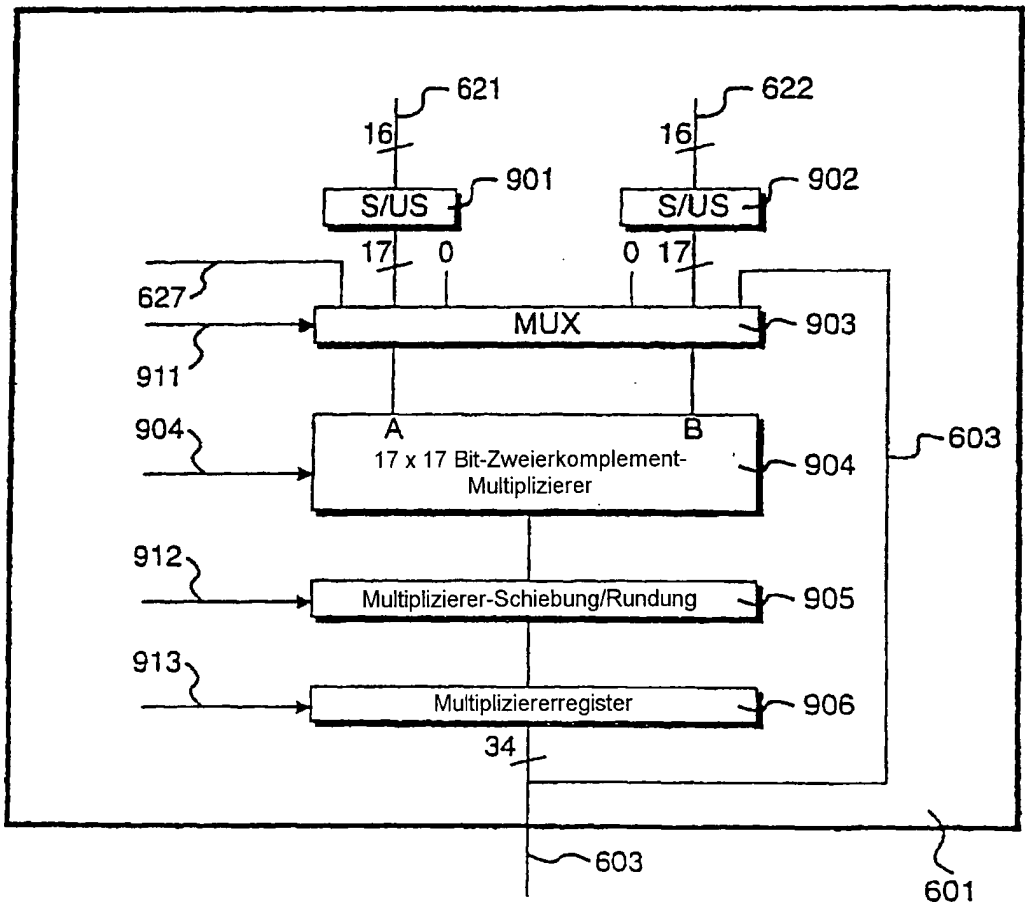
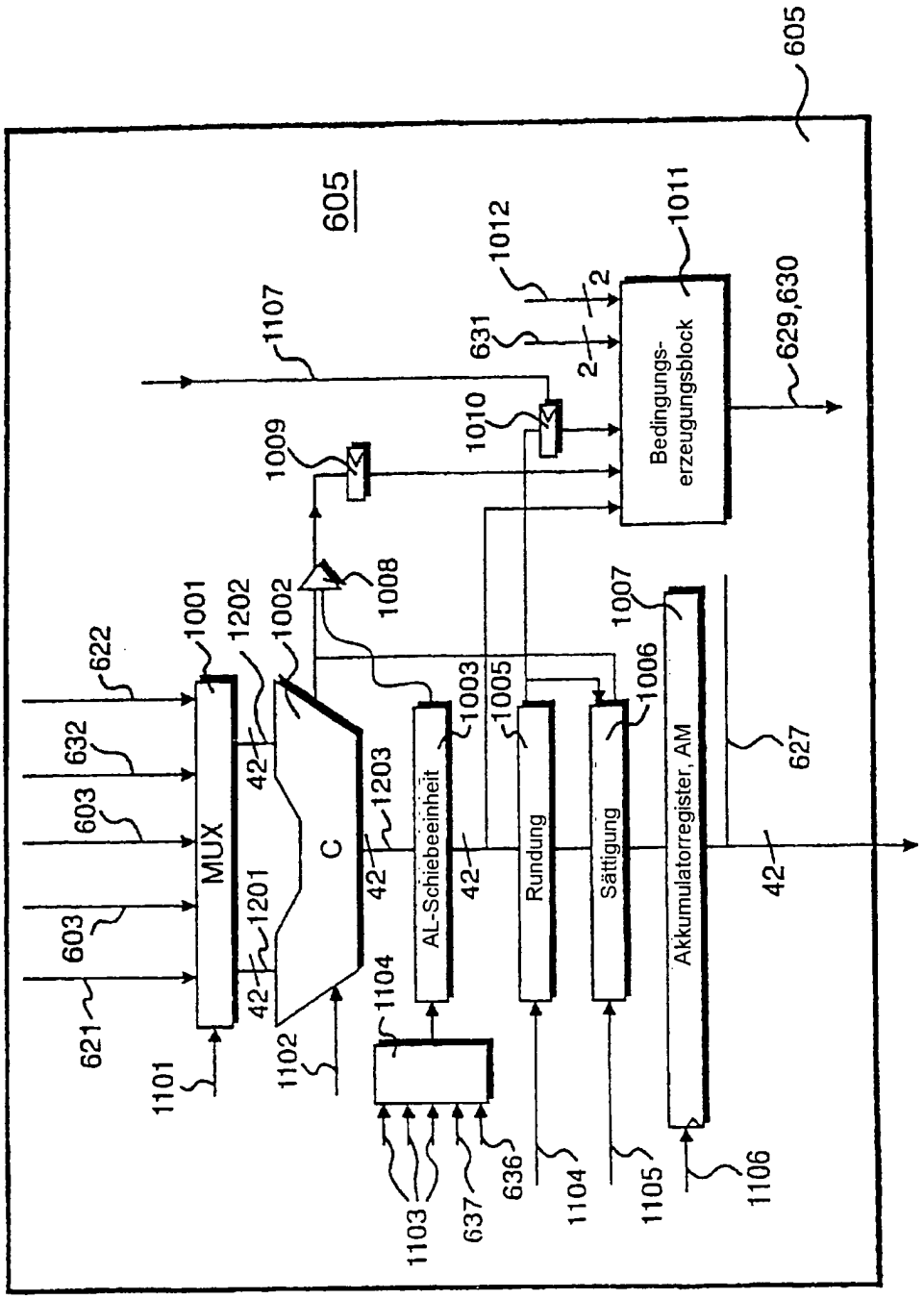

621
622
16
16
901
902
S/US
S/US
17
0
0
17
627
911
MUX
903
904
A
B
17 x 17 Bit-Zweierkomplement-Multiplizierer
904
603
912
Multiplizierer-Schiebung/Rundung
905
913
Multiplizierregister
906
34
603
601

Fig. 10


605
605
MUX
C
AL-Schiebeeinheit
Rundung
Sättigung
Akkumulatorregister, AM
Bedingungs-
erzeugungsblock
1001
1002
1003
1005
1006
1007
1008
1009
1010
1011
1012
1101
1102
1103
1104
1105
1106
1107
1201
1202
1203
621
622
627
629,630
631
632
636
637
603
42
2

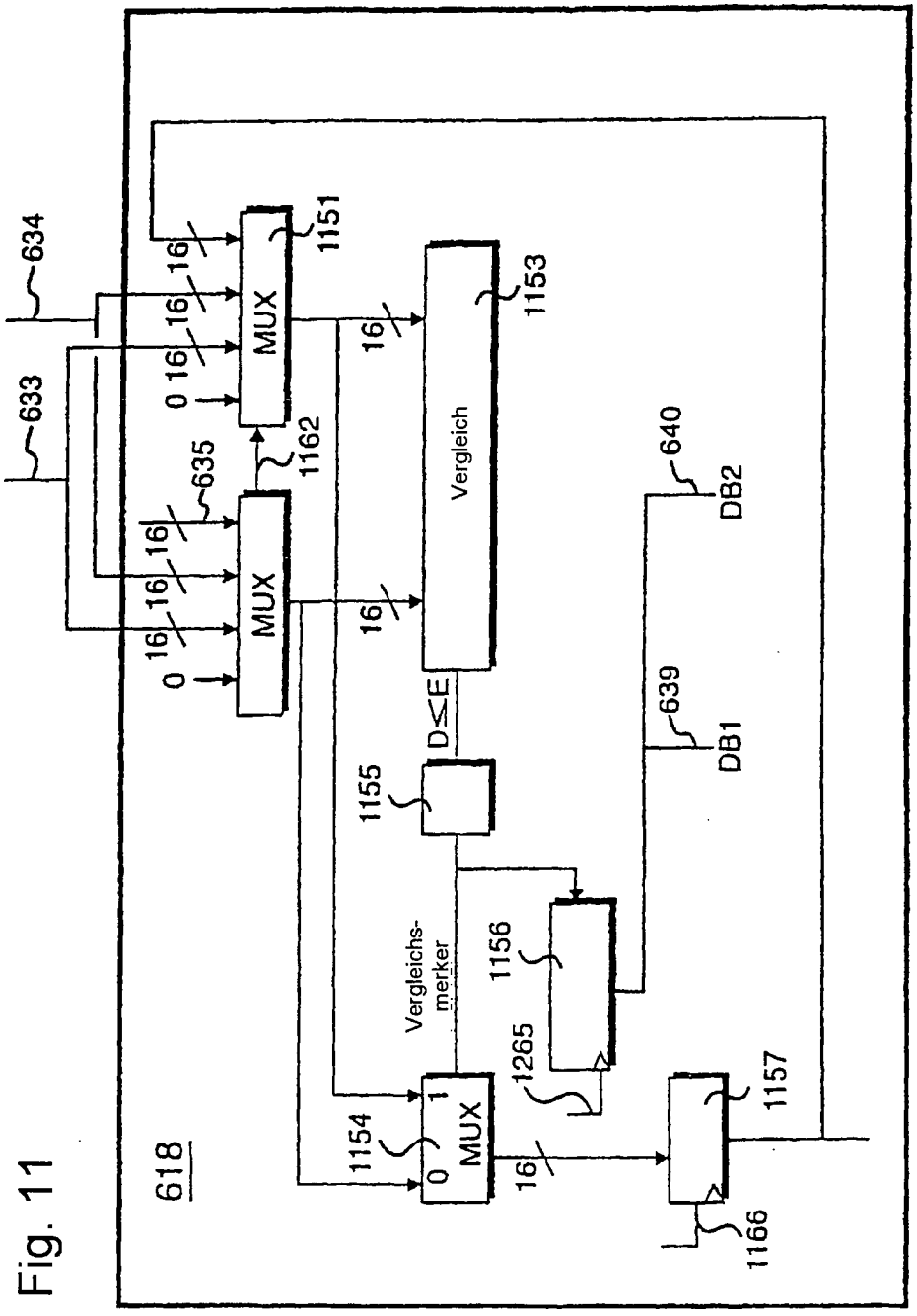

618
634
633
635
MUX
1151
1162
1153
Vergleich
640
DB2
639
DB1
D≤E
1155
Vergleichs-merker
1156
1265
1154
1157
1166
16
0
1

Fig. 11

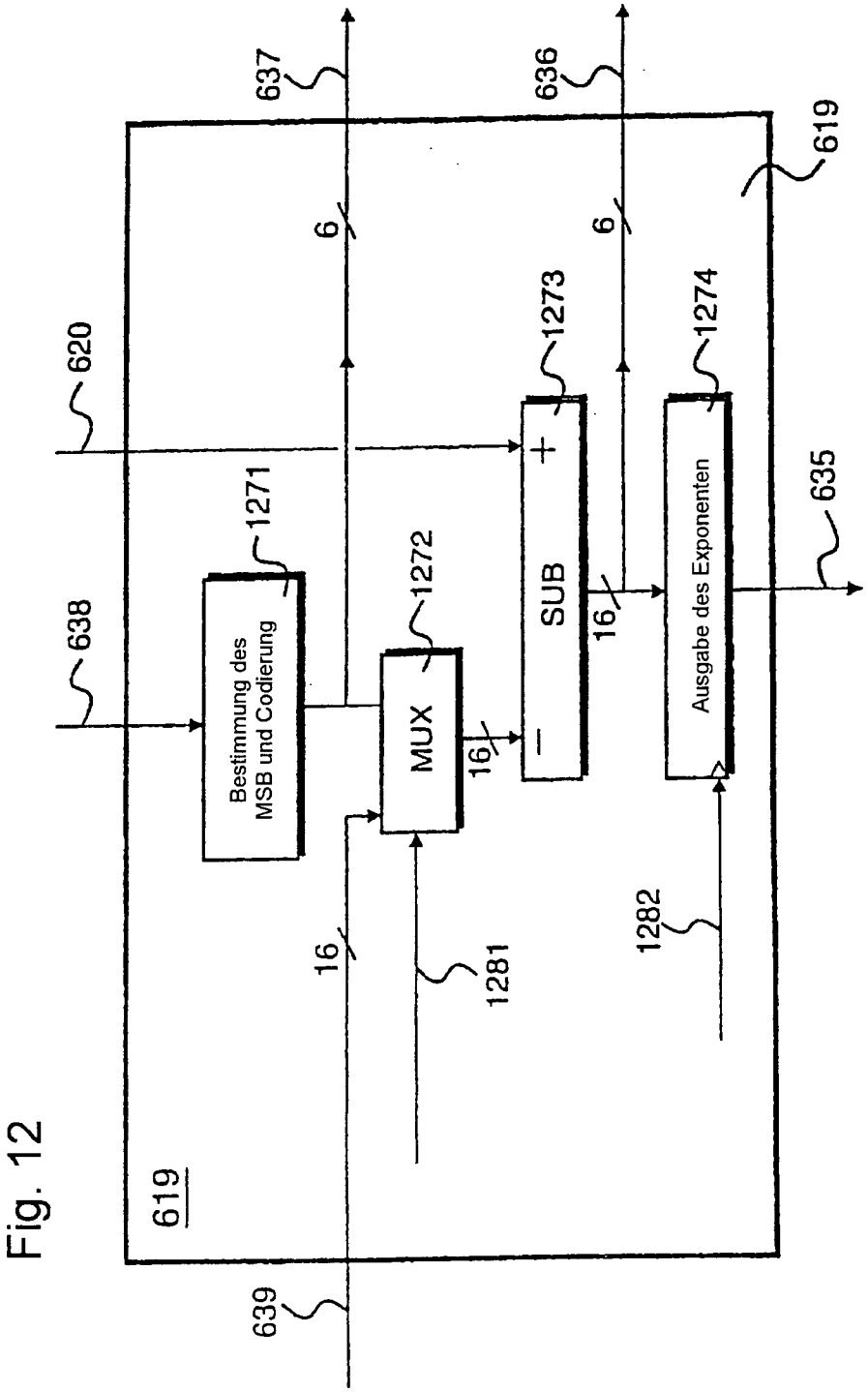

637
636
619
6
6
620
1273
1274
635
638
1271
1272
Bestimmung des MSB und Codierung
MUX
SUB
Ausgabe des Exponenten
16
16
16
1281
1282
619
639

Fig. 12