CH 632 615 A5

(19) SCHWEIZERISCHE EIDGENOSSENSCHAFT
BUNDESAMT FÜR GEISTIGES EIGENTUM

(51) Int. Cl.³: H 02 P 5/16

ETH-BIBLIOTHEK ZÜRICH

Erfindungspatent für die Schweiz und Liechtenstein
Schweizerisch-liechtensteinischer Patentschutzvertrag vom 22. Dezember 1978

(12) PATENTSCHRIFT A5

(11) 632 615

(21) Gesuchsnummer: 14648/77

(22) Anmeldungsdatum: 30.11.1977

(30) Priorität(en): 10.12.1976 US 749641
10.12.1976 US 749648

(24) Patent erteilt: 15.10.1982

(45) Patentschrift veröffentlicht: 15.10.1982

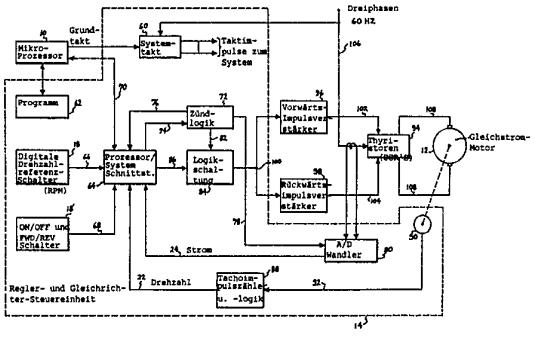
(73) Inhaber:
General Electric Company, Schenectady/NY (US)

(72) Erfinder:
Paul John Roumanis, Stratford/CT (US)
David Lindsay Lippitt, Scotia/NY (US)

(74) Vertreter:
Ritscher & Seifert, Zürich

(54) Verfahren und Vorrichtung zum Steuern eines Gleichstrommotors.

(57) Es wird ein programmierter Datenprozessor (64) verwendet, der die Stromaufnahme, die Drehzahl und die Drehrichtung des Motors (12) zu vorgegebenen Zeitpunkten ausliest und daraus den Zündwinkel für die Gleichrichter (94) in einer Gleichrichterbrücke berechnet. Weiter werden Zähler enthaltende Logikschaltungen (72, 84) verwendet, die den vom Prozessor berechneten Wert für den Zündwinkel aufnehmen und aus denen dieser Wert von einem von der Speisespannungsquelle gesteuerten Taktgeber (60) ausgelesen und ein Zündimpuls erzeugt wird. Gleichzeitig mit jedem Zündimpuls wird ein Programmanforderungssignal für den Datenprozessor erzeugt, das diesen veranlasst, den Zündwinkel für den nächsten zu zündenden Gleichrichter zu berechnen.



PATENTANSPRÜCHE

1. Verfahren zum Steuern der Drehzahl eines Gleichstrommotors mit einer Steuer- und Regelvorrichtung (Fig. 3), die einen zum Berechnen des Zündwinkels des Zündimpulses für mindestens einen zwischen einer Wechselstromquelle (106) und dem Gleichstrommotor (12) angeschlossenen, steuerbaren Gleichrichter (94) vorgesehenen Datenprozessor (64) enthält, dadurch gekennzeichnet, dass in dem Prozessor Parameter gespeichert werden, die proportional zur Motor-Solldrehzahl und den Sollzündwinkeln sind, dass dem Prozessor ein Programmforderungssignal zugeleitet wird, dass der Prozessor als Antwort auf dieses Signal einen der Motor-Istdrehzahl und einen dem Motorstrom proportionalen Wert abliest und einen den Zündwinkel bestimmenden Wert berechnet, der von den Differenzen zwischen der Motor-Solldrehzahl und der Motor-Istdrehzahl sowie zwischen den dem zuletzt abgelesenen und dem vorgängig abgelesenen, dem Motorstrom proportionalen Werten sowie dem vorgängig berechneten, den Zündwinkel bestimmenden Wert abhängig ist, dass der Prozessor ferner in Abhängigkeit vom Programmanforderungssignal auf der Grundlage des berechneten Werts einen der gespeicherten Zündwinkel auswählt und zu dem dem ausgewählten Zündwinkel entsprechenden Zeitpunkt einen Zündimpuls für einen Gleichrichter erzeugt und für die wiederholten Berechnungen des den Zündwinkel bestimmenden Werts und das Erzeugen des Zündimpulses der Zündimpuls als Programmanforderungssignal verwendet wird.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass zum Erzeugen des Zündimpulses eine dem gewählten Zündwinkel entsprechende Zahl in einem Zähler eingespeichert, der Zählerinhalt mit einer vorgegebenen Geschwindigkeit gezählt und beim Erreichen der eingespeicherten Zahl der Zündimpulse erzeugt wird.

3. Verfahren nach Anspruch 1 in einer Vorrichtung mit mehreren Gleichrichtern, dadurch gekennzeichnet, dass der Prozessor bei der Zuleitung des Programmanforderungssignals ein von der Wechselstromquelle abgeleitetes und eine Zeitspanne innerhalb der Periodendauer des Wechselstroms bezeichnendes Informationssignal ausliest, sowie den von dem berechneten Wert bestimmten, gespeicherten Zündwinkel und den von dem Informationssignal bezeichneten zu zündenden Gleichrichter sucht und zum Bestimmen des nächsten zu zündenden Gleichrichters den Wert des gelesenen Informationssignals um einen festgelegten Betrag ändert.

4. Steuer- und Regelvorrichtung zum Ausführen des Verfahrens gemäss Anspruch 1, gekennzeichnet durch Einrichtungen (10) zum Erzeugen eines Grundtaktsignals und durch Takteinrichtungen (60), die mit der Wechselstromquelle (106) verbunden sind und auf das Grundtaktsignal ansprechen, um mit der Wechselstromquelle synchronisierte Informationssignale zu erzeugen, die einen ersten Kennwert definieren, der proportional zu der von den festgestellten Nulldurchgängen des Wechselstroms abgeleiteten Periodendauer des Wechselstroms ist, und einen zweiten Kennwert, der eine Zeitspanne innerhalb der Periodendauer definiert, wobei der Nullpunkt dieser Zeitspanne mit dem Nulldurchgang der Wechselstromperiode übereinstimmt, Einrichtungen (80) zum Erzeugen eines ersten, dem Motorstrom proportionalen Parameters, Einrichtungen (18) zum Erzeugen eines zweiten, der Motor-Solldrehzahl proportionalen Parameters, Einrichtungen (90) zum Erzeugen eines dritten, der Motor-Istdrehzahl proportionalen Parameters, einen Datenprozessor (64), der beim Erscheinen eines Programmanforderungssignals die Parameter und Informationssignale selektiv ausliest und aus dem Betrag der daraus bestimmten Motor-Sollspannung einen dem Zündwinkel proportionalen Kennwert sucht, aus dem zweiten Kennwert den Zeitpunkt für die externe Übertragung des Zündwinkels berechnet und aus dem ersten Kennwert die Adresse des zu zündenden Gleichrichters bestimmt, sowie mit den Gleichrichtern zusammenwirkende Wähleinrichtungen (84), die die vom Prozessor (64) bestimmten Adressen empfangen und den zu zündenden Gleichrichter (94) aussuchen, und Einrichtungen (72) für den Empfang des vom Prozessor gewählten Zündwinkels zu dem durch die Berechnung für die externe Übertragung festgelegten Zeitpunkt, wobei jede Wähleinrichtung einen Zähler (124, 224) zum Einzählen der Taktsignale enthält und beim Erreichen eines vorgegebenen Zählerstands einen Zündimpuls für den ausgewählten Gleichrichter und ein Unterbrechungssignal für den Prozessor erzeugt.

Die Erfindung betrifft ein Verfahren zum Steuern eines Gleichstrommotors mit einer Steuer- und Regelvorrichtung, die einen zum Berechnen des Zündwinkels des Zündimpulses für mindestens einen zwischen einer Wechselstromquelle und dem Gleichstrommotor angeschlossenen, steuerbaren Gleichrichter vorgesehenen Datenprozessor enthält.

Motorregelsysteme der hier beschriebenen Art verwenden häufig Leistungsverstärker mit Regelgleichrichtern, die den elektrischen Energiefluss zwischen einer Wechselstromquelle und einem Antriebsmotor verändern. Regel- oder Steuergleichrichter sind bekannt, und sie bestehen aus einer Familie von Bauelementen, die bezüglich der elektrischen Energieströmung eine relativ hohe Impedanz darstellen, bis sie durch ein Zündsignal in Vorwärtsrichtung vorgespannt werden, welches an eine Steuerelektrode angelegt wird. Während des leitenden Zustandes besitzen die Regelgleichrichter normalerweise eine sehr kleine Impedanz gegenüber dem Stromfluss und leiten normalerweise den Strom so lange, bis sie in Rückwärtsrichtung vorgespannt werden und/oder der Strompegel durch die Gleichrichter unter einen minimalen Haltepegel verringert ist, der erforderlich ist, um diese Gleichrichter in leitendem Zustand zu halten. Die Familie der erörterten Regelgleichrichter umfasst allgemein Halbleiterbauelemente, wie zum Beispiel Thyristoren und andere Bauelemente, wie zum Beispiel Ignitrons und Thyrotrons.

In den hier erläuterten Systemen wird die Menge der einer Systemlast oder einem Gleichstrommotor zugeführten Leistung dadurch geregelt, dass die Dauer des leitenden Zustandes der regel- oder steuerbaren Gleichrichter verändert wird. Die Dauer des leitenden Zustands der steuerbaren Gleichrichter ist eine Funktion des Zeitpunkts innerhalb des Wechselstrom-Kurvenverlaufs, in dem die Gleichrichter in den leitenden Zustand gebracht werden. Dieser Punkt wird als Zündwinkel bezeichnet.

In der Vergangenheit wurden Systeme zur Regelung des Leitungszustands der steuerbaren Gleichrichter allgemein unter Verwendung von analogen Regeleinrichtungen verwirklicht, um die geforderten Regelfunktionen durchzuführen, wobei die analogen Signale in digitale Werte umgewandelt wurden, um die Gleichrichter zu zünden. Bei dieser Art von Systemen arbeiten die Zündkreise in Abhängigkeit von einem Eingangssignal, das die Leistung anzeigt, um einen Zündimpuls beim geeigneten Zündwinkel zu erzeugen. Allgemein gesprochen, ist der Zündwinkel direkt proportional zum Eingangssignal. Analogsysteme der bekannten Art arbeiten allgemein in Abhängigkeit von einem Eingangssignal, dessen Grösse den gewünschten Zündwinkel angibt.

In den letzten Jahren ist mit der Entwicklung der Digitaltechnik und der Hardware das Interesse der Ingenieure daran gewachsen, digitale Schaltungen in derartigen Regelsystemen einzusetzen. Die Anwendung der Digitaltechnik bei derartigen Regelsystemen ist insbesondere vorteilhaft, wo das System

einen Genauigkeitsgrad, eine Zuverlässigkeit oder einen driftfreien Betrieb erfordert, der nur mit digitaler Schaltungstechnik herstellbar ist. Es ist daher ziemlich üblich, Elemente eines analogen Systems durch funktional äquivalente digitale Schaltungen zu ersetzen.

Ein derartiges bekanntes digitales Regel- und Steuersystem zur Regelung des Leitungszustandes von steuerbaren Gleichrichtern ist in der US-PS 3 601 674 offenbart. In diesem Patent ist ein digitales Regelsystem offenbart, um den Leistungsfluss durch steuerbare Gleichrichter aus einer mehrphasigen Wechselstromquelle zu einer Last zu regeln. Dieses System enthält einen Zündkreis für jede Phase, wobei jeder Zündkreis einen reversiblen Zähler und einen digitalen Komparator enthält. Eine Phasenwahrnehmlogik ist vorgesehen, die die drei Phasen der Wechselstromquelle derart überprüft, um synchron ein Regelintervall für einen geeigneten Gleichrichter dadurch auszulösen, dass eine vorbestimmte positive oder negative digitale Zahl in dem reversiblen Zähler vorgesetzt wird, der der betreffenden Phase zugeordnet ist. Der reversible Zähler zählt dann während des Regelintervalls abwärts, wenn die vorgesetzte Zahl positiv ist, er zählt aufwärts, wenn die vorgesetzte Zahl negativ ist. Während des Zählvorgangs wird ein digitales Drehzahlfehlersignal, das aus einem vorausgegangenen Vergleich eines digitalen Befehls mit einem digitalen Rückkopplungssignal gewonnen wird und die Motordrehzahl angibt, kontinuierlich durch den digitalen Komparator mit dem Inhalt des reversiblen Zählers verglichen. Wenn der Fehler den Inhalt des reversiblen Zählers übersteigt, wird ein Zündimpuls erzeugt, der einem positiv oder negativ gepolten Gleichrichter zugeführt wird und den entsprechend gepolten Gleichrichter gemäss der positiven oder negativen Zahl zündet.

Ein weiteres bekanntes System der digitalen Bauart wurde in zwei Veröffentlichungen beschrieben, wobei die erste Veröffentlichung von R.D. Jackson und R.D. Weatherby stammt und den Titel trägt «Direct Digital Control of Thyristor Converters» in «IFAC Symposium on Control and Power Electronics and Electrical Drives», Oktober 1974, abgehalten in Düsseldorf, Deutschland, Vorabdruck, Band I, Seiten 431-441, und wobei die zweite Veröffentlichung von F. Fallside und R.D. Hackson stammt und den Titel trägt «Direct Digital Control of Amplifiers» in Proceedings of the Institute of Electrical Engineers, mit dem Titel «Control and Science», Band 116, Nr. 5, S. 873-878, Mai 1969.

In den genannten Veröffentlichungen beschreiben die Autoren eine Untersuchung eines Laborsystems der direkten digitalen Regelart, um die Durchführbarkeit der direkten digitalen Regelung der regelbaren Verstärker, wie zum Beispiel von Thyristoren, zu zeigen. In diesem System wird ein programmierter digitaler Rechner verwendet, um das Zünden der Gleichrichter durch ein Anschlussgerät (Interfacegerät) zu regeln, um eine resistive kapazitive Last durch Erzeugung von Zündimpulsen zu regeln, die vom Rechner erzeugt sind. Der digitale Rechner berechnet den Zündwinkel, der einen Zeitpunkt festlegt, um einen speziellen Gleichrichter zu zünden. Das System ist mit den Nulldurchgängen der Phasenspannungen der Wechselstromquelle synchronisiert, wobei diese Phasendurchgänge Befehlssignale für einen Abtasthaltekreis eines Analog-/Digitalwandlers erzeugen, der die Ausgangsspannung der Systemlast misst.

Am Ende der Analog-/Digitalwandlung wird ein Impuls von dem Konverter als ein Programmanforderungssignal für den Rechner erzeugt. Dieser Impuls startet die Zündwinkelberechnung. Im Anschluss an die Auslösung dieses Signals liest der Rechner den Analog-/Digitalwandler eine bestimmte Zeit nach dem Zeitpunkt des Phasennulldurchgangs aus. Der Rechner berechnet dann den Zündwinkel oder Zündzeitpunkt für den Gleichrichter, wobei ein von den Autoren bezeichnetes «gegebenes Zündgesetz» verwendet wird. Bei dieser Berechnung wird ein Regelsignal oder Regelwert entwickelt, der kontinuierlich so lange mit einer linearten Nachschlagetabelle verglichen wird, die das Zündgesetz definierende Werte enthält, bis eine Übereinstimmung zwischen dem Regelsignal und dem Inhalt der Tabelle existiert. Wenn ein Vergleichswert erreicht ist, wird ein Zündimpuls erzeugt, in dem die Adresse des zu zündenden Gleichrichters gesetzt wird und ein Zündsignal dem Gleichrichter zugeführt wird.

Während die Autoren die Durchführbarkeit der direkten digitalen Regelung von steuerbaren Gleichrichtern zeigten, stellten sie auch verschiedene praktische Schwierigkeiten fest, die sich beim Erstellen eines derartigen Systems ergaben. Dies war bei der auffälligsten Einrichtung innerhalb dieser Experimente, der Verwendung von Nachschlagetabellen, offensichtlich, die eine beträchtliche Rechnerzeit benötigen und die Menge an zusätzlichen Berechnungen, die durch den Rechner beim Betreiben eines Realzeitsystems dieser Art möglich sind, stark beschränken.

Es muss ferner festgestellt werden, dass direkte digitale Antriebsregelsysteme die Lastausgangsspannung zu einem gegebenen Zeitpunkt abtasten müssen, der zu einem festgestellten Phasenintervall der Wechselstromquelle in Beziehung steht, dass anschliessend derartige Systeme die erforderliche Berechnung des Zündwinkels durchführen und einen ausgewählten Gleichrichter innerhalb des festgestellten Intervalls früh genug innerhalb des speziellen Intervalls zünden müssen, um die Verzögerung des Zündwinkels genau und voll zu steuern, um einen maximalen Leistungstransfer zur Last zu erhalten.

Die genannten Systeme arbeiteten nicht als ein Gesamtregelsystem zur Regelung der Motordrehzahl eines reversiblen Antriebsmotors mit variabler Drehzahl. In Analogsystemen ist die Art und Weise der Regelung der Drehzahl eines Gleichstrommotors, der entweder im kontinuierlichen oder nichtkontinuierlichen Strombetrieb arbeitet, ebenso wie die Art und Weise der Richtungsumkehr des Motors wohl bekannt. Es ist ferner bekannt, dass ein Kriterium zur Umkehr der Richtung eines Gleichstrommotors darin besteht, dass der Motorstrom zum Zeitpunkt der Umkehr null sein muss. Um diese Umkehr durchzuführen, muss bei Analogsystemen zuerst festgestellt werden, wenn der Strom null ist, und dann muss ein gegebenes Sicherheitsintervall gewartet werden, bevor eine Drehrichtungsumkehr des Motors erfolgen kann. Es ist ferner bei analogen Motorantriebssystemen bekannt, dass derartige Systeme zwei Rückkopplungskreise im System erfordern, ein Rückkopplungskreis regelt den Motor, wenn dieser sich in einem kontinuierlichen Strombetrieb befindet, und ein Rückkopplungskreis regelt den Motor, wenn der Motor im diskontinuierlichen Strombetrieb arbeitet. Diese beiden Kreise erzeugen in Abhängigkeit von der Betriebsart des Systems unterschiedliche Verstärkungen. Dieses Betriebsverfahren in analogen Regelsystemen hat sich in gewisser Weise bei Gleichstrommotor-Antriebssystemen als nicht ganz befriedigend herausgestellt, insbesondere wenn es erwünscht ist, einen sehr hohen Grad der Motordrehzahlkonstanz unter sehr leichten Lastbedingungen zu erzielen. Es ist daher wünschenswert, ein Antriebs- und Regelsystem für die Drehzahl eines Gleichstrommotors zu schaffen, welches den gesamten Betrieb des Systems dadurch verbessert, dass ein Regler vorgesehen wird, der die Systemparameter misst und die Motordrehzahl genau berechnet, der die Betriebsart des Systems bestimmt und der die Drehrichtung des Motors und Änderungen der Drehrichtung des Motors sofort und ohne Verzögerung feststellt, wenn dies erwünscht ist.

Aufgabe der Erfindung ist es daher, ein verbessertes Verfahren zur digitalen Steuerung der Zündung von steuerbaren Gleichrichtern in der Stromzuleitung für einen Gleichstrommotor zu schaffen sowie eine zur Ausführung dieses Verfah-

rens geeignete Vorrichtung mit einem programmierten Datenprozessor, welche Vorrichtung Schnittstellen-Einrichtungen enthält, die die Systemeingangsparameter lesen und die erforderliche Motorspannung berechnen können und einen einer derartigen berechneten Spannung proportionalen Wert benutzen, um den Zündwinkel für einen Zündimpuls zu wählen.

Die vorstehende Aufgabe wird erfindungsgemäss mit einem Verfahren gelöst, das dadurch gekennzeichnet ist, dass in dem Prozessor Parameter gespeichert werden, die proportional zur Motor-Solldrehzahl und den Sollzündwinkeln sind, dass dem Prozessor ein Programmanforderungssignal zugeleitet wird, dass der Prozessor als Antwort auf dieses Signal einen der Motor-Istdrehzahl und einen dem Motorstrom proportionalen Wert abliest und einen den Zündwinkel bestimmenden Wert berechnet, der von den Differenzen zwischen der Motor-Solldrehzahl und der Motor-Istdrehzahl sowie zwischen den dem zuletzt abgelesenen und dem vorgängig abgelesenen, dem Motorstrom proportionalen Werten sowie dem vorgängig berechneten, den Zündwinkel bestimmenden Wert abhängig ist, dass der Prozessor ferner in Abhängigkeit vom Programmanforderungssignal auf der Grundlage des berechneten Werts einen der gespeicherten Zündwinkel auswählt, und zu dem dem ausgewählten Zündwinkel entsprechenden Zeitpunkt einen Zündimpuls für einen Gleichrichter erzeugt, und für die wiederholten Berechnungen des den Zündwinkel bestimmenden Werts und das Erzeugen des Zündimpulses, der Zündimpuls als Programmanforderungssignal verwendet wird.

Das neue Verfahren ermöglicht die vollständige Berechnung des Zündwinkels, die Wahl eines zu zündenden Gleichrichters und die Bestimmung und Wahl der Betriebsart der Vorrichtung und die Drehrichtung des Motors in einer Zeitspanne, die kurz genug ist, dass der Prozessor einen Zündwinkel, der eine maximale Leistungsabgabe an den Gleichstrommotor ermöglicht, berechnen kann.

Die zur Ausführung des neuen Verfahrens verwendete Vorrichtung enthält einen Datenprozessor, der eine digitale Regelung der Änderungsgeschwindigkeit des Motorstroms, der Motorspannung und der Motordrehzahl ermöglicht.

Die Verwendung des Zündimpulses als Programmanforderungssignal für den Prozessor ermöglicht insbesondere die Berechnungen in so kurzer Zeit auszuführen, dass ein Gleichrichter zum frühestmöglichen Zeitpunkt innerhalb eines Phasenintervalls der Wechselstromquelle gezündet und dadurch ein maximaler Leistungstransfer zum Gleichstrommotor erreicht werden kann.

Im folgenden werden Ausführungsbeispiele der Erfindung anhand der Zeichnungen näher erläutert. Es zeigen:

Fig. 1 ein Blockschaltbild einer Regel- und Steuervorrichtung gemäss der vorliegenden Erfindung,

Fig. 2 ein vereinfachtes Blockschaltbild eines repräsentativen Datenprozessors zur Verwendung bei der Erfindung,

Fig. 3 ein Blockschaltbild, das den Regler und die Gleichrichtersteuerung und den Gleichrichter der Vorrichtung gemäss der Fig. 1 zeigt, die mit dem Datenprozessor und der Last oder Gleichstrommotor verbunden sind,

Fig. 4 eine detaillierte Logikschaltung der Prozessor-/Systemschnittstelle in dem Blockschaltbild gemäss der Fig. 3, die die Verbindung zwischen dem Datenprozessor und den anderen Logikblöcken des Reglers und der Gleichrichtersteuerung zeigt,

Fig. 5 ein detailliertes Logikschaltbild des Blockschaltbilds der Fig. 3,

Fig. 6 ein Zeitsteuerdiagramm, das zum Verständnis des Betriebs des Systemtaktes und der erfindungsgemässen Regel- und Steuervorrichtung nützlich ist,

Fig. 7 ein detailliertes Logikschaltbild der Tachometerimpulszähler- und Logikeinheit der Fig. 3,

Fig. 8 ein Zeitsteuerdiagramm, das zum Verständnis des Betriebs der Tachometerimpulszähler- und Logikeinheit der Fig. 7 nützlich ist,

Fig. 9 ein detailliertes Logikschaltbild der Zündlogik der Fig. 3,

Fig. 10 ein Zeitsteuerdiagramm, das die Zeitsteuerung der Zündlogik der Fig. 9 zeigt und zum Verständnis des Betriebs der vorliegenden Erfindung hilfreich ist,

Fig. 11A und 11B zusammengenommen, wobei Fig. 11A an der Oberkante der Fig. 11B zu liegen kommt, zeigen ein detailliertes Logikschaltbild und einen schematischen Aufbau der Thyristorwähl- und Antriebsrichtung-Logik, der Gleichrichter und des Analog-/Digitalwandlers der Fig. 3 und die Zwischenverbindungen,

Fig. 12 Kurven, die zum Verständnis des Betriebs der vorliegenden Erfindung nützlich sind,

Fig. 13 ein Strichbalken-Flussdiagramm, das zum Verständnis der Abfolge des Betriebs der Erfindung nützlich ist, und

Fig. 14 bis 24 ein Flussdiagramm, zum Verständnis der Durchführung des erfindungsgemässen Regel- und Steuerverfahrens.

Zur Beschreibung des Gesamtbetriebs der Erfindung wird nun auf Fig. 1 des Regel- und Steuersystems Bezug genommen, die in Form eines Blockschaltbildes die wesentlichen Funktionalblöcke zeigt, aus denen das vorliegende System besteht. In Fig. 1 enthält ein allgemein als Mikroprozessor 10 dargestellter Datenprozessor ein Programm zur Steuerung des Gesamtbetriebs des Systems, wobei über eine Regler- und Gleichrichtersteuereinheit 14 Parametereingangssignale von einem herkömmlichen Gleichstrommotor 12 dem Mikroprozessor eingelesen werden. Das Programm im Prozessor 10 steuert das Lesen dieser verschiedenen Eingangssignale und enthält ein Programm zur Berechnung des Zündwinkels zur richtigen Zündung der Gleichrichter oder Thyristoren, gewöhnlich auch als SCRs bezeichnet, in einem herkömmlichen Dreiphasen-Brückengleichrichter 16. Die Regler- und Gleichrichtersteuereinheit 14 liefert ein gemeinsames Interface bzw. eine gemeinsame Schnittstelle zwischen dem Prozessor 10 und dem Rest des Regelsystems. Unter der Steuerung des Prozessors 10 liest die Gleichrichtersteuereinheit 14 Eingangssignale aus einer Drehzahlreferenzeinheit 18 über eine Vielzahl von Eingangsleitungen aus, wobei derartige Signale ein digitales Referenzsignal, gemäss dem der Motor in Umdrehungen pro Minute laufen soll, und einen Ein-/Aus-Leistungszustand des Motors, und Benutzersignale kennzeichnen, die gesetzt sind und die Richtung bezeichnen, in der der Motor 12 laufen soll. Die letzeren Signale werden über eine Vielzahl von Leitungen 20 geliefert, die als «Drehzahlreferenz» bezeichnet sind. Über die Reglersteuereinheit 14 gehen über eine Vielzahl von Leitungen 22 zusätzliche Eingangssignale zum Prozessor 10 und stellen Drehzahlsignale vom Gleichstrommotor 12 dar, die von einem Sensor am Motor 12 herkommen und die Drehzahl in Umdrehungen pro Minute kennzeichnen, mit welcher der Motor läuft. Der Motorstrom wird ebenso über die Regel- und Steuereinheit 14 vom Mikroprozessor gemessen, wofür ein Strom vom Motor über eine Vielzahl von Leitungen 24 an den Prozessor geliefert wird. Die Regel- und Steuereinheit 14 liefert unter der Steuerung von vom Prozessor 10 gelieferten Signalen Steuersignale an den Gleichrichter 16 und empfängt Daten vom Prozessor, um das Zünden der Thyristoren (SCR) im Gleichrichter zu geeigneter Zeit zu steuern, um dadurch den Gleichstrommotor zu regeln. Wie noch erläutert wird, ist der Gleichrichter 16 als ein Vorwärts-/Rückwärts-Brückengleichrichter ausgebildet, der derart ausgesteuert werden kann, dass er die Richtung der Spannung und des Stroms durch den Motor 12 umkehrt und auf diese Weise zur Drehzahlregelung und zur Umkehr der Laufrichtung dient.

Der in Fig. 1 dargestellte Mikroprozessor kann irgendein Mikroprozessor aus der Vielzahl der mikroprogrammierten

digitalen Vielzweckrechner sein, die zurzeit auf dem Markt erhältlich sind. Ein derartiger Rechner, der für die erfindungsgemässe Anwendung geeignet ist, stellt der Mikrorechner Intel 8080 dar, der von der Intel Corporation verkauft wird. Ein weiterer ideal geeigneter Mikroprozessor, der bei der Ausführung dieser Erfindung verwendet wird, ist ein digitaler mikrokodierter Vielzweckrechner, der von der General Electric Comp. als Modell CRD 8 Mikrocomputersystem verkauft wird.

In Fig. 2 ist der digitale Mikroprogrammrechner CRD 8 dargestellt, wobei die Hauptkomponenten dieses Rechners gezeigt sind. Die Hauptsteuereinheit des Rechners enthält einen Mikrokode-Steuerfestwertspeicher, ROM, 26, der mit einem Mikrokode programmiert ist, welcher aus Mikrobefehlen besteht, die im ROM gespeichert sind. Die Mikrobefehle, die als Aussteuersignale für das Register, den Speicher und I/O-Kanäle bezeichnet sind und auf einer Vielzahl von Leitungen 28 geführt werden, steuern das Holen und die Interpretation der Befehle, die im Hauptspeicher oder Speicher 30 gespeichert sind, wobei sie zuerst den Befehl erkennen und dann den Sprung zu einer Sequenz von Mikrobefehlen im Steuer-ROM bewirken, die die Vorgänge ausführen, die durch die Befehle aufgerufen sind. Die Adresse des nächsten Befehls, der vom Mikrokode-ROM interpretiert werden soll, ist in einem Programmzählerregister (PC) 32 enthalten. Vor der Interpretation jedes Befehls inkrementiert der Mikrokode-ROM den Inhalt des Programmzählers PC, um auf den folgenden Befehl zu zeigen.

Der Mikrokode im Mikrokode-ROM interpretiert Subroutines-Rufe dadurch, dass er die Adresse der Subroutine in ein Programmzähler-Zwischenregister PCS 34 gibt und dann die Rolle des Programmzählers PC mit der Rolle des Programmzähler-Zwischenregisters PCS vertauscht. Subroutine-Rückverzweigungen werden wiederum dadurch interpretiert, dass die Rolle der letzteren zwei Register vertauscht wird, wodurch veranlasst wird, dass der dem Subroutine-Aufruf folgende Befehl als nächster interpretiert wird. Wenn eine äussere Unterbrechung beim Prozessor auftritt, tauscht der Prozessor die Rolle des Programmzählers PC 32, des Programmzähler-Zwischenregisters PCS 34, und eines Seitenregisters (Page) 36 mit einem Unterbrecherprogrammzähler 38, einem Unterbrecherprogrammwähler-Zwischenregister (IPCS) 40, und einem Unterbrecher-Seitenregister (IPAGE) 42 aus. Unterbrechungsrückverzweigungen werden durch den Mikrokode im Mikrokode-ROM dadurch interpretiert, dass die Rollen dieser Register in ihre ursprünglichen Rollen zurückgetauscht werden.

Dem Prozessor gelieferte äussere Unterbrechungen können unter der Programmsteuerung durch das Setzen oder Rücksetzen eines Unterbrecher-Auslöse-Flipflops (nicht dargestellt) ausgelöst oder nicht ausgelöst werden. Wenn eine äussere Einrichtung den Prozessor zu unterbrechen wünscht, so gibt diese Einrichtung eine Anfrage an die Unterbrechungsleitung. Wenn diese Anfrage vorhanden ist, und das Unterbrechungs-Auslöse-Flipflop gesetzt ist, und der Prozessor einen unterbrechbaren Befehl ausführt, dann beginnt der Prozessor mit der Ausführung der Unterbrechung nach Abschluss des laufenden Befehls. Wenn einmal die Unterbrechungsverarbeitung beginnt, ist das Unterbrechungsprogramm dafür verantwortlich, dass es sich die äussere Eingabeeinrichtung merkt, um deren Anfrage von der Unterbrechungsleitung zu entfernen. Der Speicher des Prozessors wird in Seiten unterteilt, wobei eine bestimmte Zahl an Wörtern pro Seite vorhanden ist. Durch Verwendung des Seitenregisters 36 kann ein Befehl überall im Speicher dadurch Zugriff zu Daten erhalten, dass lediglich eine Adresse bezüglich des Kopfes der laufenden Datenseite spezifiziert wird (die Seite, auf die das Seitenregister zeigt).

Daten im Hauptspeicher 30 können ferner direkten Zugriff dadurch erhalten, dass eine Adresse des Datenworts in ein oder drei Vielzweckregister gegeben wird, die R1, R2 und R3 bezeichnet sind. Diese Register können auch zur Speicherung von Daten verwendet werden. Die Ansammlung der drei Vielzweckregister und der zusätzlichen Register 32 bis 42 wird als Zwischenspeicher (scratch pad memory) bezeichnet.

Zusätzlich zu den Zwischenspeicherregistern enthält der Prozessor ferner einen Akkumulator 44, ein Befehlsregister 46 und ein Speicheradressenregister MAR 48, wobei letzteres den Hauptspeicher 30 adressiert. Während des Betriebs des Prozessors enthält das Befehlsregister 46 ständig den Befehl, den der Mikrokode-ROM zuletzt aus dem Hauptspeicher holte und augenblicklich interpretiert. Das Hauptspeicher-Adressenregister 48 enthält ständig diejenige Adresse im Hauptspeicher, zu der durch die nächsten Speicherlese- oder Speicherschreibbefehle Zugriff erfolgt.

Durch eine Arithmetik- und Logikeinheit ALU 50 werden arithmetische und logische Operationen durchgeführt. Vom Akkumulator 44 und von einer Zweirichtungsdaten- und Steuersammelleitung 52 werden Eingangssignale der ALU 50 zugeführt.

Innerhalb des Prozessors werden Daten längs der Sammelleitung 52 übertragen. Diese Sammelleitung gestattet die Übertragung von Daten von dem Hauptspeicher 30, einem gewählten Zwischenspeicherregister 46, oder einem Eingangskanal 54 zu einem Befehlsregister 46, dem Speicheradressenregister 48 oder der ALU-Einheit 50. Wenn ein Eingabe-/Ausgabebefehl im Befehlsregister vorhanden ist, und wenn dieser Befehl festlegt, dass ein Ausgabebetrieb durchgeführt werden soll, gibt der Prozessor den Inhalt der ALU-Einheit 50 über einen Ausgangskanal 56 auf den Ausgabedatenkanal und hält die Eingabe-/Ausgabeeinrichtung (I/O-Einrichtung) fest, die diese Daten erhalten soll.

Wenn eine Leseoperation spezifiziert ist, merkt sich der Prozessor die betreffende I/O-Einrichtung und gibt Daten auf den Eingabekanal 54. Wie in Fig. 2 dargestellt ist, sind die Eingabe-/Ausgabeeinrichtungen im vorliegenden System in der Regler- und Steuereinheit 14 enthalten, die zuvor beschrieben wurde und auch in Fig. 2 enthalten ist.

Der Prozessor 10 enthält ferner einen Taktgenerator, der als Prozessortakt 58 bezeichnet ist und ein Taktgrundsignal erzeugt mit einer kennzeichnenden Wiederholfrequenz von 4,167 MHz. Wie in Fig. 2 dargestellt ist, wird das Taktgrundsignal dem Prozessor 10 zur Steuerung der Zeittaktung von Informationen und Befehlen durch den Prozessor, und ferner dem System zugeführt, um als ein Synchrongrundpuls zu dienen, um Information in und aus der Regler- und Gleichrichtersteuereinheit zu takten. Obwohl der Prozessortakt 58 in der vorliegenden Erfindung verwendet wird, um Systemtaktimpulse zu liefern, sei darauf hingewiesen, dass auch ein Taktgrundsignal gleichermassen von einer äusseren Quelle zur Durchführung derselben Funktion dem Prozessor zugeführt werden kann.

Es wird nun auf Fig. 3 Bezug genommen, die hauptsächlich in Form eines Blockdiagramms die Blöcke enthält, welche die Regler- und Gleichrichtersteuereinheit 14 bilden. Aus Klarheits- und Vereinfachungsgründen sind in Fig. 3 verschiedene, zuvor in Verbindung mit den Fig. 1 und 2 beschriebene Komponenten enthalten und mit den zuvor verwendeten Bezugszeichen versehen. Der Prozessor 10 liefert Taktgrundsignale an einen Systemtaktgenerator in der Regler- und Steuereinheit 14. Der Systemtaktgenerator 60 empfängt ferner ein dreiphasiges 60-Hz-Netzsignal von einer äusseren Netzquelle, nicht dargestellt, und liefert Taktimpulse an das System zur Verwendung bei der Synchronisierung des Systemgesamtbetriebs mit dem 60-Hz-Dreiphasennetz zur Steuerung des Zündvorgangs der Thyristoren oder SCRs, um den Motor 12 zu regeln.

Die Regler- und Steuereinheit 14 enthält ferner als einen

Teil ein Programm 62, das mit dem Prozessor 10 in Verbindung steht und den Betrieb des Reglers in der Steuereinheit 14 steuert, um letztlich die richtigen Zündimpulse an die Thyristoren oder SCRs zur Regelung des Gleichstrommotors zu liefern. Obwohl das Programm 62 im Hauptspeicher 30 der Fig. 2 enthalten sein kann, sei bemerkt, dass das Programm 62 als Teil der Regler- und Gleichrichtersteuereinheit 14 betrachtet wird, da es bestimmte Logikfunktionen ausführt, die für den Betrieb der Gesamtsteuerung des Systems wichtig sind.

In Fig. 3 ist die schon erwähnte Drehzahlreferenz 18 als digitale Schalter (RPM) und als Ein-/Aus- und Vorwärts-/Rückwärts-Schalter 18' dargestellt, die Eingangssignale über eine Prozessor-/Systemschnittstelle 64 an den Prozessor 10 liefern. Von den Schaltern 18 wird über eine Vielzahl von Leitungen 66 eine digitale Drehzahlreferenz geliefert, die in Umdrehungen pro Minute der gewünschten Motordrehzahl entspricht und in den Prozessor eingelesen wird und im Hauptspeicher oder Programm 62 unter der Steuerung des Prozessors gespeichert wird.

In ähnlicher Weise werden Signale, die den Ein-/Ausschalter des Motors und einen Schalter kennzeichnen, der die gewünschte Vorwärts- oder Rückwärtsrichtung des Motors festlegt, dem Prozessor von den Ein-/Aus- und Vorwärts-/Rückwärtsschaltern 18' über die Prozessor-/Systemschnittstelle 64 auf Leitungen 68 zugeführt. Eine Verbindung zwischen dem Prozessor 10 und der Prozessor-/Systemschnittstelle 64 erfolgt über eine Vielzahl von Leitungen 70, die aus Dateneingabe-/Ausgabeleitungen und Steuerleitungen bestehen. Wie noch erläutert wird, werden die Taktimpulse vom Systemtaktgenerator 60 auch über diese Leitungen während des Betriebs des Systems dem Prozessor zugeführt.

Eine Zündlogik 72 ist in der Regler- und Steuereinheit 14 vorgesehen, um Information zu erhalten, die einem gewünschten Zündwinkel zur Zündung der Thyristoren entspricht, um den Motor zu regeln. Diese Information wird vom Mikroprozessor über die Prozessor-/Systemschnittstelle 64 auf den Leitungen 74 geliefert. Die Zündlogik 72 liefert im wesentlichen drei Signale, ein Signal ist ein Unberbrechersignal, das auf der Leitung 76 dem Prozessor 10 zugeführt wird. Das Unterbrechersignal kann entweder die Schnittstelle 64 umgehen oder durch die Schnittstelle 64 hindurchlaufen. Ein weiteres dieser Signale stellt ein Umwandlungssignal auf einer Leitung 78 zu einem Analog-/Digitalwandler 80 dar, um diesen Wandler zu triggern, um den analogen dreiphasigen Motorstrom in einen der Stärke des Gleichstroms proportionalen Zählerwert umzuwandeln, der über die Leitung 24 und die Schnittstelle 64 dem Prozessor zugeführt wird. Zusätzlich erzeugt die Zündlogik 72 einen Zündimpuls auf einer Leitung 82, die zu einer Thyristorwähl- und Antriebsrichtungslogik 84 führt.

Die Thyristorwähl- und Antriebslogik 84 empfängt über die Schnittstelle 64 auf einer Vielzahl von Leitungen 86 digitale Information vom Prozessor 10. Diese Information stellt Wörter oder Adressen dar, um die richtige Wahl der zu zündenden Thyristoren durchzuführen, und um eine spezielle Brücke aus zwei Brücken (vorwärts oder rückwärts) im Gleichrichter 16 auszuwählen, um die Motorrichtung zu steuern. Der Betrieb der Zündlogik und der Thyristorwähl- und Antriebsrichtungslogik wird weiter unten näher erläutert.

Die genannten Drehzahlsignale auf den Leitungen 22 werden von einer Tachoimpulszähler- und Logikschaltung 88 geliefert, vergleiche Fig. 3, die Impulse von einem herkömmlichen digitalen Tachometer 90 erhält. Ein spezieller Tachometer, der in der vorliegenden Erfindung Verwendung findet, ist als Modell K827 von der Firma Avtron Corporation erhältlich. Dieser Tachometergenerator ist als optische Einrichtung ausgebildet und enthält zwei sich drehende Scheiben mit Schlitzen, die die Erzeugung von 1200 Impulsen pro Motorumdrehung durch jede Scheibe bewirken. Das Ausgangssignal von den beiden Scheiben ist im wesentlichen als Rechteckkurve ausgebildet mit 1200 Zählimpulsen pro Umdrehung der Tachometerwelle. Diese Impulse von den beiden Scheiben werden um 90° in der Phase gegeneinander versetzt, so dass die Motorrichtung dadurch wahrgenommen werden kann, dass die Verschiebung der Phasen der Impulse festgestellt wird, die vom Tachometer auf den Leitungen 92 dem Tachoimpulszähler 88 zugeführt werden. Die Art und Weise der Feststellung oder des Nachweises wird unten in Verbindung mit der Beschreibung der Tachoimpulszählerlogik 88 beschrieben.

Der in Fig. 1 erwähnte Gleichrichter 16 besteht, wie aus Fig. 3 ersichtlich ist, aus einem Block, der als Thyristoren (SCRs) 94 und Vorwärts- und Rückwärtsimpulsverstärker 96 bzw. 98 bezeichnet ist. Den Verstärkern 96 und 98 werden über eine Vielzahl von Leitungen 100 von der Thyristorwähl- und Antriebsrichtungslogik 84 Thyristorwähl- oder Adressensignale und Antriebsrichtungswählsignale zugeführt. Während des Betriebs des Systems bewirkt die vom Mikroprozessor in die Thyristorwähl- und Antriebsrichtungslogik geleitete Information, dass der richtige Verstärker der Vorwärts- oder Rückwärtsverstärker 96 bzw. 98 ausgewählt wird, um einen Zündimpuls den Thyristoren 94 zuzuführen, wenn die Zündlogik den Zündimpuls auf der Leitung 82 erzeugt. Die Ausgangszündimpulse von den Vorwärts- und Rückwärtsimpulsverstärkern 96 und 98 werden über Leitungen 102 bzw. 104 den Thyristoren 94 zugeführt. Die Leistung für den Betrieb der Thyristoren und damit des Gleichstrommotors 12 wird von einem 60-Hz-Dreiphasennetz 106 den Thyristoren 94 zugeführt. Wenn die Thyristoren gezündet sind, werden über die Leitungen 108 Impulse geliefert, um dem Gleichstrommotor 12 Strom zuzuführen, um den Motor anzutreiben. Ein Gesamtverständnis des Betriebs der vorliegenden Erfindung lässt sich am besten durch eine in Einzelheiten gehende Beschreibung aller der Logikblöcke erhalten, die in der Regler- und Gleichrichtersteuereinheit 14 der Fig. 3 beschrieben wurden. Der erste dieser zu beschreibenden Blöcke ist die Prozessor-/Systemschnittstelle, die in Fig. 4 dargestellt ist. Wie im linken Teil der Fig. 4 gezeigt ist, umfassen alle Eingangs- und Ausgangssignalleitungen zur Prozessor-/Systemschnittstelle, die links von der gestrichelten Linie verlaufen, kollektiv die Leitungen 70, wie zuvor in Verbindung mit Fig. 3 beschrieben wurde. Die vom Prozessor 10 in die Systemschnittstelle 64 übertragene Information kommt vom Ausgabekanal 56, wie zuvor in Verbindung mit Fig. 2 erläutert wurde. Im Grunde überträgt der Prozessor 10 zwei Arten von Befehlen oder Instruktionen an die Systemschnittstelle. Diese Befehle leiten die Systemschnittstelle derart, dass sie entweder gewisse Daten vom Prozessor in spezielle Register im System, zum Beispiel in die Zündlogik und die Thyristorrichtungslogik, einschreibt, oder dass sie Information von verschiedenen adressierten Eingabeeinrichtungen ausliest, die im rechten Teil der Fig. 4 dargestellt sind.

Vom Ausgabekanal 56 des Prozessors werden über Leitungen 110, 112, 114 und 116 Befehlsdaten an die Systemschnittstelle geliefert. Die Signale auf den Leitungen 112, 110 und 114 stellen Befehlsregisterbits vom Prozessor 10 dar. Wenn der Prozessor einen Lesebefehl an die Systemschnittstelle liefert, werden die Befehlsregisterbits IR1 bis IR3 auf den Leitungen 112 in einem BCD-/Dezimalwandler dekodiert, der als ein Dekoder dient, um einen als READ gekennzeichneten Leseimpuls von einem Ausgangsanschluss 6 der Einheit 118 zu erzeugen. Der Leseimpuls wird erzeugt, wenn das Befehlsregisterbit IR4 auf der Leitung 114 eine binäre Null ist, und er wird durch einen Inverter 120 in eine binäre 1 umgewandelt, um ein NOR-Tor 122 auszulösen, wenn ein READ-Befehlsregistersignal (IR-Signal) mit dem binären Wert 1 durch den Prozessor geliefert wird. Wenn das Tor 122 ausgelöst ist, liegt an dessen Ausgang ein binärer 0-Taktimpuls, der an den D-Eingangsanschluss des Dekoders 118 geleitet wird und auf diese Weise

einen READ-Impuls erzeugt, wie auf der Leitung 124 gezeigt ist. der READ-Impuls oder Leseimpuls wird zwei logischen Elementen in der Schnittstelle zugeführt, zuerst einem D-Eingangsanschluss eines zweiten BCD-/Dezimalwandlers, der als ein Dekoder 126 dient, und einem Auslöseeingangsanschluss (EN) eines 8-Bit-Multiplexers 128.

Die Dekoder 126 und der Multiplexer 128 empfangen auf der Leitung 110 die Befehlsregisterbits IR5 bis IR7. Wenn diese Bits durch einen Dekoder 126 als O-Befehl für eine Leseeinrichtung dekodiert werden, erzeugt der Dekoder ein RDVO-Signal an seinem O-Ausgangsanschluss auf einer Leitung 130, wie in Fig. 4 gezeigt ist. Das RDVO-Signal wird der Zündlogik 72 zugeführt, deren Aufgabe anschliessend erläutert wird. Wenn ferner ein READ-Befehl durch den Prozessor abgegeben wird, werden die Befehlsregisterbits IR5 bis IR7, die einem SEL-Eingang des Multiplexers 128 zugeführt werden, dekodiert, um Daten aus einer der Eingabeeinrichtungen im rechten Teil der Fig. 4 über eine gemeinsame zeitverzahnte (time shared) Sammelleitung 132 zum Datenprozessor zu leiten, wobei die als ID0 bis ID7 bezeichnete Eingangsinformation dem Eingabekanal 54 des Prozessors 10 zugeführt wird (vergleiche Fig. 2).

Wenn der Prozessor einen Schreibbefehl abgibt, wird der Befehl im Dekoder 118 in der zuvor für den READ-Impuls beschriebenen Art dekodiert, und es wird auf diese Weise ein Schreibimpuls am Ausgangsanschluss 7 auf eine Leitung 134 gegeben. Der Schreibimpuls auf der Leitung 134 wird dem Dekoder 136 und dem Logiktreiber 138 zugeführt. Der Dekoder 136 empfängt ferner die Befehlsregisterbits IR5 bis IR7 auf den Leitungen 110 und dekodiert diese Bits, um gemäss der binären Bitkonfiguration ein Ausgangssignal aus zwei Ausgangssignalen (WDV1 oder WDV3) zu erzeugen. Die letztgenannten beiden Signale, welche die Bezeichnungen WDV1, WDV3 für Schreibeinrichtungen tragen, werden der Zündlogik und der Thyristorwähl- und Antriebsrichtungslogik aus Gründen zugeführt, die noch erläutert werden. Der einem C- oder Takteingangsanschluss des Treibers 138 zugeführte Schreibimpuls gestattet, dass Daten auf einer Vielzahl von Leitungen 140 vom Prozessorausgabekanal 56 zur Zündlogik und der Thyristorwähl- und Antriebsrichtungslogik als Signale WDV0 bis WDB7 zugeführt werden.

Es wird nun auf die Eingangseinrichtungsblöcke 18, 60, 80 88 im rechten Teil der Fig. 4 Bezug genommen. Es ist ersichtlich, dass jede dieser Einrichtungen mit einer eigenen Eingabeeinrichtungszahl versehen ist, wie zum Beispiel die Eingabeeinrichtung 1 für den Systemtaktgenerator 60. Diese Einrichtungszahl entspricht der Adresse der betreffenden Einrichtung, die der Systemschnittstelle vom Prozessor geliefert wird, wenn es erwünscht ist, Information durch den Multiplexer 128 aus irgendeiner der Einrichtungen in den Prozessor zu lesen. Wenn zum Beispiel der Datenprozessor einen Lesebefehl liefert, um einen Leseimpuls auf der Leitung 124 zu erzeugen, wobei eine Adresse auf der Leitung 110 die Adresse für die Einrichtung 1 spezifiziert, so werden die Systemtakt-Eingangsdatenbits ID1B0 bis ID1B7 durch den Multiplexer 128 auf die Eingabedatensammelleiter 132 kanalisiert oder geleitet und in den Datenprozessorspeicher übertragen. Alle Übertragungen der Eingabedaten von den Eingabeeinrichtungen zum Prozessor werden in der Weise behandelt, wie gerade für den Systemtakt 60 beschrieben, mit der Ausnahme, dass die dem 8-Bit-Multiplexor 128 gelieferte spezielle Adresse die Information von der adressierten Einrichtung in den Prozessor leitet.

Es wird nun auf die Fig. 5 und 6 Bezug genommen, wobei Fig. 5 ein Blockdiagramm der Einrichtung 1, des Systemtaktgenerators 60 ist, und Fig. 6 ein Zeitsteuerdiagramm darstellt, welches zum Verständnis des Betriebs des Systemtakts nützlich ist. Die Dreiphasennetzspannung wird drei herkömmlichen Rechteckverstärkern 142 zugeführt, die entsprechende Rechteckausgangssignale mit der Bezeichnung Ø 1, Ø 2 und Ø 3 auf den Leitungen 144, 145 bzw. 146 erzeugen. Die drei Signale- Ø1 bis Ø3 werden den entsprechenden Eingängen eines D-Anschlusses eines herkömmlichen D-Typ-Flipflops von drei ähnlichen Phasennulldurchgangslogik- oder Kantendetektoren 148, 150 und 152 zugeführt. Da die Kantendetektoren 148 bis 152 ähnlich aufgebaut sind, wird nur der Kantendetektor 148 in den gestrichelten Linien in Fig. 5 in Einzelheiten gezeigt.

Die Kantendetektoren arbeiten alle in folgender Weise, wie anhand des Kantendetektors 148 beschrieben ist. Wenn das Ø 1-Signal auf der Leitung 144 auf einen positiven Wert geht, wird der D-Eingangsanschluss eines FAØ 1-Flipflops erregt, um einen Setzzustand bei Anlegen des Taktgrundsignals vom Prozessor zu einem CLK-Eingangsanschluss dieses Flipflops zu verwirklichen. Wenn das Taktgrundsignal auf einen positiven Wert geht, wird das FAØ 1-Flipflop gesetzt und bewirkt, dass der Q-Ausgangsanschluss dieses Flipflops in einen binären 1-Zustand übergeht, wodurch ein ID1B0-Signal auf der Leitung 154 erzeugt wird. Das ID1B0-Signal wird als Eingangssignal an ein negatives exklusives ODER-Tor 156 und an den D-Anschluss eines zweiten Flipflops gelegt, welches als FBØ 1 bezeichnet ist. Beim Auftreten des nächsten Taktgrundsignals erreicht das FBØ 1-Flipflop einen gesetzten Zustand, wobei der Q-Ausgangsanschluss auf eine binäre 1 geht und dadurch bewirkt, dass das exklusive ODER-Tor 156 einen Ausgangsimpuls Ø 1ZR0X auf der Leitung 158 erzeugt, vergleiche Fig. 5. Die FAØ 1- und FBØ 1-Flipflops bilden im wesentlichen ein 2-Bit-Schieberegister, dessen Ausgänge dem Tor 156 zugeführt werden. Der FAØ 1-Eingang synchronisiert das Rechtecksignal vom Ø 1-Eingang mit dem Systemtakt. Es lässt sich daher erkennen, dass der Ausgang Ø 1ZR0X des exklusiven ODER-Tors 156 einen Impuls mit der Breite des Taktbasisimpulses jedesmal dann erzeugt, wenn die Sinuskurve am Eingang durch einen Nulldurchgang mit ungefähr einer 2,7 Millisekunden dauernden Periode hindurchläuft. Das Ø 1ZR0X-Signal wird dem Eingang eines ODER-Tors 160 in Verbindung mit Signalen Ø 2ZR0X und Ø 3ZR0X von den entsprechenden Kantendetektoren 150 bzw. 152 auf den Leitungen 162 bzw. 164 zugeführt. Die Signale Ø 1ZR0X bis Ø 3ZR0X entsprechen den Phasen A, B und C der Eingangsnetzspannung.

Das Ausgangssignal des ODER-Tors 160 wird einem K-Eingangsanschluss eines ZR0X-JK-Flipflop 166 zugeführt. Das Flipflop 166 empfängt am CLK-Eingangsanschluss das Taktgrundsignal, um dieses Flipflop zu triggern, damit es gemäss dem Zustand des vom ODER-Tor 160 an den K-Anschluss angelegten Eingangssignals gesetzt oder zurückgesetzt wird. Das ZR0X-Flipflop erzeugt ein ZR0X-Signal oder Nulldurchgangssignal an seinem Q-Ausgangsanschluss, welches der Tachoimpulszähler- und Logikschaltung und zwei Zählern 168 und 170 zugeführt wird. Unter Bezugnahme auf das Zeitsteuerdiagramm der Fig. 6 lässt sich erkennen, dass das ZR0X-Flipflop 166 ein ZR0X-Signal jedesmal dann erzeugt, wenn eine Phasenspannung die Eingangsspannung kreuzt, oder es werden 6 Impulse bei einem Netzspannungszyklus von 360° erzeugt, wobei das ZR0X-Signal einen Impuls mit der Breite einer Taktgrundbreite besitzt.

Aus den Fig. 5 und 6 lässt sich erkennen, dass die drei Signale ID1B0 bis ID1B2 (die kombiniert die Leitungen 172 bilden) vom Datenprozessor benutzt werden können, um ein beliebiges 60°-Intervall innerhalb eines 360°-Phasenzyklus' der Netzspannung zu definieren. Dies ist in Fig. 6 unter Bezugnahme auf das Ø 3-Rechteck (ID1B2) dargestellt, wobei die verschiedenen Grade der Sinuseingangskurve und die verschiedenen Nulldurchgänge bei den 60°-Intervallen dargestellt sind. Wie sich durch die Beziehungen zwischen den ID1B0- bis ID1B3-Signalen ergibt, ist es relativ leicht, diese Signale so zu dekodieren, dass festgelegt wird, welches Intervall aus den 6 Intervallen eines 360°-Zyklus' während eines beliebigen gegebenen Zeitpunktes vorhanden ist. Wird zum Beispiel angenom-

men, dass das erste Intervall von 0 bis 60° läuft, so kann dieses Intervall, wenn ID1B0 eine binäre 1 ist, ID1B1 eine binäre 0 ist und ID1B2 eine binäre 1 ist, durch Dekodierung dieser 3 binären Bits als das erste Intervall des 360°-Zyklus bezeichnet werden. Eine ähnliche Dekodierung kann für die 60°- bis 120°-Intervalle, die 120°- bis 180°-Intervalle usw. durchgeführt werden.

Es wird nun wieder auf Fig. 5 Bezug genommen, in der die zuvor erwähnten Zähler 168 und 170 in Verbindung mit einem durch den Wert 45 teilenden Zähler 174 gezeigt ist. Der 4,167-MHz-Grundtakt wird dem Eingang des durch den Wert 45 teilenden Zählers 174 zugeführt, der die Taktgrundimpulse herunterteilt, um ein Signal auf einer Leitung 176 mit einer Impulsdauer von 11 Mikrosekunden zu erzeugen. Wie in Fig. 5 gezeigt ist, wird der Puls auf der Leitung 176 mit einer Dauer von 11 Mikrosekunden einem UND-Tor 178 und ferner auf einer Leitung 180 der Zündlogik zugeführt. Wie ferner auf der Leitung 180 angezeigt ist, ist der Impuls mit einer Dauer von 11 Mikrosekunden ungefähr gleich einem Viertel eines elektrischen Grads der Netzspannung, die den Rechteckverstärkern 142 zugeführt wird. Die Impulse mit 11 Mikrosekunden Dauer werden über ein UND-Tor 178 einem durch den Wert 8 teilenden Zähler 168 zugeführt, um Zeitbasisimpulse mit 88 Mikrosekunden Dauer zu erzeugen, die jeweils ungefähr 2 elektrischen Graden der Netzspannung entsprechen. Die Impulse mit 88 Mikrosekunden Dauer werden über eine Leitung 182 der Zündlogik, einem NOR-Tor 184 und dem Zähler 170 zugeführt. Der Zähler ist ein durch den Wert 32 teilender Zähler, der die Impulse mit 88 Mikrosekunden Dauer durch den Wert 32 teilt. Solange sich der Zähler 170 nicht auf einem Zählerwert von 31 befindet, liefert das NOR-Tor 184 ein Taktstoppsignal CT31 mit einer binären 1 auf die Leitung 186 als ein zweites Eingangssignal für das UND-Tor 178, um zu gestatten, dass die Impulse mit 11 Mikrosekunden Dauer durch dieses Tor zum Zähler 168 gelangen. Wenn der Zähler 170 einen Zählerwert von 31 erreicht, wird in Verbindung mit einem Impuls mit binärer 1 (88 Mikrosekunden Dauer) das NOR-Tor 104 ausgelöst, um ein Sperrsignal mit einer binären 0 an das Tor 178 zu liefern, wodurch die Zähler 168 und 170 daran gehindert werden, über die Zahl 31 hinaus zu zählen. Der aus den Zählern 168 und 170 bestehende Zähler verbleibt auf einem Zählwert 31, bis das nächste Nulldurchgangs- oder ZR0X-Signal vom Flipflop 166 erzeugt wird, um die Zähler auf 0 zurückzusetzen, wie in den zeitlichen Beziehungen der Fig. 6 gezeigt ist. Es lässt sich daher erkennen, dass der Zähler zwischen den einzelnen Nulldurchgängen der Eingangsspannung von 0 bis 31 zählt. Wie in Fig. 5 dargestellt, lässt sich feststellen, dass die Ausgangssignale ID1B7 bis ID1B3 vom Zähler 170 auf den Leitungen 188 die Zeit innerhalb des 60°-Intervalls festlegen, wie es durch das Signal ID1B0 bis ID1B2 definiert ist. Die ID1B3- bis ID1B7-Signale werden mit den ID1B0- bis ID1B2-Signalen kombiniert, um Leitungen 190 zu bilden, die dem 8-Bit-Multiplexer 128 der Prozessor-/Systemschnittstelle zugeführt werden, vergleiche Fig. 4.

Es lässt sich erkennen, dass der Prozessor 10 durch das Lesen des Systemtakts den 60°-Intervall eines 360°-Zyklus' der Eingangskurve dadurch feststellen kann, dass er die Bits ID1B0 bis ID1B2 betrachtet, während er gleichzeitig die Zahl von 2°-Inkrementen (Impulse mit 88 Mikrosekunden Dauer) der Netzphasenspannung bestimmt, die seit dem letzten Nulldurchgang (ZR0X) passiert sind.

Es wird nun auf die Fig. 7 und 8 Bezug genommen, wobei Fig. 7 ein in Einzelheiten gehendes Blockschaltbild der Tachoimpulszähler- und Logikschaltung, und Fig. 8 ein Zeitsteuerdiagramm zeigt, welches zum Verständnis des Betriebs dieser Logikschaltung nützlich ist. Wie schon in Verbindung mit Fig. 3 erwähnt, erzeugt der in der vorliegenden Ausführungsform verwendete Tachometer zwei rechteckförmige Ausgangssignale wobei jedes Ausgangssignal 1200 Zählwerte pro Umdrehung der Tachometerwelle erzeugt. Diese beiden Signale werden auf den Leitungen 92, vergleiche Fig. 7, als zwei Eingangssignale geführt, das Signal TACH-Eingangssignal 1 wird dem Operationsverstärker 192, und ein TACH-Eingangssignal 2 einem D-Eingangsanschluss eines TACH-Rev-Flipflop 194 zugeführt. Die in Fig. 8 dargestellten zeitlichen Beziehungen zeigen eine 90°-Phasenverschiebung zwischen dem TACH-Eingangssignal 1 und dem TACH-Eingangssignal 2. Das TACH-Eingangssignal 1 wird über den Verstärker 192 einem D-Eingangsanschluss eines flankengetriggerten TACH-Flipflop F/F1 vom D-Typ zugeführt, welches an seinem CLK-Anschluss das Taktgrundsignal vom Prozessor empfängt. Wie in Fig. 8 gezeigt, pendelt das TACH-Flipflop F/F1 jedesmal dann vom Setz- in den Rücksetzzustand gemäss dem Zustand des TACH-Eingangssignals 1, wenn das Taktgrundsignal vom Prozessor dieses Flipflop triggert. Der Q-Ausgangsanschluss des TACH-Flipflops F/F1 ist mit dem D-Eingangsanschluss eines zweiten Flipflops, als TACH F/F2 bezeichnet, verbunden, welches ebenso den Grundtakt an seinem CLK-Eingangsanschluss empfängt. Diese beiden Flipflops stellen im wesentlichen ein 2-Bit-Schieberegister dar, das in einer Weise funktioniert, die derjenigen der oben geschilderten Kantendetektor-Flipflops der Fig. 5 im Systemtaktgenerator ähnlich ist. Der Ausgang der TACH-Flipflops F/F1 und F/F2 wird über Leitungen 196 und 198 einem negativen exklusiven ODER-Tor 200 zugeführt. Das ODER-Tor differenziert den TACH-Eingangsimpuls 1 wirksam, der über Leitungen 196 und 198 angelegt wird, um bei jedem Übergang des TACH-Eingangssignals 1 einen Impuls mit einer Taktbreite des Grundtakts zu erzeugen. Da das TACH-Eingangssignal 1 200 Impulse pro Umdrehung der Tachomterwelle erzeugt, erzeugt der Ausgang des exklusiven ODER-Tors 200 2400 Impulse pro Umdrehung der Tachometerwelle und erzeugt ein TACH-Eingangssignal X2, das auf der Leitung 202 und in Fig. 8 dargestellt ist.

Das TACH-Eingangssignal X2 auf der Leitung 202 wird einem CLK-Eingangsanschluss eines Tachoimpulszählers 204 zugeführt, um den Zähler zu veranlassen, dass er die vom Tachometer ausgelesenen Tachoimpulse akkumuliert. Das TACH-Eingangssignal X2 wird ferner einem vorsetzbaren LSB-Eingangsanschluss des Zählers 204 zugeführt, dessen Zweck unten erläutert wird. Es sei bemerkt, dass das ZR0X-Signal vom Systemtaktgenerator auch einem vorsetzbaren Eingangsanschluss des Zählers 204 und auch einem CLK-Eingangsanschluss einer Tachoimpuls-Verriegelungsschaltung 206 zugeführt wird. Es sei aus der obigen Beschreibung des Systemtaktgenerators daran erinnert, dass immer dann ein ZR0X-Signal erzeugt wird, wenn die Eingangsphasenspannungen einen Nulldurchgang aufweisen und durch 0 laufen. Auf diese Weise lässt sich erkennen, dass der Tachozähler 204 auf einen binären Nullzustand zurückgesetzt wird, wenn ein Nulldurchgangsimpuls auftritt. Auf diese Weise ist es evident, dass der Tachoimpulszähler 204 Zählwerte akkumuliert, die für die Motorumdrehungen pro 60°-Intervall eines 60-Hz-Eingangssignals kennzeichnend sind.

Wie in Fig. 8 dargestellt ist, wird der Tachoimpulszähler 204 beim Auftreten eines ZR0X-Signals immer auf einen Nullzustand zurückgesetzt. Es ist ferner wichtig zu beachten, wie in den Fig. 7 und 8 dargestellt, dass der Inhalt des Tachoimpulszählers 204 beim Auftreten eines ZR0X-Signals an die Tachoimpuls-Verriegelungsschaltung 206 übertragen wird. Obwohl in den Fig. 6 und 7 nicht dargestellt, sei darauf hingewiesen, dass der Inhalt des Tachoimpulszählers bei der Forderflanke des ZR0X-Signals in die Tachoimpulsverriegelung übertragen wird, und dass der Tachoimpulszähler bei der Rückflanke dieses Signals zurückgesetzt wird.

Es wird nun erneut auf den vorsetzbaren LSB-Eingangsanschluss des Zählers 204 Bezug genommen. Der Zweck, dass das TACH-Eingangssignal X2 dem letzteren Anschluss zugeführt

wird, besteht darin, das am wenigsten signifikante Bit des Tachoimpulszählers auf eine binäre 1 vorzusetzen oder vorzugeben für den Fall, dass ein Tachoimpuls zur Zeit eines ZR0X-Signals oder eines Nulldurchgangs auftritt. Sofern ein gleichzeitiges Auftreten eines ZR0X-Signals mit einem TACH-Eingangssignal X2 erfolgen sollte, stellt die Vorgabe des am wenigsten signifikanten Bits sicher, dass jeder Zählwert, der während eines Nulldurchgangs auftritt, nicht unbeachtet bleibt, sondern im Tachoimpulszähler gespeichert wird. Sofern einmal der Inhalt des Tachoimpulszählers in die Tachoimpuls-Verriegelung 206 geladen ist, ist diese Information in Form der Signale ID3B0 bis ID3B7 auf den Leitungen 22 für den Prozessor verfügbar, um die Motorumdrehungen pro 60° auszulesen, wenn der Prozessor die Einrichtung 3 adressiert.

In den Fig. 7 und 8 ist ferner Logik zur Feststellung der Drehrichtung des Motors dargestellt. Die Drehrichtung des Motors wird durch ein Flipflop 194, auch TACH-Rev-Flipflop genannt, festgestellt, welches das TACH-Eingangssignal 2 an seinem D-Eingangsanschluss empfängt. Der Betrieb des Flipflops 134 ist in Fig. 8 dargestellt, die den Betrieb dieses Flipflops zeigt, wenn der Motor in Vorwärts- oder in Rückwärtsrichtung läuft. Es sei bemerkt, dass das TACH-Eingangssignal 1 dem TACH-Eingangssignal 2 ständig um 90° vorauseilt, wenn der Motor in Vorwärtsrichtung läuft. Wenn der Motor in Vorwärtsrichtung läuft, vergleiche Fig. 8, erreicht das TACH-Rev-Flipflop 194 nie den gesetzten Zustand aufgrund der Tatsache, dass das TACH-Eingangssignal 1, welches das Flipflop 194 über die Leitung 208 triggert, ständig in den gesetzten Zustand geht bevor das TACH-Eingangssignal 2 jeweils einen binären 1-Zustand erreicht. Das flankengetriggerte Flipflop 194 wird daher nie gesetzt. Bei der Rückwärtsrichtung lässt sich jedoch feststellen, vergleiche die rechte Seite der Fig. 8, dass wenn das TACH-Eingangssignal 2 dem TACH-Eingangssignal 1 um 90° vorauseilt, das TACH-Rev-Flipflop 194 einen gesetzten Zustand erreicht, wenn das TACH-Flipflop 1 einen gesetzten Zustand erreicht. Wenn das Rev-Flipflop den gesetzten Zustand erreicht, erzeugt sein Q-Ausgangsanschluss ein ID0B4-Signal mit einer binären 1 auf einer der Leitungen 22 zur Prozessor-/Systemschnittstelle. Wenn das TACH-Eingangssignal 2 dem TACH-Eingangssignal 1 vorauseilt, gibt das ID0B4-Signal mit einem binären Wert 1 dem Datenprozessor Kenntnis, dass der Motor in Rückwärtsrichtung läuft.

Es wird nun auf die Zündlogik der Fig. 9 Bezug genommen, die diese Logik in Form eines Blockschaltbildes zeigt. In Verbindung mit Fig. 9 soll ebenfalls auf Fig. 10 Bezug genommen werden, die ein Zeitablaufdiagramm zeigt, das die zeitliche Beziehung zwischen den verschiedenen Signalen innerhalb der Zündlogik 72 angibt. Wie schon erläutert, besteht der primäre Zweck der Zündlogik darin, Zündimpulse auf der Leitung 82 der Thyristorwähl- und Antriebsrichtungslogik 84 zuzuführen, wie in Fig. 3 dargestellt ist. Zusätzlich erzeugt die Zündlogik einen Umwandlungsimpuls an den Analog-/Digitalwandler auf der Leitung 78. Durch den Betrieb der Zündlogik wird dem Prozessor von einem Unterbrechungssignal auf der Leitung 210 der Fig. 9 mitgeteilt, den Berechnungsprozess für den Zündwinkel zur Erzeugung eines Zündimpulses zu beginnen, um einen Thyristor zur richtigen Zeit zu zünden.

Bei der Beschreibung des Betriebs der Zündlogik wird ferner jetzt auf Fig. 4 Bezug genommen. Aus der vorausgegangenen Beschreibung sei daran erinnert, dass der Prozessor einen Schreibbefehl und eine Adresse einer Einrichtung erzeugen muss, um einen Befehl an diese Einrichtung zu senden. Für die Zündlogik erzeugt der Dekoder 136 ein Schreibsignal für die Einrichtung 1, WDV1, das in den Fig. 4 und 10 dargestellt ist. Wie aus Fig. 10 ersichtlich, geht das WDV1-Signal von einem binären 1- auf einen binären Nullzustand über, wenn das WDV1-Signal auf der Leitung 212 bewirkt, dass ein Lastzähler-Flipflop 214 das binäre 0-Signal am CLR-Eingangsanschluss empfängt, welches dieses Flipflop zurücksetzt. Gleichzeitig wird das WDV1-Signal durch einen Inverter 216 in ein binäres 1-Signal invertiert, wobei ein Auslösesignal an einen EN-Eingangsanschluss einer Schreibdatenverriegelung 218 zugeführt wird, wodurch die Daten (WDB0 bis WDB7) von den Treibern 138 der Fig. 4 auf die Leitungen 220 geliefert werden.

Es wird nun auf die Fig. 9 und 10 Bezug genommen, und es wird darauf hingewiesen, dass das Auftreten des ersten Impulses mit 88 Mikrosekunden Dauer auf der Leitung 182 erscheint, nachdem das WDV1-Signal das Lastzähler-Flipflop 214 taktet, wodurch bewirkt wird, dass der Zähler nun einen gesetzten Zustand einnimmt und an seinem Q-Ausgangsanschluss auf der Leitung 222 ein binäres 1-Signal erzeugt. Das binäre 1-Signal auf der Leitung 222 wird einem Inverter-Eingangslastanschluss eines Abwärtszählers 224 zugeführt. Wie in Fig. 10 dargestellt ist, liefert das Lastzähler-Flipflop, wenn es sich in gesetztem Zustand befindet, und in Verbindung mit einem Impuls von 88 Mikrosekunden Dauer dem Abwärtszähler 22 entweder ein Verzögerungssignal mit 20 Sekunden Verzögerung oder TIMTG0-Signal. Das TIMTG0-Signal ist eine binäre Anordnung von Bits, die vom Datenprozessor in den Abwärtszähler eingegeben sind und den Zündwinkel der Thyristoren kennzeichnen oder diesem proportional sind. Wenn ein TIMTG0-Signal nicht in den Abwärtszähler gegeben ist, dann wird ein Datenwort eingegeben, welches eine Verzögerung von 20° repräsentiert. Eine in grössere Einzelheiten gehende Beschreibung der Aufgabe und des Zwecks der TIMTG0- und der 20°-Verzögerungssignale oder Werte wird weiter unten gegeben.

Es wird nun erneut auf ein UND-Tor 226 der Fig. 9 Bezug genommen. Das UND-Tor 226 wird durch ein Ausgangssignal mit dem binären Wert 1 vom $\bar{Q}$-Ausgangsanschluss eines ersten Detektor-Flipflops 228 ausgelöst. Wenn das Flipflop 228 in dem Rücksetzzustand ist, bewirkt der erste Impuls von 11 Mikrosekunden Dauer auf der Leitung 180, der dem Tor 226 zugeführt wird, dass der Inhalt des Zählers 224 über die Leitung 230 getaktet oder gezählt wird, und dass ein Inverter 232 den Impuls von 11 Mikrosekunden Dauer einem CLK-Anschluss des Abwärtszählers zuführt. Der Zeitablauf für die Taktung des Abwärtszählers 224 ist auf der 11-Mikrosekunden-Zeile und auf der Abwärtszählerzelle der Fig. 10 dargestellt. Der Abwärtszähler fährt mit dem Abwärtszählen bis zu einem speziellen Wert fort, bis ein Dekoder 234 für einen Zählwert 14 über eine Vielzahl von Leitungen 236 vom Zähler erkennt. Bei einem Zählwert 14, und bei einem Impuls von 11 Mikrosekunden Dauer vom Tor 226, erzeugt der Dekoder 234 einen Impuls, um einen monostabilen Umwandlungsmultivibrator 238 zu zünden. Der monostabile Multivibrator 238 erzeugt einen Umwandlungsimpuls von 8 Mikrosekunden Dauer auf der Leitung 78, der einem Analog-/Digitalwandler 80 der Fig. 3 zu dem in Fig. 10 dargestellten Zeitpunkt zugeführt wird. Dieser Impuls startet den Analog-/Digitalwandler und lässt ihn mit der Durchführung einer Analog-/Digitalwandlung des Motorstroms auf den Leitungen 24 zur nachfolgenden Verwendung durch den Prozessor beginnen.

Der Abwärtszähler setzt seine Abwärtszählung bis auf einen speziellen Wert Null fort, wie in Fig. 10 dargestellt ist. Wenn der Abwärtszähler auf den Zählwert Null gelangt, der über Leitungen 242 vom Abwärtszähler durch einen Dekoder 240 für den Zählwert Null festgestellt wird, erzeugt der Dekoder 240 für den Zählwert Null einen Impuls auf einer Leitung 244, der einem D-Anschluss des Detektor-Flipflops 1, 228, zugeführt wird. Beim Auftreten des nächsten Taktgrundsignals, welches dem CLK-Anschluss des Flipflops 228 zugeführt wird, geht dieses Flipflop in den gesetzten Zustand und bewirkt, dass ein binäres 0-Signal nun dem UND-Tor 226 zugeführt wird, um die Taktimpulse von 11 Mikrosekunden Dauer daran zu hindern, dass sie zum Abwärtszähler 224 gelangen. Dies wird

durch die Bemerkung «Stopp Abwärtszähler» in Fig. 10 gezeigt. Wenn das Detektor-Flipflop 228 in einen gesetzten Zustand geht, geht der Q-Ausgangsanschluss auf eine binäre 1, um gleichzeitig einen Eingang eines UND-Tors 246 zu erregen und ein Setzsignal mit einem binären Wert 1 einem D-Anschluss eines zweiten Detektor-Flipflops 248 zuzuführen. Es sei bemerkt, vergleiche Fig. 10, dass das UND-Tor 246 in dem Augenblick ausgelöst wird, wenn das Flipflop 228 in den gesetzten Zustand übergeht, da das Flipflop zu jener Zeit zurückgesetzt ist. Der Ausgang des UND-Tors 246 liefert nun ein Triggersignal an einen J-Eingangsanschluss eines Unterbrechungs-Flipflops 245, das die Erzeugung eines Unterbrechungssignals für den Datenprozessor bewirkt. Das Unterbrechungssignal bewirkt, dass der Datenprozessor in eine Unterbrechungssubroutine eintritt, um die Berechnungen des Zündwinkels zum anschliessenden Zünden der Thyristoren zu starten.

Es sei bemerkt, dass das erste Taktgrundsignal, welches dem Setzen des Flipflops 28 folgt, das Flipflop 248 setzt, wodurch dessen $\bar{Q}$-Ausgangsanschluss auf eine binäre 0 geht und dadurch das UND-Tor 246 entregt. Dies bewirkt die Erzeugung eines kurzen Impulses, der einem INT-Flipflop 250 zugeführt wird, wie in Fig. 10 durch die Überlappung zwischen den DET-FF1- und DET-FF2-Signalen angezeigt ist. Es lässt sich ferner erkennen, dass gleichzeitig mit dem Setzen des Unterbrechungs-Flipflops 250 das Ausgangssignal vom UND-Tor 246 einem monostabilen Zündimpuls-Multivibrator 252 (FP) zugeführt wird, um einen Zündimpuls von 23 Mikrosekunden Dauer auf der Leitung 82 der Thyristorwähl- und Antriebsrichtungslogik 84 zuzuführen. Die Erzeugung des Zündimpulses ist in Fig. 10 dargestellt, wobei zu diesem Zeitpunkt ein Thyristorpaar gleichzeitig mit der Erzeugung des Unterbrechersignals gezündet wird. Die Zündlogik bleibt im vorliegenden oder vorgegebenen Zustand, bis ein anderes WDV1-Signal auf der Leitung 212 empfangen wird und veranlasst, dass neue Daten in den Abwärtszähler 224 in der eben geschilderten Weise eingegeben werden.

Wenn der Abwärtszähler mit neuen Daten beladen wird, liefert der Dekoder für den Zählwert Null ein Rücksetzsignal auf der Leitung 244 zum Flipflop 228, wodurch diesem Flipflop nun gestattet wird, einen Rücksetzungszustand zu erreichen und gleichzeitig das Flipflop 248 zurückzusetzen. Wenn das Flipflop 228 sich zurücksetzt, geht sein $\bar{Q}$-Ausgangssignal auf der Leitung 254 auf den binären Wert 1, wodurch das UND-Tor 226 ausgelöst wird und dem Zähler 224 das Zählen gestattet, nachdem dieser Zähler beladen wurde. Wie in Fig. 10 dargestellt, muss der Datenprozessor eine Zeit nach der Zündung des Thyristorpaares ein RDV0-Leseeinrichtungen-Nullsignal auf der Leitung 130 an einen Löscheingangsanschluss CLR des Unterbechungs-Flipflops 250 senden, um dieses Flipflop zur Vorbereitung auf das Aussenden eines weiteren Unterbrechungssignals an den Prozessor unmittelbar nach Erzeugung eines Zündimpulses an die Thyristoren zurückzusetzen.

Es wird nun auf die Fig. 11A und 11B Bezug genommen, wobei Fig. 11A oben an Fig. 11B anschliesst, um eine Figur zu bilden, die die Logik der Wähl- und Antriebsrichtungslogik 84 und ein elektrisches Schema der aus Thyristoren bestehenden Vorwärts- und Rückwärtsantriebsbrücken in Einzelheiten zeigt. Ferner ist der Analog-/Digitalwandler 80 gezeigt, der den analogen Motorstrom über eine Leitung 256 aus dem herkömmlichen Dreiphasen-Brückensummiergleichrichterkreis 258 erhält. In Fig. 11A wird die dreiphasige 60-Hz-Netzspannung (auch 50-Hz-Netzspannung) als Ø A, Ø B und Ø C auf den Leitungen 106 den entsprechend zugeordneten Anoden und Kathoden der Vorwärts- und Rückwärtsthyristorbrücken zugeordnet, die je aus 6 Thyristoren bestehen, die mit P1 bis P3 und N1 bis N3 bezeichnet sind, vergleiche Fig. 11A. Der Betrieb der Vorwärts- und Rückwärtsthyristorbrücken wird nicht in Einzelheiten beschrieben, da es sich um herkömmliche Brükkenzündnetzwerke handelt, die zur Regelung eines Gleichstrommotors bekannt sind. Eine derartige herkömmliche Brücke wird von der General Electric Comp. hergestellt und unter der Bezeichnung Siltrol 1 verkauft, die als integriertes statisches Wandlungs- und Regelelement für einstellbare Drehzahlantriebe unter der Bezeichnung IC3610 bekannt ist.

Drei Stromtransformatoren, 260, 262 und 264, sind je einer Netzphasenspannung Ø A bis Ø C zugeordnet. Drei Transformatoren liefern Wechselstromeingangssignale an die Dreiphasen-Brückensummierverstärker 258 über entsprechende Leitungen, wobei das Ausgangssignal des Gleichrichters an den Wandler 80 der Durchschnitt der drei Eingangsströme ist. Wie schon erwähnt, besitzt der Analog-/Digitalwandler 80 einen herkömmlichen Aufbau, wobei ein derartiger Wandler von der Firma Analog Devices Inc. als Modell ADC-8QU hergestellt wird. Dieser spezielle Wandler ist ein vollständiger 8-Bit-Wandler mit aufeinanderfolgender Approximation und hoher Geschwindigkeit, der das Eingangsanalogsignal auf der Leitung 256 bei Empfang eines Eingangsbefehls, der als Wandlungsimpuls auf der der Leitung 78 bezeichnet ist, in einen digitalen Wert wandelt. Bei diesem speziellen Wandler bezeichnen 7 der 8 Bits die Stromgrösse, und das achte Bit bezeichnet die Polarität des Stroms. Es sei an die obige Erläuterung der Zündlogik gemäss Fig. 9 erinnert, aus der sich ergibt, dass die Zündlogik einen Wandlungsimpuls von 8 Mikrosekunden Dauer auf der Leitung 78 zum Analog-/Digitalwandler sendet, wenn der Abwärtszähler einen Zählwert von 14 erreicht. Dieser Wandlungsimpuls startet den Analog-/Digitalwandler 80, um den analogen Motorstrom auf der Leitung 256 in einen digitalen Wert zum nachfolgenden Transfer an den Datenprozessor über die Prozessorschnittstelle als Datenbits ID5B0 bis ID5B7 auf der Leitung 24 umzuwandeln.

Wie in Fig. 4 gezeigt ist, wird der Transfer des Motorstroms auf der Leitung 24 durchgeführt, wenn der Analog-/Digitalwandler 80 (Einrichtung 5) über den 8-Bit-Multiplexer 128 adressiert wird, um Daten über die Sammelleitung 132 dem Prozessor zuzuleiten. Die Adressierung des Analog-/Digitalwandlers wird dadurch verwirklicht, dass der Datenprozessor eine richtige Adresse in die Bits IR5 bis IR7 eingibt und diese Bits an den SEL-Anschluss des Multiplexers 128 zusammen mit dem READ-Impuls an dem Auslöseeingangssignal des Multiplexers anlegt. Die richtige binäre Bitkonfiguration der Bits IR5 bis IR7 leitet die Motorstromauslesung vom Analog-/Digitalwandler 80 durch den Multiplexer auf der Sammelleitung 132 zum Transfer an den Datenprozessor.

Es wird nun auf Fig. 11B und auf die Thyristorwähl- und Antriebsrichtungslogik 84 Bezug genommen. Der primäre Zweck der Wähl- und Antriebsrichtungslogik besteht darin, ein Datenwort oder eine Adresse vom Datenprozessor über Leitungen 266 auf den richtigen Datenleitungen 266 (WDB0 bis WDB7) vom Treiber 138 der Fig. 4 zu erhalten. Dieses Datenwort stellt eine binäre Bitfolge dar, die durch ein WDV3-Signal auf der Leitung 270 vom Dekoder 116 der Fig. 4 in ein Thyristorsteuer- oder Wählregister 268 geladen wird. Wenn der Prozessor einen Schreibbefehl sendet, der die Schreibeinrichtung 3 adressiert, geht das WDV3-Signal auf der Leitung 270 auf eine binäre Null über und wird durch einen Inverter 272 in eine binäre 1 invertiert, und liefert auf diese Weise ein Auslöseladesignal an das Register 268, um eine Thyristorpaaradresse in das Register zu laden. Die Ausgänge von allen Stufen oder Bits des Registers 268 sind bis auf eine Stufe mit einem entsprechenden UND-Tor aus einer Vielzahl von UND-Toren 274, 276, 278 und 280 verbunden. Das Ausgangssignal von den einzelnen UND-Toren ist mit einem Signal bezeichnet, das in jeder der Vorwärts- und Rückwärtsbrücken einem der Thyristoren entspricht. So entspricht zum Beispiel ein Ausgangssignal P1 vom UND-Tor 274 dem Thyristor P1 in den Vorwärts- oder Rückwärtsthyristorbrücken. Wenn es erwünscht ist, ein spezielles

Thyristorpaar in einer der Brücken zu zünden, wird ein binäres Wort oder Adresse in das Register 268 gebracht, um die speziellen UND-Tore 274 bis 280 auszulösen, so dass sie ihre geeigneten Steuersignale an die entsprechenden Vorwärts-/Rückwärts-Antriebsschaltverstärkerkreise (FWD/REV) liefern.

Diese FWD/REV-Antriebskreise besitzen einen herkömmlichen Aufbau und sind mit 282, 284, 286, 288 bezeichnet. Jeder Kreis entspricht einem mit gleicher Zahl versehenen Thyristor in den Vorwärts- und Rückwärtsantriebsbrücken. So ist zum Beispiel die P1-FBD/REV-Antriebsschaltung 282 über Leitungen 290 und 292 mit entsprechenden Steuerelektroden des Thyristors P1 in jeder Brücke verbunden. Ähnliche Verbindungen bestehen von der Antriebsschaltung 284 zu den P2-Torelektroden und von den Antriebsschaltungen 286 und 288 zu den Steuerelektroden N2 und N3. In Fig. 11B lässt sich erkennen, dass nur vier der UND-Tore, welche die P1- bis N3-Signale erzeugen, und der Antriebskreise gezeigt sind, die den Thyristoren P1 bis N3 zugeordnet sind. Das UND-Tor und die Antriebselektronik für die Thyristoren P3 und N3 kommen aus Vereinfachungsgründen aus dem Wählregister 268 nur als gestrichelte Linien heraus.

Es ist wichtig zu beachten, dass ein Bit des Zündregisters 268 ein FWD/REV-Signal auf einer Leitung 294 an alle FWD/REV-Antriebskreise 282 bis 288 liefert. Die Antriebskreise 282 bis 288 stellen herkömmliche Treiber- oder Schaltkreise bekannter Ausführungsform dar, die in der Lage sind, logische Eingangssignale zu empfangen, um ihre Ausgangssignale selektiv zwischen eine von zwei Leitungen zu schalten, die aus jeder Antriebsschaltung herausführen. So wird zum Beispiel beim Betrieb der Antriebsschaltung 282, um den Thyristor P1 der Vorwärtsbrücke zu aktivieren oder zu zünden, ein Signal mit dem binären Wert 1 vom Register 268 als ein Eingangssignal an das Tor 274 geliefert, und beim Auftreten des Zündimpulses auf der Leitung 82 und der Zündlogik wird das Tor 274 ausgelöst und lässt Impulse durch den Antriebskreis P1 zum Vorwärtsthyristor P1 hindurch. Andererseits wird der Antriebskreis 282 aktiviert, sofern das FWD/REV-Bit den binären Wert 0 besitzt, um den Zündimpuls auf der Leitung 292 zum Thyristor P1 der Rückwärtsthyristoren zu übertragen. In der vorliegenden Ausführungsform und beim Betrieb der Thyristoren des Gleichrichters 94 ist es wünschenswert, die Thyristoren immer paarweise, wie zum Beispiel P1 und N2 in den Vorwärts- oder Rückwärtsbrücken zu zünden. Das in das Register 268 geladene Wort besitzt immer zwei binäre Bits, die den zu zündenden Thyristoren entsprechen. So aktiviert zum Beispiel die binäre 1, die den Thyristor P1 zünden soll, das Tor 274, und die binäre 1, die den Thyristor N2 zünden soll, aktiviert das Tor 280, wobei die anderen Tore nicht erregt oder ausgesteuert sind. Um den Gesamtbetrieb des erfindungsgemässen Systems voll zu verstehen und zu würdigen, wird es als vorteilhaft erachtet, zu beschreiben, wie der Zündwinkel zur Zündung der Thyristorpaare in dem System abgeleitet wird.

Es wird nun auf Fig. 12 Bezug genommen, die den Zusammenhang zwischen den Dreiphasen-Netzspannungen Ø A, Ø B und Ø C und eine Darstellung zeigt, auf welche Weise der Zündwinkel, der als FINVAL bezeichnet wird, für die Thyristorpaare entwickelt wird, um einen variablen Wert zu erzeugen, der für ein Signal TIMTG0 kennzeichnend ist, das die zu verstreichende Zeit darstellt, die einen berechneten Wert darstellt, der proportional zum Zündwinkel ist. Es ist bekannt, dass der Zündwinkel zur Steuerung von Thyristorgleichrichtern des erfindungsgemäss verwendeten Bautyps vom Phasen-/Phasenkreuzpunkt bis zum Zündpunkt des Thyristorpaars gemessen wird. Gemäss der Erfindung wird der Wert für den Zündwinkel, FINVAL, zur Erzeugung einer Motorklemmenspannung, die gleich VT ist, dadurch erhalten, dass ein Tabellennachschlag im Speicher mit der Darstellung oder Anordnung gemäss Tabelle I erfolgt. Die im Speicher gespeicherte Grösse, der Zündwinkel, ist auf der rechten Spalte als FINVAL-Zählwert gezeigt.

Die FINVAL-Tabelle wird aus der Beziehung

$$FINVAL = 245{,}8\ \mathrm{Cos}^{-1}(3VT/\pi V_{LN})$$

berechnet, wobei 245,8 gleich der Zahl der Impulse von 11 Mikrosekunden Dauer ist, die dem Abwärtszähler zugeführt werden, um den Zähler pro elektrisches Grad abwärts zu zählen. V_{LN} ist als Leitererdspannung der Eingangsnetzspannung definiert.

Tabelle I
Spannung gegen FINVAL-Zählwerte

Motorklemmenspannung	Zählwerte
-272	715
-256	671
-240	640
-224	615
-208	592
-192	572
-176	553
-160	530
-144	519
-128	503
-112	487
-960	472
-800	457
-040	443
-480	428
-320	414
-160	400
0	386
160	372
320	358
480	343
40	329
800	314
960	300
112	284
128	269
144	253
160	236
176	218
192	199
203	179
224	157
240	132
256	101
272	57

Es wird nun wieder auf Fig. 12 Bezug genommen, wobei diese Figur die Ableitung der TIMTG0-Gleichung zeigt, und TIMTG0 ein Wert ist, der proportional zum Zündwinkel ist, der in den Abwärtszähler 224 der Fig. 9 geladen wird, um das richtige Thyristorpaar zur richtigen Zeit zu zünden. Per Definition gilt:

$$TIMTG0 = FINVAL - (NEWTIM + 1) \cdot 8 - T_P.$$

Das Verfahren oder die Reihenfolge, mit der der Abwärtszähler mit dem Wert TIMTG0 beladen wird, ist in der vereinfachtesten Form durch folgende Schritte erläutert:

1. Der Prozessor berechnet den Wert FINVAL, den Zündwinkel, der das laufende Ausgangssignal des Reglers bildet;

2. Der Prozessor liest anschliessend den Systemtakt (Einrichtung 1), wie zuvor in Verbindung mit den Fig. 4 und 5

beschrieben, um das 60°-Intervall im Eingangsleistungszyklus aufzubauen oder zu definieren, und um ferner eine Zeit innerhalb dieses Intervalls zu definieren, und anschliessend berechnet er den Wert von NEWTIM und TIMGO.

3. Der Prozessor liest anschliessend den Systemtakt wiederholt, bis der Wert des Takts gleich NEWTIM ist, anschliessend arbeitet der Prozessor, um den Abwärtszähler mit TIMTGO zu laden. NEWTIM ist der vom Prozessor berechnete Wert, der vom Programm benutzt wird, um anzugeben, zu welcher Zeit TIMTGO in den Abwärtszähler geladen werden soll, so dass der Abwärtszähler zur richtigen Zeit mit dem Zählen beginnt. Das Beladen zu dem durch NEWTIM angegebenen Zeitpunkt stellt sicher, dass das Programm mit dem Zünden der Thyristorenpaare synchronisiert ist.

Der zuvor erwähnte Prozessor CRD8 benutzt einen 300 Nanosekundenspeicher, der gestattet, den Schritt 2 in ungefähr 120 Mikrosekunden auszuführen. Diese Periode von 120 Mikrosekunden Dauer ist geringfügig kleiner als die Zeitdauer von zwei Impulsen mit 88 Mikrosekunden Dauer, die durch den Systemtakt 360 der Fig. 5 entwickelt werden. Wenn daher TCLOCK die Zeit darstellt, die durch die Bits ID1B3 bis ID1B7 des Systemtakts zu Beginn des erwähnten Schritts 2 gegeben ist, und wenn NEWTIM durch NEWTIM = TCLOCK + 2 gegeben ist (Berechnungszeit des Prozessors für NEWTIM und TIMTGO), wird Schritt 2 ständig rechtzeitig abgeschlossen, um den Abwärtszähler 224 vor dem Systemtaktübergang bei NEWTIM + 1 zu laden. Die Grösse + 1, die in Fig. 12 der Grösse NEWTIM angehängt ist, gibt die Taktperiode von 88 Mikrosekunden Dauer an, die erforderlich ist, um den Abwärtszähler vom Prozessor zu laden. Es sei an die Beschreibung des Systemtakts nach Fig. 5 erinnert, wonach der Zähler 170 von Nulldurchgang zu Nulldurchgang (ZR0X) von Null bis 31 zählt. Es ist für den Zähler möglich, auf einem Zählwert 31 für ein Intervall stehen zu bleiben, das gleich 32 Zählwerten ist, wobei der letzte Zählwert des Zählers 170 länger als die vorausgegangenen Zählwerte gemacht wird. Wenn NEWTIM gleich oder grösser 31 ist, so muss in diesem Fall ein schneller Impuls oder 11 Mikrosekundenimpuls dem Wert T_p zuaddiert werden, dass das 31. Intervall des Systemtakts länger ist. Sofern NEWTIM grösser als 31 ist, muss das System ferner zum Rücksetzen des Systemtakts beim nächsten Nulldurchgang (CR0X) korrigiert werden.

Es gilt FINVAL = T_p + (NEWTIM + 1) + TIMTGO, vergleiche Fig. 12. In der vorliegenden Ausführungsform werden FINVAL, T_p und TIMTGO in schnellen Zählwerten oder 11 Mikrosekundenimpulsen ausgedrückt, und NEWTIM + 1 wird in langsamen Zählwerten oder 88 Mikrosekundenimpulsen ausgedrückt. Um daher in äquivalente Werte umzuwandeln, gilt FINVAL = T_p + 8 (NEWTIM + 1) + TIMTGO.

Der Multiplikationsfaktor 8 soll NEWTIM + 1 an T_p und TIMTGO angleichen, da es 8 schnelle Zählwerte (Impulse von 11 Mikrosekunden Dauer) für einen langsamen Zählwert (Impulse von 88 Mikrosekunden Dauer) braucht.

Es wird mit der Entwicklung der TIMTGO-Gleichung fortgefahren, und die Substitution des Werts von TCLOCK für NEWTIM in der Gleichung liefert: TIMTGO = FINVAL − T_p − 8 (TCLOCK + 3).

Es sei daran erinnert, dass es ungefähr zwei langsame Taktimpulse benötigt, um den 360°-Systemtakt zu lesen und NEWTIM und TIMTGO zu berechnen. Diese Zeit muss daher in NEWTIM durch Addition von +2 addiert werden. Wenn daher TCLOCK die vom Prozessor gelesene Zeit ist, und wenn zwei langsame Taktimpulse an Todzeit hinzuaddiert werden, um die Rechenzeit zu kompensieren, dann gilt: NEWTIM + 1 = TCLOCK + 3, wie in der obigen Gleichung für TIMTGO gezeigt ist.

Um TIMTGO zu berechnen gilt, da T_p in langsamen Taktimpulsen vorliegt, TIMTGO = FINVAL − T_p − 8 × TCLOCK − 24. Beachte den Ausdruck 8(TCLOCK+3) oben, wobei drei langsame Impulse gleich 24 schnellen Impulsen sind.

Es wird weiterhin auf Fig. 12 Bezug genommen, in der T_p als der Winkel vom Phasen-/Phasenkreuzpunkt, der den Zündwinkel Null für den zu zündenden Thyristor festlegt, bis zum nächstfolgenden Phasennulldurchgang definiert ist. Ein anderer Weg, dies zu bestätigen, besteht darin, den nächstfolgenden Phasennulldurchgang im 0°- bis 360°-Zyklus zu betrachten und von diesem Winkel den Referenzwinkel des zu zündenden Zellenpaars abzuziehen, wobei dies die Grösse T_p ergibt. Wenn zum Beispiel der nächstliegende Nulldurchgang durch die Phase Ø C gegeben ist, die bei 60° zu negativen Werten läuft, vergleiche Fig. 12, und wenn die Thyristorpaare P1/N2 gezündet werden sollen, dann beträgt der Referenzwinkel 30° (60°−30° = T_p), wobei 30° der Winkel zwischen dem Ø ØA/0C-Kreuzungspunkt und dem Nulldurchgang von ØC ist. Sofern T_p + 24 = TABTP sei, dann gilt TIMTG0 = FINVAL − 8 × TCLOCK − TABTP − CORR, wobei CORR die Korrektur für den zuvor erwähnten langen 31. Impuls des Systemtakts ist.

Beim Betrieb des Programms wird der Wert vom TABTP aus einer Nachschlagetabelle erhalten, wie in Tabelle II dargestellt ist. Unter Bezugnahme auf Tabelle II lässt sich erkennen, dass die TABTP-Tabelle 11 Erklärungen (entry) in den schnellen Zählwerten enthält, die Gradwerte kennzeichnen, die als ein Gleichwert oder Offset bei der TIMTGO-Gleichung dienen, um das tatsächliche Zeitintervall zu der Zeit zu kompensieren, innerhalb dessen der Systematik vom Rechner ausgelesen wird.

Es wird nun auf Fig. 6 und die Tabelle II Bezug genommen, und es wird darauf hingewiesen, dass die Systemtaktbits ID1B0 bis ID1B2, die die 3 signifikantesten Bits darstellen, in 60°-Intervalle dekodiert werden können, die die Nummern 1 bis 6 tragen und in der linken Spalte der Tabelle II, ebenso wie in Fig. 6 gezeigt, als KOCT bezeichnet sind. Es wird nun auf die zweite Spalte von links der Tabelle II Bezug genommen, und es lässt sich erkennen, dass eine als TABPH bezeichnete Tabelle, die die Zahl der Nulldurchgänge kennzeichnet, in sequentiellen Plätzen, die im Speicher als PHA1 bis PHA6 bezeichnet sind, gespeichert ist, wobei jeder der Plätze einer der betreffenden Phasen entspricht, wie ebenfalls in Tabelle II angegeben ist.

Während der Zeit zum Lesen des Systemtaktes benutzt der Rechner die KOCT-Zahl, um den entsprechenden der PHA-Plätze im TABPH zu adressieren, wie in Tabelle II angezeigt ist. Es lässt sich zum Beispiel erkennen, dass KOCT5 der Fig. 6 und in der linken Spalte der Tabelle II gleich dem Phasennulldurchgang PHA1 oder ØA ist, dass KOCT4 gleich dem Phasennulldurchgang PHA2 oder ØC ist usw.

Der Prozessor enthält einen Thyristorpaarzündzähler, der in einer Spalte der Tabelle II als PH bezeichnet ist. Der PH-Zähler wird jedesmal, wenn während des Programms ein Thyristorpaar gezündet wird, um einen speziellen Wert erhöht oder aktualisiert. Der Zündvorgang findet also in einer speziellen Folge statt. Um einen richtigen TABTP-Wert für die Berechnung von TIMTGO zu erhalten, wird die Adresse, die aus der Differenz zwischen PHA und PH (PHA minus PH)-Werten entwickelt ist, verwendet, um eine Adresse für die TABTP-Tabelle zu entwickeln. Der Thyristorpaarzähler PH gibt immer ein spezielles Paar von Thyristoren an, das gezündet werden soll. Wenn zum Beispiel der Thyristorpaarzähler PH auf dem Wert 1 steht, wird das Thyristorpaar P1/N2 gezündet, während das Paar P3/N2 gezündet wird, wenn der Zähler auf dem Wert 6 steht usw.

Für jeden der Speicherplätze TABTP sind 6 Adresseneinsprünge (adress entries) vorhanden, wobei jede dieser 6 Adressen für einen der 6 Nulldurchgänge in einem vollständigen Zyklus der Eingangsspannung repräsentativ ist. Es sei bemerkt, dass jedes der Thyristorenpaare jeweils einmal alle 60° gezündet wird oder 6 Zündungen pro 360°-Zyklus der Eingangssinus-

kurve besitzt. Es sei ferner bemerkt, dass die PHAO-Nulldurchgangszahl nicht immer mit dem PH-Zählwert korrespondiert. Dies ist in der Tatsache begründet, dass jedes gegebene Zellenpaar während einer 360°-Zyklusperiode bei jedem beliebigen 60°-Intervall gezündet werden kann. Es ist diese Differenz zwischen den PHA-Zahlen und den Zahlen des PH-Zählers, die die Ableitung von Adressen für die TABTP-Tabelle gestattet, um aus der Tabelle die richtige Zählerzahl in schnellen Zählwerten zur Einfügung in die TIMTGO-Gleichung zu gewinnen.

Tabelle II

60°-Intervall KOCT	TABPH Ø-Nulldurchgang PHA	SCR Thyristorpaarzähler PH	FWD/REV Thyristorpaar, gezündet =	PHA-PH TABTP Tabelle addieren	TABTP Tabelleninhalt (schnelle Zählwerte)
5	1-Ø A -	1	P1/N2 =	0	-105 = 30°
4	2-Ø C -	2	P1/N3 =	0	
6	3-Ø B -	3	P2/N3 =	0	
2	4-Ø A -	4	P2/N1 =	0	
3	5-Ø C -	5	P3/N1 =	0	
1	6-Ø B -	6	P3/N2 =	0	
	-	2	P1/N3	-1	-362 = -90°
	-	3	P2/N3	-1	
	-	4	P2/N1	-1	
	-	5	P3/N1	-1	
	-	6	P3/N2	-1	
	-	1	P1/N2		
	-	3	P2/N3	-2	-619 = -180°
	-	4	P2/N1	-2	
	-	5	P3/N1	-2	
	-	6	P3/N2	-2	
	-	1	P1/N2	4	
	-	2	P1/N3	4	
	-	4	P2/N1	-3	667 = +180°
	-	5	P3/N1	-3	
	-	6	P3/N2	-3	
	-	1	P1/N2	3	
	-	2	P1/N3	3	
	-	3	P2/N3	3	
	-	5	P3/N1	-4	410 = +90°
	-	6	P3/N2	-4	
	-	1	P1/N2	2	
	-	2	P1/N3	2	
	-	3	P2/N3	2	
	-	4	P2/N1	2	
5	1-Ø A -	6	P3/N2 =	-5	153 = +30°
4	2-Ø C -	1	P1/N2 =	1	
6	3-Ø B -	2	P1/N3 =	1	
2	4-Ø A -	3	P2/N3 =	1	
3	5-Ø B -	4	P2/N1 =	1	
1	6-Ø C -	5	P3/N1 =	1	

Bevor mit einer Beschreibung des Programms zur Steuerung des Gesamtbetriebs des erfindungsgemässen Regel- und Steuersystems fortgefahren wird, wird nun auf Fig. 13 Bezug genommen, die in einem vereinfachten Balkendiagramm den Gesamtbetrieb des Systems zeigt, um den dem Zündwinkel proportionalen Wert TIMTGO zu erzeugen, um die Thyristorpaare im Gleichrichter 16 der Fig. 1 zu zünden. Um den Betrieb der Fig. 13 zu verstehen, wird es als vorteilhaft erachtet, zuerst anzunehmen, dass irgendein Thyristorpaar im Gleichrichter gerade gezündet hat. Wie schon erläutert, erzeugt das INT-Flipflop 250 der Fig. 9 ein Unterbrechungssignal an den Prozessor, wenn ein Thyristorpaar gezündet wird. Dieses Unterbrechungssignal veranlasst den Prozessor, zu einer Unterbrechungssubroutine zu springen, die das Einlesen des Analog-/Digitalwandlers 80 in den Rechner bewirkt. Wie gezeigt, belädt der Prozessor zu diesem Zeitpunkt den Abwärtszähler mit einem Zählwert, der proportional zu einer 20°-Verzögerung ist. Die vorliegende Erfindung ist in der Lage, entweder in einem kontinuierlichen oder nicht kontinuierlichen Strombetrieb zu arbeiten, und der Zweck, dass die 20°-Verzögerung in den Abwärtszähler 224 der Fig. 9 geladen wird, besteht darin, dem Prozessor Zeit zu geben, dass er die Betriebsart bestimmen kann, in der der Regler arbeiten soll, und dass er die Verstärkungen und Konstanten in richtiger Weise entweder für den kontinuierlichen oder den diskontinuierlichen Betrieb setzen kann. Die Art und Weise, wie dies durchgeführt wird, wird weiter unten in Verbindung mit dem Programm beschrieben.

Anhand der Fig. 13 sei bemerkt, dass bei einem Zählwert von 14 im Abwärtszähler 224 ein Umwandlungsimpuls auf der Leitung 78 an den Analog-/Digitalwandler geschickt wird, um den Wandler zu aktivieren, damit der Wandler mit der Analog-Digitalwandlung beginnt. Bei Beendigung der 20°-Verzögerung, oder wenn der Abwärtszähler 224 den vorgegebenen Zählwert 0 erreicht, sendet das INT-Flipflop 250 wieder ein zweites Unterbrechungssignal an den Prozessor. Bei Empfang des zweiten Unterbrechungssignals führt die Unterbrechungssubroutine des Prozessors nun die Berechnungen des Zündwinkels FINVAL durch, um den TIMTGO-Wert zu entwickeln. Wie aus Fig. 13 erkennbar ist, findet die gesamte Auslesung und Berechnung des Zündwinkels zwischen der Zündung aufeinanderfolgenden Thyristoren statt. Da alle 60° des Eingangssinuskurvenzyklus' eine Thyristorzündung erfolgt, wird also die gesamte Berechnung für den Zündwinkel zur Zündung des nächsten Thyristorpaars in einem 60°-Intervall vorgenommen. Die 20°-Verzögerung, die ausgewählt wurde, stellt den Maximalwert dar, die Zeit für die Berechnungen des Reglers gibt (das heisst Zeit, um den Zündwinkel zu berechnen), und um einen positiven TIMTGO-Wert zu erzeugen, wenn die Phasenvoreilrate maximal ist.

Der zweite vom Prozessor gelesene Strom wird zu den Berechnungen der Reglerantwort benutzt. Vorteilig bei der Ausführung der Berechnungen auf diese Weise ist:

1. Die Regelzeitverzögerung, wie vom Gesamtregler gesehen, wird auf ein Minimum gebracht, wodurch die Eigenschaften des Reglers maximiert werden.

2. Die zweite Stromlesung besitzt immer einen gewissen endlichen Wert bei allen praktischen Betriebsregeln des Reglers, so dass der Regler während der diskontinuierlichen Strombetriebsart arbeiten kann. Dies ergibt sich aufgrund der Tatsache, dass die zweite Stromlesung 20° nach der ersten Stromlesung vorgenommen wird.

3. Wie noch erläutert wird, wird ein einziger Abwärtszähler, wie der Abwärtszähler 224 der Fig. 9 benötigt, da der Zählvorgang immer erst gestartet wird, nachdem das vorausgegangene Thyristorpaar gezündet ist.

Es wird weiter auf Fig. 13 Bezug genommen, wenn die Berechnungen abgeschlossen sind, lädt der Prozessor den TIMTGO-Wert in den Abwärtszähler 224 der Fig. 9, und zu diesem Zeitpunkt beginnt dieser Zähler gegen den Wert 0 zu zählen. Das Programm verzweigt sich dann unmittelbar zu einer READ-TACH-Zählersubroutine RDTACH, bei der der Tachoimpulszähler 88 vom Prozessor gelesen wird und ein Wert einer Gegen-EMK für Vorwärtsspeisung (CEMF) zur Benutzung bei der Berechnung der befohlenen Motorklemmenspannung (VT) benutzt wird.

Bei Beendigung der Subroutine RDTACH verzweigt sich das Programm zurück zur Unterbrechungssubroutine, wobei diese Subroutine eine Änderungsgeschwindigkeit des Stromsetzwerts SPDESI berechnet. Das Programm tritt nun in eine Schleife ein und wartet, bis der Zündzähler den Wert 0 erreicht, wie auf der oberen Linie in Fig. 13 gezeigt ist, wobei zu diesem

Zeitpunkt das Thyristorpaar gezündet und wieder ein Unterbrechungssignal an den Prozessor gegeben wird und der gerade beschriebene Prozess wiederholt wird.

Es wird nun auf Fig. 14 Bezug genommen, die ein Flussdiagramm hoher Stufe darstellt, welches den Gesamtbetrieb des erfindungsgemässen Regel- und Steuersystems in grösseren Einzelheiten als gerade in Verbindung mit Fig. 13 beschrieben, zeigt.

Wenn das System zuerst gestartet wird, wie im linken oberen Block der Fig. 14 dargestellt ist, liefert das Programm eine Pseudounterbrechung an das System, indem eine Zahl 16 in den Abwärtszähler 224 der Fig. 9 geladen wird. Zu diesem Zeitpunkt werden auch Nullwerte in das Thyristorwählregister 268 der Fig. 11B geladen. Der Abwärtszähler beginnt dann gegen den Wert 0 zu zählen, und wenn der Wert 0 erreicht ist, erzeugt das INT-Flipflop 250 der Fig. 9 ein Unterbrechungssignal auf der Leitung 210 an den Prozessor. Der Zweck dafür, dass Nullen in das Thyristorwählregister eingegeben werden, besteht darin, zu verhindern, dass irgendein Thyristorpaar zu diesem Zeitpunkt gezündet wird.

Der Prozessor tritt nun bei Empfang des Unterbrechungssignals in die Unterbrechungssubroutine ein. Das Programm tritt in einen ersten Lesentscheidungsblock ein, der bestimmt, ob dies die erste oder zweite Stromlesung vom Analog-/Digitalwandler 80 der Fig. 3 ist. Unter der Annahme, dass es sich um die erste Stromlesung handelt, geht das Programm durch einen Ja-Zweig in einen Block, in dem der erste Strom vom Analog-/Digitalwandler gelesen wird. Das Programm bestimmt ferner in diesem Block, ob sich das System im kontinuierlichen oder diskontinuierlichen Strombetrieb befindet, wobei es die Werte des ersten Stroms mit einer Konstanten vergleicht, die proportional zu einem vorgegebenen Strom ist. Das Programm läuft dann weiter und setzt den zuvor erwähnten Zündwinkel für eine 20°-Verzögerung. Das Programm läuft weiter und liest die Systemtaktbits ID1B0 bis ID1B7 auf den Leitungen 90 der Fig. 5 aus und berechnet den Wert von NEWTIM. Nach Abschluss der Berechnung von NEWTIM läuft das Programm weiter und berechnet TIMTGO, wobei diese Grösse zu diesem Zeitpunkt eine 20°-Verzögerung einschliesst. Das Programm tritt dann in eine Schleife ein und fährt fort, den Systemtakt zu lesen, bis NEWTIM gleich den 5 am wenigsten signifikanten Bits des durch den Wert 32 teilenden Zählers 170 der Fig. 5 ist, die als ID1B3 bis ID1B7 bezeichnet sind. Wenn diese beiden Werte gleich sind, lädt der Prozessor den Wert TIMTGO, der proportional zum Zündwinkel ist, in den Abwärtszähler und fährt fort, ein Kennzeichen (flag) für die zweite Lesung zu setzen.

Das Programm fährt dann fort, zu prüfen, ob eine neue Tachometerlesung im Tachoimpulszählerregister vorhanden ist. Wenn ein neuer Lesewert vorhanden ist, wird er ausgelesen und den Tachometerlesewerten hinzugefügt, die schon im Speicher (CACTI) akkumuliert sind. Das Programm überprüft dann, ob drei aufeinanderfolgende Ablesungen oder Lesewerte akkumuliert sind. Ist dies nicht der Fall, verzweigt sich das Programm nicht und läuft zurück in das Hauptprogramm, bis ein weiteres Unterbrechungssignal vom Prozessor erhalten wird (das heisst wenn TIMTGO gleich 0 ist). Zu diesem Zeitpunkt erfolgt ein neuer Eintritt in den ersten Leseentscheidungsblock, und bei diesem Eintritt tritt das Programm, da das flag-Kennzeichen für die zweite Lesung gesetzt ist, durch den Nein-Zweig des letzten Entscheidungsblocks aus und tritt in einen Block ein, in dem der Prozessor den zweiten Strom aus dem Analog-/Digitalwandler ausliest.

Nachdem der zweite Strom gelesen ist, führt das Programm nun Reglerberechnungen aus, um die Grössen FINVAL und TIMTGO zu berechnen. Beim Abschluss dieser Berechnungen schreibt der Prozessor dann die Thyristorpaaradresse in das Thyristorwählregister 268 der Fig. 11B. Zu diesem Zeitpunkt tritt der Prozessor wieder in eine Schleife ein, um mit dem Lesen des Systemtakts fortzufahren, bis die Werte von NEWTIM und ID1B3 bis ID1B7 gleich sind. Wenn diese Werte gleich sind, wird dem Prozessor mitgeteilt, wenn TIMTGO in den Abwärtszähler geladen werden soll, was zu dieser Zeit erfolgt. Der Prozessor fährt dann fort, die Thyristorpaaradresse im zuvor erwähnten PH-Zähler zu aktualisieren und setzt dann ein flag-Kennzeichen für die erste Stromlesung, so dass beim nächsten Durchgang durch das Programm eine erste Auslesung vorgenommen wird. Das Programm kehrt dann zurück und liest erneut den Tachometer ab, um festzustellen, ob ein Lesewert verfügbar ist, und es prüft dann, ob drei aufeinanderfolgende Tachometerlesewerte akkumuliert sind. Wenn 3 Lesewerte noch nicht akkumuliert sind und ein Geschwindigkeitsregler Anfrage-flag-Kennzeichen (SPDFLG) nicht gesetzt ist, läuft das Programm durch die gerade beschriebene Schleife weiter und läuft zurück durch den ersten Leseentscheidungsblock, aus dem Ja-Zweig heraus und fährt mit den Stromreglerberechnungen fort, wie dies beim zweiten Auslesen gerade beschrieben wurde. Wenn nach der zuvor erwähnten Prüfung bezüglich eines neuen Tachometerlesewerts drei aufeinanderfolgende Lesewerte verfügbar sind, werden neue Werte für die Motorgeschwindigkeit (CACT), stetig gemachte Motorbeschleunigung (TACSMD) und Gegenelektromotorische Kraft (CEMF) berechnet. Das Geschwindigkeitsregler-flag-Kennzeichen (SPDFLG) wird auf Null gesetzt, um zu bewirken, dass eine Geschwindigkeitsreglerberechnung durchgeführt wird. Bei Abschluss dieser Berechnungen wird ein flag-Kennzeichen für eine zweite Stromlesung gesetzt, wodurch angezeigt wird, dass die erste Lesung gerade gemacht wurde, und das Programm zweigt zum Hauptprogramm zurück, wobei ein Unterbrechungssignal vom INT-Zündlogik-Flipflop 250 der Fig. 9 abgegeben wird, wie zuvor beschrieben wurde. Wenn jedoch kein flag-Kennzeichen für die zweite Auslesung gesetzt wird, wird ein Ja-Zweig zu einem Block genommen, um die Zeit zur Durchführung der Geschwindigkeitsreglerberechnung zu prüfen, die Grösse SPDFLG wird um 1 erhöht und dann bezüglich des Werts 2 geprüft. Wenn dieser Test durchgeht, wird mit der Geschwindigkeitsreglerberechnung begonnen. Wenn nicht, wird, wie schon zuvor, wieder in das Hauptprogramm eingetreten. Diese Prozedur stellt sicher, dass Regler- und Verstetigungsberechnungen zwischen dem Zünden der Thyristoren nicht im gleichen Intervall durchgeführt werden. Dies wurde getan, um eine Überlastung des Rechners zu verhindern. Bei Abschluss der Geschwindigkeitsreglerberechnungen läuft das Programm bei schwebendem Empfang des Unterbrechungssignals vom Zündzähler zurück in das Hauptprogramm.

Auf dem breiten Hintergrund der Beschreibung des Systembetriebs in Verbindung mit den Fig. 13 und 14 wird nun auf die Fig. 15 bis 24 Bezug genommen, die in Form detaillierter Flussdiagramme die Ausführung des Stromreglerprogramms zur Steuerung und Regelung des erfindungsgemässen Regler- und Steuersystems zeigen. Es wird zuerst auf Fig. 15 Bezug genommen, die ein Flussdiagramm darstellt, welches das erfindungsgemässe Hauptprogramm wiedergibt. Nicht enthalten in Fig. 15 ist eine Standardauslöseroutine, die normalerweise jedes Programm durchläuft, um alle verschiedenen Register und Speicherplätze im Speicher für einen Programmlauf vorzubereiten. Da diese Art der Auslösung bekannt ist, ist sie in Fig. 15 nicht dargestellt, vielmehr wird angenommen, dass das Programm an einer Einsprungstelle beginnt, die BEGIN bezeichnet ist. Wenn das System zuerst startet, wobei der Start an der Einsprungstelle BEGIN erfolgt, liest der Prozessor zuerst die Einrichtung 3, den Tachoimpulszähler 88, wie in den Fig. 4 und 7 dargestellt ist. Die durch den Rechner gelesenen Bits sind die Bits ID3B0 bis ID3B7 auf den Leitungen 22. Diese Bits werden durch den 8-Bit-Multiplexer 128 der Fig. 4 in Abhängigkeit von einer Leseadresse, die durch die Bits IR5 bis

IR7 bezeichnet ist, und einen READ-P-Impuls auf der Auslöseleitung ausgelesen, die zum Multiplexer 128 führt.

Der Prozessor prüft dann in einem Entscheidungsblock TACH COUNT = 0, um zu bestimmen, ob der Motor sich dreht. Sofern der Tachozähler-(tach count-)Lesewert (ID3B0-ID3B7) nicht 0 ist, zeigt der Prozessor an, dass CEMF nicht 0 ist und der Motor sich dreht, und das Programm nimmt eine Nein-Verzweigung von diesem Entscheidungsblock und läuft in der Schleife zurück zu BEGIN, bis CEMF oder TACH COUNT 0 ist. Wenn TACH COUNT 0 ist, tritt das Programm durch eine Ja-Verzweigung in den nächsten Aktionsblock, in welchem der Prozessor die Einrichtung 0 (18' der Fig. 4) liest. ID4B0 ist das zu diesem Zeitpunkt vom Prozessor gelesene Bit, um im Ein-/Ausschalter zu lesen und um festzustellen, ob der Motor eingeschaltet wurde. Zusätzlich sendet der Prozessor einen «Leseeinrichtung-0»-Befehl an die Prozessor-/Systemschnittstelle, wobei das RDVO-Signal auf der Leitung 130 zum INT-Flipflop 250 erzeugt wird, wodurch dieses Flipflop zurückgesetzt wird. Das INT-Flipflop 250 befindet sich nun in einem Zustand, in dem es ein Unterbrechungssignal zur richtigen Zeit während des Betriebs des Systems erzeugt.

Das Programm läuft nun in einen Entscheidungsblock ON/OFF SCHALTER EIN. Wenn, in diesem Entscheidungsblock, der ON/OFF-Schalter, der gerade von der Einrichtung 0 gelesen wurde, nicht im eingeschalteten Zustand liegt, nimmt das Programm eine Nein-Verzweigung zurück zum Beginn des Programms und läuft im Programm so lange um, bis der ON/OFF-Schalter eingeschaltet ist. Es wird nun angenommen, dass der ON/OFF-Schalter eingeschaltet sei, wobei dann das Programm durch eine Ja-Verzweigung austritt und in einen Aktionsblock eintritt, in dem der Prozessor einen «Schreibe-Einrichtung-1»-Befehl zusammen mit Datenbits WDB0-WDB7 zur Prozessor-/Systemschnittstelle der Fig. 4 überträgt, um die Erzeugung des WDV1-Signals auf der Leitung 212 zu veranlassen, das zur Zündlogik der Fig. 9 gesendet werden soll, und um den Zählwert 16 in die Schreibdatenverriegelung und in den Abwärtszähler 224 in der zuvor beschriebenen Weise zu laden. Der Zweck, den Zählwert 16 in den Abwärtszähler 224 zu laden, besteht darin, eine Pseudounterbrechung an den Prozessor zu liefern, so dass der Prozessor mit der Ausführung des Hauptprogramms und allen anschliessenden Subroutinen beginnen kann, in die ein Eintritt vom Hauptprogramm aus erfolgt.

Zu diesem Zeitpunkt beginnt der Abwärtszähler mit der Abwärtszählung, während das Programm umgehend an eine START-Einsprungstelle weiterläuft, vergleiche Fig. 15. Der Prozessor sendet einen «Lies-Einrichtung-6»-Befehl zur Prozessor-/Systemschnittstelle, um das Ablesen des Drehzahlreferenz-Änderungsschalters durchzuführen, der durch Bit ID6B0 auf den Leitungen 66 gezeichnet ist, vergleiche Fig. 4. Der Zustand des Bits ID6B0 wird nun vom Prozessor abgefragt, um festzustellen, ob der Drehzahländerungsschalter sich im Ein-Zustand befindet. Der Drehzahländerungsschalter ist als ein vom Benutzer gesteuerter Schalter auf einer Konsole ausgebildet, der nicht dargestellt ist, und bildet einen Teil der Drehzahlreferenzschalter 18 (Eingangseinrichtungen 6 und 7), die von einem Benutzer betätigt werden, wenn der Benutzer wünscht, die Drehzahlreferenz-Eingabe an den Datenprozessor zu ändern, um die Drehzahl des Motors zu ändern. Solange dieser Schalter im Ein-Zustand liegt, fährt das Programm fort und tritt durch den Ja-Zweig des Entscheidungsblocks CHANGE SPEED SW ON aus und kehrt im Programm zurück zum START-Punkt.

Es werde nun angenommen, dass der Drehzahländerungsschalter nicht im Ein-Zustand liegt, wobei dann das Programm durch eine Nein-Verzweigung in einen Aktionsblock eintritt, in dem der Prozessor Befehle zur Prozessor-/Systemschnittstelle sendet, um die Einrichtungen 6 und 7 über Leitungen 66 in den Prozessor einzulesen. In diesem Augenblick werden die zuvor erwähnten Drehzahlreferenzschalter, die einen Drehzahlsollwert des Motors in Umdrehungen pro Minute kennzeichnen (Bits ID6B3 bis ID6B7 und ID7B0 bis ID7B7) in einem Speicherplatz im Prozessorprogramm 62 gespeichert, der als CHALF bezeichnet ist und den Speicherplatz für den Drehzahlsollwert darstellt.

Das Programm fährt nun fort, das Vorzeichen des Platzes CHALF gemäss der Einstellung des FWD/REV-Schalters zu setzen, wobei zuerst ein «Leseeinrichtung-0»-Signal zur Systemschnittstelle gesendet wird und Bit ID0B5 aus der Einrichtung 0 ausgelesen wird. Wenn ID0B5 angibt, dass der Motor in Vorwärtsrichtung laufen soll, wird der Platz CHALF nicht verändert, wenn jedoch ID0B5 angibt, dass die Richtung des Motors in Rückwärtsrichtung laufen soll, dann wird das binäre Komplement (2s-Komplement) von CHALF genommen und folglich für CHALF substituiert.

Das Programm fährt nun fort, festzustellen, ob der ON/OFF-Schalter sich in der OFF-Stellung befindet. Wenn der Schalter in der OFF-Stellung liegt, tritt das Programm durch eine Ja-Verzweigung aus und geht zurück zu BEGINN, wodurch die gerade beschriebenen Operationen wiederholt werden. Wird jedoch angnommen, dass der ON/OFF-Schalter nicht in der OFF-Stellung liegt, tritt das Programm aus dem letzten Entscheidungsblock durch eine Nein-Verzweigung aus und kehrt zur START-Einsprungstelle zurück, wie in Fig. 15 dargestellt ist. Das Programm setzt nun den Durchlauf durch die Schleife vom START-Punkt nach unten bis zum ON/OFF-Schalter AUS (OFF)-Entscheidungsblock fort, bis ein Unterbrechungssignal durch den Datenprozessor vom INT-Flipflop 250 in der Zündlogik 9 erhalten wird.

Wie schon in Verbindung mit dem Zündbetrieb der Logik beschrieben wurde, wird das INT-Flipflop 250 gesetzt, um ein INT-Signal auf der Leitung 210 zum Prozessor zu schicken, wenn der Abwärtszähler einen Zählwert 0 erreicht. Es ist wichtig, darauf hinzuweisen, dass das Unterbrechungssignal von der Zündlogik zu jeder Zeit während der Ausführung dieser letztgenannten Schleife (das heisst zwischen der START-Einsprungstelle und dem ON/OFF-SCHALTER-AUS-Entscheidungsblock) auftreten kann. Wenn die Unterbrechung auftritt, springt der Prozessor vom Hauptprogramm der Fig. 15 in eine Startstelle INTPT der Fig. 16, den Beginn des Unterbrechungsprogramms. Wie noch ersichtlich wird, kehrt bei Beendigung des Unterbrechungsprogramms, wenn alle Berechnungen abgeschlossen sind, zurück zum Hauptprogramm der Fig. 15, zu demjenigen Punkt, an dem die Unterbrechung auftrat.

Es werde nun angenommen, dass der Prozessor das Unterbrechungssignal auf der Leitung 210 erzeugt hat, wodurch das Programm in die Stelle START INTPT der Fig. 16 eintritt. Die erste Operation durch den Prozessor besteht darin, die Stromwerte der verschiedenen Prozessorregister, insbesondere des zuvor in Verbindung mit Fig. 2 beschriebenen Zwischenspeichers, zu speichern. Dies stellt eine Standardprozedur bei allen Betriebsprogrammen dar, wenn von einer Subroutine oder Programm zu einer anderen gesprungen wird, so dass diese Werte später wieder gespeichert werden können, wenn eine Rückkehr zum Programm erfolgt, von dem der Sprung aus gemacht wurde.

Dann fährt der Prozessor fort, einen «Lies-Einrichtung-0»-Befehl zur Prozessorschnittstelle der Fig. 4 zu senden, um das Bit ID0B0 des ON/OFF-Schalters auszulesen und gleichzeitig das Unterbrechungs-Flipflop zurückzusetzen oder zu löschen, wobei das RDVO-Signal vom Dekoder 126 der Fig. 4 zur Zündlogik in der zuvor beschriebenen Weise gesendet wird. Der ON/OFF-Schalter wird nun geprüft, um festzustellen, ob er im AUS- oder OFF-Zustand ist. Sofern der Schalter im OFF-Zustand liegt, wodurch angezeigt wird, dass Leistung vom Motor abgenommen werden soll, tritt das Programm durch einen Ja-Zweig aus, und die zuvor gespeicherten Register wer-

den nun wieder auf ihre ursprünglichen Werte zurückgespeichert, und das Programm kehrt zur Fig. 15 zurück, in der die Operationen ablaufen, wie schon beschrieben wurde. Es werde jedoch angenommen, dass der ON/OFF-Schalter nicht im OFF(AUS)-Zustand liegt, wobei dann das Programm durch eine Nein-Verzweigung in einen ersten CURRENT-READING-Entscheidungsblock (CURFLG = 0) eintritt. Im letztgenannten Entscheidungsblock wird ein Test ausgeführt, um festzustellen, ob dies der erste Stromlesewert ist. Der Test wird hier mit variablem Kennzeichen (variable flag) im Speicher vorgenommen, das CURFLG für das erste Stromlesewert-Kennzeichen bezeichnet wird. Wenn D'CURFLG gleich 0 ist, so zeigt dies an, dass dies der erste Stromlesewert ist, wenn CURFLG eine binäre 1 ist, so zeigt dies an, dass es der zweite Stromlesewert ist.

Unter der Annahme, dass zurzeit CURFLG gleich 0 ist, tritt das Programm durch eine Ja-Verzweigung aus und tritt in einen Aktionsblock ein, in welchem der Prozessor einen «Lies-Einrichtung-5»-Befehl zur Prozessor-/Systemschnittstelle sendet, der anweist, dass der Analog-/Digitalwandler 80 die Bits ID5B0 bis ID7B7 durch den 8-Bit-Multiplexer in den Prozessor auf den Eingabedatenleitungen ID0 bis ID7 einliest. Der durch die Bits ID5B0 bis ID5B7 spezifizierte Wert wird in einem CRNT bezeichneten Platz im Speicher gespeichert, wobei dieser Platz ein Speicherplatz für den gemessenen Motorstrom ist. Das Programm läuft dann zu einem Entscheidungsblock weiter, in dem ein im Speicher gespeicherter konstanter Wert CURTOL mit dem Absolutwert von CRNT verglichen wird. Der Wert von CURTOL ist einem Wert von 1 bis 2 Prozent des Motornennstroms proportional und wird benutzt, um eine Prüfung bezüglich des diskontinuierlichen Strombetriebs vorzunehmen. Sofern CURTOL kleiner als CRNT ist, tritt das Programm durch eine Ja-Verzweigung aus und geht in den diskontinuierlichen Betrieb, während dann, wenn CURTOL grösser als CRNT ist, der kontinuierliche Betrieb vorliegt, und das Programm durch die Nein-Verzweigung austritt.

Es wird zuerst angenommen, dass der Motor im diskontinuierlichen Betrieb arbeitet. Wenn also durch den Ja-Zweig ausgetreten wird, setzt der Prozessor ein Betriebsartkennzeichen MODFLG im Speicher auf einen Wert 1, wodurch angezeigt wird, dass das System nun im diskontinuierlichen Strombetrieb arbeitet. Im Speicher sind vier Konstanten gespeichert, die G1 und G2 bezeichnet sind. Es sind zwei Konstanten G1 und zwei Konstanten G2 vorhanden, wobei ein Paar verwendet wird, wenn das System im diskontinuierlichen Betrieb arbeitet, und das andere Paar der Grössen G1 und G2 verwendet wird, wenn das System im kontinuierlichen Betrieb arbeitet. Diese Konstanten, die für kontinuierlichen und diskontinuierlichen Strombetrieb benutzt werden, sind Verstärkungskonstanten, die gewählt sind, um die von der Motorantriebsschleife geforderte Gesamtverstärkung zu schaffen, wenn der Motor in dem einen oder dem anderen Betrieb arbeitet. So wählt zum Beispiel im diskontinuierlichen Betrieb das Programm die richtigen G1 und G2 aus, die Verstärkungen von 32 bzw. 0 besitzen. Im letzteren Aktionsblock werden auch negative und positive obere und untere Grenzwerte (VRLIMN und VRLIMP) vom Speicher ausgelesen und in die Unterbrechungssubroutine gebracht, um beim Aufbau von oberen und unteren Grenzwerten für die Motorspannung verwendet zu werden, die durch den Stromregler berechnet werden.

Bei Abschluss dieser letzten Operationen tritt das Programm in die Anschlussstelle B der Fig. 17 ein. Unter Bezugnahme auf Fig. 17 sei bemerkt, dass die Anschlussstelle A von der Fig. 16 ebenfalls in die Fig. 17 läuft. Wie schon beschrieben, erfolgt ein Einsprung an der Anschlussstelle A in die Fig. 17, sofern das System im kontinuierlichen Betrieb arbeitet. Beim Einsprung in die Anschlussstelle A sind die im ersten Aktionsblock stattfindenden Operationen dieselben wie die im letzten Aktionsblock der Fig. 16 beschriebenen Aktionen, mit der Ausnahme, dass MODFLG für den kontinuierlichen Betrieb gleich 0 gesetzt ist. Das Programm wählt auch die richtigen G1 und G2 für den kontinuierlichen Strombetrieb (ein Beispiel für diese Verstärkungswerte ist G1=15 und G2=11).

Beim Einsprung in die Anschlussstelle B der Fig. 17 setzt der Prozessor den Zündwinkel derart, dass eine Unterbrechung von 20° nach dem Zünden des letzten Thyristorpaars bewirkt wird. Dies wird dadurch verwirklicht, dass der Zündwinkel FINVAL im Speicher gleich FINVAL minus einer Zahl gesetzt wird, die proportional zu 40° ist. Wenn 40° von der Grösse FINVAL subtrahiert werden, wird eine Unterbrechung zur richtigen Zeit für den zweiten Stromlesewert bewirkt. Sofern TIMTGO unter Benutzung des alten Werts von FINVAL berechnet wäre, würde das Thyristorpaar 60° später gezündet werden. Durch Subtraktion von 40° vom Wert FINVAL wird der Abwärtszählerwert derart gesetzt, dass er eine Unterbrechung 20° nach der Zündung des letzten Thyristorpaars erzeugt. Das Programm läuft nun in einen Aktionsblock weiter, in welchem ein Speicherplatz DESI, der den gewünschten Stromsollwert kennzeichnet, gleich sich selbst plus einem berechneten Wert SPDESI gesetzt wird, der eine gewünschte Änderungsgeschwindigkeit des Stromsollwerts angibt.

Das Programm läuft nun in eine Anschlussstelle E der Fig. 20 und tritt in einen Aktionsblock ein, in dem der Prozessor nun einen «Lies-Einrichtung-1»-Befehl zur Prozessorschnittstelle schickt, um die Systemtaktbits ID1B0 bis ID1B7 auf den Leitungen 190 zu lesen, wie in Fig. 4 und 5 dargestellt ist. Im nächsten Aktionsblock wird das 60°-Intervall, das durch die Bits ID1B0 bis ID1B7 gekennzeichnet ist, im Platz KOCT gespeichert (vergleiche Tabelle 2), und die Zeit innerhalb des Intervalls, durch die Bits ID1B3 bis ID1B7 gekennzeichnet, wird in einem als TCLOCK bezeichneten Speicherplatz gespeichert.

Der Prozessor fährt nun fort, den Wert von NEWTIM zu berechnen und setzt diesen Speicherplatz im Speicher gleich TCLOCK plus 2, was die Verzögerungszeit darstellt, die zuvor zur Berechnung bei der Ableitung der TIMTGO-Gleichung beschrieben wurde. Zu diesem Zeitpunkt wird ferner die lange Taktzählkorrektor CORR gleich 0 gesetzt. Das Programm tritt nun in einen NEWTIM > 30 Entscheidungsblock ein. Sofern NEWTIM grösser als 30 ist, tritt das Programm durch eine Ja-Verzweigung aus und setzt das CORR-Bit auf den Wert 1. Das Programm läuft dann in einen weiteren Entscheidungsblock NEWTIM > 31. Wenn NEWTIM 32 oder grösser ist, tritt das Programm durch eine Ja-Verzweigung in einen Aktionsblock oben im rechten Teil der Fig. 20 ein, in welchem NEWTIM entweder auf den Wert 0 oder 1 gesetzt wird, indem NEWTIM = NEWTIM-32 gesetzt wird. Sofern NEWTIM den Wert 32 besitzt, wird diese Grösse auf den Wert Null gesetzt, während NEWTIM auf den Wert 1 gesetzt wird, wenn diese Grösse gleich 33 (das heisst TCLOCK = 31 + 2 = 33) ist.

Es wird nun wiederum auf die Entscheidungsblöcke NEWTIM > 30 und NEWTIM > 31 der Fig. 20 Bezug genommen. Wenn eine dieser Entscheidungen negativ ist, tritt das Programm durch eine Nein-Verzweigung des betreffenden Entscheidungsblocks aus und tritt in einen Aktionsblock ein, in welchem die Nulldurchgangszahl PHA im Speicher benutzt wird, um TIMTGO zu berechnen, wobei der Wert von KOCT als eine Adresse zur PH-Tabelle (TABPH) verwendet wird, in dem PHA gleich TABPH gesetzt wird (vergleiche Tabelle 2). Der Prozessor fährt nun fort, TIMTGO dadurch zu berechnen, dass TIMTGO gleich FINVAL (dem Zündwinkel), minus TABTP (die Offsetkorrektur der Tabelle 2, adressiert durch die Differenz zwischen PHA und PH) minus 8×TCLOCK (das gerade gelesene Zeitintervall) minus dem Wert von CORR gesetzt wird. CORR ist zu dieser Zeit entweder 0 oder 1, je nachdem ob NEWTIM grösser oder kleiner 31 war. Der Prozessor tritt

dann in einen Entscheidungsblock CURFLG = 0 ein, in dem wiederum ein Test durchgeführt wird, um festzustellen, ob dies der erste Stromlesewert ist. Sofern CURFLG nicht gleich 0 ist, was anzeigt, dass es der zweite Stromlesewert ist, dann erfolgt ein Einsprung in Fig. 21 an der Anschlussstelle F, wobei in einen Aktionsblock eingetreten wird, in welchem der Prozessor die Thyristorpaar- und Brückenadresse an die Einrichtung 3, das Thyristorwähl- und Antriebsrichtungsregister 268 der Fig. 11B, schreibt, in dem ein WDV3-Befehl über die Leitungen 270 ausgesendet wird und die Thyristorpaar- und Brückenadresse über die Leitungen 266 als Schreibdatenbits WDB0 bis WDB7 vom Treiber 138 der Prozessor-/Systemschnittstelle 64 gesendet wird. Die Thyristorpaar- und Brückenadressen kommen von einer als OCTF bezeichneten Tabelle im Speicher, die 12 separate Adresseinsprungstellen, 6 für die Vorwärtsthyristorbrücke und 6 für die Rückwärtsthyristorbrücke, enthält. Die Plätze in der Tabelle OCTF werden durch den Inhalt des PH-Zählers adressiert, der das zu zündende Thyristorpaar und das Richtungskennzeichen DIRFLG spezifiziert, welches ein flag-Kennzeichen im Speicher ist, das angibt, ob die Vorwärts- oder die Rückwärtsbrücke zu zünden ist.

Es sei daran erinnert, dass die Zündung der Thyristoren tatsächlich nach der Durchführung der Berechnung von TIMTGO (das heisst im Anschluss an das Lesen des zweiten Stroms) stattfindet, es ist daher notwendig, die Thyristorpaar- und Brückenadresse zu ändern, um die richtigen Thyristoren zu zünden. Handelt es sich dagegen um einen ersten Stromlesewert, so ist es nicht wünschenswert, die Thyristorpaar- und Brückenadresse zu ändern, da zu diesem Zeitpunkt kein Zünden erfolgt. Handelt es sich also nicht um den ersten Stromlesewert, so erfolgt ein Einsprung in Fig. 21 am Punkt G aus Fig. 20, und die Aktualisierung der Thyristorpaar- und Brückenadresse wird umgangen.

Das Programm läuft nun in einen Entscheidungsblock TIMTGO < 16. Wenn TIMTGO kleiner als 16 ist, tritt das Programm durch eine Ja-Verzweigung aus und in einen Aktionsblock ein, in welchen der Prozessor die Zahl 16 zur Einrichtung 1, den Abwärtszähler, schreibt, in dem ein WDV1-Signal von der Prozessor-/Systemschnittstelle zusammen mit der Zahl 16 auf den Schreibdaten-Sammelleitungen WDB0 bis WDB7 erzeugt wird, wie zuvor erläutert wurde. Der Grund für den Test TIMTGO > besteht darin, dass ein minimaler Grenzwert für den Wert von TIMTGO gesetzt ist, um sicher zu stellen, dass ständig mindestens 4° Verzögerung vor der Erzeugung eines Unterbrechungssignals an den Datenprozessor vorhanden ist, so dass ein Umwandlungsbefehl zum Analog-/Digitalwandler 80 gesendet wird, um die Durchführung einer neuen Umwandlung zu veranlassen.

Sofern in Fig. 21 TIMTGO nicht kleiner als 16 ist, zweigt das Programm durch eine Nein-Verzweigung ab und tritt in einen Aktionsblock READ DEVICE 1 ein, in welchem der Prozessor wieder die Systemtaktbits ID1B0 bis ID1B7 liest. Das Programm tritt dann über einen Entscheidungsblock ID1B3 bis ID1B7 = NEWTIM in eine Schleife ein, die durch eine Nein-Verzweigung zurück zum Aktionsblock READ DEVICE 1 führt, und das Programm läuft so lange in dieser Schleife um, bis der Systemtakt gleich NEWTIM ist. Wenn diese beiden Werte gleich sind, ist es Zeit, den Abwärtszähler zu beladen, und das Programm nimmt eine Ja-Verzweigung und tritt in einen Aktionsblock ein, in dem der Prozessor nun TIMTGO in den Abwärtszähler 224 der Fig. 9 schreibt.

Wie schon beschrieben, startet der Abwärtszähler, um TIMTGO abwärts gegen 0 zu zählen, wenn das nächste Taktsignal von 88 Mikrosekunden Dauer (vergleiche Fig. 9 und 10) auftritt, das dem Laden des Abwärtszählers folgt. Wenn der Abwärtszähler den Wert 0 erreicht, erzeugt der Prozessor wieder ein Unterbrechungssignal auf der Leitung 210 der Fig. 9 und erzeugt auf diese Weise eine weitere Unterbrechung für den Prozessor, wie schon beschrieben wurde. Unmittelbar nach der Übertragung der Grösse TIMTGO in den Abwärtszähler geht der Prozessor von der Anschlussstelle H in Fig. 21 zur Anschlussstelle H in Fig. 22 und tritt in einen Entscheidungsblock CURFLG = 0 ein.

In diesem Entscheidungsblock wird dann ein Test durchgeführt, um festzustellen, ob dies der erste Stromlesewert ist. Sofern es der erste Lesewert ist, tritt der Prozessor durch eine Nein-Verzweigung aus und in einen Aktionsblock ein, in dem der Platz CURFLG, das Stromlesekennzeichen gleich dem Wert 1 gesetzt ist, um anzugeben, dass der zweite Lesewert im nächsten Durchgang durch das Programm an der Reihe ist. Handelt es sich um den ersten Lesewert, so tritt das Programm aus dem Entscheidungsblock CURFLG = 0 durch eine Ja-Verzweigung aus und tritt in einen Aktionsblock ein, in dem der Reihenfolgezähler oder Sequenzzähler PH gleich PH + 1 gesetzt wird, wodurch die Thyristorpaaradresse erhöht wird, so dass das richtige Thyristorpaar bei der nächsten Berechnung von TIMTGO gezündet wird.

Der Prozessor tritt nun in einen Entscheidungsblock PH > 6 ein. Sofern PH grösser als 6 ist, tritt das Programm durch eine Ja-Verzweigung aus und in einen Aktionsblock ein, in dem PH nun in Vorbereitung der Zündung des Thyristorzellenpaars, welches der PH-Adresse 1 entspricht, gleich dem Wert 1 gesetzt wird. Sofern andererseits PH nicht grösser als 6 ist, wird PH nicht verändert, und das Programm tritt durch eine Nein-Verzweigung aus und in einen Aktionsblock ein, in dem der Platz CURFLG gleich 0 gesetzt wird, zur Vorbereitung auf den ersten Stromlesewert beim nächsten Durchgang durch das Programm. Die Unterbrechungssubroutine ruft nun eine Subroutine, die RDTACH bezeichnet ist, vergleiche Fig. 24.

Es wird nun auf Fig. 24 Bezug genommen, in der der Prozessor in eine Einsprungstelle START RDTACH zur READ-Tachometerroutine eintritt. In der Subroutine RDTACH liest der Prozessor zuerst die Eingangseinrichtung 1, den Systemtakt, in dem die ersten signifikantesten Bits (ID1B0–ID1B2) dieses Takts gelesen werden. Es sei daran erinnert, dass diese Bits ein 60°-Intervall der Eingangsspannung definieren, wenn die Auslesung durch den Prozessor vorgenommen wird. Der Prozessor tritt nun in einen Entscheidungsblock PHOCT = ID1B0 bis ID1B2 ein. In diesem Entscheidungsblock wird ein Test durchgeführt, um festzustellen, ob ein Phasennulldurchgang seit dem letzten Durchlauf durch die Subroutine stattgefunden hat. Dies wird durch einen Vergleich der drei signifikantesten Bits des 360°-Systemtakts (ID1B0–ID1B2) mit dem Platz PHOCT verwirklicht, der den Lesewert oder Wert des 60°-Intervalls vom vorausgegangenen Durchgang durch die Subroutine enthält. Eine Änderung in ID1B0 bis ID1B2 bedeutet, dass ein Nulldurchgang aufgetreten ist, und dass ein neuer Wert in PHOCT gespeichert werden soll, um diesen Speicherplatz für nachfolgende Tests zu aktualisieren. Dies wird in einem Aktionsblock durchgeführt, in den ein Eintritt von einer Nein-Verzweigung des Entscheidungsblocks PHOCT ist gleich ID1B0 bis ID1B2 erfolgt, in dem PHOCT gleich ID1B0 bis ID1B2 gesetzt wird. Hat andererseits keine Änderung bei den Nulldurchgängen stattgefunden, dann tritt das Programm durch eine Ja-Verzweigung aus und kehrt von dem Punkt zu Fig. 22 zurück, den es verlassen hatte, um in einen Entscheidungsblock CURFLG = 1 einzutreten.

Es wird nun wieder auf Fig. 24 Bezug genommen, und es wird angenommen, dass eine Änderung in den Nulldurchgängen erfolgte, wodurch der Wert von ID1B0 bis ID1B2 geändert ist. Als Ergebnis tritt das Programm aus dem Entscheidungsblock PHOCT = ID1B2 aus, wobei PHOCT in der zuvor angegebenen Weise gesetzt ist, und tritt in einen Aktionsblock ein, in welchem der Prozessor nun die Einrichtung 3, den Tachoimpulszähler, liest, in dem er die Bits ID3B0 bis ID3B7 und ID0B4 in den Prozessor liest. Unter Bezugnahme auf Fig. 7 sei

daran erinnert, dass das Bit ID0B4 als dasjenige Bit identifiziert war, welches die Laufrichtung des Motors angibt. In diesem Aktionsblock wird der Wert des Tachoimpulszählers in den Prozessor gelesen, und das Vorzeichen dieses Werts wird gemäss dem Zustand von ID0B4 gesetzt. Der Wert des Tachoimpulszählers stellt als solcher entweder eine positive oder negative Zahl dar, die angibt, dass der Motor entweder in Vorwärts- oder in Rückwärtsrichtung läuft. Im vorliegenden System bewirkt die Adressierung der Eingabeeinrichtung 3 durch den 8-Bit-Multiplexer der Fig. 4 auch, dass das Bit ID0B4 durch diesen Multiplexer gelesen und gleichzeitig mit den Bits ID3B0 bis ID3B7 in den Prozessor eingegeben wird.

In Fig. 24 tritt der Prozessor nun in einen Aktionsblock ein, in dem der Tachometerlesewert der Summe der zuvor abgegriffenen Lesewerte hinzuaddiert wird, in dem ID3B0 bis ID3B7 einem Platz CACTI hinzuaddiert wird, der als Tachozählerakkumulator im Speicher festgelegt ist. Es lässt sich daher erkennen, dass bei jedem Durchgang durch die RDTACH-Subroutine die Tacholesewerte vom Tachoimpulszähler 88 als eine Summe im Speicherplatz CACTI akkumuliert werden. Der Prozessor schreitet dann in den nächsten Aktionsblock, in dem eine Zahl des Lesewertzählers CKNT im Speicher aktualisiert oder um den Wert 1 erhöht wird, wobei CKNT = CKNT + 1 gesetzt wird. Der Zweck des Zählers CKNT besteht darin, die Zahl der akkumulierten Lesewerte in CACTI zu verfolgen. Dies wird durch einen Entscheidungsblock CKNT = 3 im rechten oberen Teil der Fig. 24 angezeigt, der einen Test durchführt, um festzustellen, ob 3 Lesewerte akkumuliert sind. Sofern CKNT nicht gleich 3 ist, werden die Drehzahlstetigkeitsberechnungen nicht durchgeführt, das Programm tritt daher durch eine Nein-Verzweigung aus und kehrt zur Unterbrechungssubroutine zurück, die es verlassen hat, um in den Entscheidungsblock CURFLG = 1 der Fig. 22 einzutreten, wie erläutert wurde.

Es wird nun in Fig. 24 angenommen, dass drei Lesewerte akkumuliert sind, wobei dann das Programm durch eine Ja-Verzweigung austritt und in einen Entscheidungsblock eintritt, in dem eine nichtverstetigte Motordrehzahl berechnet wird. Dies wird dadurch bewirkt, dass der Platz TEMP im Speicher gleich der Summe der über die letzten zwei Durchgänge akkumulierten Tachoimpulse gesetzt wird. Dies ist die mittlere Motordrehzahl. Die Summe wird dadurch verwirklicht, dass TEMP = CACT gesetzt wird, wobei CACT einen Speicherplatz darstellt, der die alte Summe der Tachometerdrehzahllesewerte mit dem Speicherplatz CACTI speichert, welcher die Summe der neuen Tachometerlesewerte enthält. In diesem Aktionsblock wird der Speicherplatz CACT auch gleich CACTI gesetzt, so dass er die Summe der alten Lesewerte reflektiert oder wiedergibt. Ferner wird CACTI gleich dem Wert 0 gesetzt, so dass CACTI zur Akkumulierung der Summe der nächsten Lesewerte in nachfolgenden Durchgängen ausgelöst werden kann. Ferner wird ein Drehzahlkennzeichen SPDFLG auf eine binäre 0 gebracht. SPDFLG wird benutzt, wie noch erläutert wird, um dem Prozessor mitzuteilen, ob Drehzahlreglerberechnungen durchgeführt werden sollen oder die Drehzahlreglerberechnungen ausgelassen werden sollen. Wenn SPDFLG gleich Null ist, gibt dieses Kennzeichen dem Programm an, dass die Drehzahlreglerberechnungen ausgelassen werden sollen. Das Programm schreitet dann in den nächsten Aktionsblock der Fig. 24, in dem eine stetige Drehzahl berechnet wird. Dies wird dadurch bewirkt, dass ein Speicherplatz TACSMD = Platz TEMP-TACSUM gesetzt wird. Zusätzlich wird in diesem Aktionsblock ein Speicherplatz TACSUM gleich TACSUM + TACSMD gesetzt. Der Platz TACSUM enthält einen Wert, der proportional zu der stetigeren Drehzahl ist, und TACSMD ist die Drehzahlrate, die eine Ableitung von TACSUM ist. Das Programm fährt nun fort, die gegenelektromotorische Kraft CEMF zu berechnen, die nachfolgend bei der Berechnung der Motorklemmenspannung VT benutzt wird. CEMF wird dadurch berechnet, dass ein Platz CEMF im Speicher gleich KV, multipliziert mit dem Speicherplatz TEMP, gesetzt wird. KV ist eine Konstante, die im Speicher gespeichert ist und einen Wert besitzt, der aus der Formel KV = CEMF (VOLTS), dividiert durch Umdrehungen pro Minute, abgeleitet ist.

Wenn die Drehzahlberechnungen nun angeschlossen sind, kehr der Prozessor zur Unterbrechungssubroutine in Fig. 22 zurück und tritt in den Entscheidungsblock CURFLG = 1 ein. In diesem Entscheidungsblock führt das Programm keine Stromdrehzahl-Reglerberechnungen aus, wenn CURFLG gleich 1 ist, wodurch angezeigt wird, dass dies der zweite Stromlesewert ist. Das Programm nimmt dann eine Ja-Verzweigung, die zur Stelle J am oberen Ende der Fig. 22 eintritt, in der die Zwischenregister wieder hergestellt werden, wie beschrieben wurde, und das Programm zum Hauptprogramm an der Unterbrechungsstelle in Fig. 15 zurückkehrte. Es soll jedoch angenommen werden, dass CURFLG nicht gleich 1 ist, wodurch angezeigt wird, dass dies der erste Stromlesewert ist, und der Prozessor tritt dann in eine Anschlussstelle I der Fig. 23 ein, in der das Drehzahlkennzeichen SPDFLG gleich SPDFLG + 1 gesetzt wird. In einem Entscheidungsblock SPDFLG = 1 wird nun ein Test durchgeführt, um festzustellen, ob das Drehzahlkennzeichen gesetzt ist. Wenn das Drehzahlkennzeichen gesetzt ist, werden Drehzahlreglerberechnungen durch das Programm durchgeführt, welches durch eine Ja-Verzweigung dieses Entscheidungsblocks austritt und in einen Aktionsblock eintritt, um einen Drehzahlfehler zu berechnen.

Der Drehzahlfehler wird dadurch berechnet, dass ein Platz ERRACT im Speicher, der gleich dem Speicherplatz zur Speicherung des Drehzahlfehlers ist, gleich dem Inhalt des Platzes CHALF, dem Drehzahlsollwert, minus dem Inhalt des Platzes CACT, der alten Summe der Drehzahllesewerte oder der Drehzahl vor der Verstetigung, gesetzt wird. Beim Weiterschreiten durch das Programm, löst der Prozessor nun die Berechnung des Stromsollwerts aus, wobei gesetzt wird: Speicherplatz ERRACT = G3 × ERRACT −G4 × TACSMD. G3 und G4 sind Reglerverstärkungen, die gemäss dem speziellen Antriebsmotor gesetzt sind, um die gewünschte Drehzahlantwort abzugeben. In dieser Ausführungsform wurden Werte G3 = 1 und G4 = 16 verwendet.

Der Prozessor fährt nun fort, den Stromsollwert zu berechnen, wobei ein Wert TDESI = TDESI + ERRACT gesetzt wird. Das Programm läuft nun weiter in einen Entscheidungsblock TDESI > CURLMP am oberen Ende der Fig. 23. Im vorliegenden Antriebssystem wird ein maximaler Grenzwert für den Motorstrom festgelegt, es wird daher ein Test durchgeführt, um festzustellen, ob der Wert von TDESI, der berechnete Motorstrom, grösser oder kleiner als die festgelegten Stromgrenzwerte CURLMP und CURLMN sind. CURLMP ist der positive Stromgrenzwert, und CURLMN ist der negative Grenzwert, wie in einem Entscheidungsblock TDESI > CURLMN der Fig. 23 angegeben ist. Sofern TDESI grösser als CURLMP ist, tritt das Programm durch eine Ja-Verzweigung aus und tritt in einen Aktionsblock ein, in welchem TDESI gleich dem maximalen Stromgrenzwert CURLMP gesetzt wird. Ist andererseits CURLMP nicht grösser als TDESI, so tritt das Programm durch eine Nein-Verzweigung aus und in einen Entscheidungsblock TDESI < CURLM ein. Sofern dieser Test positiv ist, tritt das Programm durch eine Ja-Verzweigung aus, in der TDESI gleich CURLMN gesetzt wird. Sofern andererseits der Test negativ ist, tritt das Programm durch eine Nein-Verzweigung aus und in einen Aktionsblock ein, in welchem der Stromänderungssollwert nun berechnet wird. Der Stromänderungssollwert wird dadurch berechnet, dass ein Platz im Speicher, der SPDESI bezeichnet ist (Stromsollwertänderung) gleich TDESI (dem berechneten Stromsollwert) minus DESI (Stromsollwert) gesetzt wird und die Differenz

durch 3 geteilt wird. Der Divisor 3 wird verwendet, um die Durchschnittsbildung des Stromänderungssollwerts über 3 Durchgänge durch das Stromreglerberechnungsprogramm für jede Drehzahlreglerberechnung in Betracht zu ziehen.

Für den Stromänderungssollwert SPDESI wird ebenfalls ein Stromänderungsgrenzwert (current rate limit) festgelegt. Dies wird dadurch bewirkt, dass das Programm nun in einen Entscheidungsblock SPDESI > RTLMP eintritt, in dem ein Test durchgeführt wird, um festzustellen, ob SPDESI grösser als RATLMP ist, was eine positive Änderungsgrenze darstellt. Sofern es grösser ist, dann tritt das Programm durch eine Ja-Verzweigung in einen Aktionsblock SPDESI = RATLMP, der die maximale positive Änderungsgrenze für SPDESI festlegt. Ist andererseits SPDESI kleiner als RATLAMP, dann tritt das Programm durch eine Nein-Verzweigung aus und tritt in einen Entscheidungsblock SPDESI < RATLMN ein. In diesem Entscheidungsblock tritt das Programm, sofern SPDESI kleiner als RATLMN ist, durch eine Ja-Verzweigung aus, wodurch SPDESI = RATLMN gesetzt wird und eine minimale Änderungsgrenze hergestellt wird. Sofern SPDESI nicht kleiner als RATLMN ist, dann tritt das Programm durch eine Nein-Verzweigung aus und tritt in die Stelle J der Fig. 22 ein, in der die zuvor bewahrten Prozessorregister wieder hergestellt werden und das Programm an der Unterbrechungsstelle zurück zum Hauptprogramm zurückkehrt, wie zuvor beschrieben wurde.

Es wird weiter in Fig. 23 auf den Entscheidungsblock SPDFLG = 1 Bezug genommen. Sofern SPDFLG gleich 1 ist, zeigt dies an, dass die Drehzahlreglerberechnungen ausgelassen werden sollen, so dass das Programm von diesem Block durch eine Nein-Verzweigung austritt und zur Stelle J der Fig. 22 zurückkehrt, wie gerade beschrieben wurde.

Es wird nun auf Fig. 16 und zwar den Entscheidungsblock erster Stromlesewert CURFLG = 0 Bezug genommen, in dem ein Test durchgeführt wird, um festzustellen, ob das Programm den ersten oder den zweiten Stromlesewert nimmt. Wenn CURFLG nicht gleich Null ist, zeigt dies an, dass der erste Stromlesewert gerade genommen wurde, wie schon erläutert wurde, dass der zweite Stromlesewert genommen werden soll, und dass die Stromreglerberechnungen durchgeführt sind. Unter dieser Bedingung tritt der Prozessor nun durch eine Nein-Verzweigung an der Anschlussstelle C aus und tritt in Fig. 18 ein.

Die erste in Fig. 18 stattfindende Operation besteht darin, dass der Prozessor einen «Lies-Einrichtung-5»-Befehl sendet, um den Analog-/Digitalstromwandler 80 auszulesen und die Bits ID5B0 bis ID5B7 im Speicherplatz CRNT zu speichern, wobei dieser Speicherplatz zur Speicherung des tatsächlichen Motorstroms oder des Ist-Wert des Motorstroms dient. Der Prozessor tritt nun in einen Aktionsblock und berechnet den Stromfehler dadurch, dass der Speicherplatz IDIFF gleich dem Speicherplatz DESI, den Stromsollwert, minus CRNT, den tatsächlichen Motorstrom, gesetzt wird. In einem Entscheidungsblock IDIFF> +IDLIM wird ein Test durchgeführt, um festzustellen, ob der Stromfehler grösser als ein positiver Stromfehlergrenzwert ist, der durch die Konstante +IDLIM gegeben ist. Sofern IDIFF grösser als +IDLIM ist, zweigt das Programm durch einen Ja-Ausgang ab und tritt in einen Aktionsblock, in dem ein IDIFF gleich +IDLIM gesetzt wird. Sofern andererseits IDIFF nicht grösser als +IDLIM ist, tritt das Programm durch eine Nein-Verzweigung aus und tritt in einen Entscheidungsblock IDIFF-IDLIM ein. In diesem Block wird dieselbe Art von Entscheidung getroffen, um festzustellen, ob IDIFF kleiner als ein negativer oder minimaler Stromfehlergrenzwert ist. Ist dies der Fall, tritt das Programm durch eine Ja-Verzweigung aus, in der IDIFF gleich −IDLIM gesetzt wird. Ist andererseits IDIFF nicht grösser als −IDLIM, tritt das Programm durch eine Nein-Verzweigung aus und tritt in einen Aktionsblock ein, in dem die Motorklemmenspannung durch den Regler berechnet wird.

Die Motorklemmenspannung wird dadurch berechnet, dass ein Speicherplatz VR, der einen mittleren Wert bei der Berechnung darstellt, gleich (G1 × IDIFF) bis (G2 × IDIFF0) gesetzt wird. Die Verstärkungen G1 und G2 wurden oben festgelegt. Die Verstärkung 1 (G1) für den diskontinuierlichen Strombetrieb ist normalerweise das 2- bis 3fache des Werts für den kontinuierlichen Strombetrieb, und die Verstärkung 2 ist für den diskontinuierlichen Strombetrieb gleich 0. Der Term IDIFF0 stellt einen Speicherplatz im Speicher dar, der den alten Wert von IDIFF speichert. Das Programm schreitet dann zu einem Entscheidungsblock DIRFLG = 0 fort. In diesem Entscheidungsblock wird ein Test vorgenommen, um festzustellen, ob die Vorwärtsbrücke gezündet werden soll, wobei der Zustand der Grösse DIRFLG getestet wird, die ein Kennzeichen im Speicher darstellt, welches angibt, welche Brücke gezündet werden soll. Wenn DIRFLG nicht gleich Null ist, wodurch angezeigt wird, dass die Thyristoren der Rückwärtsbrücke gezündet werden sollen, erfolgt der Ausgang durch eine Nein-Verzweigung und VR wird gleich VR0−VR gesetzt, wobei VR0 von einem Platz im Speicher herkommt, der den alten Wert von VR speichert. Sofern DIRFLG gleich 0 ist, wodurch angezeigt wird, dass die Thyristoren der Vorwärtsbrücke gezündet werden sollen, tritt das Programm durch eine Ja-Verzweigung in einen Aktionsblock, in dem VR gleich VR0 + VR gesetzt wird.

Bei Vollendung der Berechnung von VR tritt das Programm in einen Entscheidungsblock VR > VRLIMP. Es gibt zwei Konstanten (VRLIMP und VRLIMN), die im Speicher gespeichert sind und positive und negative Grenzwerte für das Maximum und das Minimum der berechneten Spannung angeben. Sofern VR grösser als VRLIMP ist, erfolgt ein Ausgang von diesem Entscheidungsblock durch eine Ja-Verzweigung in einen Aktionsblock, in dem VR gleich VRLIMP gesetzt wird. Andererseits wird, sofern VR nicht grösser als VRLIMP ist, eine Nein-Verzweigung genommen, und es erfolgt ein Einsprung in einen Entscheidungsblock VR < VRLIMN. In diesem Block wird eine Ja-Verzweigung genommen, in der VR gleich VRLIMN gesetzt wird, sofern VR kleiner als VRLIMN ist. Ist dies nicht der Fall, wird eine Nein-Verzweigung genommen, und VR0, der Platz zur Speicherung des alten Werts von VR, wird durch das Setzen von VR0 = VR aktualisiert.

Das Programm tritt nun in die Stelle D ein, wobei es Fig. 18 verlässt und in die Stelle D der Fig. 19 eintritt. Beim Eintreten in Fig. 19 tritt der Prozessor in einen Entscheidungsblock MODFLG = 0 ein, in dem ein Test durchgeführt wird, um festzustellen, ob das System im kontinuierlichen oder diskontinuierlichen Strombetrieb arbeitet. Sofern MODFLG nicht gleich 0 ist, zeigt dies an, dass das System im kontinuierlichen Betrieb arbeitet. Der Prozessor tritt dann aus einer Nein-Verzweigung aus und tritt in einen Entscheidungsblock DIRFLG = 0 ein. Im Flussdiagramm der Fig. 19 wird die Entscheidung getroffen, ob es passend ist, die Richtung des Motors umzukehren. Das Kriterium für die Richtungsumkehr des Motors besteht darin, dass das System im diskontinuierlichen Strombetrieb arbeiten muss und das Vorzeichen des Stromsollwerts (DESI) dem Richtungskennzeichen (DIRFLG) entgegen gesetzt sein muss. Diese Bestimmung der Stromumkehr wird folgendermassen erläutert. Sofern der Entscheidungstest MODFLG = 0 positiv ist, tritt der Prozessor durch eine Ja-Verzweigung aus und zeigt damit einen diskontinuierlichen Betrieb in einem Vorzeichen des Entscheidungsblocks DESI OPP DIRFLG an. Im letzteren Entscheidungsblock wird die Feststellung getroffen, ob DESI der Grösse DIRFLG entgegengesetzt ist. Sofern DESI nicht entgegengesetzt ist, wird eine Nein-Verzweigung genommen, und das Programm tritt in den Entscheidungsblock DIRFLG = 0, wie schon beschrieben wurde. Wenn jedoch DESI der Grösse DIRFLG entgegengesetzt ist, nimmt das Programm

eine Ja-Verzweigung und tritt in einen Aktionsblock, in dem das Richtungskennzeichen DIRFLG gegenüber dem augenblicklichen Zustand umgekehrt wird. Wie in diesem Aktionsblock gezeigt ist, wird, wenn DIRFLG gleich dem Wert 1 gesetzt ist, dadurch angezeigt, dass der Strom in der Rückwärtsbrücke und nicht in der Vorwärtsbrücke fliesst. Wenn DIRFLG gleich 0 gesetzt ist, dann wird die Vorwärtsbrücke gezündet. Das Programm läuft nun weiter in den Entscheidungsblock DIRFLG = 0, um die relative Polarität der Grösse CEMF und der Spannung von der Brücke festzustellen. Sofern die Rückwärtsbrücke gezündet werden soll, wird die Nein-Verzweigung von diesem Entscheidungsblock genommen, und es erfolgt ein Einsprung in einen Aktionsblock, in dem die gewünschte Klemmenspannung VT dadurch berechnet wird, dass VT = CEMF (die Gegen-EMK) minus VR (die Motorspannung, da die Polaritäten entgegengesetzt sind) gesetzt wird. Wenn die Vorwärtsbrücke gezündet werden soll, wie durch DIRFLG = 0 angezeigt wird, dann erfolgt ein Einsprung durch den Ja-Zweig in einen Aktionsblock, in dem die gewünschte Motorklemmenspannung VT dadurch berechnet wird, dass VT = CEMF + VR gesetzt wird, um die richtige Polarität zu liefern.

Im vorliegenden System sind der Motorklemmenspannung positive und negative Grenzwerte auferlegt, die der Berechnung von VT unmittelbar folgenden Tests müssen also bestimmen, ob VT gleich oder kleiner als die positiven und negativen Grenzwerte ist. Der erste Entscheidungsblock nach der Berechnung von VT hat zum Inhalt VT > VTLIMP. Sofern VT über dem positiven Grenzwert liegt, wird ein Ja-Zweig in einen Aktionsblock genommen, in dem VT gleich dem maximalen positiven Grenzwert VTLIMP gesetzt wird. Sofern andererseits VT kleiner als VTLIMP ist, wird eine Nein-Verzweigung genommen, und ein ähnlicher Test für VT < VTLIMN ausgeführt. Sofern VT kleiner als die minimale Grenze ist, wird eine Ja-Verzweigung genommen, und VT gleich VTLIMN gesetzt. Sofern dies nicht der Fall ist, wird eine Nein-Verzweigung vom Entscheidungsblock VT < VTLIMN genommen und ein Einsprung in einen Aktionsblock der Fig. 19 vorgenommen, in welchem der Zündwinkel FINVAL zur Erzeugung der gewünschten VT aus der Wertetabelle genommen wird, die aufgrund der zuvor angegebenen Beziehung

$$FINVAL = 245{,}8\ COS^{-1}(3VT/\pi V_{LN})$$

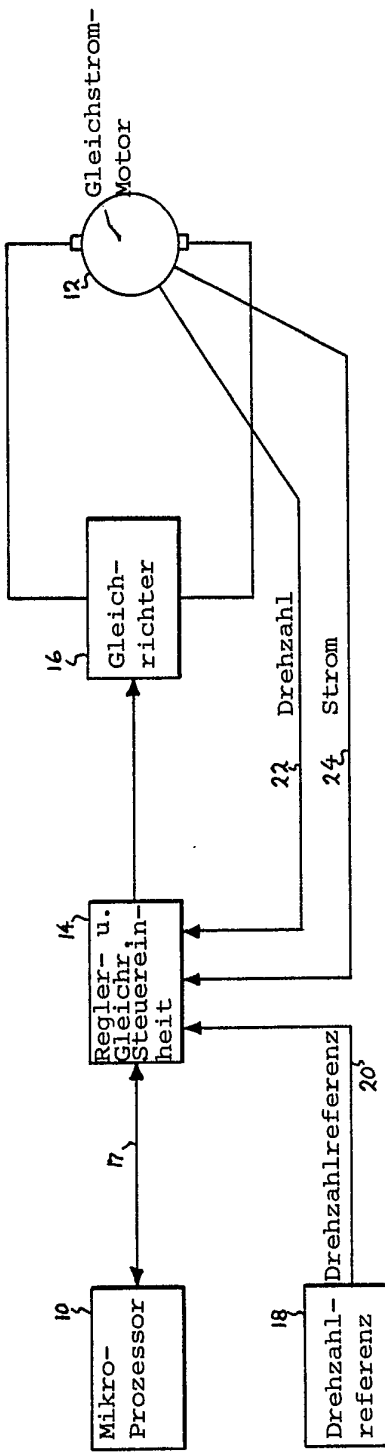
berechnet ist, wie oben angegeben ist. Es sei daran erinnert, dass diese Werte der Grösse FINVAL oben geschildert und in Tabelle 1 dargestellt sind. Diese Art des Einsprungs in eine Tabelle ist bekannt und stellt eine geradlinige Art dar, die Tabelle mit einer Adresse zu adressieren, die durch den Wert von VT angegeben ist, und wobei der Wert des adressierten Platzes als Zündwinkel FINVAL benutzt wird.

Das Programm verlässt nun Fig. 19 an der Anschlussstelle E und tritt in die Anschlussstelle E der Fig. 20 ein, in der der Systemtakt durch den Prozessor in einer schon geschilderten Weise ausgelesen wird. Das Programm setzt seine Ausführung in Fig. 20 nicht fort und kehrt bei seiner Ausführung schliesslich zurück zur Unterbrechungsstelle des Hauptprogramms in der schon geschilderten Weise.

Nachdem die Erfindung in Einzelheiten beschrieben ist, lässt sich würdigen, dass die Gesamtstruktur und das Gesamtverfahren des vorliegenden Systems ein Hauptprogramm umfasst, das kontinuierlich in einer Schleife umläuft, um die Drehzahlreferenzschalter und die Motorrichtungsschalter auszulesen und den Geschwindigkeitssollwert für den Motor zu berechnen. Das Unterbrechungsprogramm erhält die Drehzahlsollwertdaten vom Hauptprogramm und liest die Motordrehzahl, den Ankerstrom und die Zeit, die durch den 360°-Systemtakt gemessen werden. Das Unterbrechungsprogramm berechnet ferner den gewünschten Zündwinkel der Thyristoren und steuert den Prozessor, dass dieser Daten, deren Wert proportional zum Zündwinkel ist, an einen Zähler im System, und ein Adressenwort an eine Thyristorwähl- und Drehrichtungslogik sendet, um die Generatoren von Steuerimpulsen für die direkte digitale Zündung der Thyristoren, die durch die Adressendaten ausgewählt sind, zu veranlassen, ein umkehrbares Dreiphasen-Motorantriebssystem zu regeln und zu steuern. Das Programm ist mit dem Zünden der Thyristoren dadurch synchronisiert, dass eine Unterbrechung bei jedem Thyristorzündvorgang erzeugt wird, um die Reglerberechnungen zu starten, um den Zähler zur richtigen Zeit zu beladen, um dadurch die Zündzeit für einen nächsten, zu zündenden Thyristor zu steuern.

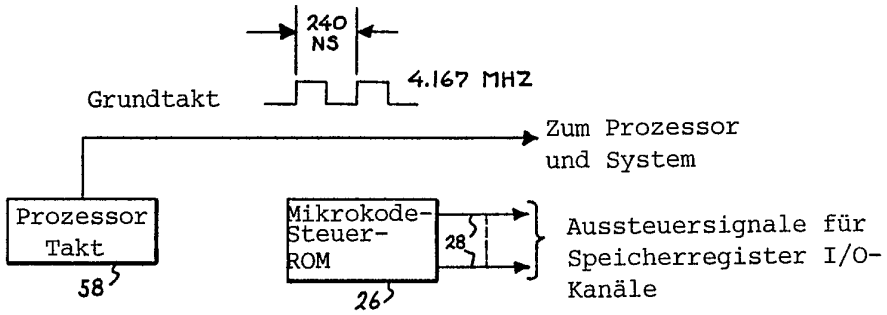
Der Prozessor des Systems liest den Ankerstrom zweimal pro 60 elektrischen Graden aus. Eine erste Ankerstromauslesung wird zu einer vorgegebenen Zeit (zum Beispiel 4 Grad) bevor der nächste Thyristor gezündet werden soll, vorgenommen. Diese erste Stromauslesung bzw. Stromlesewert wird verwendet, um die Art des Stromreglerbetriebs (kontinuierlich oder diskontinuierlich) zu bestimmen. Eine zweite Stromauslesung bzw. Stromlesewert wird vorgenommen, und die Reglerberechnungen werden ungefähr 20° nach der vorausgegangenen Zündung eines Thyristors gestartet. Der zweite Stromlesewert wird durch das Stromreglerprogramm als Stromrückkopplung zur Steuerung des Gesamtstromreglers verwendet.

Es wurde ein Regel- und Steuersystem zur Regelung einer Last, zum Beispiel eines Gleichstrommotors beschrieben, welches sich der Vorteile eines Prozessors, zum Beispiel eines Mikrocomputers, bedient und die Möglichkeiten bekannter analoger Regel- und Steuersysteme weit übertrifft, wobei nur begrenzte zusätzliche Ausgaben erforderlich sind.



Regel- und Steuersystem

FIG. 1


240 NS
4.167 MHZ
Grundtakt
Zum Prozessor und System
Prozessor Takt
58
Mikrokode-Steuer-ROM
28
26
Aussteuersignale für Speicherregister I/O-Kanäle

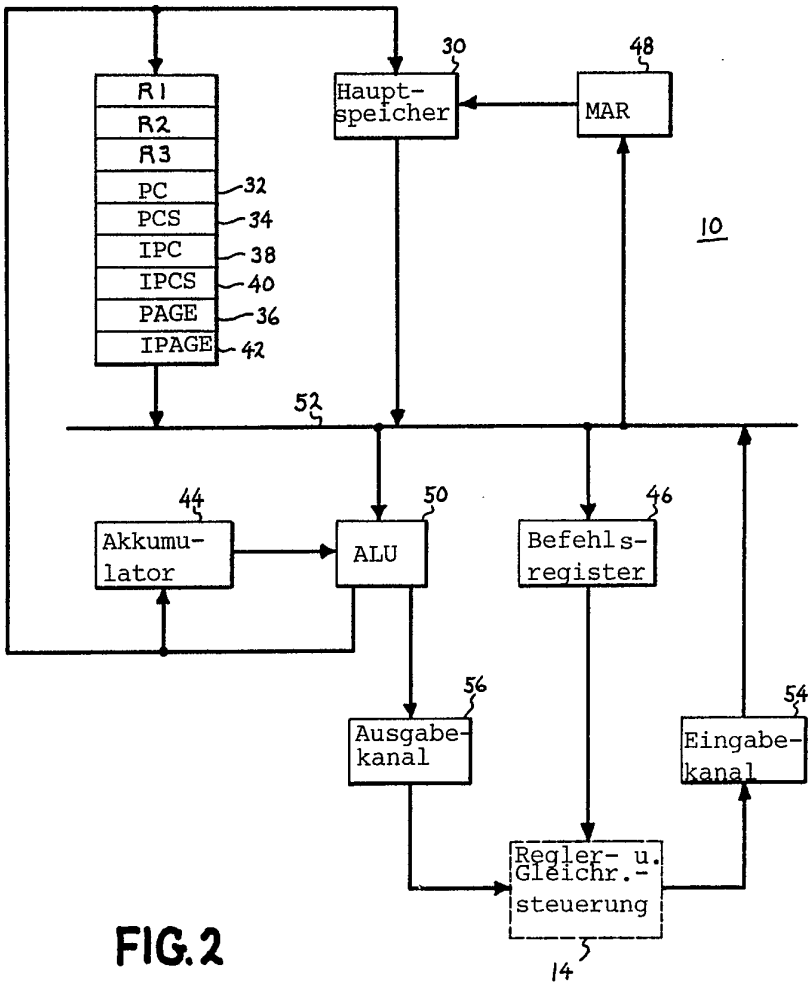

R1
R2
R3
PC
PCS
IPC
IPCS
PAGE
IPAGE
32
34
38
40
36
42
Haupt-speicher
30
MAR
48
10
52
Akkumu-lator
44
ALU
50
Befehls-register
46
Ausgabe-kanal
56
Eingabe-kanal
54
Regler- u. Gleichr.-steuerung
14

FIG. 2

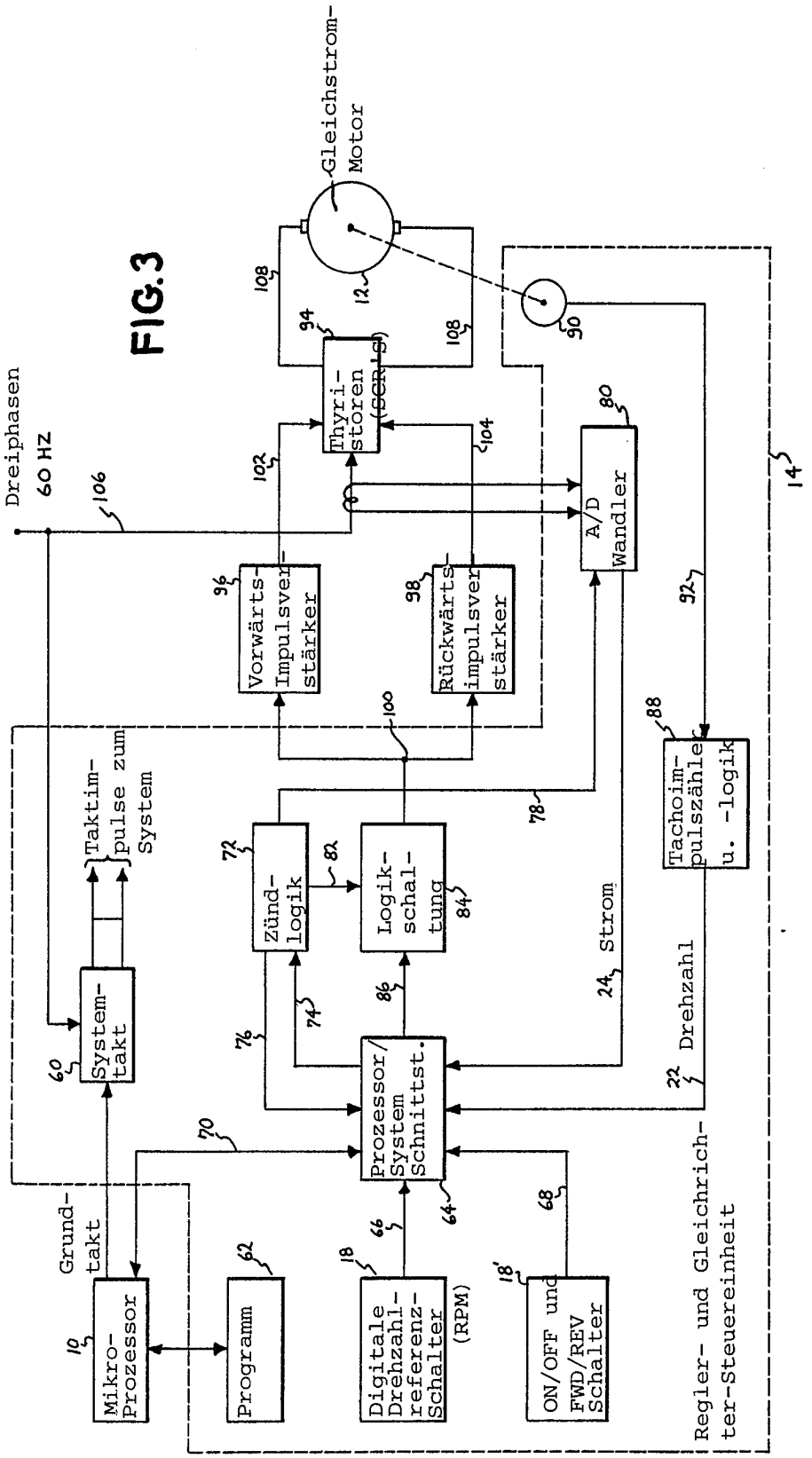

Gleichstrom-Motor
Dreiphasen
60 HZ
Thyri-storen (SCR's)
A/D Wandler
Vorwärts-Impulsver-stärker
Rückwärts-impulsver-stärker
Taktim-pulse zum System
System-takt
Zünd-logik
Logik-schal-tung
Tachoim-pulszähler u. -logik
Strom
Drehzahl
Prozessor/ System Schnittst.
Grund-takt
Mikro-Prozessor
Programm
Digitale Drehzahl-referenz-Schalter (RPM)
ON/OFF und FWD/REV Schalter
Regler- und Gleichrich-ter-Steuereinheit

FIG. 3

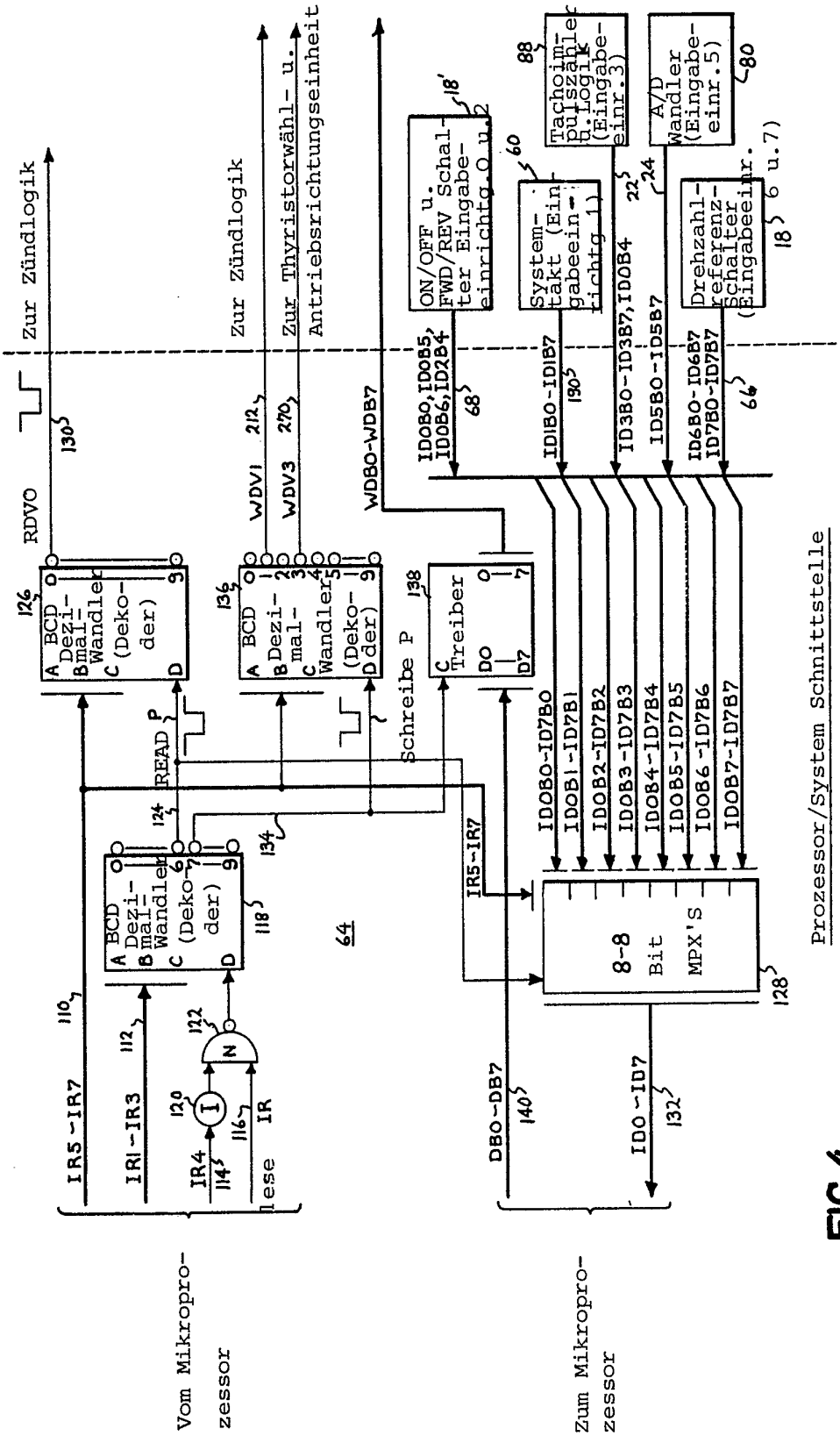
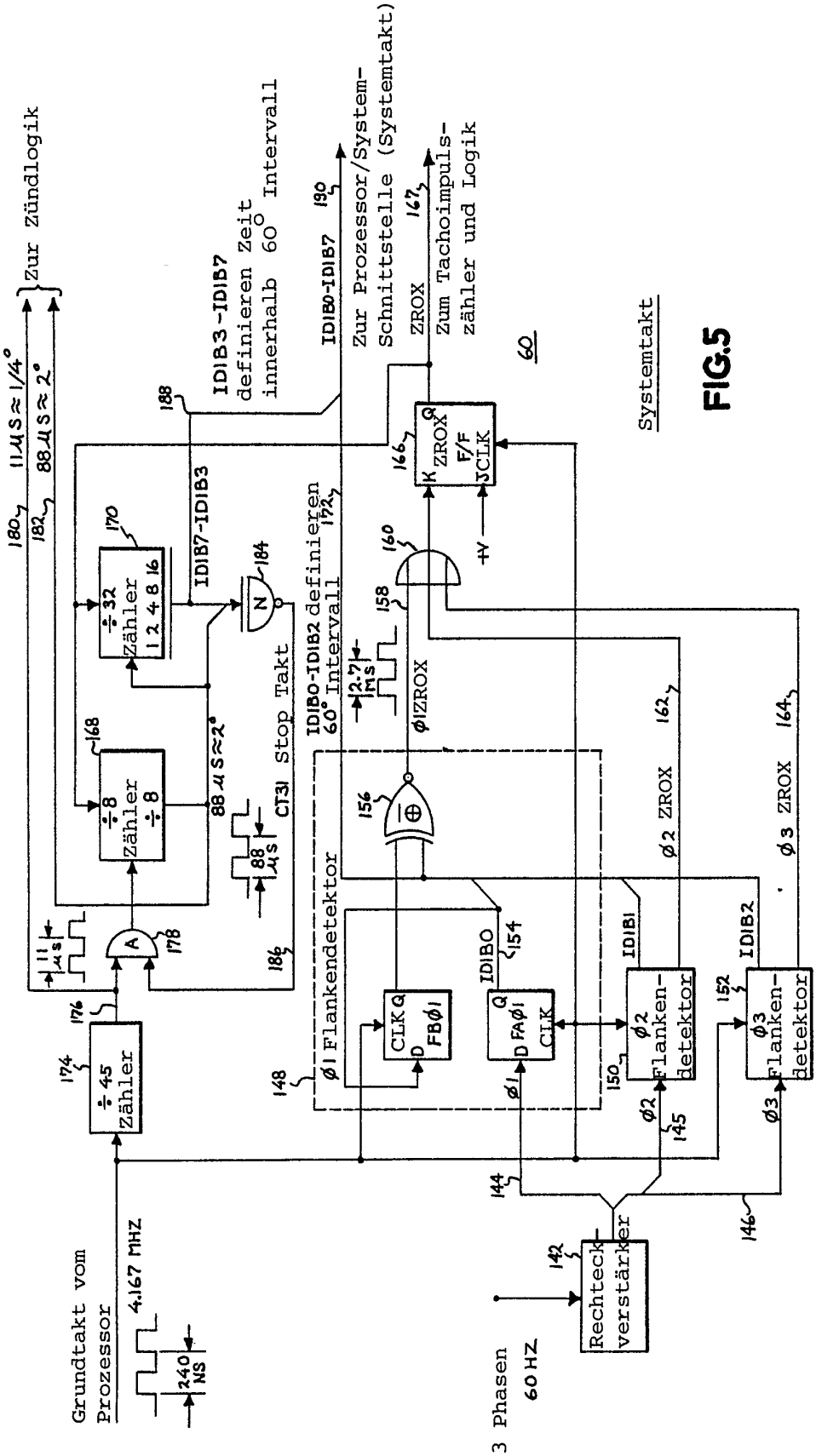

Zur Zündlogik
Zur Zündlogik
Zur Thyristorwähl- u. Antriebsrichtungseinheit
RDVO
WDV1
WDV3
WDB0-WDB7
ON/OFF u. FWD/REV Schalter Eingabeeinrichtg.0 u.2
Tachoimpulszähler u. Logik (Eingabeeinr.3)
A/D Wandler (Eingabeeinr.5)
Systemtakt (Eingabeeinrichtg.1)
Drehzahlreferenzschalter (Eingabeeinr. 6 u.7)
ID0B0,ID0B5, ID0B6,ID2B4
ID1B0-ID1B7
ID3B0-ID3B7,ID0B4
ID5B0-ID5B7
ID6B0-ID6B7 ID7B0-ID7B7
BCD Dezimal-Wandler (Dekoder)
BCD Dezimal-Wandler (Dekoder)
Treiber
READ P
Schreibe P
IR5-IR7
IR1-IR3
IR4
lese IR
IR5-IR7
8-8 Bit MPX'S
DB0-DB7
ID0-ID7
Vom Mikroprozessor
Zum Mikroprozessor
Prozessor/System Schnittstelle

FIG.4


Grundtakt vom Prozessor
4.167 MHZ
240 NS
÷ 45 Zähler
÷ 8 Zähler ÷ 8
÷ 32 Zähler 1 2 4 8 16
11 µS
88 µS ≈ 2°
11 µS ≈ 1/4°
Zur Zündlogik
ID1B3-ID1B7 definieren Zeit innerhalb 60° Intervall
ID1B7-ID1B3
ID1B0-ID1B7
Zur Prozessor/System-Schnittstelle (Systemtakt)
ZROX
Zum Tachoimpuls-zähler und Logik
88 µS ≈ 2°
CT31 Stop Takt
ID1B0-ID1B2 definieren 60° Intervall
2.7 µS
ØIZROX
Ø1 Flankendetektor
CLK Q D FBØ1
IDIBO
Q D FAØ1 CLK
Ø2 Flanken-detektor
Ø3 Flanken-detektor
IDIB1
IDIB2
Ø2 ZROX
Ø3 ZROX
ZROX F/F JCLK
Systemtakt
3 Phasen 60 HZ
Rechteck-verstärker
60
FIG.5

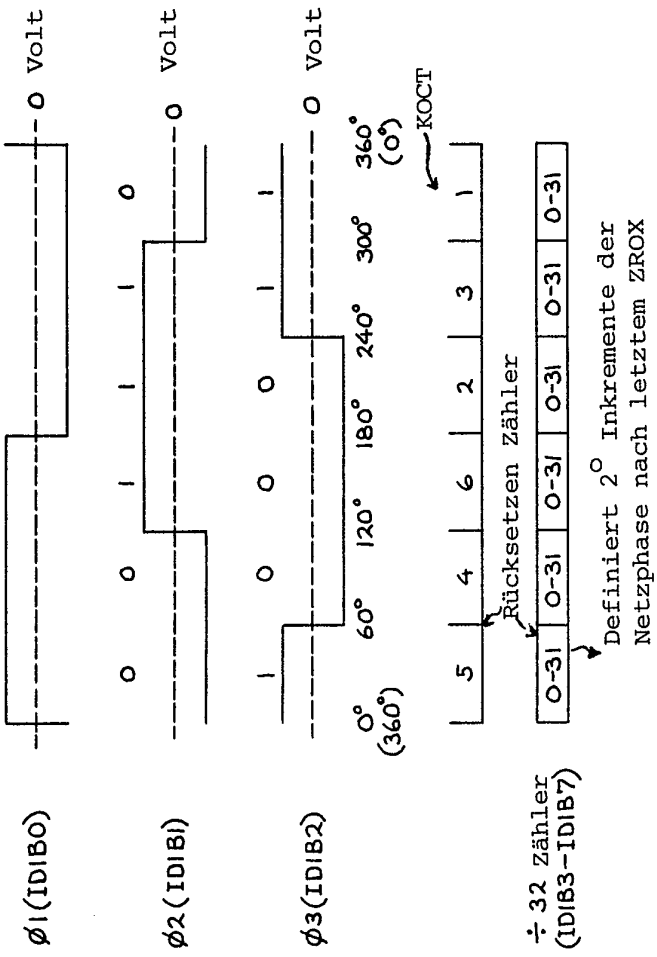
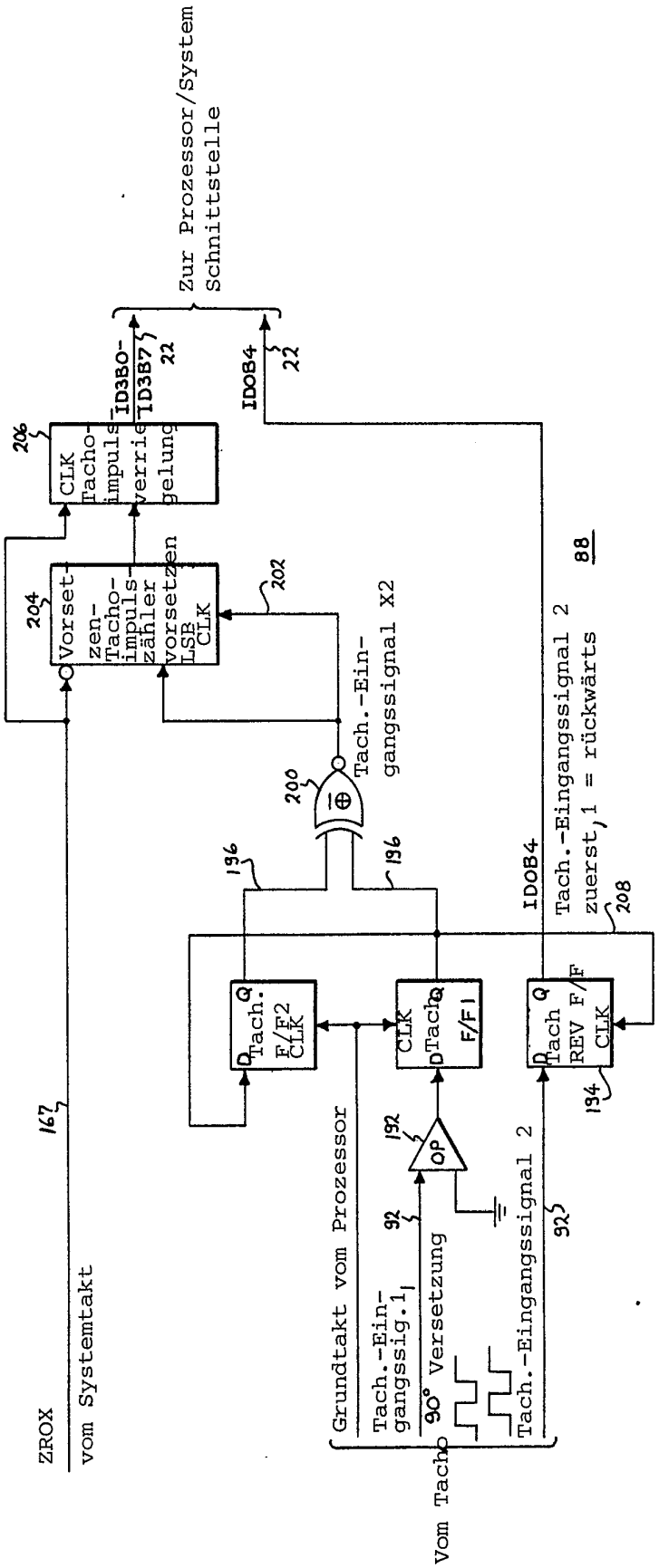

ø1(ID1B0)
ø2(ID1B1)
ø3(ID1B2)
0 Volt
0 Volt
0 Volt
1 1 1 0 0 0 0
0 0 1 1 1 1 0
1 0 0 0 1 1 1
0° (360°)
60°
120°
180°
240°
300°
360° (0°)
KOCT
5 4 6 2 3 1
Rücksetzen Zähler
÷ 32 Zähler (ID1B3-ID1B7)
0-31 0-31 0-31 0-31 0-31
Definiert 2° Inkremente der Netzphase nach letztem ZROX

FIG.6



Tachoimpulszähler und -logik

FIG.7

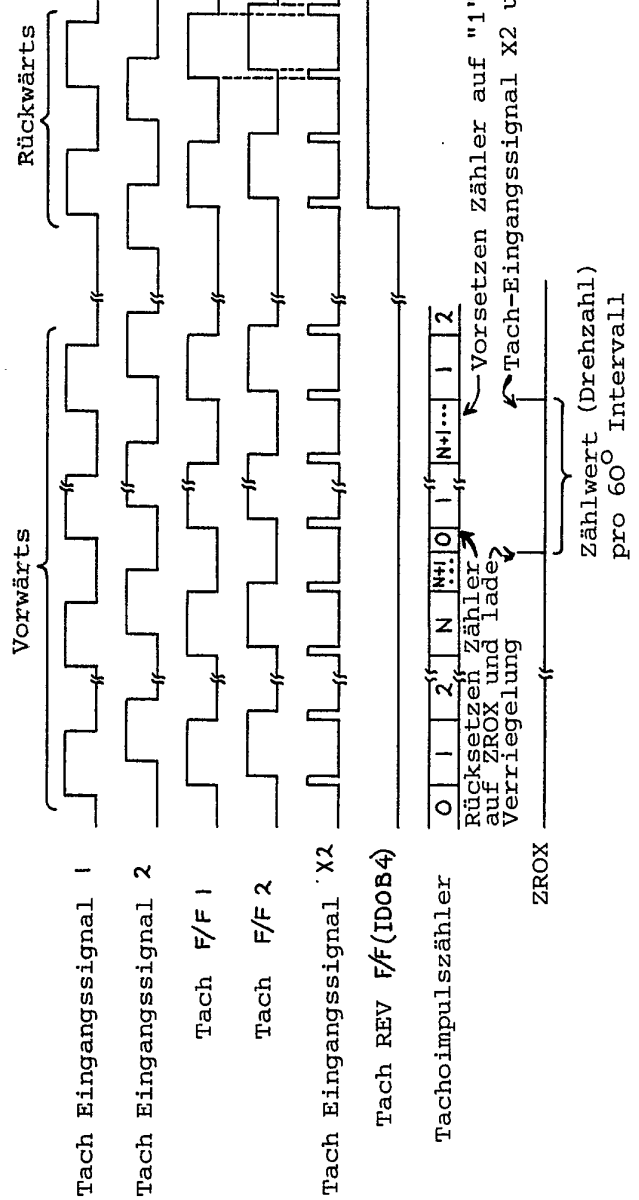

Vorwärts
Rückwärts
Tach Eingangssignal 1
Tach Eingangssignal 2
Tach F/F 1
Tach F/F 2
Tach Eingangssignal X2
Tach REV F/F (IDOB4)
Tachoimpulszähler
0
1
2
N
N+1
0
1
N+1
1
2
Rücksetzen Zähler auf "1", sofern
auf ZROX und lade Vorsetzen
Verriegelung
Tach-Eingangssignal X2 und ZROX
ZROX
Zählwert (Drehzahl)
pro 60° Intervall

FIG. 8

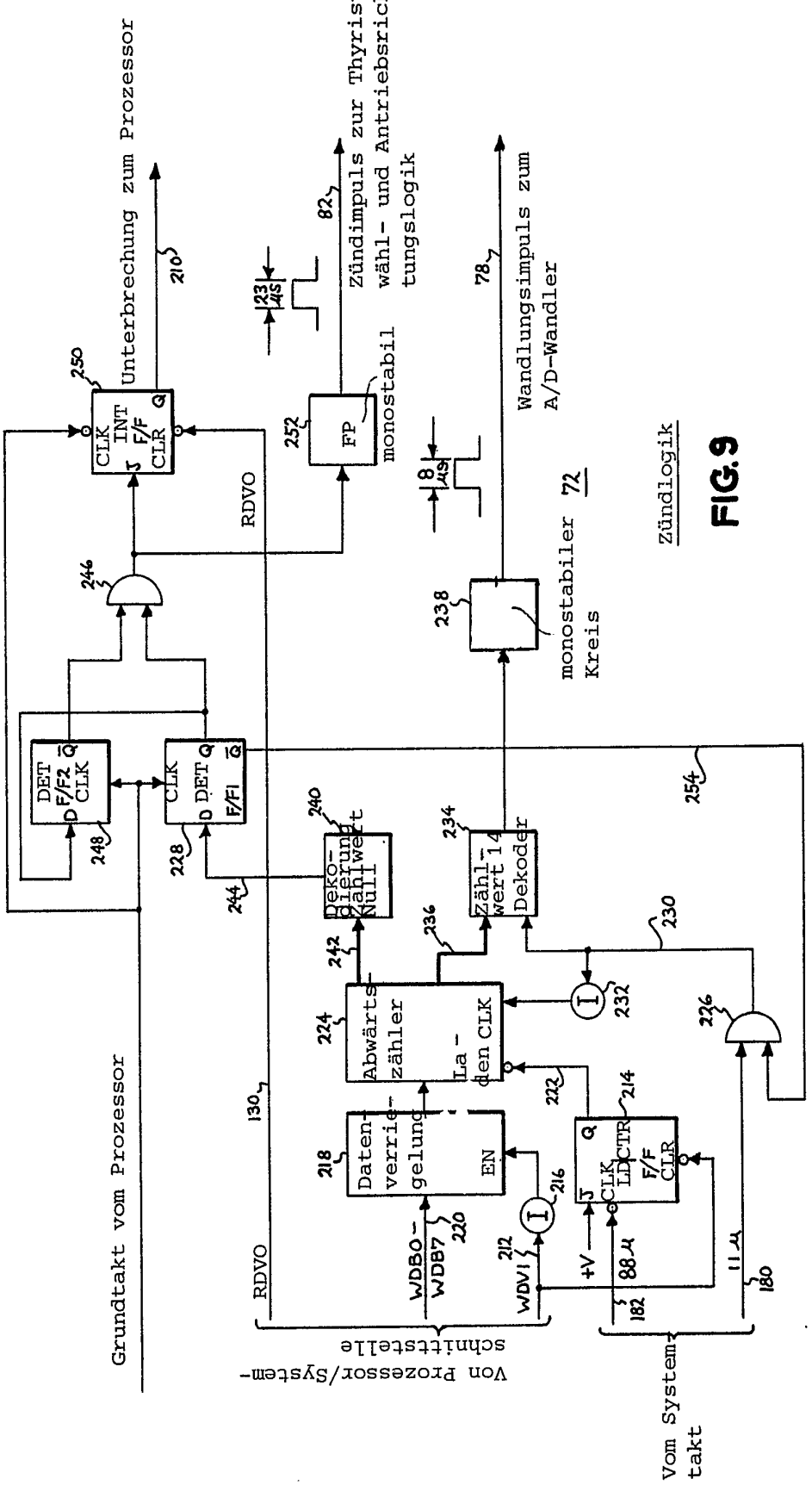

Unterbrechung zum Prozessor
Zündimpuls zur Thyristor-wähl- und Antriebsrichtungslogik
Wandlungsimpuls zum A/D-Wandler
monostabil
monostabiler Kreis
Zündlogik
Grundtakt vom Prozessor
Von Prozessor/Systemschnittstelle
Vom Systemtakt
RDVO
CLK INT F/F CLR
FP
DET F/F2 CLK
CLK DET F/F1
Deko-dierung Zählwert Null
Zähl-wert 14 Dekoder
Abwärts-zähler
La-den CLK
Daten-verrie-gelung
EN
CLK LD CTR F/F CLR
WDB0-WDB7
WDV1
+V
88 µ
11 µ
23 µs
8 µs
210
250
252
82
78
246
238
248
228
240
234
254
244
242
236
230
224
232
226
222
218
214
216
220
212
130
182
180

FIG. 9

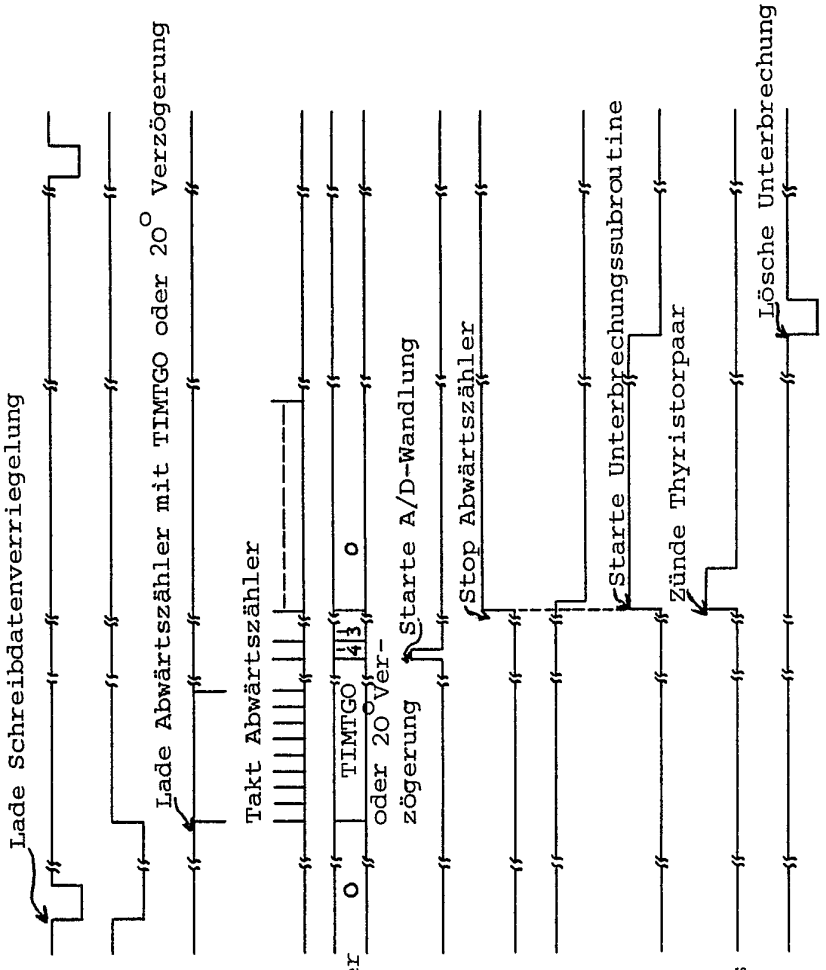

Lade Schreibdatenverriegelung
WDVI
Lastzähler Flip-Flop
Lade Abwärtszähler mit TIMTGO oder 20° Verzögerung
88 µs
Takt Abwärtszähler
11 µs
Abwärtszähler
TIMTGO
oder 20° Verzögerung
Starte A/D-Wandlung
Wandlungsimpuls
Stop Abwärtszähler
DET F/F 1
DET F/F 2
Starte Unterbrechungssubroutine
INT F/F
Zünde Thyristorpaar
Zündimpuls
Lösche Unterbrechung
RDVO

FIG.10

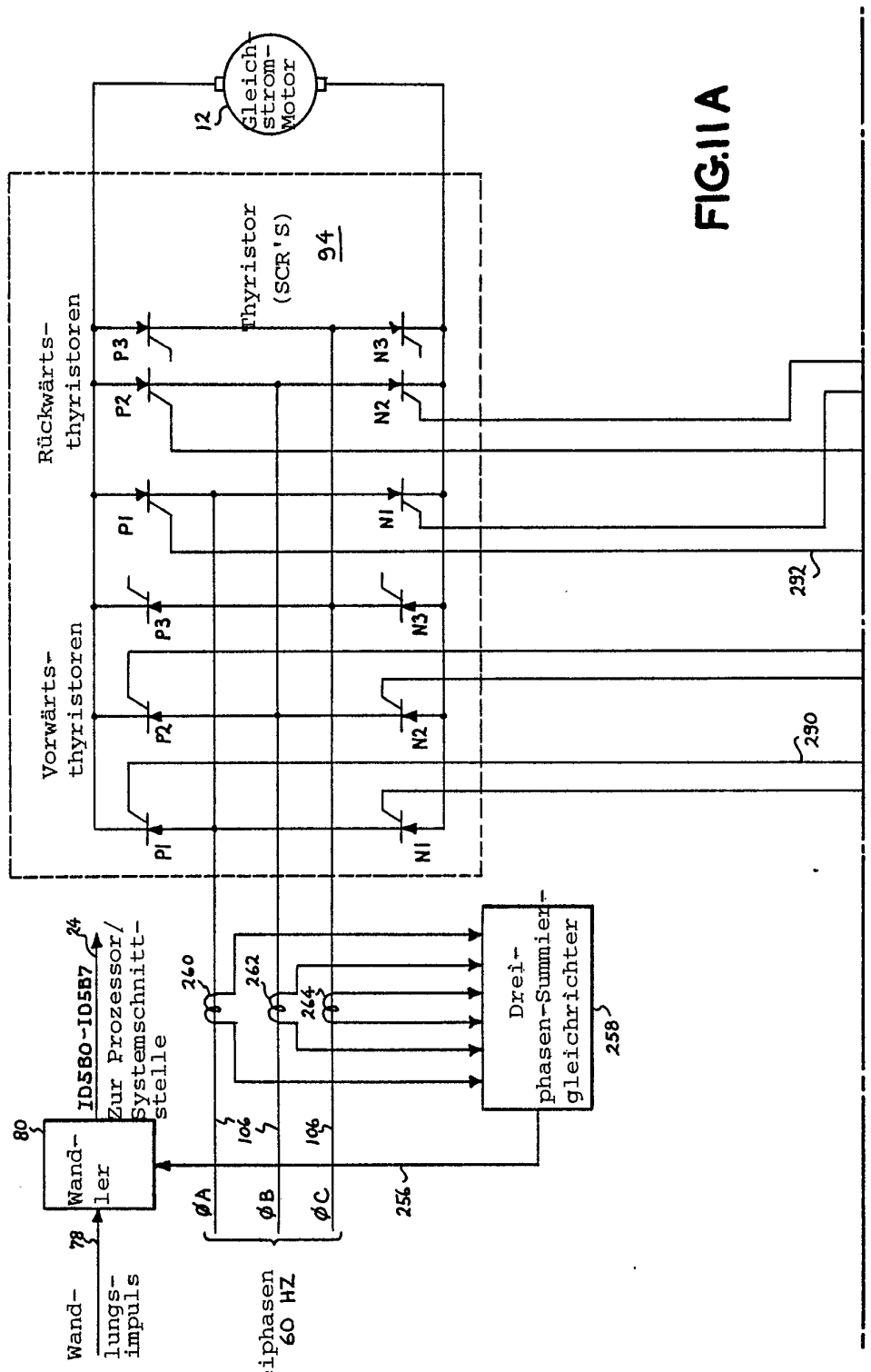

12
Gleich-strom-Motor
Thyristor (SCR'S) 94
Rückwärts-thyristoren
Vorwärts-thyristoren
P1
P2
P3
N1
N2
N3
232
230
Drei-phasen-Summier-gleichrichter
258
256
260
262
264
106
106
ØA
ØB
ØC
Dreiphasen 60 HZ
80
Wand-ler
78
Wand-lungs-impuls
ID580-ID587
24
Zur Prozessor/ Systemschnitt-stelle

FIG.11A

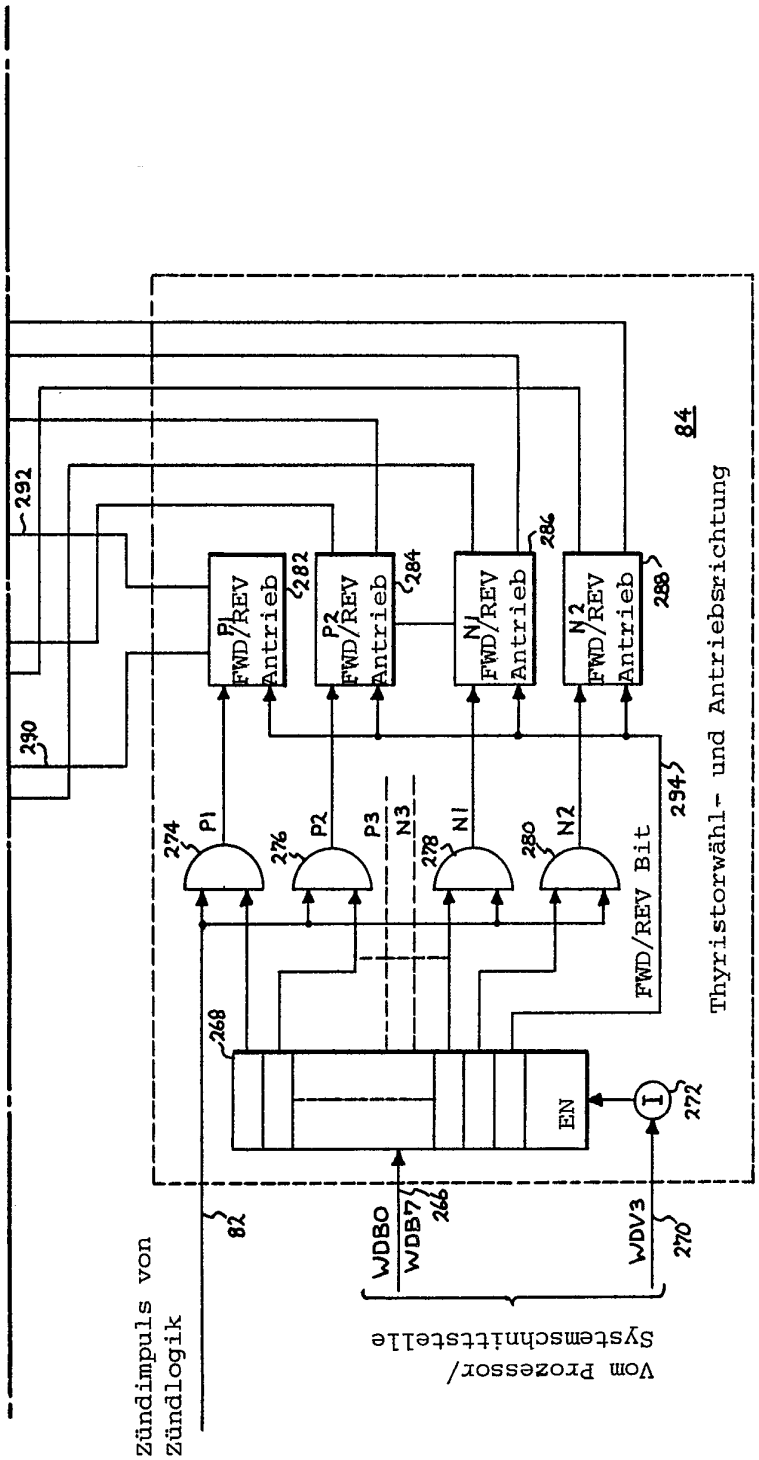

292
290
P1 FWD/REV Antrieb
282
P2 FWD/REV Antrieb
284
N1 FWD/REV Antrieb
286
N2 FWD/REV Antrieb
288
84
Thyristorwähl- und Antriebsrichtung
274
276
278
280
P1
P2
P3
N3
N1
N2
294
FWD/REV Bit
268
EN
I
272
82
WDB0
WDB7
266
WDV3
270
Zündimpuls von Zündlogik
Vom Prozessor/ Systemschnittstelle

FIG11B

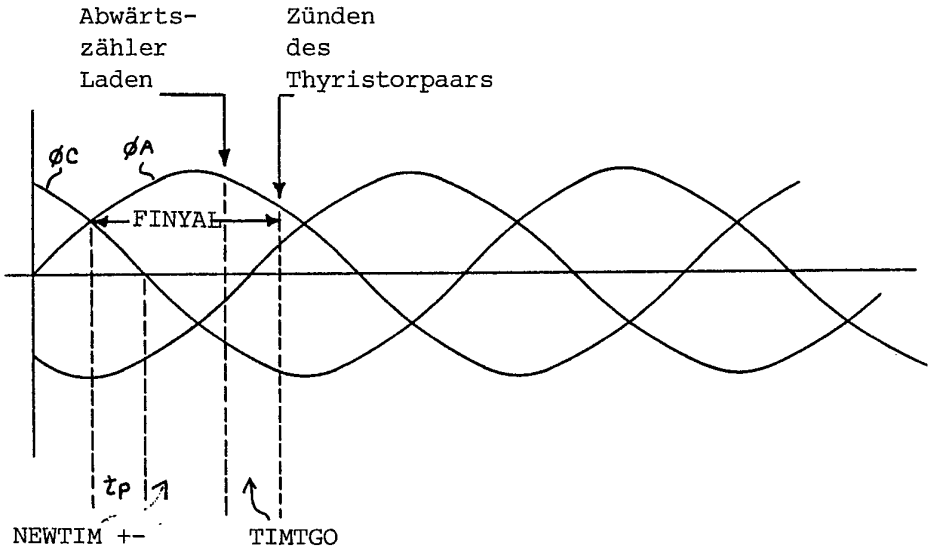

Abwärts-
zähler
Laden
Zünden
des
Thyristorpaars
øC
øA
FINYAL
tp
NEWTIM +-
TIMTGO

FIG.12

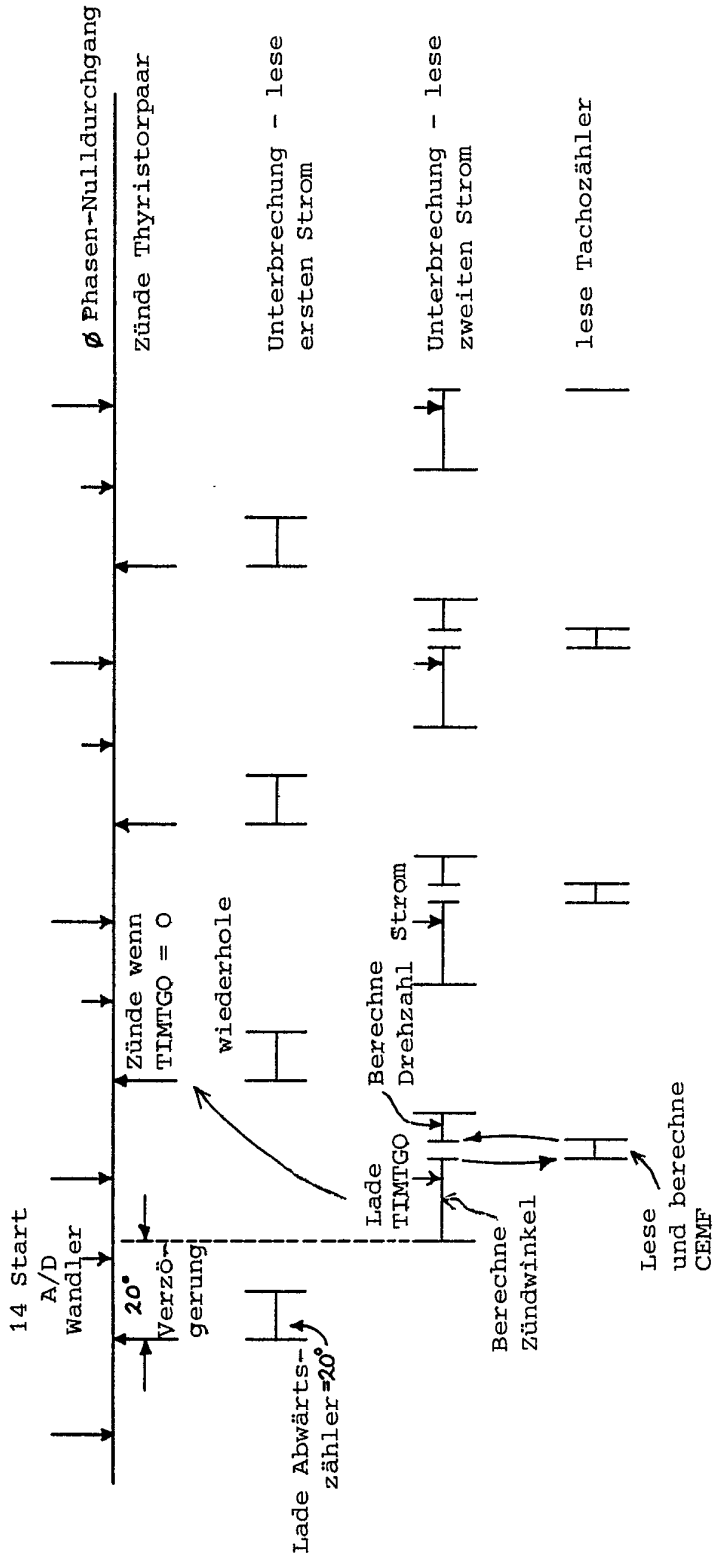

14 Start A/D Wandler
20° Verzögerung
Lade Abwärtszähler=20°
Zünde wenn TIMTGO = 0
wiederhole
Ø Phasen-Nulldurchgang
Zünde Thyristorpaar
Unterbrechung - lese ersten Strom
Unterbrechung - lese zweiten Strom
lese Tachozähler
Berechne Drehzahl Strom
Lade TIMTGO
Berechne Zündwinkel
Lese und berechne CEMF

FIG.13

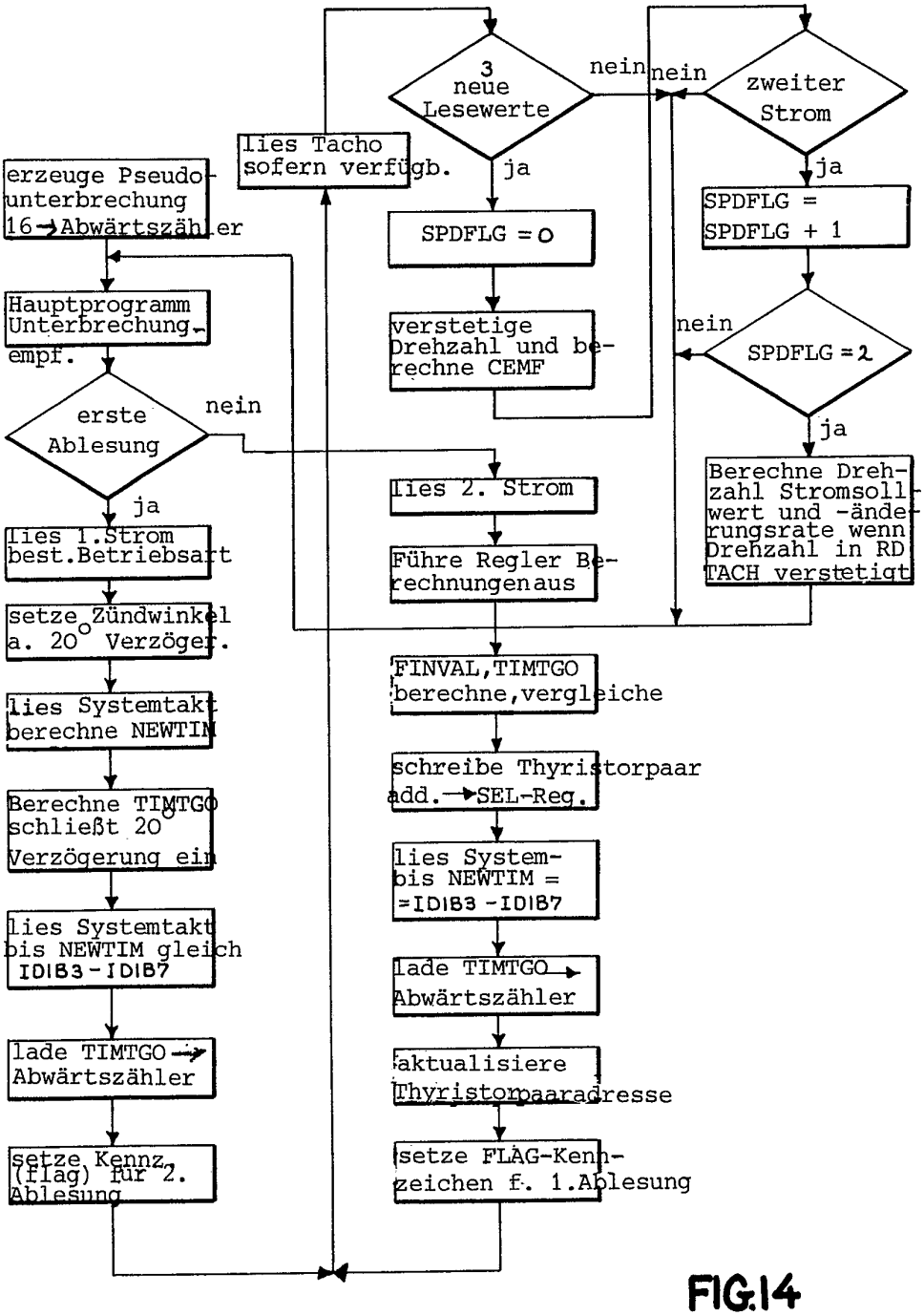

erzeuge Pseudo-unterbrechung 16→Abwärtszähler
Hauptprogramm Unterbrechung-empf.
erste Ablesung
nein
ja
lies 1.Strom best.Betriebsart
setze Zündwinkel a. 20° Verzöger.
lies Systemtakt berechne NEWTIM
Berechne TIMTGO schließt 20° Verzögerung ein
lies Systemtakt bis NEWTIM gleich IDIB3-IDIB7
lade TIMTGO→ Abwärtszähler
setze Kennz. (flag) für 2. Ablesung
lies Tacho sofern verfügb.
3 neue Lesewerte
nein
ja
SPDFLG = 0
verstetige Drehzahl und berechne CEMF
lies 2. Strom
Führe Regler Berechnungen aus
FINVAL,TIMTGO berechne,vergleiche
schreibe Thyristorpaar add.→SEL-Reg.
lies System-bis NEWTIM = =IDIB3 -IDIB7
lade TIMTGO→ Abwärtszähler
aktualisiere Thyristorpaaradresse
setze FLAG-Kenn-zeichen f. 1.Ablesung
nein
zweiter Strom
ja
SPDFLG = SPDFLG + 1
nein
SPDFLG = 2
ja
Berechne Dreh-zahl Stromsoll-wert und -ände-rungsrate wenn Drehzahl in RD TACH verstetigt

FIG.14

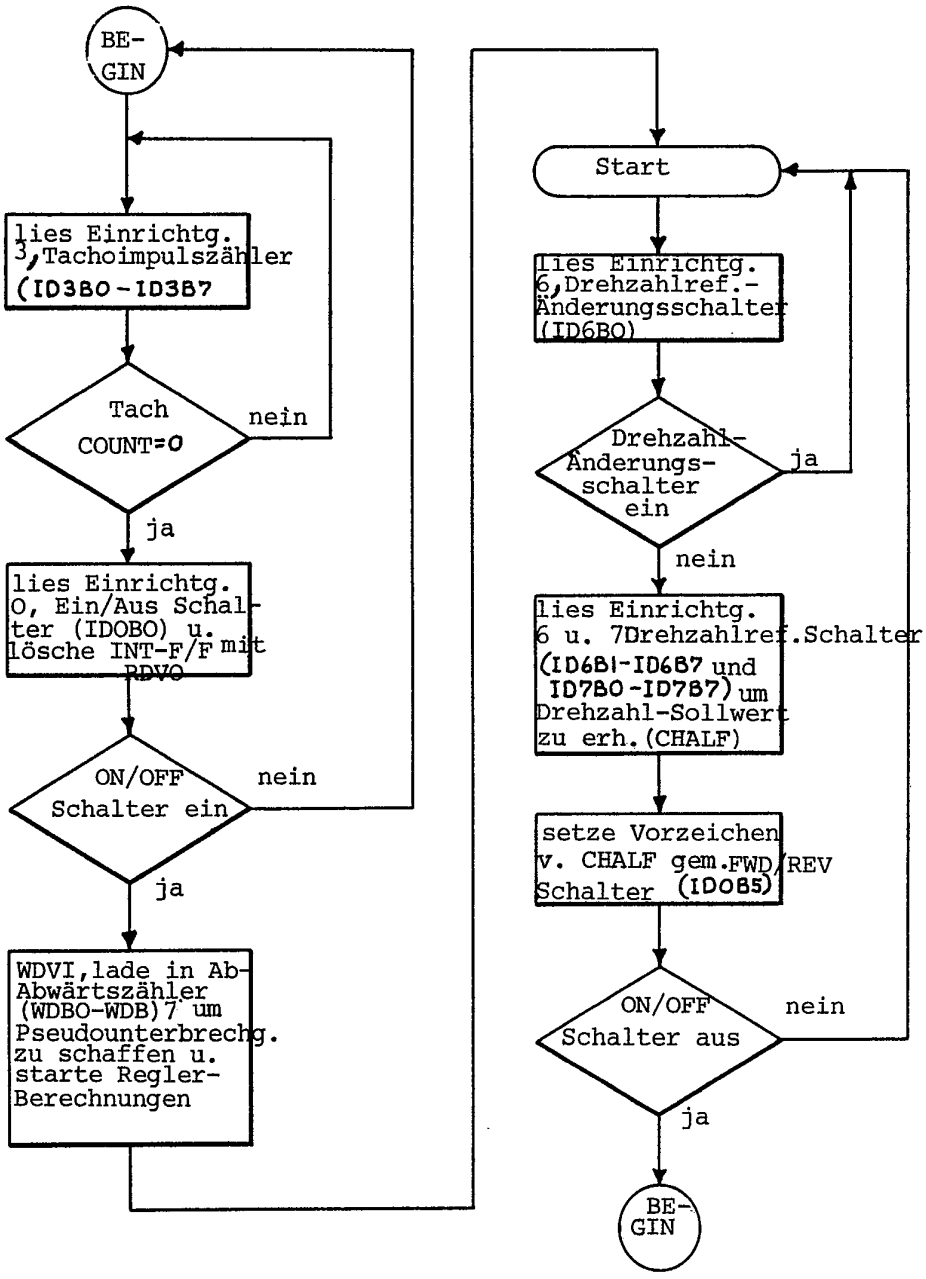

BE-
GIN
lies Einrichtg. 3, Tachoimpulszähler (ID3BO - ID3B7
Tach COUNT=0
nein
ja
lies Einrichtg. O, Ein/Aus Schalter (IDOBO) u. lösche INT-F/F mit RDVO
ON/OFF Schalter ein
nein
ja
WDVI, lade in Ab-Abwärtszähler (WDBO-WDB) 7 um Pseudounterbrechg. zu schaffen u. starte Regler-Berechnungen
Start
lies Einrichtg. 6, Drehzahlref.-Änderungsschalter (ID6BO)
Drehzahl-Änderungs-schalter ein
ja
nein
lies Einrichtg. 6 u. 7 Drehzahlref. Schalter (ID6B1-ID6B7 und ID7BO-ID7B7) um Drehzahl-Sollwert zu erh. (CHALF)
setze Vorzeichen v. CHALF gem. FWD/REV Schalter (IDOB5)
ON/OFF Schalter aus
nein
ja
BE-
GIN

FIG.15

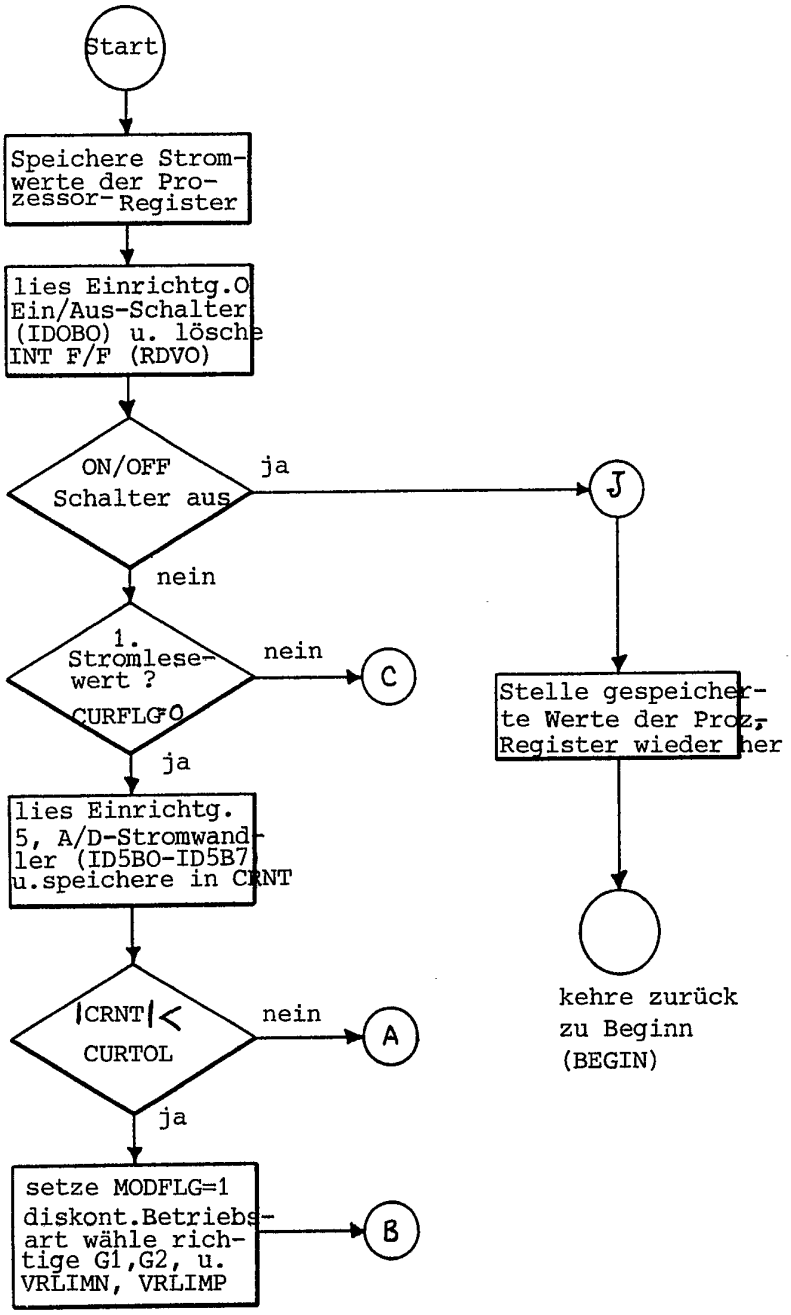

Start
Speichere Strom-werte der Pro-zessor-Register
lies Einrichtg.0 Ein/Aus-Schalter (IDOBO) u. lösche INT F/F (RDVO)
ON/OFF Schalter aus
ja
J
nein
1. Stromlese-wert ? CURFLG=0
nein
C
ja
lies Einrichtg. 5, A/D-Stromwand-ler (ID5BO-ID5B7) u.speichere in CRNT
Stelle gespeicher-te Werte der Proz. Register wieder her
|CRNT| < CURTOL
nein
A
ja
setze MODFLG=1 diskont.Betriebs-art wähle rich-tige G1,G2, u. VRLIMN, VRLIMP
B
kehre zurück zu Beginn (BEGIN)

FIG. 16

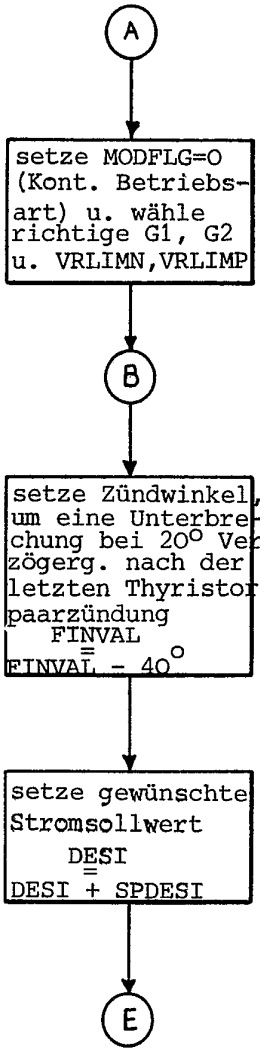

A
setze MODFLG=0 (Kont. Betriebsart) u. wähle richtige G1, G2 u. VRLIMN,VRLIMP
B
setze Zündwinkel, um eine Unterbrechung bei 20° Verzögerg. nach der letzten Thyristorpaarzündung FINVAL = FINVAL - 40°
setze gewünschte Stromsollwert DESI = DESI + SPDESI
E

FIG.17

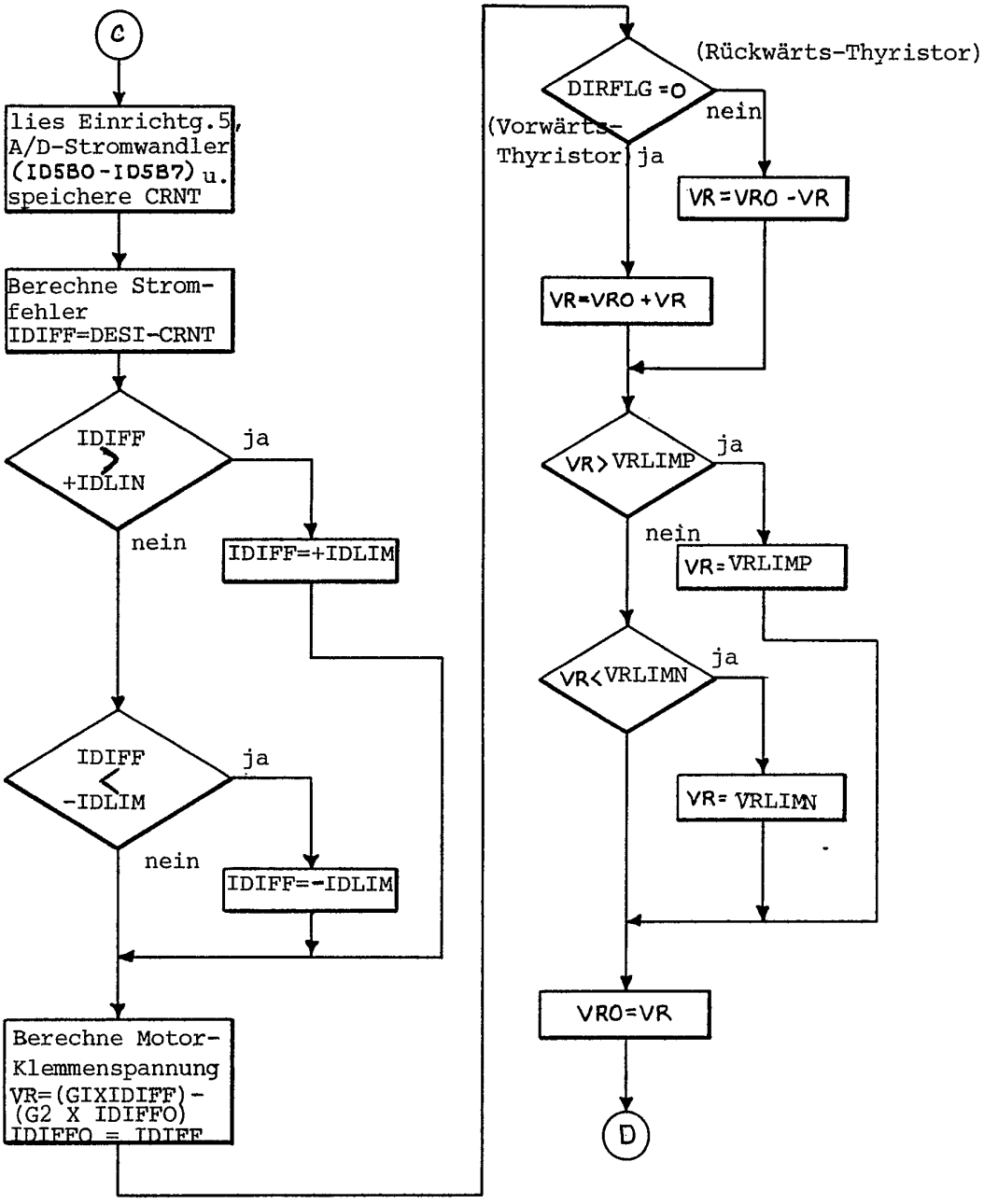

C
lies Einrichtg.5, A/D-Stromwandler (ID5B0-ID5B7) u. speichere CRNT
Berechne Strom-fehler IDIFF=DESI-CRNT
IDIFF > +IDLIN
ja
nein
IDIFF=+IDLIM
IDIFF < -IDLIM
ja
nein
IDIFF=-IDLIM
Berechne Motor-Klemmenspannung VR=(GIXIDIFF)-(G2 X IDIFFO) IDIFFO = IDIFF
DIRFLG=0
(Vorwärts-Thyristor)
ja
nein
(Rückwärts-Thyristor)
VR=VR0+VR
VR=VR0-VR
VR>VRLIMP
ja
nein
VR=VRLIMP
VR<VRLIMN
ja
VR=VRLIMN
VR0=VR
D

FIG. 18

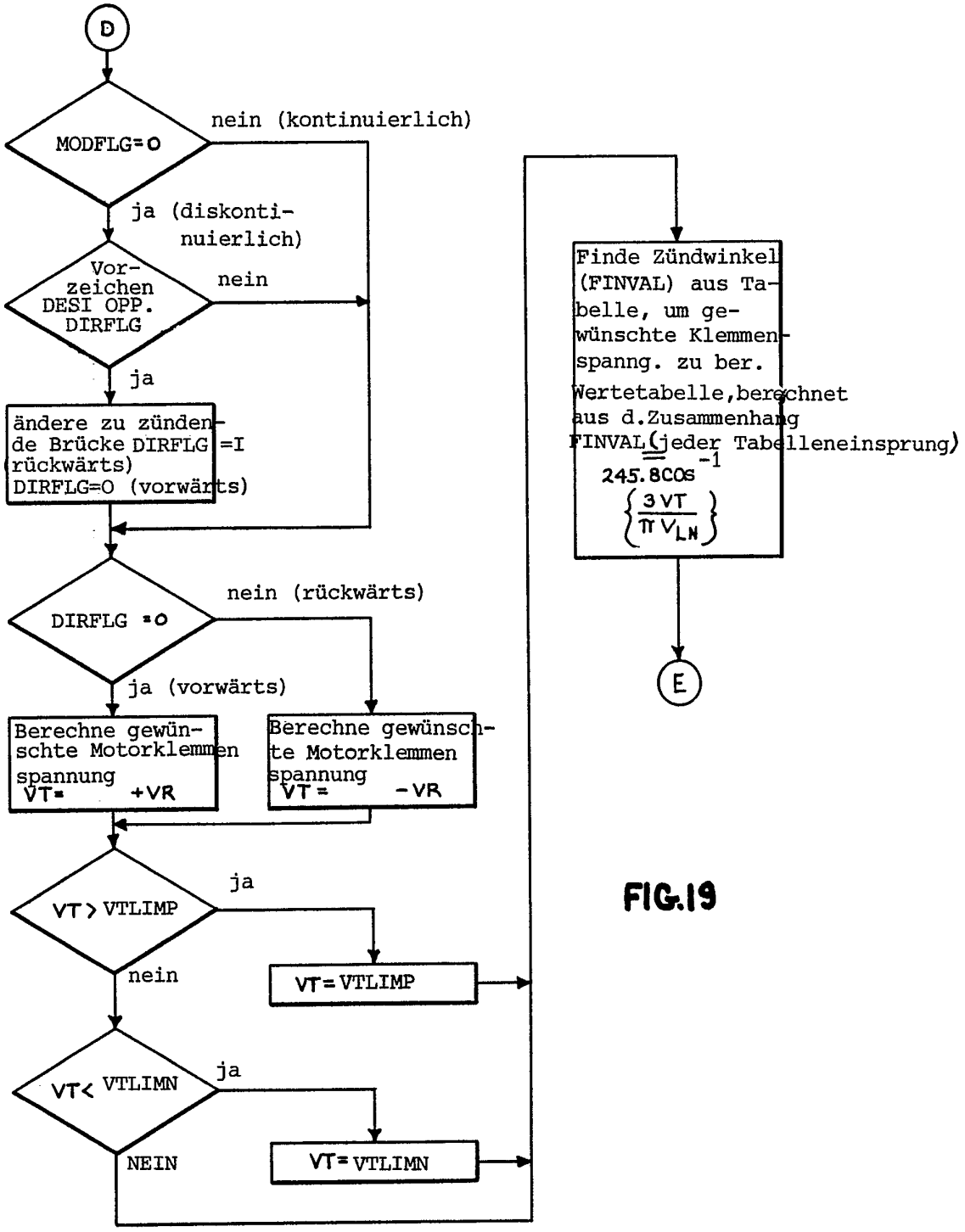

D
MODFLG=0
nein (kontinuierlich)
ja (diskontinuierlich)
Vorzeichen DESI OPP. DIRFLG
nein
ja
ändere zu zündende Brücke DIRFLG =I (rückwärts) DIRFLG=0 (vorwärts)
DIRFLG =0
nein (rückwärts)
ja (vorwärts)
Berechne gewünschte Motorklemmen spannung VT= +VR
Berechne gewünschte Motorklemmen spannung VT = -VR
VT > VTLIMP
ja
nein
VT= VTLIMP
VT< VTLIMN
ja
NEIN
VT= VTLIMN
Finde Zündwinkel (FINVAL) aus Tabelle, um gewünschte Klemmenspanng. zu ber. Wertetabelle, berechnet aus d.Zusammenhang FINVAL (jeder Tabelleneinsprung)
245.8COs -1
3 VT
π V LN
E

FIG.19

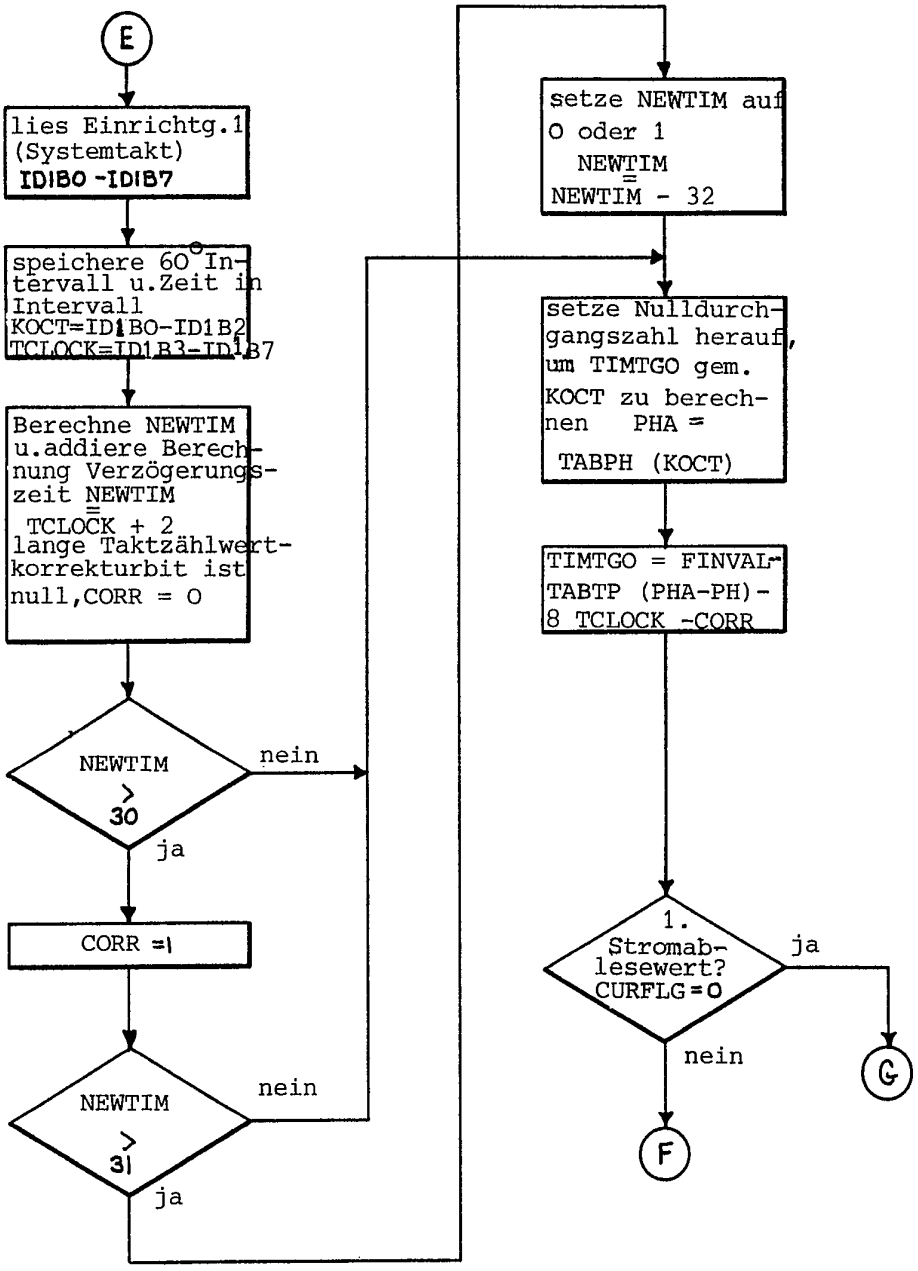

E
lies Einrichtg.1 (Systemtakt) IDIBO -IDIB7
speichere 60 Intervall u.Zeit in Intervall KOCT=ID1BO-ID1B2 TCLOCK=ID1B3-ID1B7
Berechne NEWTIM u.addiere Berechnung Verzögerungszeit NEWTIM = TCLOCK + 2 lange Taktzählwertkorrekturbit ist null,CORR = O
NEWTIM > 30
nein
ja
CORR =1
NEWTIM > 31
nein
ja
setze NEWTIM auf O oder 1 NEWTIM = NEWTIM - 32
setze Nulldurchgangszahl herauf, um TIMTGO gem. KOCT zu berechnen PHA = TABPH (KOCT)
TIMTGO = FINVAL-TABTP (PHA-PH)-8 TCLOCK -CORR
1. Stromablesewert? CURFLG=O
ja
nein
G
F

FIG.20

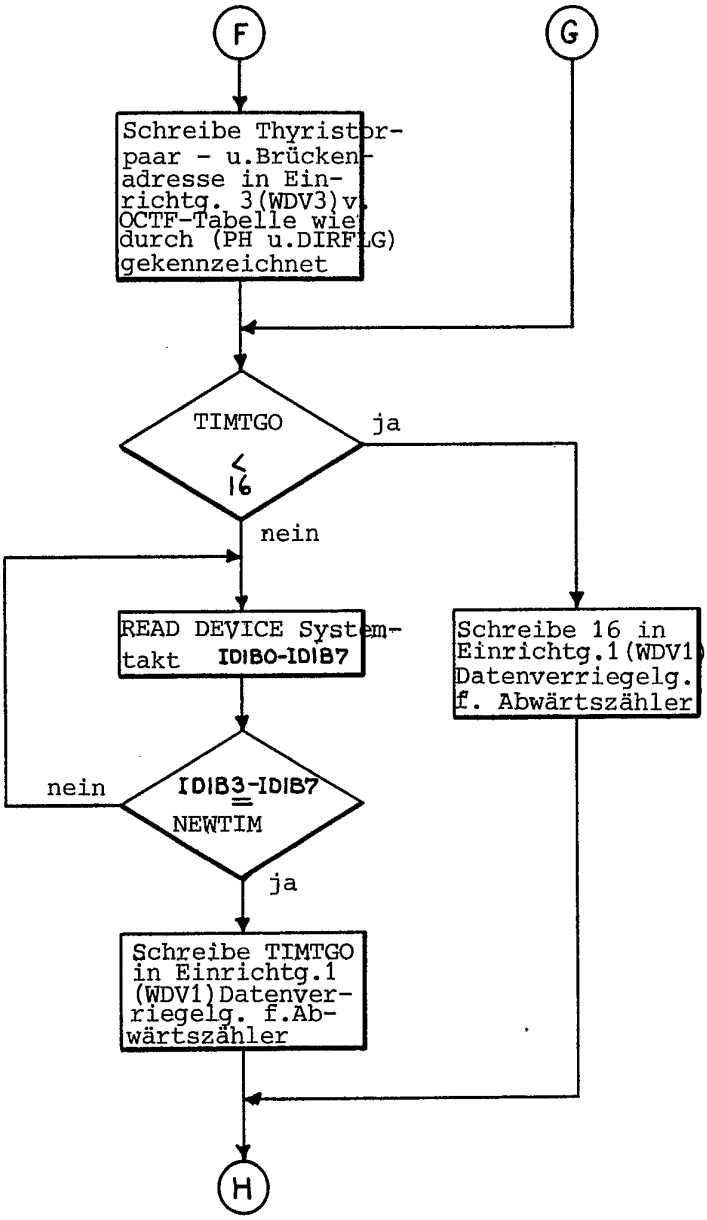

F
G
Schreibe Thyristorpaar - u.Brückenadresse in Einrichtg. 3(WDV3)v. OCTF-Tabelle wie durch (PH u.DIRFLG) gekennzeichnet
TIMTGO
< 16
ja
nein
READ DEVICE Systemtakt IDIBO-IDIB7
Schreibe 16 in Einrichtg.1(WDV1) Datenverriegelg. f. Abwärtszähler
IDIB3-IDIB7 = NEWTIM
nein
ja
Schreibe TIMTGO in Einrichtg.1 (WDV1)Datenverriegelg. f.Abwärtszähler
H

FIG. 21

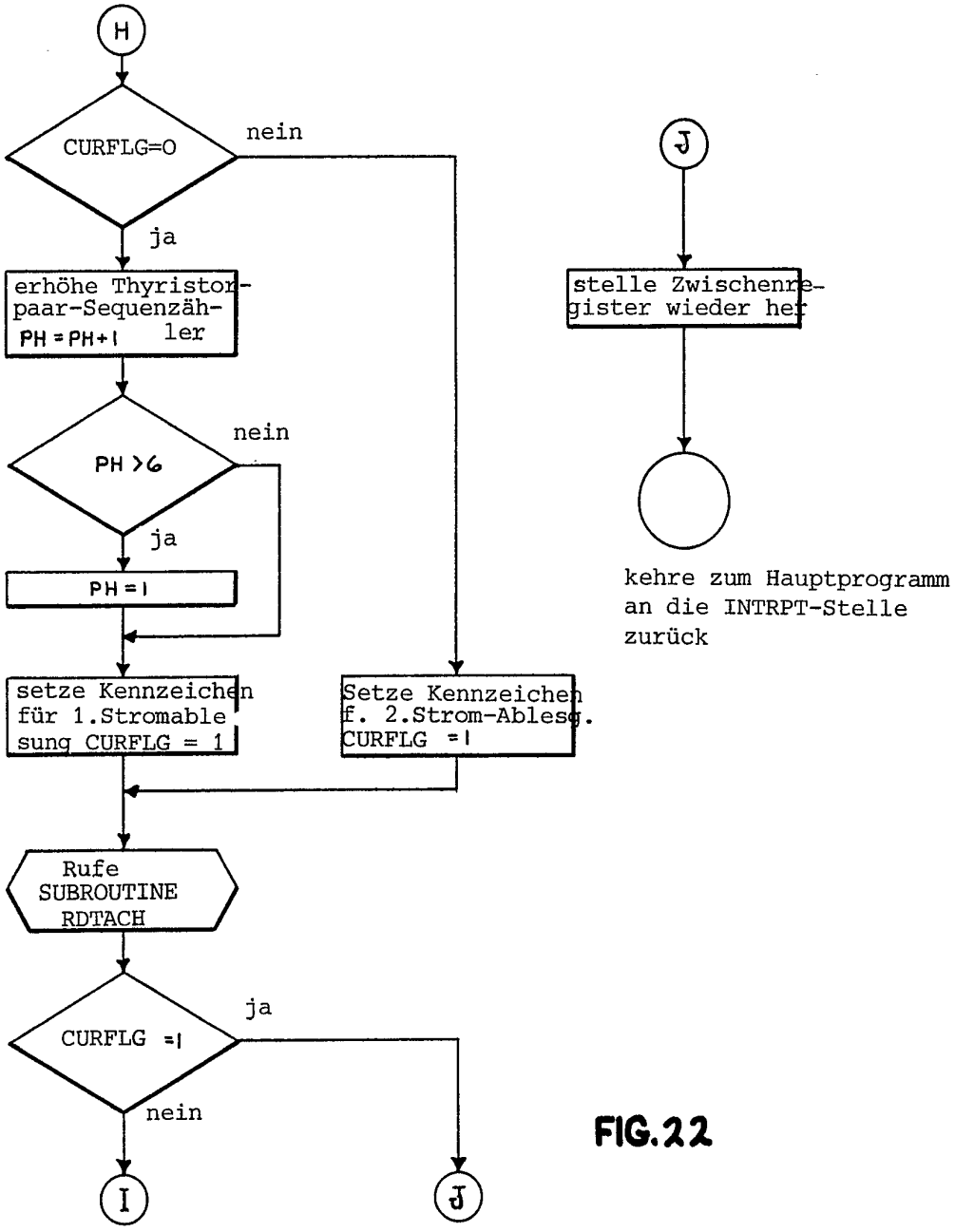

H
CURFLG=O
nein
ja
erhöhe Thyristor-paar-Sequenzäh-ler
PH = PH+1
PH >6
nein
ja
PH = 1
setze Kennzeichen für 1.Stromable sung CURFLG = 1
Setze Kennzeichen f. 2.Strom-Ablesg. CURFLG = 1
Rufe SUBROUTINE RDTACH
CURFLG = 1
ja
nein
I
J
J
stelle Zwischenre-gister wieder her
kehre zum Hauptprogramm an die INTRPT-Stelle zurück

FIG. 22

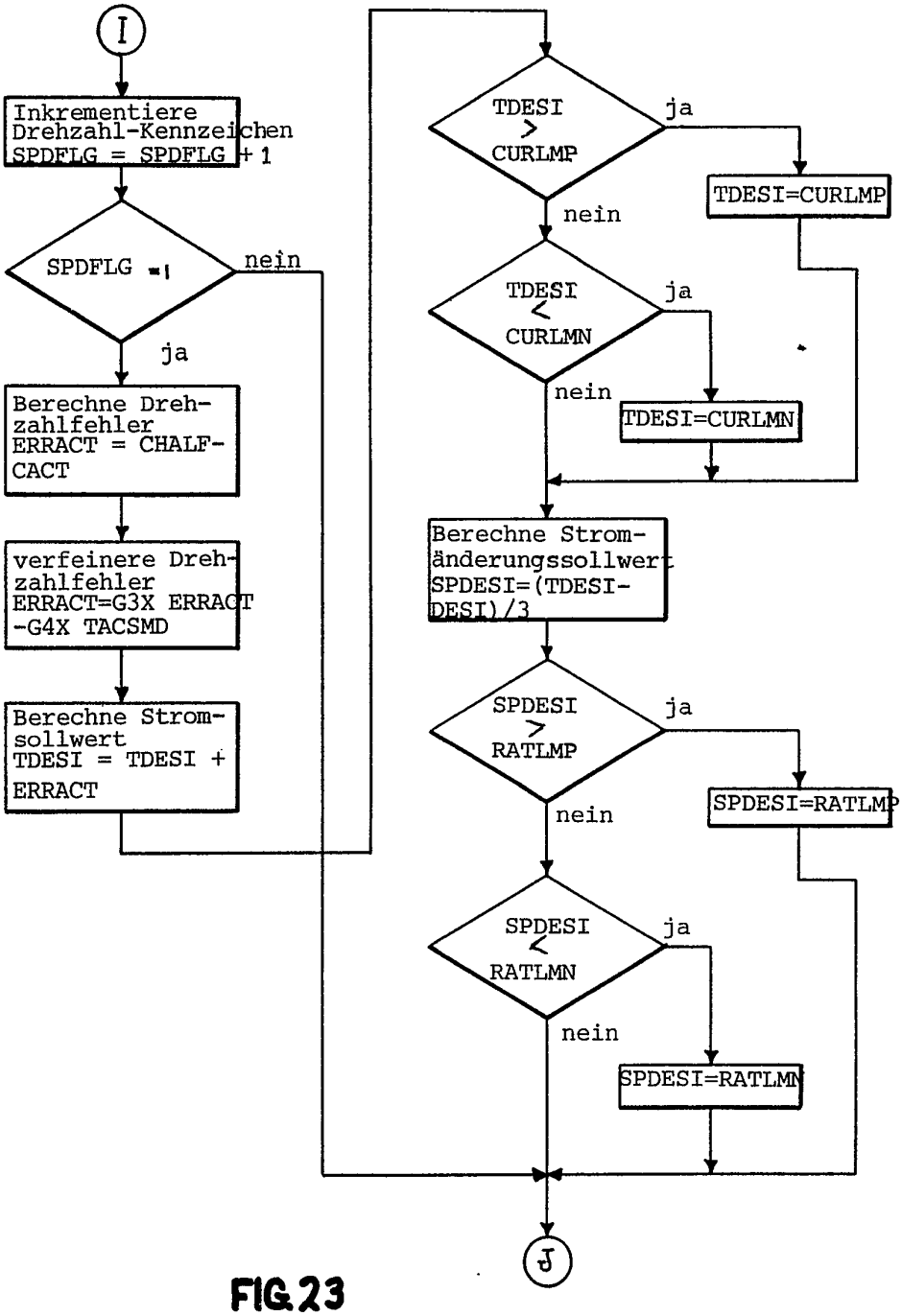

I
Inkrementiere Drehzahl-Kennzeichen SPDFLG = SPDFLG + 1
SPDFLG =1
nein
ja
Berechne Drehzahlfehler ERRACT = CHALF-CACT
verfeinere Drehzahlfehler ERRACT=G3X ERRACT -G4X TACSMD
Berechne Stromsollwert TDESI = TDESI + ERRACT
TDESI > CURLMP
ja
TDESI=CURLMP
nein
TDESI < CURLMN
ja
TDESI=CURLMN
nein
Berechne Stromänderungssollwert SPDESI=(TDESI-DESI)/3
SPDESI > RATLMP
ja
SPDESI=RATLMP
nein
SPDESI < RATLMN
ja
SPDESI=RATLMN
nein
J

FIG.23

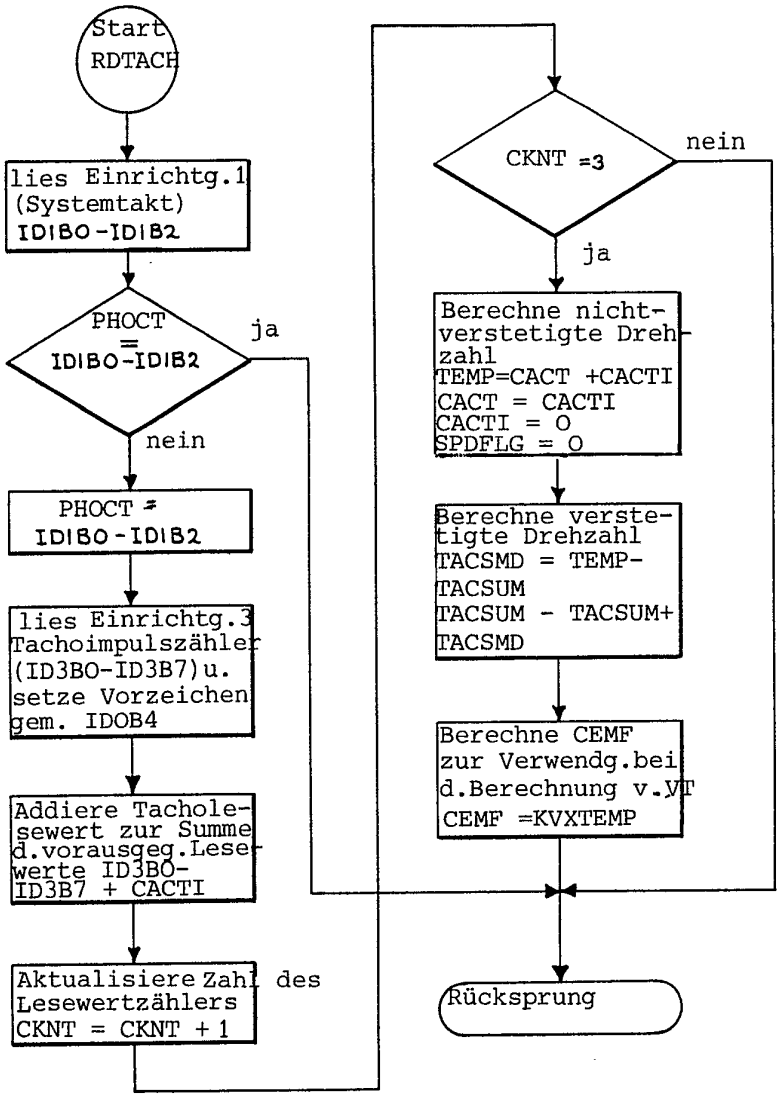

Start RDTACH
lies Einrichtg.1 (Systemtakt) IDIBO-IDIB2
PHOCT = IDIBO-IDIB2
ja
nein
PHOCT = IDIBO-IDIB2
lies Einrichtg.3 Tachoimpulszähler (ID3BO-ID3B7) u. setze Vorzeichen gem. IDOB4
Addiere Tachole-sewert zur Summe d.vorausgeg.Lese-werte ID3BO-ID3B7 + CACTI
Aktualisiere Zahl des Lesewertzählers CKNT = CKNT + 1
CKNT =3
nein
ja
Berechne nicht-verstetigte Dreh-zahl TEMP=CACT +CACTI CACT = CACTI CACTI = O SPDFLG = O
Berechne verste-tigte Drehzahl TACSMD = TEMP-TACSUM TACSUM - TACSUM+ TACSMD
Berechne CEMF zur Verwendg.bei d.Berechnung v.VT CEMF =KVXTEMP
Rücksprung

FIG.24