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

(10) **DE 602 23 555 T2** 2009.06.10

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 421 704 B1**
(21) Deutsches Aktenzeichen: **602 23 555.3**
(86) PCT-Aktenzeichen: **PCT/US02/27462**
(96) Europäisches Aktenzeichen: **02 759 483.7**
(87) PCT-Veröffentlichungs-Nr.: **WO 2003/021800**
(86) PCT-Anmeldetag: **29.08.2002**
(87) Veröffentlichungstag der PCT-Anmeldung: **13.03.2003**
(97) Erstveröffentlichung durch das EPA: **26.05.2004**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **14.11.2007**
(47) Veröffentlichungstag im Patentblatt: **10.06.2009**

(51) Int Cl.[8]: ***H04B 1/16*** (2006.01)
G06F 1/32 (2006.01)

(30) Unionspriorität:
315655 P **29.08.2001** **US**

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:
Mediatek Inc., Hsinchu, TW

(74) Vertreter:
Kuhnen & Wacker Patent- und Rechtsanwaltsbüro, 85354 Freising

(72) Erfinder:
SOERENSEN, Joern, DK-9600 Aars, DK; ALLEN, Michael, Austin, TX 78750, US; ANAND, Hitesh, Durham, NC 27007, US

(54) Bezeichnung: **VERFAHREN UND APPARAT ZUR TAKT- UND LEISTUNGSSTEUERUNG IN DRAHTLOSEN SYSTEMEN**

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

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

Beschreibung

GEBIET DER ERFINDUNG

[0001] Die vorliegende Erfindung betrifft das Gebiet der drahtlosen Kommunikation und genauer Takt- und Leistungs- bzw. Clock-and-Power-Regelung in drahtlosen Systemen.

HINTERGRUND DER ERFINDUNG

[0002] Mit der rapiden Weiterentwicklung drahtloser Kommunikationssysteme geht häufig die Erzeugung von neuen Standards für die drahtlose Kommunikation einher, mit denen veraltete Standards ersetzt werden. Es braucht jedoch häufig Zeit, um ein neues drahtloses Netzwerk auf der Basis eines neuen Standards über ein großes räumliches Gebiet zu implementieren. Somit ist es häufig erwünscht, über ein drahtloses Endgerät zu verfügen, das mit vorhandenen drahtlosen Netzwerken ebenso wie mit neuen drahtlosen Netzwerken kommunizieren kann. Wegen des rapiden Wachstums der drahtlosen Computer-Datennetzwerke ist es außerdem häufig erwünscht, über ein drahtloses Endgerät zu verfügen, das mit diesen Netzwerken kommunizieren kann, damit ein Nutzer im Internet browsen oder E-Mails versenden und empfangen kann. Darüber hinaus kann es nützlich sein, gleichzeitig mit unterschiedlichen drahtlosen Systemen zu kommunizieren, so dass ein Nutzer beispielsweise E-Mails in einen drahtlosen Datennetz durchsehen kann, während er ein Telefongespräch in einem drahtlosen 2G-Netzwerk führt.

[0003] Solche drahtlosen Systeme verwenden häufig unterschiedliche Zeitbasen. Beispielsweise verwendet das 2G GSM-Netz eine Zeitbasis, wo Rahmen eine Dauer von 4,615 Millisekunden haben und in 8 Zeitschlitze geteilt sind. Jedoch verwendet 3G WCDMA-Netze eine Zeitbasis, wo Rahmen eine Dauer von 10 Millisekunden haben und in 15 Zeitschlitze geteilt sind. Ereignisse im mobilen Endgerät müssen zeitlich exakt festgelegt und mit Bezug auf jedes der drahtlosen Systeme synchronisiert werden, unabhängig davon, ob das mobile Endgerät mit einem drahtlosen System oder gleichzeitig mit zwei oder mehr drahtlosen Systemen operiert.

[0004] Damit man sie mit sich tragen kann, werden drahtlose Endgeräte in der Regel durch Batterien mit Leistung bzw. Energie versorgt, wobei das Wiederaufladungsintervall eine Umkehrfunktion des Stromverbrauchs ist. Da der Nutzer im Stande sein soll, das drahtlose Endgerät bis zur nächsten Wiederaufladung so lange wie möglich zu betreiben, ist das Energie- bzw. Power-Management ein wichtiger Gesichtspunkt.

[0005] EP 0939495 A1 beschreibt Energiesparverfahren für tragbare elektronische Geräte. **Fig.** 2 zeigt ein tragbares Funkkommunikationsgerät, das ein Basisband IC **109**, ein Power-Management IC **121**, einen 13 MHz-Oszillator **106** und einen 32 kHz-Oszillator **134** aufweist. Der Oszillator **106** versorgt den Prozessor **109**, aber nicht den Prozessor **121**. Der Prozessor **121** benötigt den 32 kHz-Taktgeber immer, um das Power-Management durchführen zu können. US 5,592,173 offenbart einen GPS-Empfänger mit einem Normalbetriebsmodus, um GPS-Satellitensignale zu verarbeiten, und mit einem Standby-Modus mit niedrigem Energieverbrauch. Im Standby-Modus ist die Betriebsleistung in der GPS-Antenne und im GPS-Frequenz-Abwärtsumsetzer blockiert, der Systemtaktgeber im Digitalverarbeitungssystem ist blockiert, und der Mikrorechner-Taktgeber im Mikroprozessorsystem ist blockiert.

ZUSAMMENFASSUNG DER ERFINDUNG

[0006] Gemäß einem ersten Aspekt der Erfindung wird ein Basisbandprozessor für drahtlose Anwendungen geschaffen, der aufweist:
ein erstes Modul, das ein erstes Taktsignal benötigt, das von einem Systemoszillator abgeleitet wird, wobei das erste Modul ein erstes Indikatorsignal liefert;
ein zweites Modul, das ein zweites Taktsignal benötigt, wobei das zweite Modul
ein zweites Indikatorsignal liefert; und
eine Power-Management-Schaltung;
dadurch gekennzeichnet, dass
das erste Indikatorsignal einen erstes Zustand aufweist, der anzeigt, dass das erste Taktsignal benötigt wird, und einen zweiten Zustand, der anzeigt, dass das erste Taktsignal nicht benötigt wird;
das zweite Taktsignal einen ersten Zustand aufweist, der anzeigt, dass das zweite Taktsignal benötigt wird, und einen zweiten Zustand, der anzeigt, dass das zweite Taktsignal nicht benötigt wird;
die Power-Management-Schaltung spricht auf die ersten und zweiten Indikatorsignale an, um den Systemoszillator zu deaktivieren, wenn das erste Indikatorsignal den zweiten Zustand aufweist und das zweite Indikatorsignal den zweiten Zustand aufweist; und

die Power-Management-Schaltung spricht auf ein Aktivierungssignal an, tun den Systemoszillator zu aktivieren.

[0007] Gemäß einem weiteren Aspekt der Erfindung wird ein Verfahren geschaffen, das den Betrieb eines Digitalbasisband-Prozessors betrifft. Das Verfahren umfasst den Empfang eines ersten Indikatorsignals von einem ersten Modul, das ein erstes Taktsignal benötigt, das von einem Systemoszillator abgeleitet ist, wobei das erste Indikatorsignal einen ersten Zustand aufweist, der anzeigt, dass das erste Taktsignal benötigt wird, und einen zweiten Zustand, der anzeigt, dass das erste Taktsignal nicht benötigt wird; den Empfang eines zweiten Indikatorsignals von einem zweiten Modul, das ein zweites Taktsignal benötigt, das vom Systemoszillator abgeleitet wird, wobei das zweite Indikatorsignal einen ersten Zustand aufweist, der anzeigt, dass das zweite Taktsignal benötigt wird, und einen zweiten Zustand, der anzeigt, dass das zweite Taktsignal nicht benötigt wird, das Inaktivieren des Systemoszillators, wenn das erste Indikatorsignal den zweiten Zustand aufweist und wenn das zweite Indikatorsignal den zweiten Zustand aufweist, und das Aktivieren des Systemoszillators, wenn die Systemoszillatorleistung von mindestens einer Gruppe benötigt wird, welche das erse Modul und das zweite Modul umfasst.

KURZE BESCHREIBUNG DER ZEICHNUNG

[0008] Die Zeichnung:

[0009] **Fig.** 1 ist ein Blockschema eines Kommunikationsprozessors gemäß einer Ausführungsform der Erfindung;

[0010] **Fig.** 2 ist ein Blockschema eines Zeit- und Ereignis- bzw. TEP-Prozessors im Kommunikationsprozessor von **Fig.** 1 gemäß einer Ausführungsfom der Erfindung;

[0011] **Fig.** 3 ist ein Blockschema einer Ablaufsteuerung im TEP-Prozessor von **Fig.** 2 gemäß einer Ausführungsform der Erfindung;

[0012] Fig. 4 ist ein Beispiel für ein Befehlsformat, das zur Verwendung in einer Ablaufsteuerung gemäß einer Ausführungsform der Erfindung geeignet ist;

[0013] Fig. 5 ist eine Skizze, welche die Funktionen des TEP-Prozessors gemäß einer Ausführungsform der Erfindung darstellt;

[0014] **Fig.** 6 ist ein Blockschema einer Schnittstelle zwischen einem Direktspeicherzugriffs-Controller und dem TEP-Prozessor gemäß einer Ausführungsform der Erfindung;

[0015] Fig. 7 ist ein Blockschema, das ein Beispiel für ein Verfahren für Direktspeicherzugriffs-Transfers mittels des TEP-Prozessors gemäß einer Ausführungsform der Erfindung zeigt;

[0016] **Fig.** 8 ist ein Blockschema einer Schnittstelle eines externen Busses mit dem TEP-Prozessor gemäß einer Ausführungsform der Erfindung;

[0017] Fig. 9A ist ein Blockschema eines Beispiels für einen Bruch-N-Taktteiler gemäß einer Ausführungsform der Erfindung;

[0018] Fig. 9B ist ein Schema, das eine kalibierte Takt- und Phasenkompensation zeigt, die vom Bruch-N-Taktteiler gemäß einer Ausführungsform der Erfindung erzeugt wird;

[0019] Fig. 10 ist ein Blockschema eines Absolutzählers und von konfigurierbaren Periodic-Trigger-Generatoren gemäß einer Ausführungsform der Erfindung;

[0020] Fig. 11A ist ein Blockschema einer Snapshot-Zustandsmaschine gemäß einer Ausführungsform der Erfindung;

[0021] Fig. 11B ist ein Ablaufschema des Betriebs der Snapshot-Zustandsmaschine von Fig. 11A gemäß einer Ausführungsform der Erfindung;

[0022] **Fig.** 12 ist ein Blockschema eines Clock-and-Power-Regelungsmoduls im TEP-Prozessor von **Fig.** 2

gemäß einer Ausführungsform der Erfindung;

[0023] Fig. 13A ist ein Schema eines Takterzeugungsmoduls gemäß einer Ausführungsform der Erfindung;

[0024] Fig. 13B ist ein Schema eines Taktverteilungsmoduls gemäß einer Ausführungsform der Erfindung;

[0025] Fig. 14A ist ein schema eines Taktgattermoduls gemäß einer Ausführungsform der Erfindung;

[0026] Fig. 14B ist eine Tabelle, die den Inhalt der Register von Fig. 14A zeigt, gemäß einer Ausführungsform der Erfindung; und

[0027] Fig. 15 ist ein Blockschema, das Oszillatorausschaltanforderungen gemäß einer einer Ausführungsform der Erfindung darstellt.

AUSFÜHRLICHE BESCHREIBUNG

[0028] Ein drahtloses Endgerät kann eine Funkeinheit, einen Digitalbasisband-Prozessor, eine Nutzerschnittstelle und eine Batterie einschließen. Der Basisbandprozessor kann einen Digitalsignalprozessor zum Ausführen von Signalverarbeitungsalgorithmen und anderen komplexen Berechnungen und einen Mikrocontroller zur Ausführung von Steuerfunktionen und relativ einfachen Berechnungen einschließen. Viele der Aufgaben, die von Basisbandprozessoren in drahtlosen Endgeräten ausgeführt werden, erfordern eine präzise Zeitsteuerung. Beispielsweise werden in einem drahtlosen Kommunikationsnetz Aktionen in einem drahtlosen Kanal zeitlich so geplant, dass sie mit einer festgelegten Präzision zu bestimmten Zeiten stattfinden. Ein eigener Zeitsteuerungs- und Ereignisprozessor (TEP) kann verwendet werden, um eine solche Zeitsteuerungspräzision zu erreichen. Beispielsweise kann der TEP dafür zuständig sein, Zeitsteuerungssignale zu erzeugen, Ereignisse zeitlich zu planen, Unterbrechungssignale bzw. Interrupts für Prozessoren zu erzeugen, Operationen in anderen Modulen zu initiieren und Steuersignale für Off-Chip-Schaltungen, wie die Funkeinheit, zu erzeugen. Der TEP kann mit dem Digitalsignalprozessor, dem Mikrocontroller und anderen Komponenten des Basisbandprozessors zusammenarbeiten, um alle Zeitsteuerungen und Ereignisse im drahtlosen Endgerät zu steuern.

[0029] Manchmal ist es gewünscht, mit mehreren unterschiedlichen drahtlosen Systemen gleichzeitig zu kommunizieren. Beispielsweise kann der Kommunikationsprozessor mit einem drahtlosen Datennetz, wie einem Bluetooth-Netz, kommunizieren, um ein E-Mail-Konto eines Nutzers nach neuen E-Mails zu durchsuchen, während er gleichzeitig den Paging-Kanal eines drahtlosen CDMA-Netzes nach neuen Sprachtelefonanrufen durchsucht. Häufig verwenden die unterschiedlichen drahtlosen Systeme, mit denen der Prozessor kommuniziert, verschiedene Zeitbasen. Der TEP kann Ereignisse für drahtlose Systeme zeitlich planen, indem er einen gemeinsamen Bezugstakt als Zeitbasis zum zeitlichen Planen von Ereignissen für eines der drahtlosen Systeme, mit denen der Kommunikationsprozessor kommuniziert, verwendet.

[0030] Ein Blockschema eines Basisband-Kommunikationsprozessors **100** gemäß einer Ausführungsform der Erfindung ist in **Fig.** 1 dargestellt. Der in **Fig.** 1 dargestellte Prozessor weist zwei Processing-Cores auf. Ein Digitalsignalprozessor (DSP)-Core **102** kann verwendet werden, um Digitalsignalverarbeitungsfunktionen des Kommunikationsprozessors **100** durchzuführen, wie Verarbeitungen, die mit Zellsuche, Signalkorrelation und Kanal-Codierung und -Decodierung zusammenhängen. Viele andere Signalverarbeitungsfunktionen können vom DSP-Core **102** durchgeführt werden. Ein Beispiel für einen DSP-Core, der sich zur Verwendung in der vorliegenden Erfindung eignet, ist in der PCT-Veröffentlichung Nr. WO 00/687783, veröffentlicht am 16. November 2000, offenbart. Es sei jedoch darauf hingewiesen, dass viele andere Arten von Digitalsignalprozessoren verwendet werden können, und dass die Erfindung nicht auf irgendeinen bestimmten Digitalsignalprozessor beschränkt ist. Ein Processing-Core einer Mikrocontroller-Einheit (MCU) **104** kann verwendet werden, um einen Steuer-Code für den Kommunikationsprozessor **100** auszuführen, beispielsweise Protokollstapelanweisungen. Ein Beispiel für eine im Handel erhältliche MCU, die sich zur Verwendung in der vorliegenden Erfindung eignet, ist der ARM/TDMI-Core, verkauft von Advanced RISC Machines, Ltd. Es sei jedoch darauf hingewiesen, dass viele andere Arten von Mikrocontrollern verwendet werden können, und die Erfindung ist nicht auf einen bestimmten Mikrocontroller beschränkt.

[0031] Der Kommunikationsprozessor **100** schließt auch einen Systemspeicher **106** ein. Der Systemspeicher **106** kann ein statischer Speicher mit wahlfreiem Zugriff (static random access memory, SRAM) oder jede andere Art von flüchtigem oder nichtflüchtigem Speicher sein, wie ein dynamischer Speicher mit wahlfreiem Zugriff (DRAM), ein synchroner dynamischer Speicher mit wahlfreiem Zugriff (SDRAM) oder ein ferroelektrischer Speicher mit wahlfreiem Zugriff (FRAM). Der DSP-Core **102** und die MCU **104** verwenden ein gemeinsames

Speicherkennfeld. Daher können sich diese Prozessoren einen Zugriff auf den Systemspeicher **106** teilen und können über den Systemspeicher **106** miteinander kommunizieren.

[0032] Jede der in **Fig.** 1 dargestellten Komponenten kann als einzelne integrierte Schaltung oder als mehrere integrierte Schaltungen implementiert werden. In einigen Ausführungsformen wird der gesamte Kommunikationsprozessor **100** auf einem einzigen Chip gefertigt. Es sei klargestellt, dass die Erfindung in dieser Hinsicht nicht beschränkt ist.

[0033] Direktspeicherzugriffs-(DMA-)Controller **134** und **136** sind vorgesehen, um die Datenübertragung im Kommunikationsprozessor **100** zu erleichtern. Die DMA-Controller **134** und **136** ermöglichen eine direkte Speicherübertragung zwischen Geräten und Speicher (z. B. dem Systemspeicher **106**) ohne eine Intervention des Prozessors. Den Geräten können DMA-Kanäle zugeordnet sein, damit diese Geräte DMA-Übertragungen bzw. -Transfers anfordern können. Die Konfiguration der Kanäle kann vom DSP-Core **102** und der MCU **104** bestimmt werden. Obwohl beide Prozessoren auf jeden DMA-Kanal zugreifen können, kann die Kanalkonfiguration für eine Gruppe von Kanälen vom DSP-Core **102** gesteuert werden und die Kanalkonfiguration für die andere Gruppe von Kanälen kann vom MCU-Core **104** gesteuert werden. Ebenso kann der DMA-Controller **134** DMA-Transfers für Kanäle steuern, die vom DSP-Core **102** konfiguriert werden, während der DMA-Controller **136** DMA-Transfers für Kanäle steuern kann, die von der MCU **104** gesteuert werden.

[0034] Der DSP-Core **102** kann einen Level 1 (L1)-Befehls-Cache **144** und einen L1-Daten-Cache **146** aufweisen, um für eine geringe Latenz beim Zugriff auf im Cache gespeicherte Daten zu sorgen. Der DSP-Core **102** kann zwei Datenbusse aufweisen, die mit dem L1-Daten-Cache **146** verbunden sind, einen Befehlsbus, der mit dem L1-Befehls-Cache **144** verbunden ist, und einen DMA-Bus, der mit den Caches **144** und **146** verbunden ist. Ein Level 2 (L2)-Speicher **148** kann als SRAM zur Verwendung durch den DSP-Core **102** gewidmet sein. Der Speicher **148** kann vom DMA-Controller **134** aus zugänglich sein. Auf den Speicher **148** kann auch von der MCU **104**, dem DMA-Controller **136** und der Verarbeitungsschnittstelle für externe Anwendungen (External Application Processing Interface, EAPI) **142** zugegriffen werden.

[0035] Eine Systembus-Schnittstelleneinheit (System Bus Interface Unit, SBIU) **132** führt Verbrückungsfunktionen aus. Beispielsweise kann die SBIU **132** als asymmetrischer Kreuzschienenschalter dienen, der Anfragen vom DSP-Core **102**, vom DMA-Controller **134**, von der MCU **104**, vom DMA-Controller **136** und von der EAPI **142** zu geeigneten Systemressourcen, wie einem L1-Cache **144**, einem L1-Cache **146**, einem L2-Speicher **148** und anderen Systemressourcen, routet. Die SBIU **132** ermöglicht einen parallelen und gleichzeitigen Datentransfer zwischen verschiedenen Bussen.

[0036] Mensch/Maschinen-Schnittstellen-(Man-Machine Interface, MMI-)Module **150** sorgen für Hardware-Nutzer-Schnittstellen mit dem Kommunikationsprozessor **100** und sind über den PBUS-Bus **128** zugänglich. Die Module der MMI **150** können eine Schnittstelle mit Mehrzweck- bzw. General-Purpose-I/O (GPIO)-Anschlussstiften des Kommunikationsprozessors **100** aufweisen. Solche Anschlussstifte können für verschiedene Zwecke verwendet werden, einschließlich der Schnittstellenbildung mit einer Funkeinheit und anderen externen Geräten. Andere MMI-Module können eine Schnittstelle für einen Anzeigebildschirm, eine Schnittstelle für einen seriellen Port, eine Schnittstelle für einen universellen asynchronen Empfänger/Sender (universal asynchronous receiver transmitter, UART), eine USB-Schnittstelle und ein Teilnehmeridentitätsmodul (subscriber identiy module, SIM) aufweisen, das eine einzigartige Seriennummer des drahtlosen Endgeräts, in das der Kommunikationsprozessor **100** eingebettet ist, enthält. In der MMI **150** können viele andere Schnittstellenmodule enthalten sein.

[0037] Housekeeping- bzw. Organisationsmodule führen verschiedene Organisationsfunktionen für den Kommunikationsprozessor **152** aus und sind über den PBUS-Bus **128** zugänglich. Solche Funktionen schließen einen Überwachungszeitgeber (WDT), der abläuft und einen Reset erzeugt, wenn es zu gegenseitigen Software-Blockierungen im Kommunikationsprozessor kommt, falls diese nicht in Ordnung gebracht werden, allgemeine Zeitnehmer, die verwendet werden können, um Trigger für eine Mehrzweck-Zeitgeberfunktionalität zu erzeugen, und einen IRQ-Controller, um Interrupts für den DSP-Core **102** und die MCU **104** zu verwalten.

[0038] Drahtlose Systemmodule **154** sorgen für Schnittstellen mit Komponenten des drahtlosen Systems, die sich außerhalb des Kommunikationsprozessors **100** befinden und auf die über den PBUS-Bus **128** zugegriffen werden kann. Beispielsweise können drahtlose Systemmodule **154** einen CSport, bei dem es sich um eine Steuerungsschnittstelle eines seriellen Ports mit einem analogen Basisband-Chip handelt, und eine Schnittstelle mit einem Frequenzaufbereiter aufweisen.

[0039] DSP-Peripheriegeräte (ihren verschiedene Digitalsignal-Verarbeitungsfunktionen in Zusammenarbeit mit einem DSP-Core **102** aus und sind über den DPBUS-Bus **110** zugänglich. DSP-Peripheriegeräte können beispielsweise eine Coprozessor-Schnittstelle **162**, einen BSport **164**, ein Flag I/O **166**, eine Hochgeschwindigkeits-Registrierungseinrichtung **168**, eine Schlüsselmaschine **170** und einen DSP IRQ-Controller **172** einschließen.

[0040] Daten können zwischen verschiedenen Komponenten des Kommunikationsprozessors und zwischen dem Kommunikationsprozessor und Off-Chip-Geräten unter Verwendung eines oder mehrerer Busse übertragen werden. Jeder Bus kann ein paralleler oder ein serieller Bus sein. Außerdem kann jeder Bus unidirektional oder bidirektional sein. Darüber hinaus kann jeder Bus einen Adressbus und/oder einen Datenbus und/oder einen Steuerbus einschließen. Die Buskonfiguration des Kommunikationsprozessors **100**, der in **Fig.** 1 dargestellt ist, schließt mehrere Bussysteme ein. Die Funktion jedes Bussystems wird nachstehend im Allgemeinen beschrieben. Viele Variationen, Modifikationen und Verbesserungen der Buskonfiguration, die in **Fig.** 1 dargestellt ist, liegen für den Fachmann nahe und sind im Gedanken und Gebiet der Erfindung eingeschlossen.

[0041] Ein SYSL2-Bus **108** ist zwischen die SBIU **132** und die Schnittstellen mit dem L2-Speicher **148** geschaltet. Der Speicher **144** gehört sowohl zur MCU **104**, zum System-DMA-Controller **136** als auch zum DSP **102**. Ein DPBUS-Bus **110** ist der periphere DSP-Bus und bildet eine Schnittstelle mit verschiedenen DSP-Peripheriegeräten, wie einem BSPort **164**, bei dem es sich um einen seriellen Basisbandport handeln kann, einer Coprozessor-Schnittstelle **162**, einem Flag I/O **166**, einer Hochgeschwindigkeits-Registriereinrichtung **168**, einer Schlüsselmaschine **170** und einem DSP IRQ-Controller **172**. Den Zugriff auf den DPBUS-Bus **110** teilen sich die MCU **104**, der System-DMA-Controller **136** und der DSP DMA-Controller **134**. Der DSP-Core **102** kann auch über die SBIU **132** auf den DPBUS-Bus **110** zugreifen. Ein DSPBUS-Bus **112** ist die Schnittstelle für einen DSP-Core **102** und den PBUS-Bus **128**, den Systemspeicher **106** und den EBUS-Bus **100**. Ein DABUS **114** dient als DSP DMA-Controller-Schnittstelle mit der SBIU **132**. Ein DMABUS-Bus **116** ist die Schnittstelle zwischen dem System-DMA-Controller **136** und Ressourcen am PBUS-Bus **128**, am RBUS-Bus **118** und am EBUS-Bus **120**. Ein RBUS-Bus **118** ist die Schnittstelle mit dem Systemspeicher **106**. Den Zugriff auf den RBUS-Bus **118** teilen sich die MCU **104**, der System-DMA-Controller **136**, der DSP DMA-Controller **134** und der DSP-Core **102**. Ein EBUS-Bus **120** dient als Schnittstelle mit einem FLASH-Speicher und einem SRAM, der sich außerhalb des Kommunikationsprozessors **100** befindet. Ein SBUS-Bus **122** ist der Hauptsystembus für die MCU **104**. Ein EAPI-Bus **124** dient als Schnittstelle mit den Ressourcen des Kommunikationsprozessors **100** von einem Anwendungsprozessor, der außerhalb des Kommunikationsprozessors **100** angeordnet ist. Ein EABUS-Bus **140** ist die Schnittstelle zwischen der EAPI **142** und einem Anwendungsprozessor, der sich außerhalb des Kommunikationsprozessors **100** befindet. Es sei darauf hingewiesen, dass es nicht nötig ist, einen externen Anwendungsprozessor vorzusehen. Ein CBUS-Bus **126** ist die Schnittstelle mit einem externen Coprozessor. Der PBUS-Bus **128** ist ein peripherer Bus, der eine Schnittstelle mit drahtlosen Peripheriegeräten **154**, Organisationsperipheriegeräten **152** und MMI-Peripheriegeräten **150** mit der MCU **104**, dem System-DMA-Controller **136**, dem DSP DMA-Controller **134** und dem DSP-Core **102** bildet.

[0042] Da sich mehrere Komponenten den Zugriff auf einige Busse teilen, beispielsweise den PBUS-Bus **128** und den RBUS-Bus **118**, sind Busvermittler **130a**, **130b** und **130c** vorgesehen, um den Zugriff auf diese Busse zu verwalten.

[0043] Der Kommunikationsprozessor **100** weist einen TEP-Prozessor **138** auf, der verwendet werden kann, um Ereignisse für den Kommunikationsprozessor **100** zeitlich zu planen. Solche Ereignisse können beispielsweise das Verbinden und Trennen von I/O-Pins, die Erzeugung von Interrupts für den DSP-Core **102** und die MCU **104** und die Initiierung von DMA-Speicherübertragungen zwischen TEP **138** und anderen Modulen des Kommunikationsprozessors **100** einschließen. Der TEP **138** wird mit anderen Modulen des Kommunikationsprozessors **10** über einen DPBUS-Bus **110** verbunden und ist auch mit einem DSP-DMA-Controller **134** und einem DSP IRQ-Controller **172** verbunden.

[0044] Im TEP **138** werden unterschiedliche drahtlose Systemzeitbasen in eine unifizierte Zeitbasis umgewandelt, die nicht spezifisch für irgendein drahtloses System ist. Ereignisse werden zeitlich anhand der unifizierten Zeitbasis als Trigger auf einen Absolutzeitpunkt geplant. Der TEP **138** erzeugt einen kalibrierten langsamen Takt als Bezug für die unifizierte Zeitbasis, indem er einen schnellen freilaufenden Hochpräzisionstakt als Kalibrierungsbezug nutzt, um eine Langzeitstabilität des kalibrierten langsamen Taktes zu erhalten. Der kalibrierte langsame Takt, der als Takt für die unifizierte Zeitbasis verwendet wird, wird durch Herausnehmen von Taktimpulsen ais einem freilaufenden langsamen Takt erhalten. Dies führt einen Phasenfehler ein, der kompensiert wird, um präzise Zeitsteuerungssignale zu erhalten. Eine Phasenkompensation wird für jeden Taktzyklus des freilaufenden langsamen Taktes berechnet. Die Phasenkompensation wird als Zahl der Taktzyklen

des freilaufenden schnellen Taktes ausgedrückt und wird, zusammen mit dem kalibrierten langsamen Takt, verwendet, um eine exakte Zeitsteuerung zu liefern. Ein Merkmal ist, dass der Phasenkompensationswert auch beibehalten wird, obwohl der schnelle Takt ausgeschaltet wird. Diese Merkmale werden nachstehend ausführlich erläutert.

[0045] Fig. 5 ist eine schematische Darstellung von Beispielen der Funktionen des TEP **138**. Der TEP **138** kann GPIO-Pins bzw. -Anschlussstifte verbinden und trennen, um Schnittstellen mit externen Geräten zu steuern. Der TEP **138** kann auch mit dem System-DMA-Controller **136** und dem DSP DMA-Controller **134** kommunizieren, um DMA-Kanäle zu aktivieren. Unter Verwendung eines gewidmeten DMA-Kanals **516** kann der TEP **138** auch aus einem beliebigen Speicherabbildungsort lesen und in diesen schreiben, wodurch der TEP **138** mit anderen Modulen, wie beispielsweise einem drahtlosen System **154**, kommunizieren kann, um den Frequenzaufbereiter unter Verwendung der Frequenzaufbereiter-Schnittstelle **154a** zu programmieren. Der TEP **138** kann eine Schnittstelle mit den DSP- und MCU IRQ-Controllers **506** bilden, um Interrupts für jeden Processing-Core zu erzeugen, wodurch die Processing-Cores in der Lage sind, den Leerlauf- bzw. Ruhezustand einzunehmen, wenn sie nicht gebraucht werden, und den Ruhezustand zu verlassen, wenn nötig, indem sie Interrupts vom TEP **138** empfangen. Alle TEP-Funktionen können präzise zeitlich gesteuert und geplant werden wie nachstehend beschrieben.

[0046] **Fig.** 2 ist ein Blockschema eines Beispiels für eine TEP-Architektur **138** gemäß einer Ausführungsform der Erfindung. Der TEP **138** kann als Zeitsteuerungs- und Laufzeitzuweisungsmechanismus für den Kommunikationsprozessor **100** dienen. Im Betrieb eines drahtlosen Systems finden alle Funksteuerungsereignisse zu geplanten Zeiten statt und erfordern eine exakte Zeitsteuerung. Zu bestimmten Zeiten während des Betriebs des Kommunikationsprozessors **100**, insbesondere bei einer drahtlosen Kommunikationsanwendung, kann es sein, dass weder die MCU **104** noch der DSP-Core **102** irgendwelche Verarbeitungsfunktionen durchführen müssen und in einen Leerlauf- oder „Schlaf"-Modus eingehen. In diesem Modus müssen die Processing-Cores nicht mehr getaktet werden, wodurch die Oszillatoren ausgeschaltet werden können. Der Kommunikationsprozessor **100** kann in ein drahtloses Endgerät eingebettet sein und durch eine Batterie mit Leistung versorgt werden. Die Energieeinsparung durch den Ruhezustand von Prozessoren oder das Ausschalten von Oszillatoren, wenn diese nicht gebraucht werden, kann die Zeit bis zum nächsten Wiederaufladen der Batterie verlängern. Bevor die Processing-Cores in den Ruhezustand gehen, können sie jedoch dem TEP **138** einen Zeitpunkt mitteilen, zu dem sie erneut gestartet werden müssen.

[0047] Der TEP **138** kann eine Vielzahl von Folgesteuereinrichtungen **202a**–**202n** aufweisen, die im Allgemeinen dazu dienen, Befehle, die vom TEP **138** genutzt werden, um zeitspezifische Aktionen durchzuführen, auszuführen. Der TEP **138** weist auch einen Speicher **206** auf, bei dem es sich beispielsweise um einen statischen RAM (SRAM) handeln kann. Die Folgesteuereinrichtungen **202a**–**202n** können den Speicher **206** nutzen, um Codes und Daten zu speichern. Ein Speicherzugriffsresolver **208** verarbeitet ankommende Speicherzugriffsanfragen von den Folgesteuereinrichtungen **202a**–**202n** und dem DPBUS-Bus **110**. Ein DPBUS-Bus-Schnittstellenmodul **210** sorgt für Verbrückungen zwischen dem Systemtaktgeber und den DPBUS-Bus-Taktdomänen innerhalb des TEP **138**. Das DPBUS-Bus-Schnittstellenmodul **210** wird nachstehend ausführlicher beschrieben. Der TEP **138** weist ferner einen Taktkalibrierungsblock **212** auf, der zur Taktkalibrierung der unifizierten Zeitbasis im TEP verwendet werden kann. Der Taktkalibrierungsblock **212** ist nachstehend ausführlicher erläutert. Der TEP **138** kann auch einen Absolutzähler **214** aufweisen, der von den Folgesteuereinrichtungen **202a**–**202n** für Zeitsteuerungszwecke genutzt werden kann. Der Absolutzähler **214** ist nachstehend ausführlicher beschrieben. Der TEP **138** kann einen Absolutzähler **214** zum Auflösen von miteinander in Konflikt stehenden Signalen, die von den Folgesteuereinrichtungen **202a**–**202n** empfangen werden, aufweisen. Ein I/O-Konfliktresolver **204** ist nachstehend ausführlicher beschrieben. Ein Takt- und Leistungssteuerungsblock **216**, der nachstehend ausführlicher erörtert wird, wird verwendet, um den Systemtaktgeber abzuschalten, wenn möglich.

[0048] Die Folgesteuereinrichtungen **202a**–**202n** können Prozessoren sein, wie RISC-Prozessoren, mit einem eigenen Befehlssatz und können für eine Zeitsteuerung mehrerer drahtloser Systeme gleichzeitig sorgen. Das heißt, die Folgesteuereinrichtungen **202a**–**202n** können Signale erzeugen, um GPIO-Pins zu verbinden oder zu trennen, Signal-DMA-Controller, und um Interrupts für den DSP-Core **102** und die MCU **104** zu erzeugen. Eine Folgesteuereinrichtung kann vorgesehen sein, um Befehle für jedes drahtlose System auszuführen, das gleichzeitig mit anderen ausgeführt werden soll. Eine verbesserte Leistung kann dadurch erhalten werden, dass man zwei oder mehr Folgesteuereinrichtungen für jedes drahtlose System vorsieht. Beispielsweise könne in einer Ausführungsform der Erfindung zwei Folgesteuereinrichtungen für jedes drahtlose System vorgesehen sein, das gleichzeitig mit anderen unterstützt werden soll. Bei dieser Konfiguration kann eine Folgesteuereinrichtung Befehle ausführen, während die andere Folgesteuereinrichtung mit Befehlen beladen wird. Es sei klar-

gestellt, dass eine einzelne Folgesteuereinrichtung in der Lage ist, mehrere drahtlose Systeme zu unterstützen. Eine einzelne Folgesteuereinrichtung kann mit Befehlen beladen werden, die sich auf zwei unterschiedliche drahtlose Systeme beziehen. Eine echte Gleichzeitigkeit kann jedoch mittels einer einzigen Folgesteuereinrichtung nicht erzielt werden, da eine Ausführungszeit für einen Befehl, der sich auf das erste drahtlose System bezieht, sich mit einer Ausführungszeit für einen anderen Befehl, der sich auf das zweite drahtlose System bezieht, überschneiden kann. Da diese Befehle in einer Sequenz durch eine einzige Folgesteuereinrichtung ausgeführt werden, können sie nicht gleichzeitig ausgeführt werden. Es sei jedoch klargestellt, dass es nicht notwendig ist, zwei Folgesteuereinrichtungen für jedes drahtlose System zu verwenden. Es kann eine Folgesteuereinrichtung pro drahtlosem System verwendet werden, oder es können drei oder mehr Folgesteuereinrichtungen für jedes drahtlose System verwendet werden. Eine zusätzliche Folgesteuereinrichtung, die nicht auf die Verarbeitung drahtloser Systeme bezogen ist, kann verwendet werden, um für eine Mehrzweckzeitsteuerung zu sorgen. Beispielsweise kann die zusätzliche Folgesteuereinrichtung verwendet werden, um Zeitsteuerungsereignisse zu planen, die mit dem Aktualisieren eines Taktgebers am Anzeigebildschirm des drahtlosen Endgeräts in Zusammenhang stehen. In einigen Ausführungsformen weist der TEP **138** zwei Folgesteuereinrichtungen für jedes drahtlose System auf, das gleichzeitig mit anderen unterstützt werden soll, sowie eine zusätzliche Folgesteuereinrichtung.

[0049] Durch die Verwendung von mehreren Folgesteuereinrichtungen kann der Kommunikationsprozessor **100** gleichzeitig mit mehreren verschiedenen drahtlosen Systemen kommunizieren. Beispielsweise kann ein drahtloses Endgerät den Paging-Kanal eines GSM-Netzes überwachen, während es gleichzeitig Daten von einem drahtlosen LAN, einem Bluetooth-Netz oder einem anderen 802.11b-Netz empfängt. Ebenso kann ein drahtloses Endgerät mit einem Kommunikationsprozessor mit mehreren Folgesteuereinrichtungen beim Starten gleichzeitig Zellsuchen sowohl für ein 2G GSM-Netz als auch ein 3G WCDMA-Netz durchführen.

[0050] Wie oben angegeben, kann TEP **138** GPIO-Pins verbinden und trennen, DMA-Kanäle aktivieren, Interrupts erzeugen und eine Taktkalibrierung durchführen. Jedoch können zwei oder mehr Folgesteuereinrichtungen miteinander in Konflikt stehende Signale ausgeben. Beispielsweise kann eine Folgesteuereinrichtung ein gesetztes Signal für einen bestimmten I/O-Stift ausgeben, während eine andere Folgesteuereinrichtung gleichzeitig ein Löschsignal für den gleichen Pin ausgeben kann. Der I/O-Konfliktresolver **204**, der in **Fig.** 2 dargestellt ist, verwaltet solche Konflikte. Eine Regel kann beispielsweise sein, dass jedes Trennsignal Vorrang vor einem Verbindungssignal hat. Eine Ausnahme kann erzeugt werden, um den Software-Prozess von dem Konflikt in Kenntnis zu setzen, und ein Interrupt kann an den Processing-Core gesendet werden. Für Interrupts und DMA-Kanalaktivierungen können miteinander in Konflikt stehende Signale einfach miteinander kombiniert werden, beispielsweise mittels einer logischen ODER-Operation.

[0051] Der in **Fig.** 2 dargestellte Speicher **206** kann über die DPBUS-Schnittstelle **210**, die eine Schnittstelle mit dem DPBUS-Bus **110** bildet, zugänglich sein. Gemäß einer Ausführungsform der Erfindung ist der Speicher **206** 26 Bits breit und weist mehrere Ports auf, um einen gleichzeitigen Zugriff durch die Folgesteuereinrichtungen **202a**–**202n** und den DPBUS-Bus **110** zu ermöglichen. Die Zahl der Lese- und Schreib-Ports, die für den Speicher **206** vorgesehen sind, kann auf der Basis der Zahl der Folgesteuereinrichtungen im TEP **138** gewählt werden. Beispielsweise können ein Lese-Port und ein Schreib-Port für jede Folgesteuereinrichtung vorgesehen sein. Jedoch verbraucht eine große Zahl von Ports Platz auf dem Chip und kann eine erhöhte Zahl von Befehls-Decodern benötigen. Alternativ dazu kann ein Befehls-Decoder für jede Folgesteuereinrichtung vorgesehen sein. Außerdem ist es nicht wahrscheinlich, dass alle Folgesteuereinrichtungen einen Speicherzugriff innerhalb des gleichen Taktzyklus erfordern. Daher kann die Zahl der Lese-Ports für den Speicher **206** auf der Basis der Zahl der drahtlosen Systeme, die gleichzeitig unterstützt werden, gewählt werden. Beispielsweise kann ein Lese-Port für jedes unterstützte drahtlose System vorgesehen sein. Da Schreibzugriffe seltener stattfinden als Lesezugriffe, könnten weniger Schreib-Ports als Lese-Ports vorgesehen sein. Die Zahl der Ports für den Speicher **206** kann auf der Basis beliebiger Kriterien ausgewählt werden, und die Erfindung ist nicht auf irgendeine bestimmte Zahl von Ports für den Speicher **206** beschränkt.

[0052] Wie oben angegeben, verarbeitet der Speicherzugriffs-Resolver **208** eingehende Zugriffsanfragen an den Speicher **206** von den Folgesteuereinrichtungen **202a**–**202n** und vom DPBUS-Bus **110**. Der Speicherzugriffs-Resolver **208** verwaltet auch Konflikte, beispielsweise im Falle von mehr Zugriffsanfragen als Lese-Ports. Der Speicherzugriffs-Resolver **208** kann solche Situationen durch Priorisieren von Anfragen beispielsweise aufgrund eines Round-Robin-Schemas bewältigen. In einem solchen Round-Robin-Schema kann ein Shift-Back-Register verwendet werden, um die Priorität zu bestimmen. In einer Ausführungsform wird das Register verschoben, wenn ein Konflikt um einen Speicherzugriff eintritt. In einer anderen Ausführungsform kann das Shift-Back-Register jedes Mal verschoben werden, wenn irgendeine Folgesteuereinrichtung einen Speicherzugriff durchführt. Es sei jedoch klargestellt, dass viele andere Verfahren zur Verwaltung von Anfragekon-

flikten angewendet werden können.

[0053] Das DPBUS-Bus-Schnittstellenmodul **210** sorgt für Verbrückungen zwischen dem Systemtaktgeber und den Taktgeberdomänen des DPBUS-Busses innerhalb des TEP **138**. Das DPBUS-Bus-Schnittstellenmodul **210** verwaltet auch die 16/32-Bit-Schnittstellenbildung zwischen dem DPBUS-Bus **110** und dem internen TEP-Bus.

[0054] Ein Blockschema des DPBUS-Bus-Schnittstellenmoduls **210** gemäß einer Ausführungsform der Erfindung ist in **Fig.** 8 dargestellt. Wie oben angegeben, führt die DPBUS-Bus-Schnittstelle **210** Intertaktsynchronisationen zwischen dem DPBUS-Bus-Takt und dem Systemtakt aus. Jede Taktdomäne kann unter Verwendung von Abstimmungszeichen für die Intersynchronisation separat gesteuert werden. Die DPBUS-Protokoll-FSM **802** verwaltet die Abstimmungssignale für den DPBUS. Die TEP-Zugriffs-FSM **804** verwaltet die Abstimmungssignale für den internen TEP-Bus.

[0055] **Fig.** 3 ist ein Blockschema, das ein Beispiel für eine Architektur der Folgesteuereinrichtung **202a** gemäß einer Ausführungsform der Erfindung erläutert. Die Folgesteuereinrichtung **202a** kann eine Vektorprozessor sein, der Abrufungs-, Decodierungs- und Ausführungsstufen aufweist. Ein Befehlsdecoder **328** decodiert Befehle, die von einem Multiplexer **342** empfangen werden. Die Befehle werden aus dem Speicher **206** des TEP **138** abgerufen. Folgesteuereinrichtungsbefehle können unter der Steuerung des DSP-Cores und der MCU **104** in den Speicher **206** geladen werden. Der Multiplexer **342** kann Daten, die mit Befehlen im Zusammenhang stehen, in ein Register **326** lenken. Die im Register **326** gespeicherten Daten können Daten sein, die in einer Schreib- oder Modifizierungsoperation geschrieben werden sollen, oder es können Daten sein, die in einer Leseoperation abgefragt werden. Die Folgesteuereinrichtung **202a** kann ferner ein DMA-Steuermodul **348** zur Schnittstellenbildung mit einem DMA-Controller **134** und einem DMA-Controller **136** aufweisen. Die Folgesteuereinrichtung **202a** kann eine Vielzahl von DMA-Registern (z. B. **302**, **304**, **306**, **308**, **310**) aufweisen, die verwendet werden, um DMA-Kanäle zu konfigurieren. Die Folgesteuereinrichtung **202a** kann ein Takt-Frequenzteilermodul **346** aufweisen, das verwendet wird, um einen Zeitimpuls zu erzeugen, um einen Delta-Zeitmesser **336** zu inkrementieren. Ein Folgesteuereinrichtungs-Steuermodul **348** verwaltet den gesamten Betrieb der Folgesteuereinrichtung und wird nachstehend ausführlicher erörtert.

[0056] Fig. 4 stellt ein Beispiel für ein Befehlsformat dar, das von den Folgesteuereinrichtungen **202a**–**202n** ausgeführt werden kann. Der Befehl **400** beinhaltet ein Sechsbit-Opcode-Feld **402**, welches den Befehlstyp identifiziert. Ein Vierbit-Datenfeld **404** kann Daten beinhalten, die benötigt werden, um den Befehl zu verarbeiten. Beispielsweise kann ein Befehl, einen Mehrzweck-I/O (GPIO)-Pin zu verbinden, ein Datenfeld beinhalten, das einen zu verbindenden GPIO-Pin identifiziert. Ein erweitertes Feld **406** kann optional als Achtbit-Erweiterung des Datenfelds **404** verwendet werden. Wenn die Daten im Zusammenhang mit der Erweiterung für das Vierbit-Datenfeld zu groß sind, kann das Erweiterungsfeld **406** verwendet werden, um den Überlauf aufzunehmen. Ein Deltazeitfeld **408** kann verwendet werden, um eine Verzögerungszeit vor der Ausführung des Befehls anzuzeigen. Das Deltazeitfeld **408** kann eine Wartezeit nach der Ausführung des vorhergehenden Befehls und vor der Ausführung des aktuellen Befehls (d. h. des Befehls **400**) anzeigen.

[0057] Nachdem ein Befehl vom Befehlsdecoder **328** decodiert wurde, kann der in **Fig.** 3 dargestellte Deltazeitgeber **336** verwendet werden, um die Wartezeit, die im Deltazeitfeld des Befehls angezeigt ist, festzulegen. Wenn die Verzögerungszeit die Zeit im Deltazeitnehmer **336** erreicht, kann der Befehl von der Ausführungseinheit **330** ausgeführt werden. Die Ausführung von Befehlen auf der Basis von Deltazeiten kann es den Folgesteuereinrichtungen ermöglichen, zeitabhängige Funktionen auszuführen, d. h. Funktionen, die zeitlich so geplant sind, dass sie zu bestimmten Zeiten stattfinden. Die Folgesteuereinrichtungen können beispielsweise zeitlich gesteuerte Interrupts für die Processing-Cores **102** und **104** (**Fig.** 1) erzeugen, damit die Prozessoren in den Ruhezustand gehen können, wenn sie nicht verwendet werden, und sie können zeitlich gesteuerte Interrupts erzeugen, um diese Ruhezustände zu geeigneten Zeiten zu verlassen. Folgesteuereinrichtungsbefehle können die Verbindung von Pins zur Steuerung von externen Geräten steuern und können die Funkeinheit ein- oder ausschalten. Folgesteuereinrichtungsbefehle können auch zu festgelegten Zeiten DMA-Kanäle aktivieren.

[0058] Das in **Fig.** 3 dargestellte Takt-Frequenzteilermodul **346** wird verwendet, um einen Zeitimpuls zum Inkrementieren des Deltazeitnehmers **336** zu erzeugen. Das Taktfrequenzteilermodul **346** führt eine Taktteilung des Systemtakts durch, um den Zeitimpuls zu erzeugen. Um Energie zu sparen, ist es günstig, eine Frequenz zu verwenden, die so niedrig wie möglich ist und die trotzdem eine ausreichende zeitliche Genauigkeit für den Betrieb mit dem drahtlosen System liefert. Da die Taktfrequenz von der Zeitsteuerung des drahtlosen Systems abhängt, kann das Taktfrequenzteilermodul **346** den Systemtakt durch irgendeinen Frequenzteilungswert zwi-

schen zwei und vierundsechzig teilen. Der Frequenzteilungswert kann in einem Register **314** gespeichert werden.

[0059] Ein Beispiel für einen Folgesteuereinrichtungsbefehl, der gemäß einer Ausführungsform der Erfindung gesetzt wird, ist in Tabelle 1 angegeben.

Tabelle 1

Opcode	Befehl (6 Bits)	Datenfeld (4 Bits)	Erweiterungs-feld (8 Bits)	Deltazeit (8 Bits)	Beschreibung
0 x 00	Clr GPIOA	Wähle Pin aus	Nicht verwendet	Deltazeit	Trennt ausgewählten GPIOA-Pin, synchronisiert mit GPSigA
0 x 01	Set GPIOA	Wähle Pin aus	Nicht verwendet	Deltazeit	Verbindet ausgewählten GPIOA-Pin, synchronisiert mit GPSigA
0 x 02	Clr GPIOB	Wähle Pin aus	Nicht verwendet	Deltazeit	Trennt ausgewählten GPIOB-Pin
0 x 03	Set GPIOB	Wähle Pin aus	Nicht verwendet	Deltazeit	Verbindet ausgewählten GPIOB-Pin
0 x 04	Clr GPIOC	Wähle Pin aus	Nicht verwendet	Deltazeit	Trennt ausgewählten GPIOC-Pin
0 x 05	Set GPIOC	Wähle Pin aus	Nicht verwendet	Deltazeit	Verbindet ausgewählten GPIOC-Pin
0 x 06	Toggle GPSigA	Wähle Signal aus	Nicht verwendet	Deltazeit	Erzeugt einen Impuls auf der ausgewählten GPSigA-Leitung

DE 602 23 555 T2 2009.06.10

0 x 07	Toggle GPSigB	Wähle Signal aus	Nicht verwendet	Deltazeit	Erzeugt einen Impuls auf der ausgewählten GPSigB-Leitung
0 x 08	Toggle ARMInt	Wähle Signal aus	Nicht verwendet	Deltazeit	Erzeugt einen Impuls auf der ausgewählten MCU-Interrupt-Leitung
0 x 09	Toggle DSPInt	Wähle Signal aus	Nicht verwendet	Deltazeit	Erzeugt einen Impuls auf der ausgewählten DSP-Interrupt-Leitung
0 x 0A	Toggle DMAInt	Wähle Kanal aus	Nicht verwendet	Deltazeit	Erzeugt einen Impuls auf der ausgewählten Sys.-DMA-Aktivierungsleitung, die einen zugehörigen DMA-Kanal aktiviert
0 x 0B	Toggle DSPDMA Int	Wähle Kanal aus	Nicht verwendet	Deltazeit	Erzeugt einen Impuls auf der ausgewählten DSP DMA-Aktivierungsleitung, die einen zugehörigen DMA-Kanal aktiviert
0 x 0C	ShortWait	Nicht verwendet	4 Bits nicht verwendet 4 Bits Warten	Warten	Warten auf <Wait>-Zeitimpulse
0 x 3x	LogWait (2Bits) Absolute Time (24 Bits)				Warten bis <Abs. Time> gekommen ist plus zusätzliche Verzögerung von 3 – 4 SysClk-Zyklen, bevor der nächste Befehl gestartet wird. Der Befehl implementiert implizit die Kalibrierungsphasenkompensation
0 x 10	If … Skip Next	Adresse (12 Bits)		Bit	Prüft <Bit> bei <Address> des internen TEP-Speichers und, falls gesetzt, wird der nächste Befehl durch einen NOP mit der gleichen Ausführungsverzögerung ersetzt

0 x 11	ClrBit	Adresse (12 Bits)	Bit	Löscht <Bit> bei <Address>
0 x 12	SetBit	Adresse (12 Bits)	Bit	Setzt <Bit> bei <Address>
0 x 15	Shift	Adresse (12 Bits)	Nicht verwendet	Links verschiebt den Inhalt bei <Address> um ein Bit
0 x 0D	Wait	Trigger auswählen, Bit 6: DSP DMA ch0 Bits 5 – 0: GPSigB[5:0] 6 Bits nicht verwendet	Nicht verwendet	Warten, bis ausgewählter Trigger GPSigB[5:0] oder der DSP DMA-Kanal 0 IRQOut gekommen sind.
0 x 14	NOP	Nicht verwendet		Kein Betrieb
0 x 0E	Jump	Absolute Adresse	Nicht verwendet	Standardzweig, springt zu <absolute Address>
0 x 0F	Die	Nicht verwendet		Stoppt die Ausführung und schickt die Hardware der Ablaufsteuerungseinrichtung in den Ruhezustand
0 x 18	WriteRegF	Reg.-Dateiadresse / Schreibe Daten (16 Bits)		Schreibt gegebene Daten unmittelbar in das interne Register der Ablaufsteuerungseinrichtung
0 x 13	DataMovE	Nicht verwendet		Initiiert DMA-Transfer, sing DMA-Setup, in RegFile gespeichert

[0060] Manchmal kann es nötig sein, zwei oder mehr I/O-Pins gleichzeitig zu verbinden oder zu trennen. Obwohl der Folgesteuereinrichtungs-Befehlssatz Befehle zum Verbinden oder Trennen von I/O-Pins liefern kann, werden solche Befehle nacheinander und nicht gleichzeitig ausgeführt. Um zwei oder mehr Pins gleichzeitig zu verbinden oder zu trennen, können die Verbindungs- und Trennungsbefehle auf ein bestimmtes Signal synchronisiert werden. Wenn beispielsweise Pin GPIOA und Pin GPIOB gleichzeitig verbunden werden müssen, kann eine Folgesteuereinrichtung diese Befehle auf GPSigA synchronisieren. Dann kann die Folgesteuereinrichtung zuerst den Set GPIOA-Befehl ausführen, gefolgt vom Set GPIOB-Befehl. Diese Pins werden eigentlich nicht verbunden, bis ein Toggle GPSigA-Befehl ausgeführt wird, der bewirkt, dass beide Pins gleichzeitig verbunden werden.

[0061] Ein LongWait-Vergleichsmodul **340** wird verwendet, wenn ein LongWait-Befehl von der Folgesteuereinrichtung ausgeführt wird. Ein LongWait-Befehl kann ausgeführt werden, wenn von der Folgesteuereinrichtung für eine bestimmten Zeitraum keine anschließenden Befehle ausgeführt werden müssen. Ein LongWait-Befehl erlaubt die Ausschaltung des Systemtaktgebers und ermöglicht es der Folgesteuereinrichtung, einen langsamen Takt für die Zeitsteuerung zu verwenden, um dadurch Energie zu sparen.

[0062] Das LongWait-Vergleichsmodul **340** vergleicht die Wartezeit, die im LongWait-Befehl angezeigt ist, mit dem Wert des Absolutzählers **214** (**Fig.** 2), was nachstehend ausführlicher erörtert wird. Die Wartezeit kann ein 24 Bit-Wert sein und somit die Verwendung des Achtbit-Deltazeitfelds, des Achtbit-Erweiterungsfelds, des Vierbit-Datenfelds und von vier Bits des Sechsbit-Opcode-Felds benötigen. Das LongWait-Vergleichsmodul **340** empfängt einen Input von einem 24 Bit-Absolutzähler **214** und vergleicht den Wert mit der 24 Bit-Wartezeit vom LongWait-Befehl. Wenn die Werte übereinstimmen, kann die Folgesteuereinrichtung den nächsten Befehl ausführen. Das LongWait-Vergleichsmodul **340** gibt auch Ruheinformationen aus, die vom Takt- und Leistungssteuerblock **216** des TEP **138** genutzt werden können, um zu bestimmen, ob die Folgesteuereinrichtungen LongWait-Befehle ausführen, so dass der Systemtaktgeber ausgeschaltet werden kann, wenn alle Folgesteuereinrichtungen im Ruhezustand sind.

[0063] Ein PreAbs32-Register **338** wird verwendet, um einen Zeitpunkt zum Einschalten des Oszillators zu bestimmen, wenn der Oszillator ausgeschaltet worden ist. Das PreAbs32-Register **338** zeigt den absoluten Zeitpunkt für die Einschaltung des Oszillators an, der dem Oszillator ausreichend Zeit lässt, sich zu stabilisieren, bevor der aktuell ausgeführte LongWait-Befehl beendet ist und die Ausführung des nächsten Befehls beginnt.

[0064] Das Folgesteuerungs-Steuermodul **334** steuert den Programmablauf und verwaltet Interrupts für die Folgesteuereinrichtungen **202a**–**202n**. Das Folgesteuerungs-Steuermodul **334** ruft auf der Basis des Inhalts des Programmregisters **322** Befehle vom Speicher ab. Ein Programmzählerregister **322** bewahrt die Adresse des nächsten auszuführenden Befehles auf. Das Folgesteuerungs-Steuermodul **334** kann Interrupts über eine Leitung **344** vom Interrupt-Selektor **332** empfangen, der die Interrupt-Abfrage mit der höchsten Priorität von einer Vielzahl von Interrupt-Quellen auswählen kann. Wenn ein Interrupt empfangen wird, kann das Interrupt-Aktivierungsbit in einem Register **316** gesetzt werden und die Adresse des Interrupt-Vektors kann in ein Register **318** geladen werden. Die Folgesteuereinrichtung springt zur Adresse des Interrupt-Vektors im Register **318** und setzt die Ausführung von dort aus fort.

[0065] Wenn eine Folgesteuereinrichtung einen harten Reset empfängt oder einen Ausschwingungs- bzw. Die-Befehl ausführt, geht die Folgesteuereinrichtung in den Ruhezustand. Ein weicher Reset wird verwendet, um einer Folgesteuereinrichtung zu befehlen, einen ersten Befehl abzurufen und mit der Ausführung von Befehlen zu beginnen. Wenn die Folgesteuereinrichtung einen weichen Reset oder einen Interrupt empfängt, kann sie mit der normalen Ausführung fortfahren. Wenn die Folgesteuereinrichtung einen weichen Reset empfängt, wird die Adresse, zu der die Folgesteuereinrichtung springt, um mit der Ausführung zu beginnen, in einem Register **320** aufgehoben. Wenn die Folgesteuereinrichtung einen Interrupt empfängt, wird die Adresse des Interrupt-Vektors, zu dem die Folgesteuereinrichtung springt, um die Ausführung zu beginnen, in einem Register **318** aufbewahrt.

[0066] DMA-Register **302**, **304**, **306**, **308** und **310** werden von der Folgesteuereinrichtung verwendet, um DMA-Kanalkonfigurationsinformationen zu speichern. Beispielsweise können diese DMA-Register eine Quelladresse, eine Zieladresse und eine Zahl von Bytes, die übertragen werden sollen, speichern. Ein DMA-Steuermodul **348** bildet eine Schnittstelle mit einem DSPDMA-Controller **134** (**Fig.** 1) und einem System-DMA-Controller **136**, um DMA-Transfers zu initiieren.

[0067] **Fig.** 6 zeigt ein Beispiel für eine Schnittstelle zwischen TEP **138** und DMA-Controller **134**. Ein

DMA-Kanal kann der Verwendung durch den TEP **138** gewidmet sein und von anderen Ressourcen nicht genutzt werden. Zum Beispiel kann Kanal 0 der TEP-Verwendung zugewiesen sein, aber es kann jeder beliebige DMA-Kanal verwendet werden. Folgesteuereinrichtungen **202a**–**202n** können DMA-Transfers unter Verwendung des festen Kanals initiieren, indem sie einen Befehl ausführen, beispielsweise den DataMoveE-Befehl, der im Folgesteuereinrichtungs-Befehlssatz von Tabelle 1 gezeigt ist. Der DataMoveE-Befehl ruft DMA-Kanalkonfigurationsinformationen von den DMA-Registern **302**, **304**, **306**, **308** und **310** ab und kopiert die Informationen in einen internen RAM **604** des DMA-Controllers **134**. Mehrere Folgesteuereinrichtungen können gleichzeitig Zugriff auf den gewidmeten DMA-Kanal verlangen. Eine Anfragenresolver-Zustandsmaschine (FSM) **218** verwaltet diese gleichzeitigen Anfragen Beispielsweise kann die Anfragenresolver FSM **218** ein Round-Robin-Prioritätsschema verwenden, um den Folgesteuereinrichtungen **202a**–**202n** Zugriff auf DMA-Kanäle zu gewähren. Wenn Zugriff gewährt wird, werden die Werte der DMA-Register in den Speicher **604** des DMA-Controllers kopiert, und der Anfragenresolver FSM **218** setzt ein Kanalaktivierungs-Flag, um den gewidmeten Kanal zu aktivieren. Wenn der DMA-Transfer ausgeschlossen ist, schickt der DMA-Controller **134** einen Interrupt zur Anfragenresolver-FSM **218** zurück. Ebenso kann die Anfragenresolver-FSm **218**, während sie in Betrieb ist, ein DRReqSysClk-Flag (nicht dargestellt) ausgeben, um sicherzustellen, dass der Systemtaktgeber während eines DMA-Transfers nicht abgestellt wird.

[0068] Fig. 7 zeigt schematisch ein Beispiel für einen TEP-initiierten DMA-Transfer, wie oben mit Bezug auf **Fig.** 6 beschrieben. Zuerst sendet der TEP **138** unter Verwendung des gewidmeten Kanals Kanalkonfigurationsinformationen an den DMA-Controller **134** und sendet ein Kanalaktivierungssignal an den DMA-Controller **134**. Der DMA-Controller **134** führt die Datenübertragung durch und erzeugt einen Interrupt für TEP **138**, der den Abschluss der Datenübertragung anzeigt.

[0069] Die in **Fig.** 2 dargestellte Taktkalibrierungseinheit **212** wird verwendet, um einen langsamen Takt des Kommunikationsprozessors **100** zu kalibrieren. Der Kommunikationsprozessor **100** kann Taktsignale von einem Systemtakt bei einer Frequenz von beispielsweise 13 MHz und von einem freilaufenden langsamen Takt bei einer Frequenz von beispielsweise 32 kHz empfangen. Während ein Hochfrequenz-Takt, wie der Systemtakt, notwendig sein kann, um die Processing-Cores des Kommunikationsprozessors **100** zu takten, kann der langsame Takt für die Zeitsteuerung verwendet werden, um Energie zu sparen, wenn die Processing-Cores im Ruhezustand sind und nicht vom Hochfrequenztakt getaktet werden müssen. Der TEP **138** kann eine Systemzeitsteuerung vom langsamen Takt ableiten. Die vom TEP **138** verwalteten Zeitsteuerungsereignisse beruhen auf einer Zeit des langsamen Taktes und einer Deltazeit (gezählt in Systemtaktzyklen) relativ zu einer Zeit des langsamen Taktes. Der Systemoszillator kann ausgeschaltet werden, wenn er nicht von einem der Module des Kommunikationsprozessors **100** gebraucht wird, und der langsame Takt kann die Aufgabe übernehmen, die Einschaltung des Systemtaktgebers zu initiieren, wenn dieser für die nächste zeitlich geplante Operation benötigt wird. Die Ausschaltung des Systemoszillators wird nachstehend ausführlicher erörtert.

[0070] Der langsame Takt ist nicht so genau wie der Hochfrequenz-Systemtakt und ist empfindlicher gegenüber Temperaturschwankungen. Somit kann der langsame Takt so kalibriert werden, dass ein gewünschter Grad an Genauigkeit sichergestellt ist. Der langsame Takt kann entweder mittels des Systemtakts oder der Zeitsteuerung des drahtlosen Systems, die über Funk empfangen wird, kalibriert werden. Wenn der Systemtakt zur Kalibrierung verwendet wird, kann die Zahl der Systemtaktzyklen über eine ausgewählte Zahl von langsamen Taktzyklen gezählt werden. Wenn die Zeitsteuerung von einem drahtlosen System zur Kalibrierung verwendet wird, kann die Zahl der drahtlosen Systemtaktzyklen (über Funk empfangen) über eine ausgewählte Zahl von langsamen Taktzyklen gezählt werden.

[0071] Um die Energie, die ansonsten durch Kalibrieren des langsamen Taktes unter Verwendung eines Frequenzaufbereiters oder eines VCO verbraucht würde, zu sparen, kann der langsame Taktgeber dadurch kalibriert werden, dass man Taktzyklen aus dem freilaufenden langsamen Takt herausnimmt, um für einen kalibrierten langsamen Takt zu sorgen. Das heißt, man kann eine Frequenz, die niedriger ist als die erwartete Frequenz des freilaufenden langsamen Taktes als kalibrierte Taktfrequenz wählen (z. B. 31 kHz im Falle eines langsamen 32 kHz-Taktes), und das kalibrierte Taktsignal kann durch Herausnehmen von Taktimpulsen aus dem freilaufenden langsamen Taktsignal erzeugt werden. Der freilaufende langsame Takt kann durch einen Bruch-N-Taktteiler eingestellt werden, der periodisch einen Taktzyklus aus dem freilaufenden langsamen Takt herausnimmt. Der Zeitraum, in dem ein Taktzyklus aus dem freilaufenden langsamen Takt herausgenommen wird, hängt von spezifischen Bruch- und Modulwerten und von Informationen ab, die aus dem Vergleich des langsamen Taktes und des Systemtaktes erhalten werden. Wenn beispielsweise der Zeitraum, in dem Taktzyklen entfernt werden, neun Zyklen des langsamen Taktes entspricht, werden acht Zyklen des kalibrierten langsamen Taktes für jeweils neun Zyklen des unkalibrierten langsamen Taktes erzeugt.

[0072] Jedoch werden durch das Herausnehmen von Taktzyklen aus dem freilaufenden langsamen Takt Phasenfehler in den kalibrierten langsamen Takt eingeführt. Solche Phasenfehler beruhen auf der Tatsache, dass der kalibrierte langsame Takt nicht wirklich periodisch ist. Man nehme beispielsweise an, dass eine kalibrierter langsamer Takt von 40 kHz aus einem freilaufenden 50 kHz-Takt erzeugt wird. Der 50 kHz-Takt weist alle 20 µs eine ansteigende Flanke auf. Das heißt, der 50 kHz-Takt weist ansteigende Flanken bei 20 µs, 40 µs, 60 µs, 80 µs, 100 µs, 120 µs usw. auf. Der kalibrierte langsame 40 kHz-Takt kann durch periodisches Herausnehmen eines Zyklus erzeugt werden. Somit weist der kalibrierte langsame Takt ansteigende Flanken bei 20 µs, 40 µs, 60 µs, 100 µs, 120 µs usw. auf. Der kalibrierte langsame Takt bildet im Durchschnitt eine 40 kHz-Takt, d. h. 40.000 ansteigende Taktflanken pro Sekunde, aber stimmt mit der Phase eines echten 40 kHz-Taktes nicht überein. Ein echter 40 kHz-Takt würde alle 25 µs eine ansteigende Taktflanke aufweisen. Beispielsweise würde ein echter 40 kHz-Takt ansteigende Taktflanken bei 25 µs, 50 µs, 75 µs, 100 µs, 125 µs usw. aufweisen. Somit treten die ansteigenden Flanken im kalibrierten 40 kHz-Takt und die ansteigenden Flanken im echten 40 kHz-Takt zu unterschiedlichen Zeiten auf, und eine Phasenkompensation wird verwendet, um den Phasenunterschied zwischen dem kalibrierten langsamen Takt und einem echten Takt mit der gleichen Frequenz zu berücksichtigen, wie nachstehend erörtert.

[0073] Die Fig. 9A und Fig. 9B zeigen eine Implementierung eines Bruch-N-Taktteilers mit Phasenkompensation bzw. ein Zeitssteuerungsdiagramm gemäß einer Ausführungsform der Erfindung. Ein fraktionales Inkrementierungsregister **902** speichert das Verhältnis von schnellen Taktzyklen zu freilaufenden 32 kHz-Taktzyklen und dient als ein Input bzw. eine Eingabe für einen Addierer **904**. Ein Phasenkompensationsregister **906** ist ein Akkumulator, der den Output bzw. die Ausgabe eines Addierers **904** akkumuliert und dient als ein Input für einen Moduloperator **912**. Ein Modulregister **908** speichert einen Wert, der von einem Komparator **914** mit den oberen 10 Bits im Phasenkompensationsregister **906** verglichen wird. Der Komparator **914** dient als Input für ein UND-Gatter **910** und steuert, ob der freilaufende 32 kHz-Takt das Gatter **910** passiert. Das Modulregister **908** ist ein zweiter Input für den Moduloperator **912**. Der Moduloperator **912** berechnet einen Modulwert, der verwendet wird, um das Phasenkompensationsregister **906** zu inkrementieren, wenn der Komparator gesetzt wird.

[0074] Im Betrieb wird, wenn der Wert im Register **906** den Wert im Modulregister **908** erreicht, die Ausgabe des Komparators **914** gesetzt, wodurch die Ausgabe des Gatters **910** verhindert wird. Wie man sieht, wird der Umfang der Phasenkompensation (d. h. der Wert des Registers **906**) akkumuliert und steigt linear über jeden freilaufenden 32 kHz-Taktzyklus an. Wenn der Akkumulator das Modulregister **908** erreicht, wird die Eingabe des freilaufenden 32 kHz-Taktes bis zum nächsten Taktzyklus ausgegattert. Dann wird der Phasenkompensationsakkumulator vom Moduloperator über den berechneten Modulwert zyklisch wiederholt.

[0075] Wie oben angegeben, werden durch die Entfernung von Taktimpulsen Phasenfehler in das kalibrierte Taktsignal eingeführt. Die Phasenfehler resultieren aus der Tatsache, dass der kalibrierte langsame Takt **930**, aus dem Impulse herausgenommen wurden, wie in Fig. 9B dargestellt, Taktflanken aufweist, die zu anderen Zeiten auftreten als bei einem freilaufenden Takt der gleichen Frequenz Wie von der Wellenform **932** in Fig. 9B dargestellt, nimmt der Phasenfehler mit jedem langsamen Taktzyklus zu, bis ein Impuls herausgenommen wird, und nimmt dann auf null ab. Ohne die Phasenkompensation würden diese Phasenfehler Zeitsteuerungsfehler in drahtlosen Systemen erzeugen. Durch Nutzen des kalibrierten langsamen Takts und eines Phasenkompensationssignals, welches den Phasenfehler darstellt, wird eine exakte Zeitsteuerung mit dem kalibrierten langsamen Takt erreicht.

[0076] Wenn das kalibrierte Taktsignal verwendet wird, um den Absolutzähler **214** anzusteuern, werden somit die Phasenfehler im kalibrierten Taktsignal unter Verwendung der Phasenkompensation, die im Phasenkompensationsregister **906** berechnet wird, kompensiert. Unter Bezugnahme auf das oben erörterte Beispiel, in dem ein freilaufender langsamer 50 kHz-Takt und ein kalibrierter langsamer 40 kHz-Takt verwendet werden, nehme man an, dass ein Ereignis zeitlich so geplant wird, dass es bei der dritten ansteigenden Flanke des 40 kHz-Taktsignals eintritt. Wie oben erörtert, tritt in einem echten 40 kHz-Takt die dritte ansteigende Flanke bei 75 µs auf. Im kalibrierten 40 kHz-Takt tritt die dritte ansteigende Flanke jedoch bei 60 µs auf. Somit unterscheidet sich die Phase des kalibrierten langsamen Takts von derjenigen des echten 40 kHz-Takts um 15 µs. Wenn der kalibrierte langsame Takt bei 60 µs die dritte ansteigende Flanke erreicht, wird eine weitere Verzögerung von 15 µs, gerechnet in Systemtaktzyklen, hinzuaddiert, bevor das geplante Ereignis ausgeführt wird. Auf diese Weise werden die Folgesteuereinrichtungen im Hinblick auf die eingestellte Frequenz des kalibrierten Taktsignals kompensiert.

[0077] Manchmal kann beispielsweise wegen schnellen Temperaturschwankungen keine ausreichende Frequenzstabilität des langsamen Takts erreicht werden. Es kann jedoch trotzdem notwendig sein, eine kalibrier-

tes langsames Taktsignal zum Ansteuern des Absolutzählers und zur zeitlichen Steuerung der Ausführung von LongWait-Befehlen zu erzeugen. In solchen Situationen kann eine Frequenzteilung des Systemtakts angewendet werden. Beispielsweise kann eine Taktteiler-FSM **916** den Systemtakt auf einen kalibrierten langsame Takt herunterbrechen.

[0078] Der in **Fig.** 2 dargestellte Absolutzähler **214** kann vom kalibrierten langsamen Takt getaktet werden. Der Absolutzähler ist ausführlicher in Fig. 10 dargestellt. In einer Ausführungsform kann es sich bei dem Absolutzähler **214** um einen 24 Bit-Zähler handeln und dieser kann von den Folgesteuereinrichtungen **202a**–**202n** verwendet werden, wenn diese einen LongWait-Befehl ausführen, um zu bestimmen, wann die Wartezeit abgelaufen ist. Beispielsweise können die Folgesteuereinrichtungen **202a**–**202n** den Wert des Absolutzählers **214** mit der Wartezeit des LongWait-Befehls vergleichen, um zu bestimmen, wann die Wartezeit abgelaufen ist.

[0079] Zwei Trigger-Generatoren **1002** und **1004** werden geliefert und können für verschiedene Zwecke verwendet werden, beispielsweise zum Triggern von Interrupts oder zum Triggern eines Snapshot. Ein Snapshot ist eine Messung des langsamen Takts gegen den Systemtakt oder eine Messung des langsamen Takts gegen die Zeitsteuerung des drahtlosen Systems, die über Funk erhalten wird und die verwendet werden kann, um den langsamen Takt zu kalibrieren. Ein Snapshot beinhaltet das Zählen der Systemtaktzyklen in einer bestimmten Zahl von langsamen Taktzyklen.

[0080] Die Fig. 11A und Fig. 11B zeigen ein Blockschema bzw. ein Zustandsübergangsschema, um einen Snapshot zu erhalten. Ein Snapshot kann durch mehrere verschiedene Eingaben initiiert werden. Beispielsweise kann ein Snapshot von irgendeinem der beiden periodischen Trigger des Absolutzählers initiiert werden oder kann durch Software initiiert werden, die auf irgendeinem der beiden Processing-Cores läuft, auf der Basis von im SeqCtrl-Register **312** (**Fig.** 3) gesetzten Bits in irgendeiner der Registerdateien der Folgesteuereinrichtungen.

[0081] Wenn ein Snapshot initiiert wird, wird ein Kalibrierungssignal ausgegeben, um eine Ausschaltung des Systemtaktgebers zu verhindern. Dann nimmt eine Snapshot-FSM **1108** einen Setup-Zustand **1103** ein, in dem sie auf den Empfang eines SysClkOk-Signals **1110** wartet, welches anzeigt, dass der Systemtaktoszillator nicht ausgeschaltet wurde. Wenn das SysClkOk-Signal **1110** empfangen wird, geht die Snapshot-FSM **1108** in einen Snapshot-Zustand **1105** über, in dem die Zahl der Systemtaktzyklen während einer Anzahl von langsamen Taktzyklen gezählt wird. Die Zahl der langsamen Taktzyklen ist in einem TCLR-Register **1112** festgelegt, das durch Software konfiguriert werden kann. Nachdem die Zahl der Taktzyklen, die im TCLR-Register **1112** festgelegt ist, von einem Zähler **1114** des langsamen Takts gezählt wurde, wird ein Interrupt erzeugt und die Snapshot-FSM **1108** geht in den Wiederholungszustand **1107** über. Während die Snapshot **1108** im Wiederholungszustand **1107** ist, kann ein Systemtaktzyklus-Zähler **1116** über die DPBUS-Bus-Schnittstelle **210** gelesen werden, um etwaige Register, die zur Kalibrierung des langsamen Takts nötig sind, zu aktualisieren. Nachdem der Zähler **1116** gelesen wurde, kehrt die Snapshot-FSM **1108** in den Ruhezustand **1101** zurück.

[0082] Das Takt- und Leistungssteuermodul **216** ist in **Fig.** 2 dargestellt. Wenn bestimmte Module des TEP **138** nicht verwendet werden, können die Taktsignale für diese Module ausgegattert werden, um Energie zu sparen. Wenn eine oder mehrere Folgesteuereinrichtungen LongWait-Befehle ausführen und keine anderen Module die Verwendung des Systemtakts verlangen, kann das Takt- und Leistungssteuermodul **216** bestimmen, ob die Dauer des LongWait-Befehls ausreicht, um eine Ausschaltung des Systemtaktgebers zu erlauben.

[0083] **Fig.** 12 ist ein Blockschema des Takt- und Leistungssteuermoduls **216**. Mehrere TEP-Module, wie ein Speicherzugriffsresolver **208** und ein DMA-Anfrageresolver **218**, können die Verwendung des Systemtaktes erfordern und können dem Takt- und Leistungssteuermodul **216** über Signale **1206** bzw. **1208** anzeigen, dass der Systemtakt benötigt wird. Ein externes Signal, ReqSysClk **1210**, kann von einer externen Quelle geschickt werden, um anzuzeigen, dass eines oder mehrere Module außerhalb des TEP die Verwendung des Systemtakts erfordern. Die Bestimmung, wann das ReqSysClk-Signal zu schicken ist, wird nachstehend ausführlicher erläutert. Das Kalibrierungssignal **1212** wird vom Taktkalibrierungsmodul ausgegeben, wenn eine Kalibrierung des langsamen Takts durchgeführt wird, und verlangt, dass der Systemtakt aktiv bleibt.

[0084] Jede Folgesteuereinrichtung **202a**–**202n** kann dem Takt- und Leistungssteuermodul **216** über ein Set/Reset-Flip-Flop (SRFF) **1218** anzeigen, dass der Systemtakt benötigt wird. Die Q-Ausgabe von SRFF **1218** ist ein TEPReqSysClk-Signal **1216**. Jede Folgesteuereinrichtung, die keinen Systemtakt benötigt, gibt ein KillSysOsc-Signal über das ein UND-Gatter **1222** aus. Wenn keine der Folgesteuereinrichtungen den Systemtakt benötigt, geht das SRFF **1218** in den Reset-Zustand über und das Signal **1216** wird nicht ausgegeben. Wenn irgendeine Folgesteuereinrichtung die Verwendung des Systemtakts benötigt, gibt sie über ein

ODER-Gatter **1224** ein Restart-Systemoszillatorsignal aus. Als Antwort darauf geht das SRFF **1218** in den Set-Zustand über, und das Signal **1216** wird ausgegeben. Das PreAbs32-Register **338** kann verwendet werden, um die späteste Zeit zu speichern, zu der der Oszillator ausgeschaltet bleiben kann, wobei die geplante Ausführungszeit des nächsten Befehls und die notwendige Aufwärmzeit des Oszillators berücksichtigt werden. Eine Folgesteuereinrichtung kann ein KillSysOsc-Signal ausgeben, wenn die aktuelle Zeit kürzer ist als die Zeit in ihrem PreAbs32-Register **338**. Wenn die aktuelle Zeit der Zeit in ihrem PreAbs2-Register **338** gleich ist, dann kann eine Folgesteuereinrichtung ein RestartSysOsc-Signal ausgeben. Wenn die aktuelle Zeit der Ablaufzeit des LongWait-Befehls gleich ist, sollte der Systemtakt stabilisiert sein.

[0085] Eine Einschalt-Folgesteuereinrichtung **122** empfängt vom ODER-Gatter **1220** ein Eingangssignal, das anzeigt, ob irgendein Modul, ob innerhalb oder außerhalb des TEP, die Verwendung des Systemtakts benötigt. Wenn dieses Signal ausgegeben wird, kann die Einschalt-Folgesteuereinrichtung **1226** den Systemtaktoszillator durch Ausgeben des SysOscOn-Signals **1236** einschalten. Ein Clock-Pad-Power-Up-Register (CPPUR) **1228** speichert die Ausregelzeit für den Clock-Pad-Puffer und ein Oszillator-Warm-Up-Register OWUR **1230** speichert die Aufwärmzeit für den Oszillator. Wenn das Eingangssignal für die Einschaltungs-Folgesteuereinrichtung **1226** vom ODER-Gatter **1220** ausgegeben wird, startet eine FSM **1234** einen 10 Bit-Zähler **1232** bei null und gibt das SysOscOn-Signal **1236** aus, wodurch bewirkt wird, dass der Systemoszillator eingeschaltet wird. Wenn der Zähler **1232** die Zeit erreicht, die im OWUR **1230** spezifiziert wurde, wird ein ClkBufOn-Signal **1238** ausgegeben, wodurch ein Clock-Pad-Puffer aktiviert wird. Wenn der Zähler **1232** die im OWUR **1230** festgelegte Zeit plus der Zeit, die im CCPUR **1228** spezifiziert ist, erreicht, wird ein SysClkGate-Signal **1240** ausgegeben, das anzeigt, dass die Ausgabe des Systemtaktoszillators gültig ist, und das UND-Gatter **1242** wird aktiviert. Das UND-Gatter **1242** blockiert die Ausgabe des Systemtaktoszillators, bis der Oszillator genügend Zeit zum Stabilisieren gehabt hat. Der Oszillator ist stabil, nachdem die Oszillator-Aufwärmzeit, die im OWUR **1230** festgelegt ist, plus der Closk-Pad-Einschaltverzögerungszeit, die im CPPUR **1228** gespeichert ist, erreicht ist. Wenn diese Zeit erreicht ist, aktiviert das SysClkGate-Signal **1240** das UND-Gatter **1242**, und Taktsignale vom Oszillator werden durch das Gatter hindurch gelassen.

[0086] Wie oben angegeben, empfängt das Takt- und Leistungssteuermodul **216** ein ReqSysClk-Signal von einer Quelle außerhalb des TEP **138**. Das Signal zeigt an, ob irgendwelche Module außerhalb des TEP **138**, wie ein DSP-Core **102** und eine MCU **104**, die Verwendung des Systemoszillators erfordern. Fig. 13A zeigt, wie Taktsignale im Kommunikationsprozessor **100** erzeugt werden können. Eine Leistungsquelle **1300** versorgt einen Systemoszillator **1301** mit Energie. Die Leistungsquelle kann wie oben beschrieben von einem vom TEP empfangenen SysOscOn-Signal gesteuert werden. Dieses Signal kann verwendet werden, um zu steuern, ob der Oszillator **1301** an- oder ausgeschaltet wird. Der Oszillatorausgang wird in einen Kontaktfleckpuffer-Verstärker **1303** eingegeben. Der Pufferverstärker **1303** kann über ein Steuersignal vom TEP an- und ausgeschaltet werden. Der Taktsignalausgang vom Pufferverstärker **1303** wird in ein UND-Gatter **1305** eingegeben. Die zweite Eingabe in das Gatter **1305** ist ein SysClkGate-Signal, das vom TEP **138** empfangen wird und das eine Ausgatterung des Oszillatorausgangs während eines Aufwärmens des Oszillators zulässt.

[0087] Taktsignale vom Gatter **1305** werden in eine Nachlaufsynchronisation bzw. Phase-Locked-Loop (PLL) **1307** eingegeben, die das Taktsignal auf eine Frequenz multipliziert, die sich für die Taktung des DSP-Core **102** eignet. In Situationen, wo der DSP-Core in Ruhestellung ist, muss es nicht notwendig sein, das Taktsignal mittels PLL **1307** zu multiplizieren, und das Taktsignal, das vom Gatter **1305** ausgegeben wird, wird der PLL **1307** nicht geschickt. Der Multiplexer **1309** wählt entweder das multiplizierte Taktsignal von der PLL **1307** oder die Ausgabe des Gatters **1305** aus. Wie in Fig. 13B dargestellt, können mehrere Taktsignale aus der Ausgabe des Multiplexers **1309** erzeugt werden. Zuerst kann ein DCLK-Takt als Ausgabe der PLL **1307** erzeugt werden. Der DCLK-Takt kann verwendet werden, um den DSP-Core **102** zu takten. Ein nicht-gegatterter DCLK(nGDCLK)-Takt **1319** kann in ein UND-Gatter **1311** eingegeben werden. Der DCLK-Takt kann mittels des UND-Gatters **1311** ausgegattert werden, wenn er nicht vom DSP-Core benötigt wird. Dann kann ein DSCLK-Takt mittels eines Frequenzteilers **1321**, mit dem der DCLK-Takt geteilt wird, erzeugt werden. Der Frequenzteiler **1321** kann über Software programmierbar sein und kann den DCLK-Takt durch 1 oder 2 teilen. Der DSCLK-Takt kann verwendet werden, um das DSP-Untersystem zu takten, welches die DSP-Peripheriegeräte und den DSP DMA-Controller **134** einschließt. Der DSCLK-Takt kann vom UND-Gatter **1312** ausgegattert werden, wenn er nicht benötigt wird. Der nicht-gegatterter DSCLK(nGDSCLK)-Takt **1323** kann einem programmierbaren Taktteiler **1325** zugeführt werden, der sein Eingangssignal durch eine Zahl zwischen 1 und 8 teilt, um einen BCLK-Takt zu erzeugen. Der BCLK-Takt kann verwendet werden, um die Busse des Kommunikationsprozessors **100** anzusteuern. Der BCLK-Takt kann von einem UND-Gatter ausgegattert werden, wenn er nicht benötigt wird. Ein MCLK-Takt kann die gleiche Frequenz aufweisen wie der BCLK-Takt und kann verwendet werden, um die MCU **104** zu takten. Der MCLK-Takt kann vom UND-Gatter **1317** ausgegattert werden, wenn er nicht gebraucht wird.

[0088] Die Fig. 14A und Fig. 14B zeigen, wie die Taktsignale der Fig. 13B ausgegattert werden können, wenn sie nicht gebraucht werden. Wie in Fig. 14A dargestellt, multipliziert die PLL **1307** die Oszillatorausgabe, um ein Taktsignal zu erzeugen. Der Taktteiler **1419** kann die gleichen Operationen durchführen wie die Taktteiler **1321** und **1325** von Fig. 13B. Die Taktsignale, die vom Taktteiler **1419** ausgegeben werden, können das NgDCLK-Signal, das nGDSCLK-Signal und das nGDBCLK-Signal sein. Jedes dieser Taktsignale wird dann zu einem der Multiplexer **1309a**–**1309c** gelenkt und dann zum geeigneten UND-Gatter **1311**–**1315**.

[0089] Das Register **1405** ist ein MCU-Schlaftaktanforderungsregister (MSCRR). Das MSCRR-Register **1405**, wie in Fig. 14B dargestellt, zeigt an, welche Takte benötigt werden, während die MCU schläft oder im Ruhezustand ist. Ebenso speichert ein MCU-Aktivtaktanforderungsregister(MACRR)-Register **1407** Informationen darüber, welche Takte benötigt werden, während die MCU aktiv ist. Ein MCU-Aktiv-Signal **1427**, das von der MCU **104** erzeugt wird, wird von einem Multiplexer **1423** verwendet, um zu bestimmen, ob der Inhalt des MSCRR-Registers **1405** oder des MACRR-Registers **1407** ausgegeben werden soll. Ein PLL-Bypass-Bit im MSCRR-Register **1405** erlaubt, wenn es gesetzt ist, eine Umgehung der PLL **1307**, während die MCU schläft. Da die MCU **104** möglicherweise nicht getaktet werden muss, während sie im Ruhezustand ist, muss es nicht notwendig sein, dass die PLL **1307** den Oszillator auf eine hohe Frequenz multipliziert, um die MCU **104** anzusteuern. Somit können Energieeinsparungen durch Umgehen der PLL **1307** erreicht werden. Ebenso können in manchen Fällen, wo keine hohe Verarbeitungsgeschwindigkeit nötig ist, der DSP-Core **102** und die MCU **104** auf dem Systemtakt laufen, der in den Kommunikationsprozessor ohne PLL-Multiplikation eingegeben wird.

[0090] Ähnlich wie die MCU **104** sind zwei DSP-Register vorgesehen: ein DSP-Schlaftaktanforderungsregister (DSCRR) **1401** und ein DSP-Aktivtaktanforderungsregister (DACRR) **1403**. Die Register **1401** und **1403** zeigen an, welche Takte benötigt werden, während der DSP-Core **102** schläft, bzw. welche Takte benötigt werden, wenn der DSP-Core **102** aktiv ist. Ein DSP-Aktiv-Signal, das vom DSP-Core **102** erzeugt wird, wird von einem Multiplexer **421** verwendet, um zu bestimmen, ob der Inhalt des DSCRR-Registers **1401** oder des DACRR-Registers **1403** ausgegeben werden soll. ODER-Gatter **1409**, **1411** und **1413** kombinieren die Ausgänge der MCU-Anforderungsregister **1405** und **1407** und der DSP-Anforderungsregister **1401** und **1403**. UND-Gatter **1415**, **1311**, **1313**, **1315** und **1317** können verwendet werden, um die entsprechenden Taktsignale gemäß dem Inhalt der Register **1401**, **1403**, **1405** und **1407** zu aktivieren oder zu blockieren.

[0091] Zusätzlich zur Blockierung bestimmter Taktsignale, wenn diese nicht benötigt werden, um dadurch Energie zu sparen, kann der Systemtaktoszillator ausgeschaltet werden, so dass keine Systemtaktsignale erzeugt werden, wenn keine Module des Kommunikationsprozessors **100** einen Takt benötigen Die Fig. 15 zeigt, wie der Oszillator ausgeschaltet werden kann. Der DSP-Core **102** und der MCU-Core **104** aktualisieren ein Register **1503**, das anzeigt, ob die entsprechenden Processing-Cores Taktsignale benötigen und ob irgendwelche Peripheriegeräte Taktsignale benötigen. Ein Taktsteuermodul **1501** überwacht das Register **1503**, um zu bestimmen, ob eines der Taktsignale von irgendwelchen Modulen im Kommunikationsprozessor **100** benötigt wird. Wenn keines dieser Taktsignale benötigt wird, kann das Taktsignalmodul **1501** ein SysClkReq-Signal an den TEP **138** ausgeben. Wie oben mit Bezug auf **Fig.** 12 erörtert, kann der TEP **138** dann bestimmen, ob der Systemoszillator ausgeschaltet werden kann. Auf diese Weise kann der Systemoszillator eingeschaltet werden, wenn er benötigt wird, und ausgeschaltet werden, wenn er nicht benötigt wird, um Energie zu sparen.

Patentansprüche

1. Basisbandprozessor für drahtlose Anwendungen, der folgendes aufweist:
ein erstes Modul (**208**), das ein erstes Taktsignal benötigt, das von einem Systemoszillator (**1301**) abgeleitet wird, wobei das erste Modul ein erstes Anzeigesignal (**1208**) liefert;
ein zweites Modul (**218**), das ein zweites Taktsignal benötigt, wobei das zweite Modul ein zweites Anzeigesignal (**1206**) liefert; und
eine Leistungsverwaltungsschaltung (**1220**, **1226**),
dadurch gekennzeichnet, dass
das erste Anzeigesignal einen ersten Zustand aufweist, der anzeigt, dass das erste Taktsignal erforderlich ist, und einen zweiten Zustand, der anzeigt, dass das erste Taktsignal nicht erforderlich ist;
das zweite Taktsignal vom Systemoszillator (**1301**) abgeleitet wird;
das zweite Anzeigesignal einen ersten Zustand aufweist, der anzeigt, dass das zweite Taktsignal erforderlich ist, und einen zweiten Zustand, der anzeigt, dass das zweite Taktsignal nicht erforderlich ist;
die Leistungsverwaltungsschaltung (**1220**, **1226**) für die ersten und zweiten Anzeigesignale zuständig ist, um den Systemoszillator (**1301**) zu deaktivieren, wenn das erste Anzeigesignal den zweiten Zustand angenommen hat und das zweite Anzeigesignal den zweiten Zustand angenommen hat; und

die Leistungsverwaltungsschaltung (**1220**, **1226**) für ein Aktivierungssignal für die Aktivierung des Systemoszillators zuständig ist.

2. Prozessor nach Anspruch 1, wobei die Leistungsverwaltungsschaltung ferner eine Schaltung aufweist zur Blockierung eines Durchgangs des ersten Taktsignals zum ersten Modul, wenn das erste Anzeigesignal den ersten Zustand angenommen hat, und zur Blockierung eines Durchgangs des zweiten Taktsignals zum zweiten Modul, wenn das zweite Anzeigesignal den zweiten Zustand angenommen hat.

3. Prozessor nach Anspruch 1, wobei das zweite Modul einen Zeitsteuerungs- und Ereignisprozessor aufweist und die Leistungsverwaltungsschaltung ferner eine Schaltung zur Aktivierung des Systemoszillators, wenn das zweite Anzeigesignal den ersten Zustand annimmt, aufweist.

4. Prozessor nach Anspruch 1, wobei die Leistungsverwaltungsschaltung den Systemoszillator als Antwort auf den Empfang eines Unterbrechungssignals aktiviert.

5. Prozessor nach Anspruch 1, wobei die Leistungsverwaltungsschaltung bewirkt, dass das erste Modul einen Leerlaufzustand verlässt, indem sie ein Unterbrechungssignal an das erste Modul sendet.

6. Prozessor nach Anspruch 5, wobei das erste Modul bewirkt, dass das erste Anzeigesignal den ersten Zustand annimmt, sobald es den Leerlaufzustand verlässt.

7. Prozessor nach Anspruch 1, wobei die Leistungsverwaltungsschaltung bewirkt, dass das zweite Modul einen Leerlaufzustand verlässt, indem es ein Unterbrechungssignal an das erste Modul sendet.

8. Prozessor nach Anspruch 7, wobei das zweite Modul bewirkt, dass das zweite Anzeigesignal den ersten Zustand annimmt, sobald es den Leerlaufzustand verlässt.

9. Prozessor nach Anspruch 1, wobei die Leistungsverwaltungsschaltung eine Schaltung zur Blockierung des Ausgangssignals des Systemoszillators für einen Zeitraum nach der Aktivierung des Oszillators aufweist.

10. Prozessor nach Anspruch 9, wobei die Schaltung zur Blockierung des Ausgangssignals des Systemoszillators ein Register zur Speicherung des Zeitraums als Oszillatoraufwärmzeit und einen Zähler zur Bestimmung, ob die Oszillatoraufwärmzeit abgelaufen ist, aufweist.

11. Prozessor nach Anspruch 10, wobei die Schaltung zur Blockierung des Ausgangssignals des Systemoszillators Mittel einschließt, die ein Ausgangssignal des Systemoszillators nach Ablauf der Aufwärmzeit durchlassen.

12. Prozessor nach Anspruch 1, der ferner einen Zeitsteuerungs- und Ereignisprozessor zur Erzeugung von Zeitsteuerungs- und Ereignissignalen für Zeitsteuerungsoperationen für die ersten und zweiten Module aufweist, wobei der Zeitsteuerungs- und Ereignisprozessor ein drittes Taktsignal verwendet, das vom Systemoszillator abgeleitet wird, und ein drittes Anzeigesignal liefert, das einen ersten Zustand annimmt, der anzeigt, dass das dritte Taktsignal nicht benötigt wird, und einen zweiten Zustand, der anzeigt, dass das dritte Taktsignal benötigt wird, wobei der Zeitsteuerungs- und Ereignisprozessor ein viertes Taktsignal, das von einem Langsamtaktoszillator abgeleitet wird, empfängt, und wobei der Zeitsteuerungs- und Ereignisprozessor seinen Betrieb fortsetzt, wenn das dritte Anzeigesignal den zweiten Zustand angenommen hat.

13. Prozessor nach Anspruch 12, wobei der Zeitsteuerungs- und Ereignisprozessor den Systemoszillator zu vorgegebener Zeit aktiviert, indem er den Zustand des dritten Anzeigesignals in den vierten Zustand ändert, wenn mindestens eines aus der Gruppe, die das erste Modul und das zweite Modul einschließt, die Verwendung des Systemoszillators verlangt.

14. Prozessor nach Anspruch 12, wobei der Zeitsteuerungs- und Ereignisprozessor bewirkt, dass das erste Modul einen Leerlaufzustand verlässt, indem er ein Unterbrechungssignal an das erste Modul sendet.

15. Prozessor nach Anspruch 14, wobei das erste Modul bewirkt, dass das erste Indikatorsignal den ersten Zustand annimmt, nachdem es den Leerlaufzustand verlassen hat.

16. Prozessor nach Anspruch 12, wobei der Zeitsteuerungs- und Ereignisprozessor bewirkt, dass das zweite Modul einen Leerlaufzustand verlässt, indem er ein Unterbrechungssignal an das zweite Modul sendet.

17. Prozessor nach Anspruch 16, wobei das zweite Modul bewirkt, dass das Anzeigesignal den ersten Zustand annimmt, nachdem es den Leerlaufzustand verlassen hat.

18. Verfahren zur Steuerung eines Basisbandprozessors für drahtlose Anwendungen, das Folgendes umfasst:
Empfangen eines ersten Indikatorsignals (**1208**) von einem ersten Modul (**208**), das ein erstes Taktsignal benötigt, das von einem Systemoszillator (**1301**) abgeleitet wird, wobei das erste Anzeigesignal einen ersten Zustand aufweist, der anzeigt, dass das erste Taktsignal benötigt wird, und einen zweiten Zustand, der anzeigt, dass das erste Taktsignal nicht benötigt wird;
Empfangen eines zweiten Anzeigesignals (**1206**) von einem zweiten Modul (**218**), das ein zweites Taktsignal benötigt, das vom Systemoszillator (**1301**) abgeleitet wird, wobei das zweite Anzeigesignal einen ersten Zustand aufweist, der anzeigt, dass das zweite Taktsignal erforderlich ist, und einen zweiten Zustand, der anzeigt, dass das zweite Taktsignal nicht erforderlich ist;
Deaktivierung des Systemoszillators (**1301**), wenn das erste Anzeigesignal (**1208**) den zweiten Zustand angenommen hat, und das zweite Anzeigesignal (**1206**) den zweiten Zustand angenommen hat; und
Aktivieren des Systemoszillators (**1301**), wenn ein Ausgangssignal vom Systemoszillator von mindestens einem aus der Gruppe, die das erste Modul (**208**) und das zweite Modul (**218**) einschließt, benötigt wird.

19. Verfahren nach Anspruch 18, das ferner folgendes aufweist:
Blockierung des Durchgangs des ersten Taktsignals zum ersten Modul, wenn das erste Anzeigesignal den zweiten Zustand angenommen hat, und Blockieren des Durchgangs des zweiten Taktsignals zum zweiten Modul, wenn das zweite Anzeigesignal den zweiten Zustand angenommen hat.

20. Verfahren nach Anspruch 18, wobei der Vorgang der Deaktivierung des Oszillators ferner eine Blockierung des Ausgangssignals vom Systemoszillator für einen Zeitraum nach der Aktivierung des Oszillators umfasst.

21. Verfahren nach Anspruch 20, wobei der Vorgang der Blockierung des Ausgangssignals des Systemoszillators ferner eine Speicherung des Zeitraums als Oszillatoraufwärmzeit und die Verwendung eines Zählers, um zu bestimmen, wann die Oszillatoraufwärmzeit abgelaufen ist, umfasst.

22. Verfahren nach Anspruch 21, wobei der Vorgang der Blockierung des Ausgangssignals des Systemoszillators ferner umfasst, dass das Ausgangssignal des Systemoszillators durchgelassen wird, wenn die Aufwärmzeit abgelaufen ist.

23. Verfahren nach Anspruch 22, das ferner die Bereitstellung eines Zeitsteuerungs- und Ereignisprozessors zur Erzeugung von Zeitsteuerungs- und Ereignissignalen für Zeitsteuerungsoperationen für die ersten und zweiten Module umfasst, wobei der Zeitsteuerungs- und Ereignisprozessor ein drittes Taktsignal verwendet, das vom Systemoszillator abgleitet wird, und ein drittes Anzeigesignal liefert, das einen ersten Zustand aufweist, der anzeigt, dass das dritte Taktsignal nicht erforderlich ist, und einen zweiten Zustand, der anzeigt, dass das dritte Taktsignal erforderlich ist, und wobei der Zeitsteuerungs- und Ereignisprozessor ein viertes Taktsignal empfängt, das von einem Langsamtaktoszillator abgeleitet wird, und das Verfahren ferner den Betrieb des Zeitsteuerungs- und Ereignisprozessors unter Verwendung des vierten Taktsignals, wenn das dritte Anzeigesignal den zweiten Zustand angenommen hat, umfasst.

24. Verfahren nach Anspruch 23, das ferner umfasst, dass der Zeitsteuerungs- und Ereignisprozessor die Inkraftsetzung des Systemoszillators zu einer präzise vorgegebenen Zeit umfasst, indem der Zustand des dritten Anzeigesignals in den ersten Zustand geändert wird, wenn mindestens eines aus der Gruppe, die das erste Modul und das zweite Modul umfasst, die Verwendung des Systemoszillators erfordert.

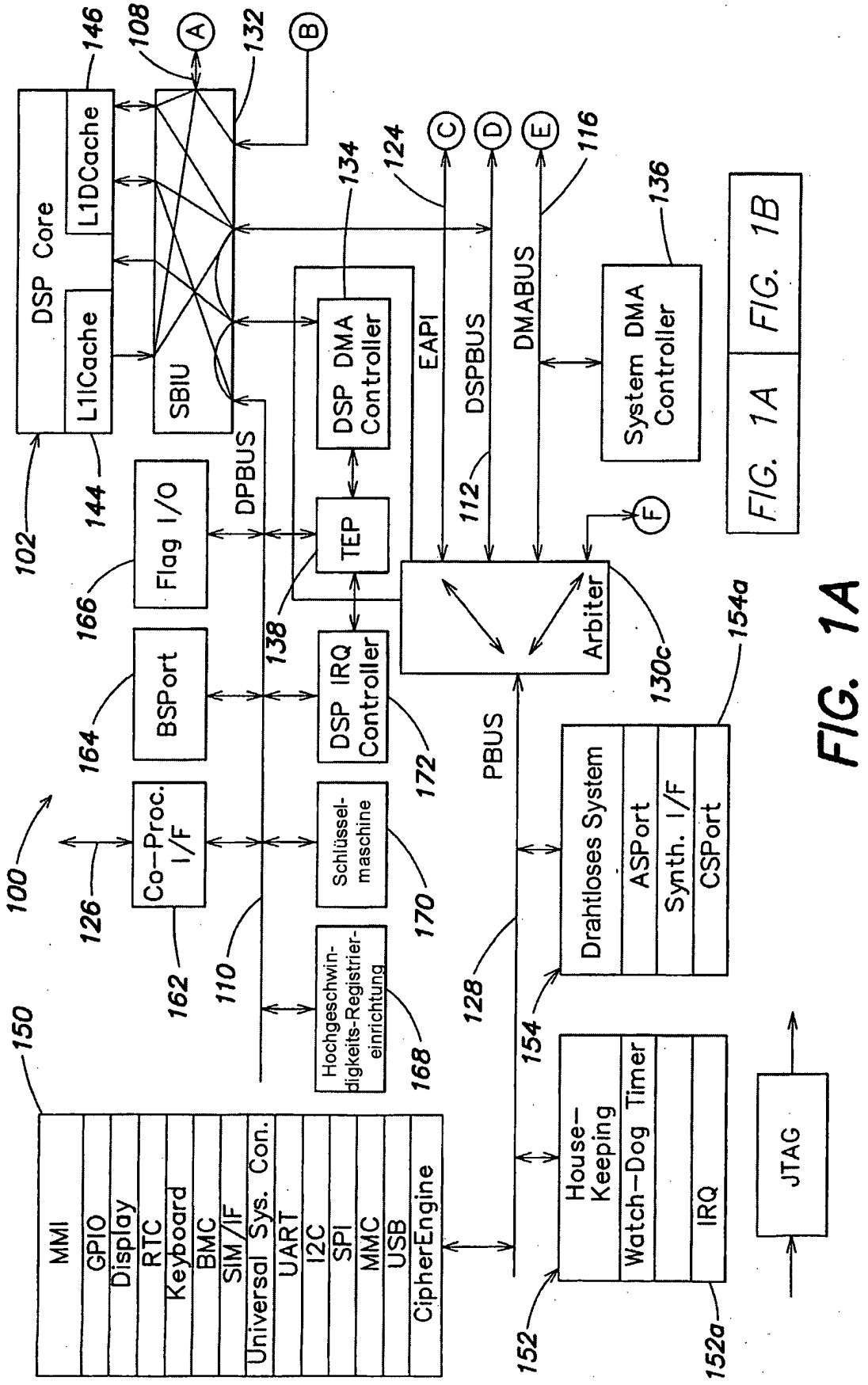
25. Prozessor nach Anspruch 1, der ferner Folgendes aufweist:
eine Phase-Locked-Loop-Schaltung zur Erhöhung der Frequenz eines dritten Taktsignals, das vom Systemoszillator erzeugt wird, um das erste Taktsignal zu erzeugen;
eine Phase-Locked-Loop-Umgehung, um zu bewirken, dass das dritte Taktsignal die Phase-Locked-Loop-Schaltung umgeht, und das als erstes Taktsignal dient, wenn das erste Anzeigesignal den zweiten Zustand angenommen hat.

26. Prozessor nach Anspruch 18, der ferner Folgendes aufweist:
Schaffung einer Phase-Locked-Loop-Schaltung zur Erhöhung einer Frequenz eines dritten Taktsignals, das vom Systemoszillator erzeugt wird, um das erste Taktsignal zu erzeugen; und

Bewirken, dass das dritte Taktsignal die Phase-Locked-Loop-Schaltung umgeht, und als erstes Taktsignal dient, wenn das erste Anzeigesignal den zweiten Zustand angenommen hat.

Es folgen 24 Blatt Zeichnungen

Anhängende Zeichnungen



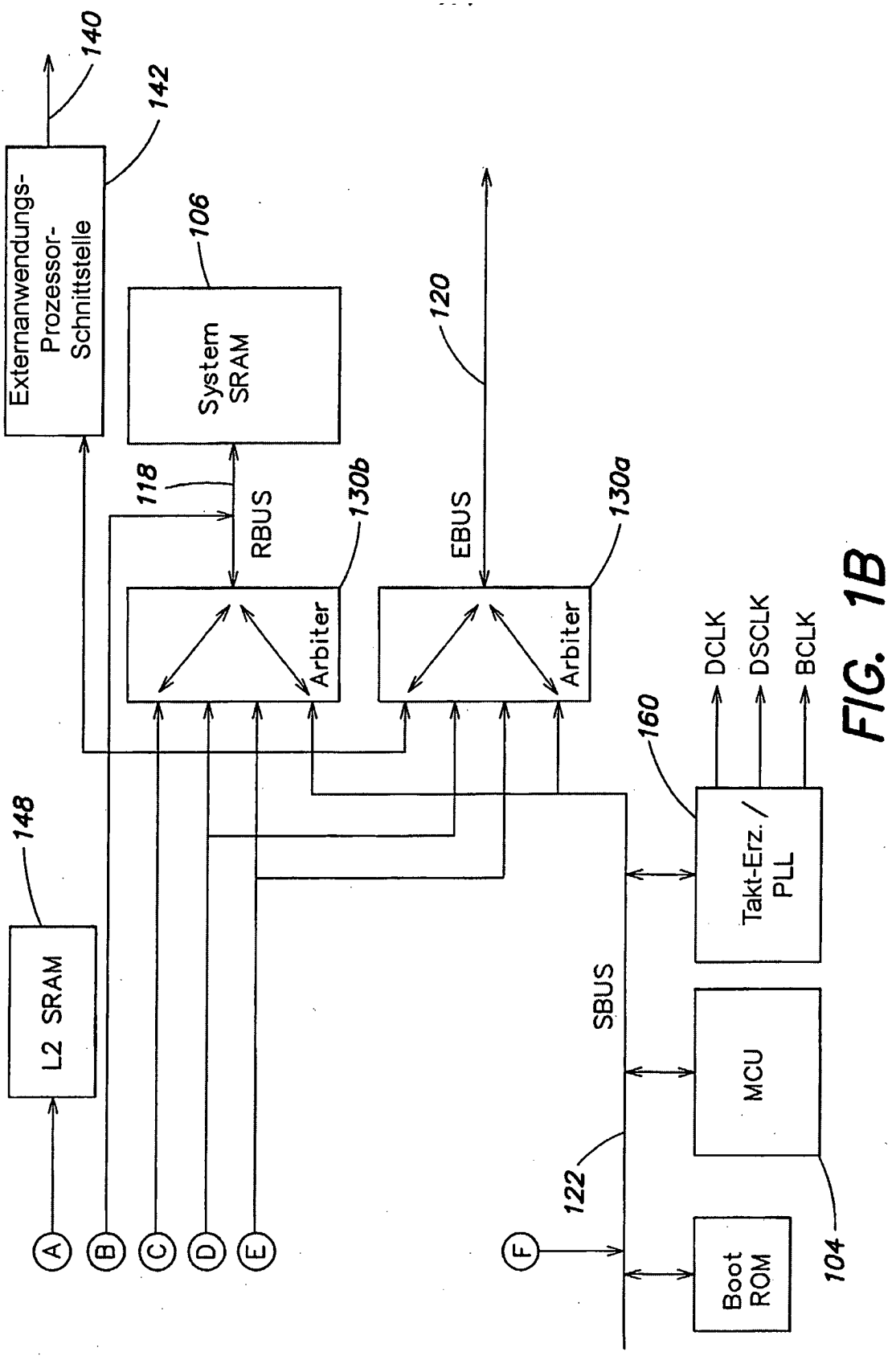

140
142
Externanwendungs-Prozessor-Schnittstelle
106
System SRAM
120
118
RBUS
130b
EBUS
130a
Arbiter
Arbiter
DCLK
DSCLK
BCLK
160
148
L2 SRAM
Takt-Erz. / PLL
SBUS
MCU
122
Boot ROM
104
A
B
C
D
E
F

FIG. 1B

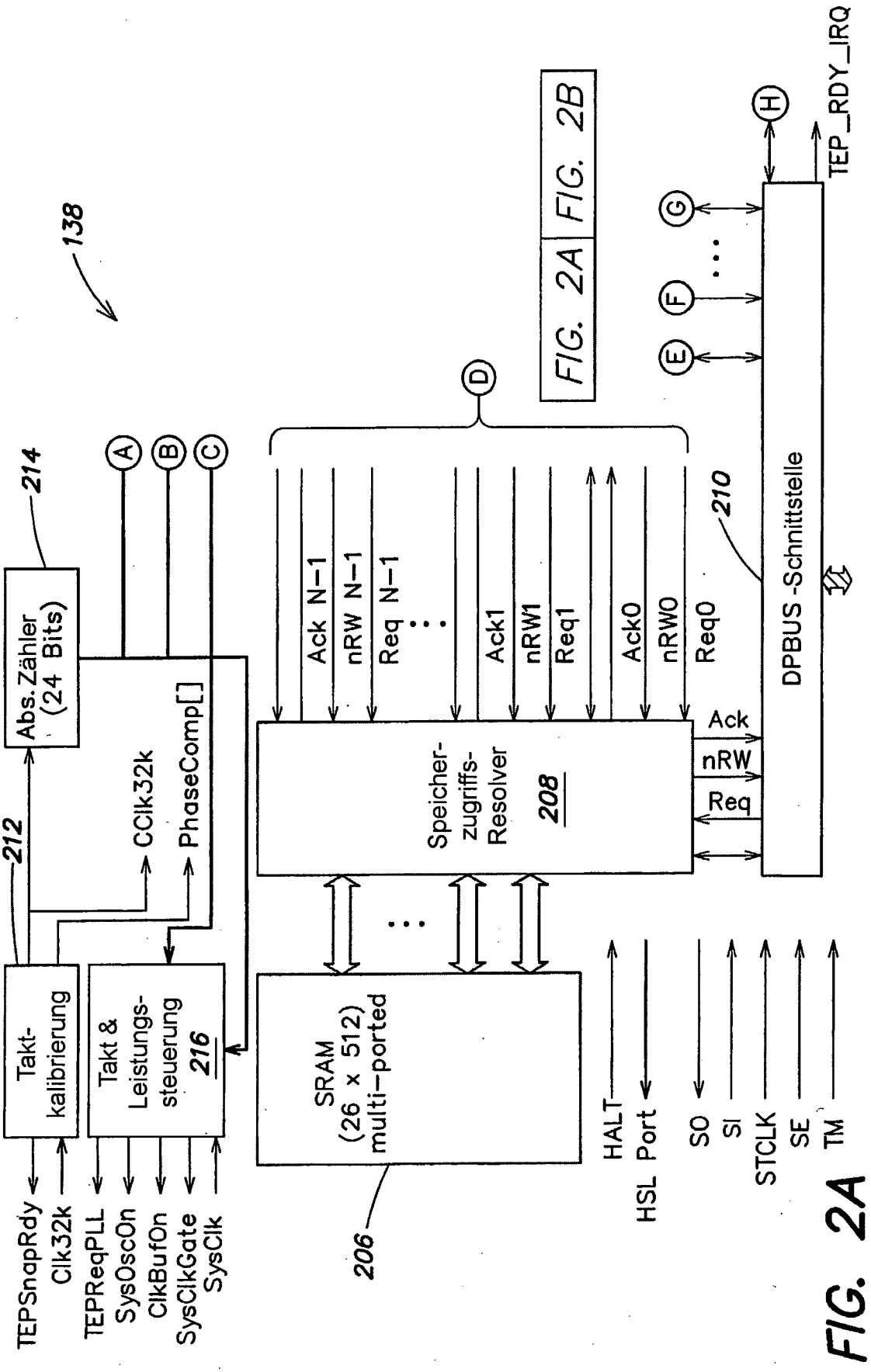

138
214
Abs. Zähler (24 Bits)
212
CClk32k
PhaseComp[]
Takt-kalibrierung
Takt & Leistungs-steuerung 216
TEPSnapRdy
Clk32k
TEPReqPLL
SysOscOn
ClkBufOn
SysClkGate
SysClk
206
SRAM (26 x 512) multi-ported
Speicher-zugriffs-Resolver 208
Ack N-1
nRW N-1
Req N-1
Ack1
nRW1
Req1
Ack0
nRW0
Req0
Ack
nRW
Req
210
DPBUS-Schnittstelle
TEP_RDY_IRQ
HALT
HSL Port
SO
SI
STCLK
SE
TM
A
B
C
D
E
F
G
H
FIG. 2A
FIG. 2B

FIG. 2A

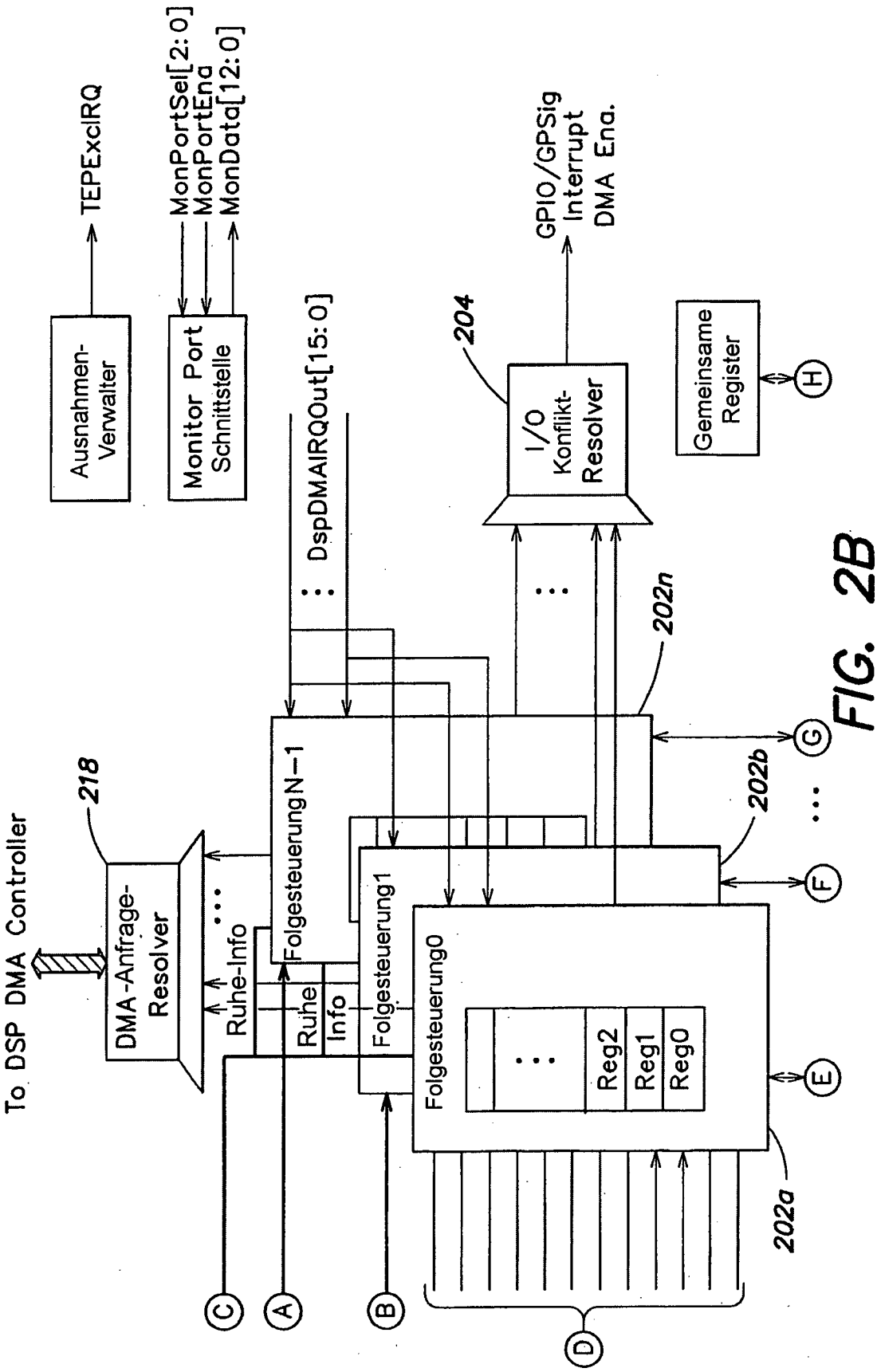

TEPExcIRQ
MonPortSel[2:0]
MonPortEna
MonData[12:0]
GPIO/GPSig
Interrupt
DMA Ena.
Ausnahmen-Verwalter
Monitor Port Schnittstelle
DspDMAIRQOut[15:0]
204
I/O Konflikt-Resolver
Gemeinsame Register
202n
202b
202a
218
To DSP DMA Controller
DMA-Anfrage-Resolver
Ruhe-Info
Ruhe
Info
FolgesteuerungN-1
Folgesteuerung1
Folgesteuerung0
Reg2
Reg1
Reg0
A
B
C
D
E
F
G
H

FIG. 2B

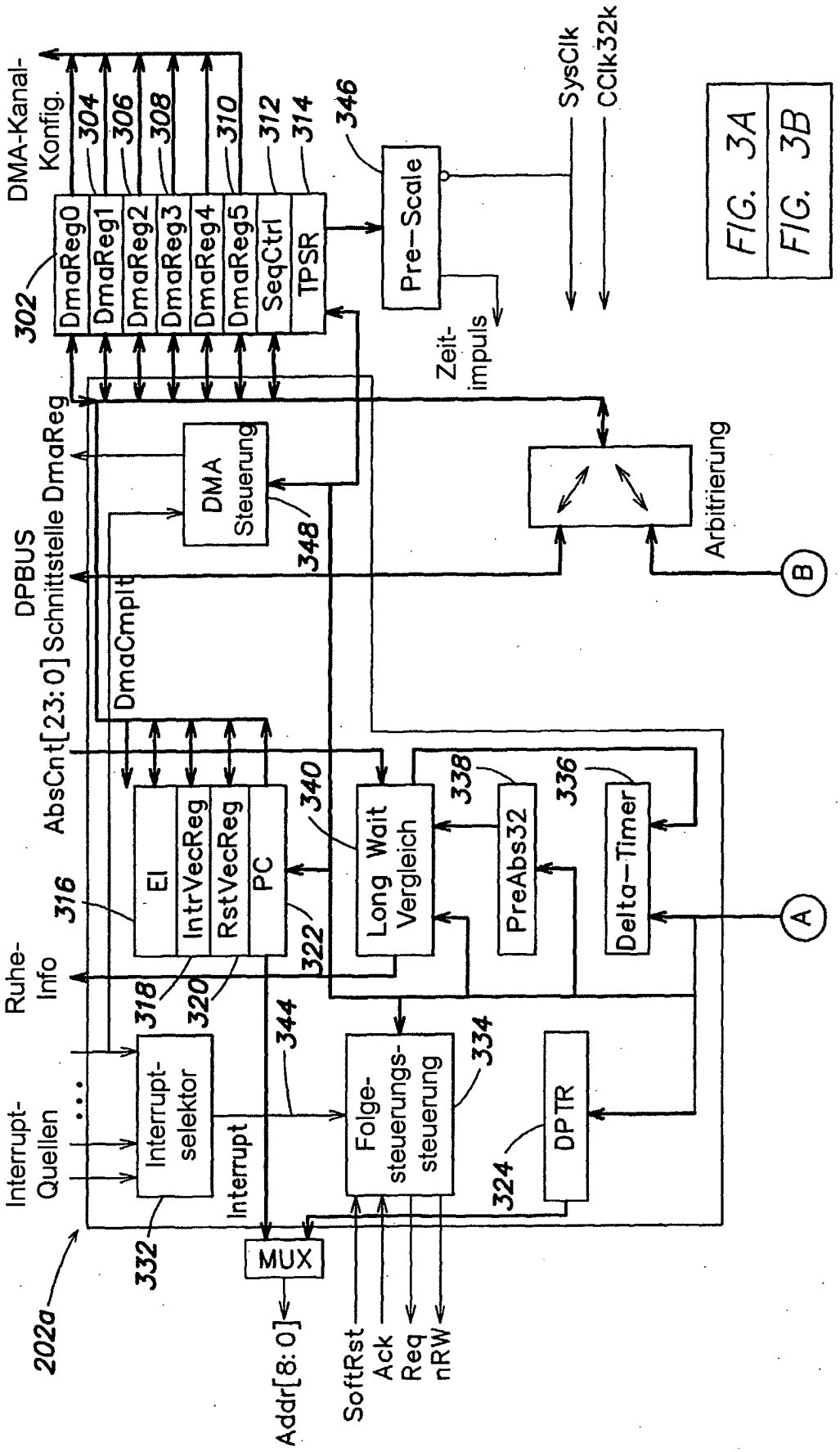


FIG. 3A

FIG. 3A
FIG. 3B

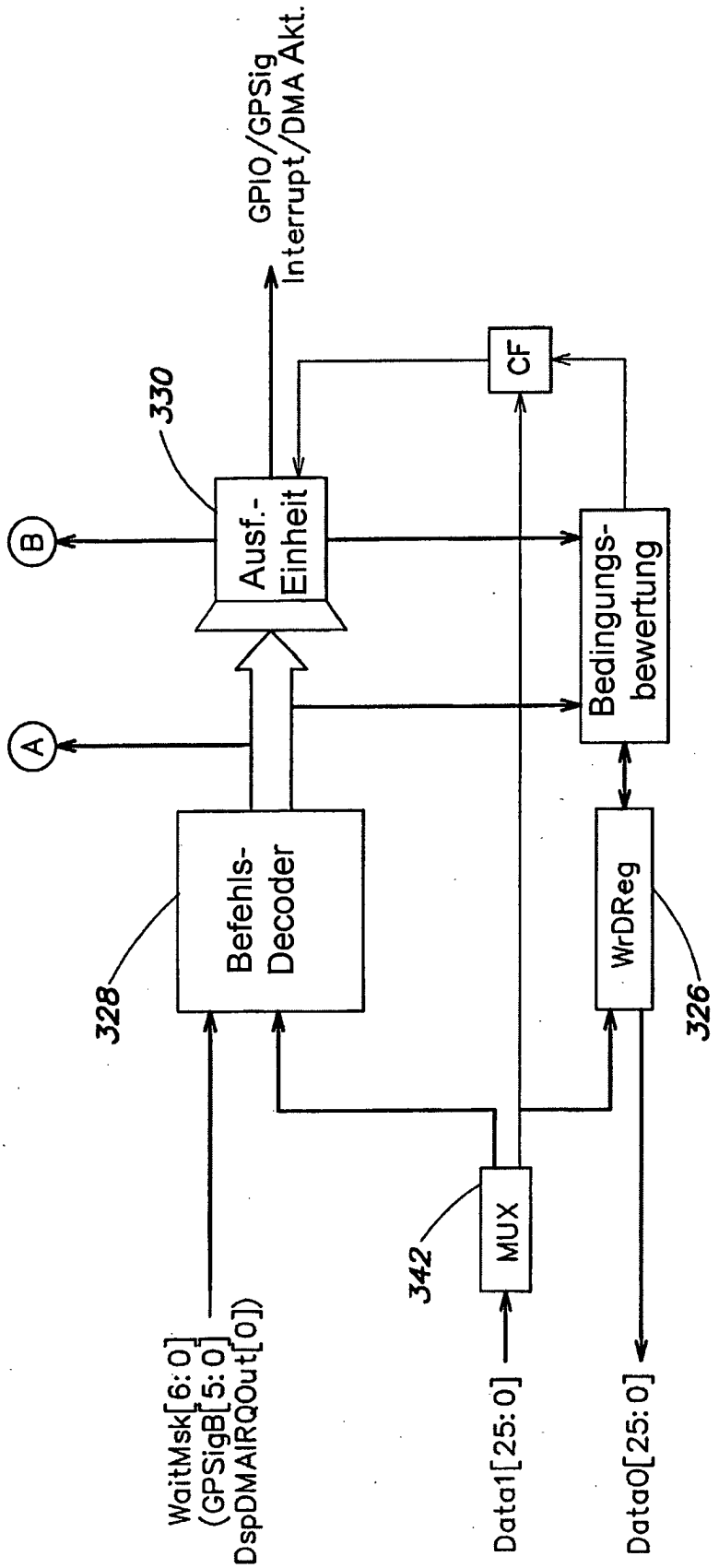

GPIO/GPSig
Interrupt/DMA Akt.
330
CF
B
Ausf.-
Einheit
Bedingungs-
bewertung
A
Befehls-
Decoder
WrDReg
328
326
342
MUX
WaitMsk[6:0]
(GPSigB[5:0]
DspDMAIRQOut[0])
Data1[25:0]
Data0[25:0]

FIG. 3B

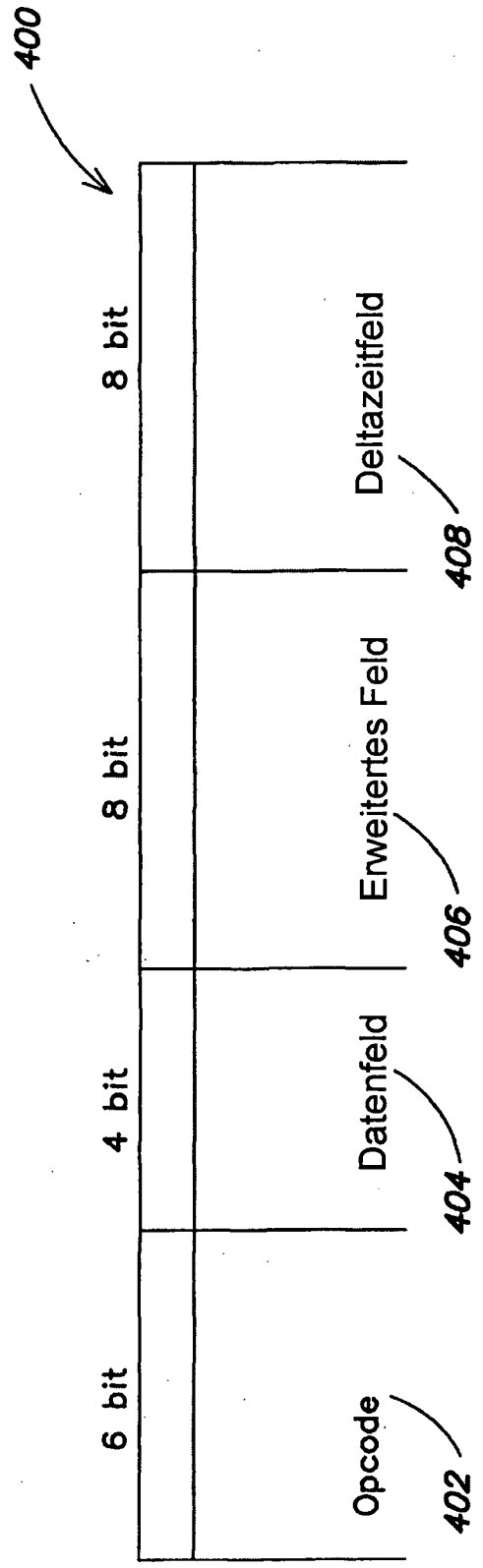

400
6 bit
4 bit
8 bit
8 bit
Opcode
Datenfeld
Erweitertes Feld
Deltazeitfeld
402
404
406
408

FIG. 4

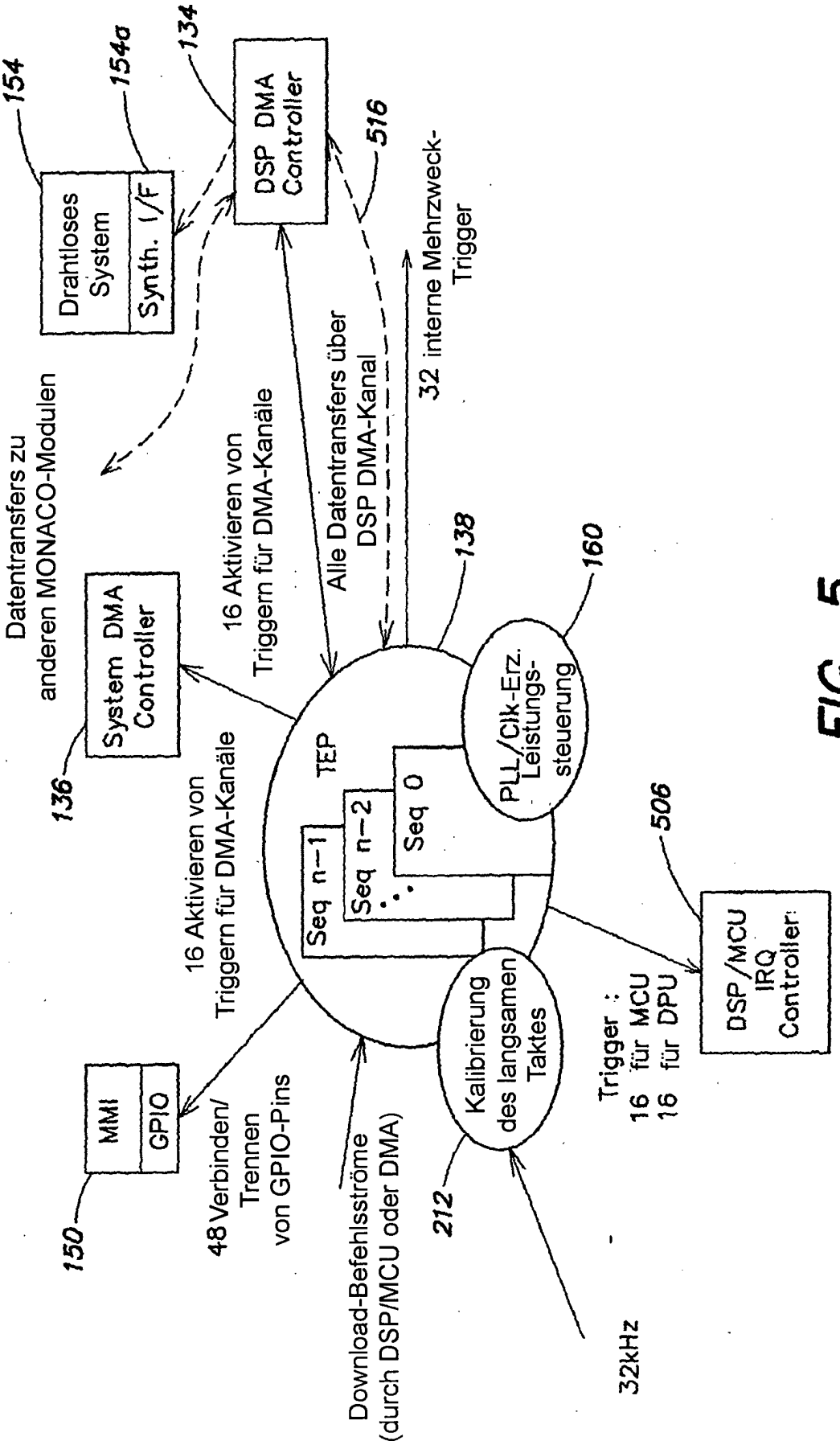

154
154a
134
Drahtloses System
Synth. I/F
DSP DMA Controller
516
Datentransfers zu anderen MONACO-Modulen
16 Aktivieren von Triggern für DMA-Kanäle
Alle Datentransfers über DSP DMA-Kanal
32 interne Mehrzweck-Trigger
System DMA Controller
136
138
160
TEP
Seq n-1
Seq n-2
Seq 0
PLL/Clk-Erz. Leistungs-steuerung
506
16 Aktivieren von Triggern für DMA-Kanäle
DSP/MCU IRQ Controller
MMI
GPIO
150
48 Verbinden/ Trennen von GPIO-Pins
Download-Befehlsströme (durch DSP/MCU oder DMA)
212
Kalibrierung des langsamen Taktes
Trigger : 16 für MCU 16 für DPU
32kHz

FIG. 5

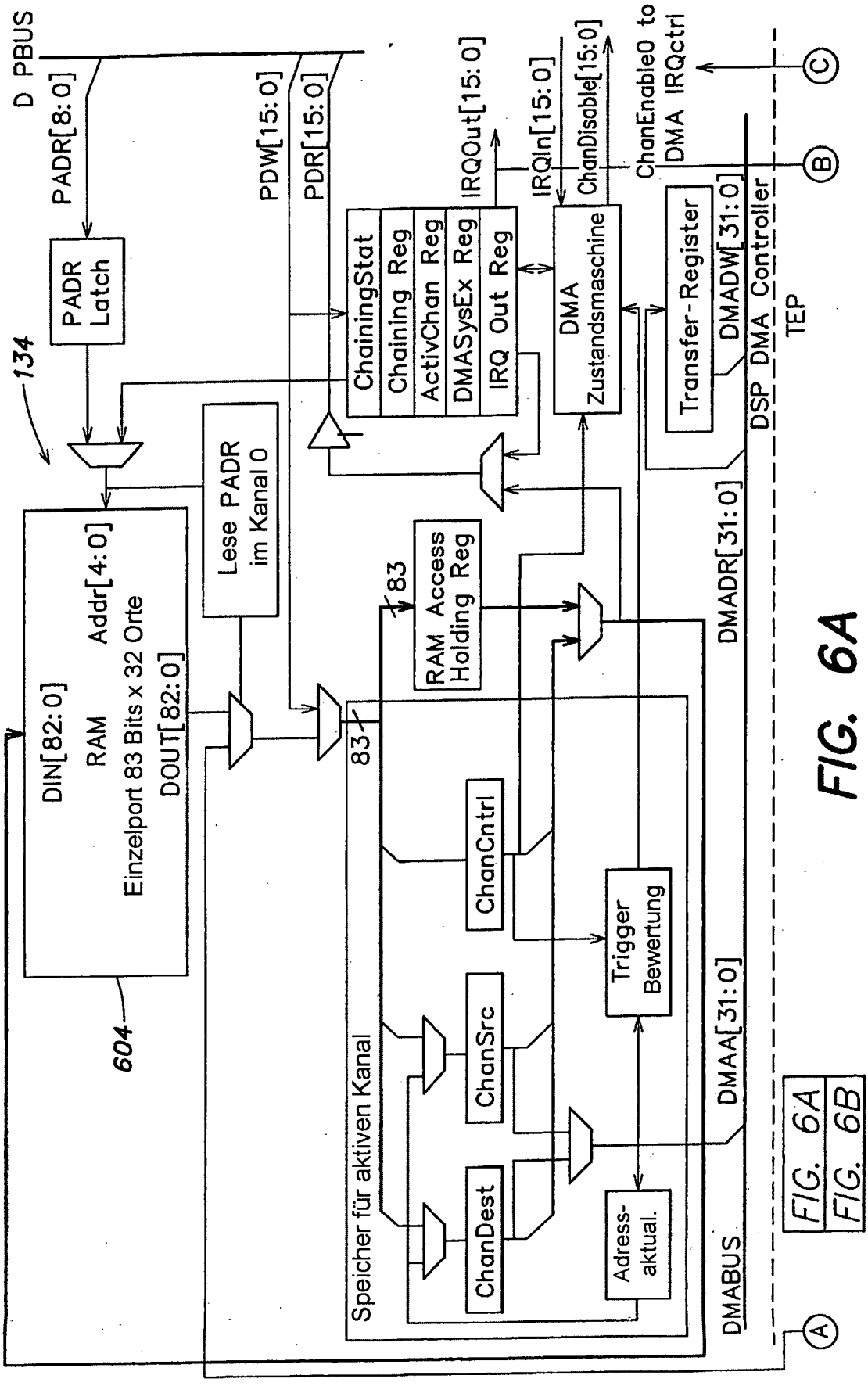


FIG. 6A

FIG. 6A
FIG. 6B

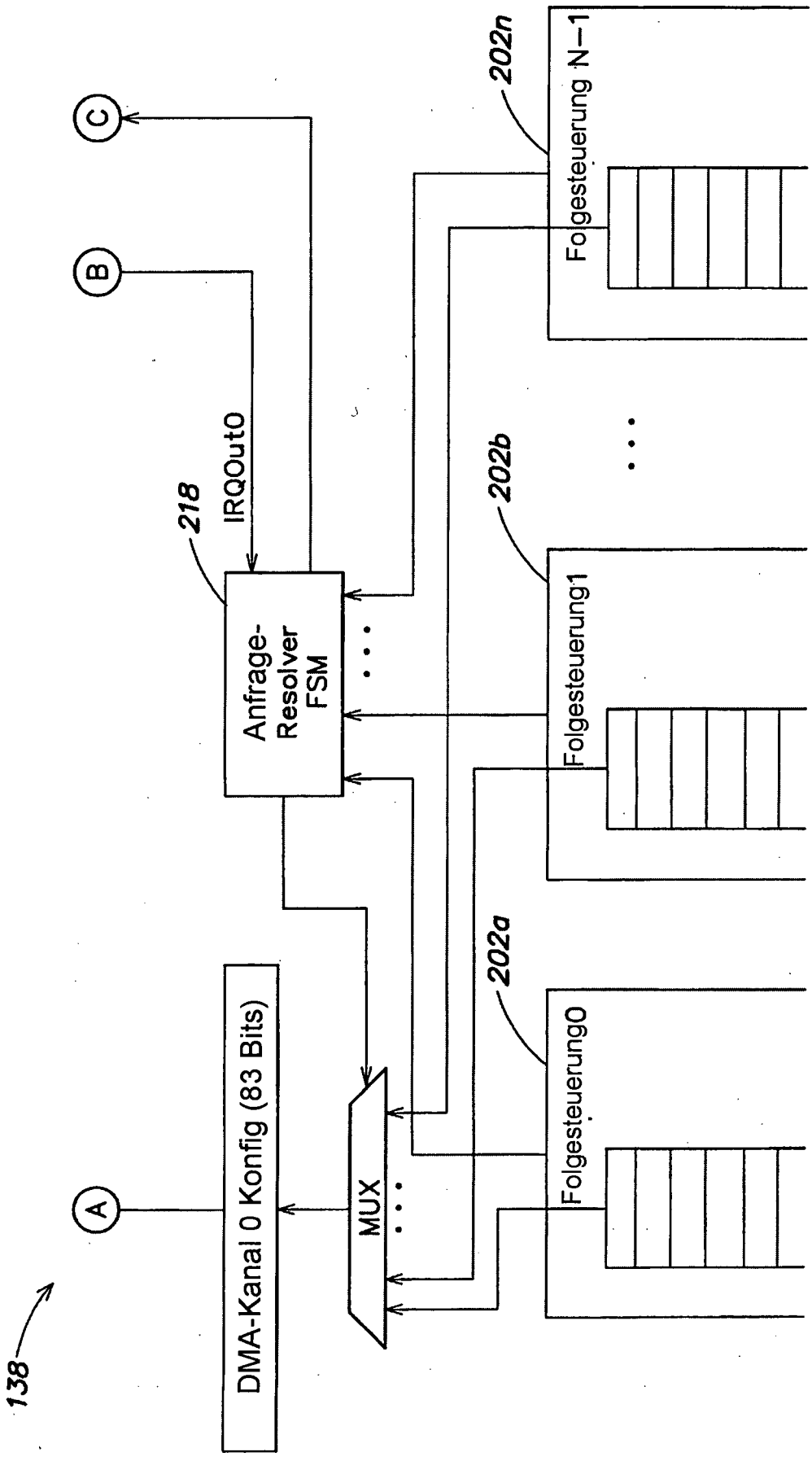

138
A
B
C
DMA-Kanal 0 Konfig (83 Bits)
MUX
Anfrage-Resolver FSM
218
IRQOut0
Folgesteuerung0
202a
Folgesteuerung1
202b
Folgesteuerung N-1
202n

FIG. 6B

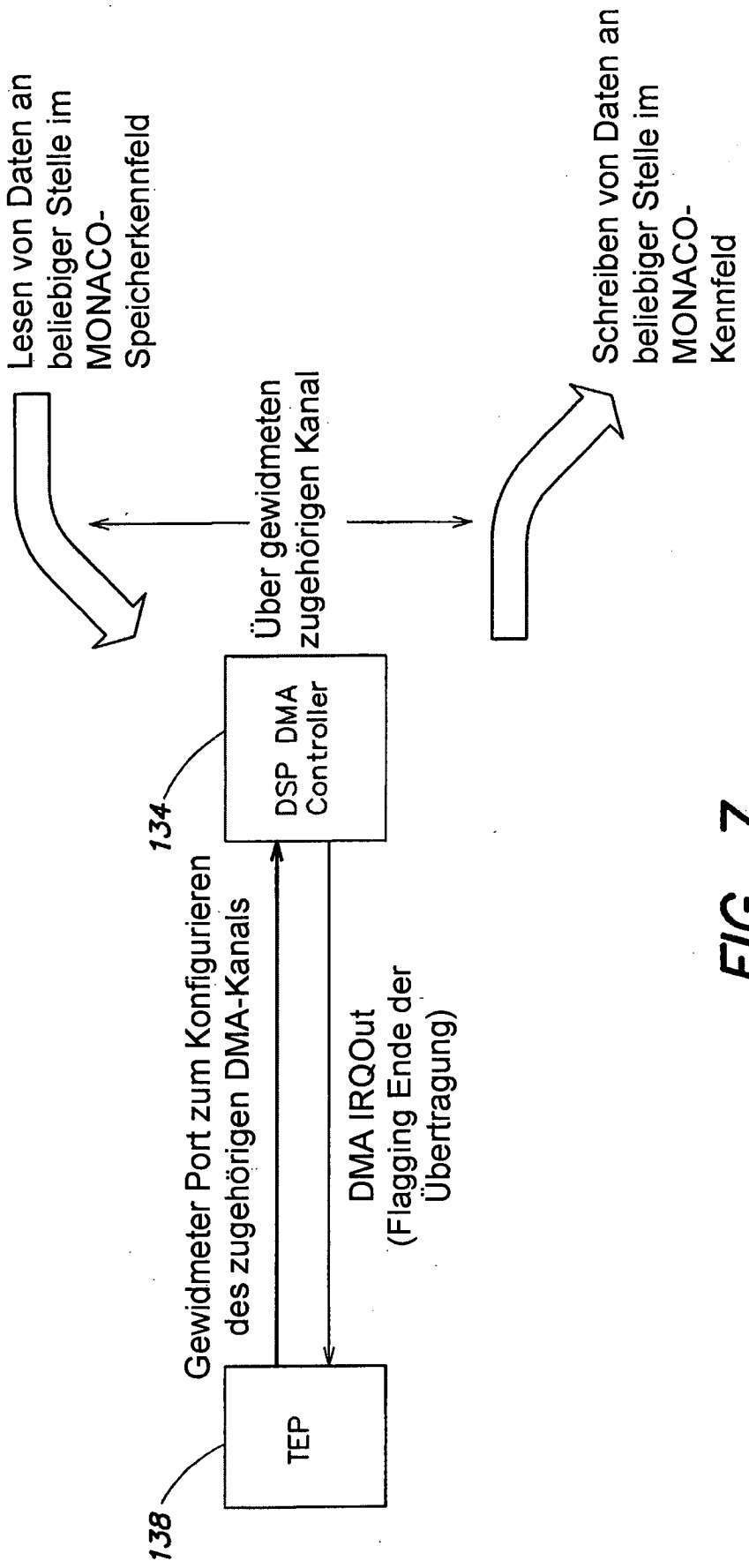

Lesen von Daten an beliebiger Stelle im MONACO-Speicherkennfeld
Über gewidmeten zugehörigen Kanal
Schreiben von Daten an beliebiger Stelle im MONACO-Kennfeld
DSP DMA Controller
134
Gewidmeter Port zum Konfigurieren des zugehörigen DMA-Kanals
DMA IRQOut (Flagging Ende der Übertragung)
TEP
138

FIG. 7

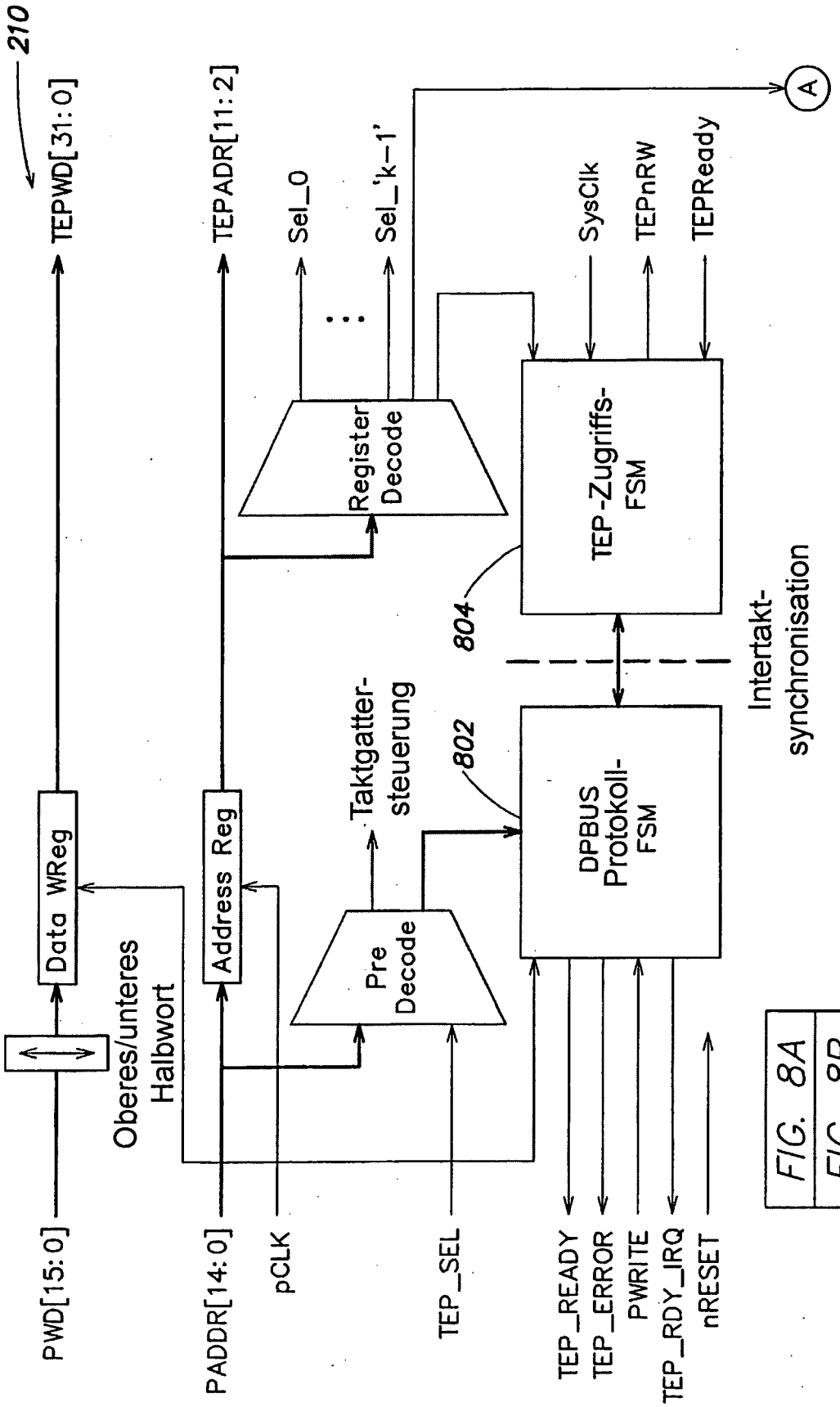


FIG. 8A

FIG. 8A
FIG. 8B

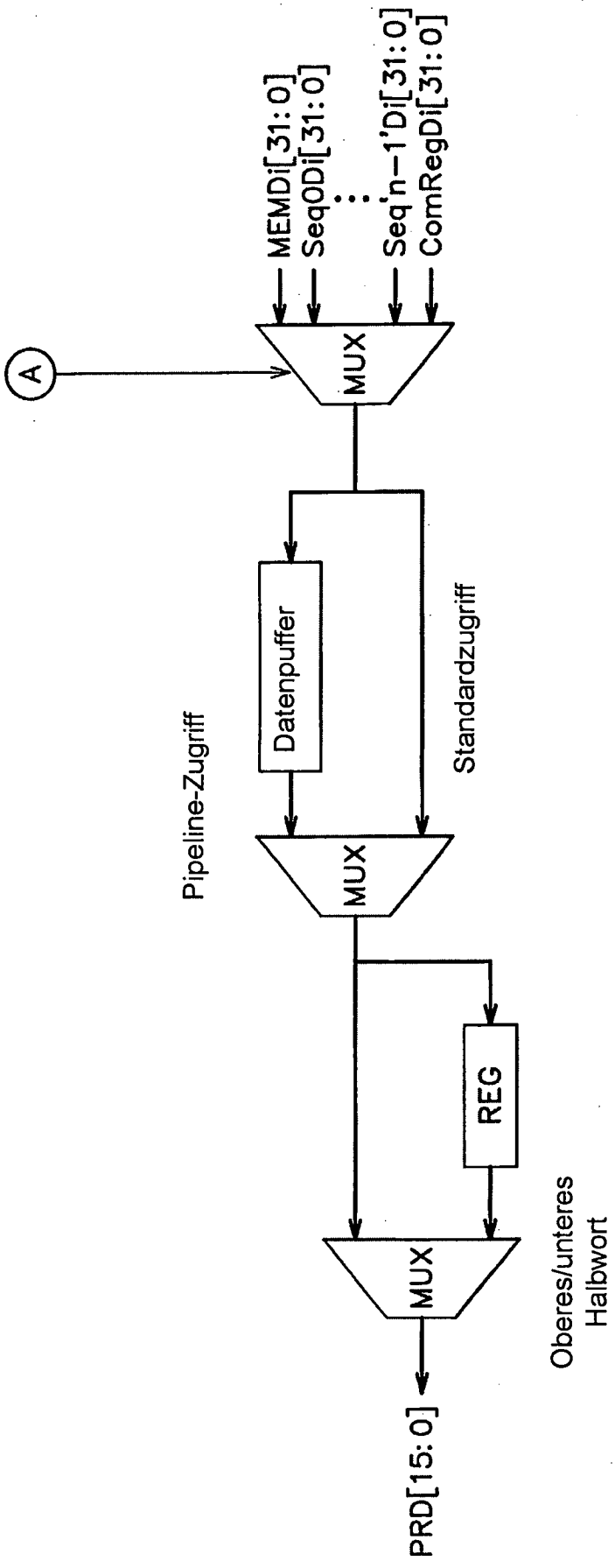

MEMDi[31:0]
SeqODi[31:0]
Seq'n−1'Di[31:0]
ComRegDi[31:0]
A
MUX
Datenpuffer
Pipeline-Zugriff
Standardzugriff
MUX
REG
MUX
Oberes/unteres Halbwort
PRD[15:0]

FIG. 8B

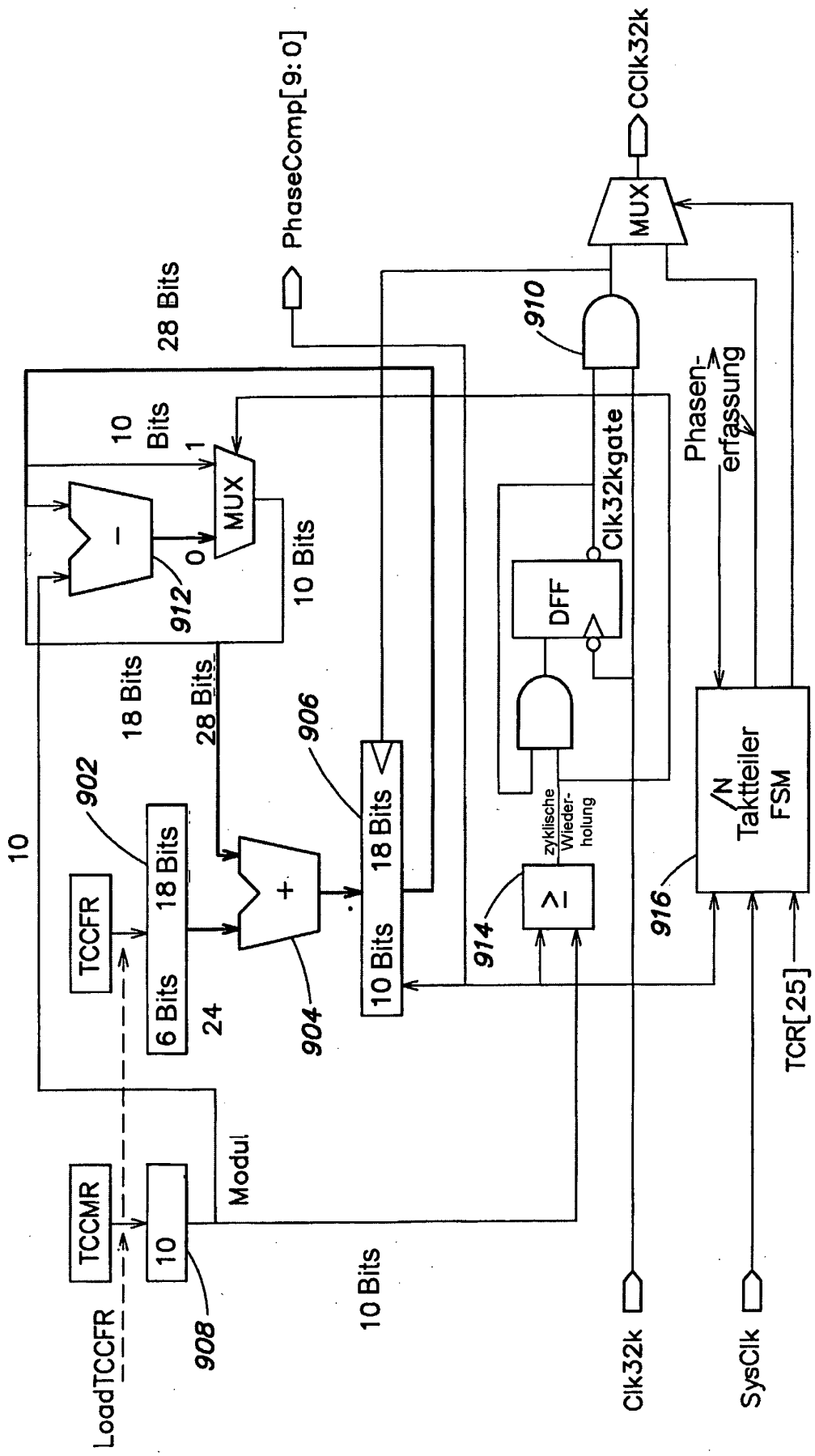

PhaseComp[9:0]
CClk32k
MUX
910
28 Bits
10 Bits
1
MUX
0
912
10 Bits
Clk32kgate
Phasen-erfassung
DFF
18 Bits
28 Bits
906
902
10
TCCFR
18 Bits
6 Bits
24
904
18 Bits
10 Bits
zyklische Wieder-holung
914
916
/N Taktteiler FSM
TCR[25]
TCCMR
10
908
Modul
10 Bits
LoadTCCFR
Clk32k
SysClk

FIG. 9A

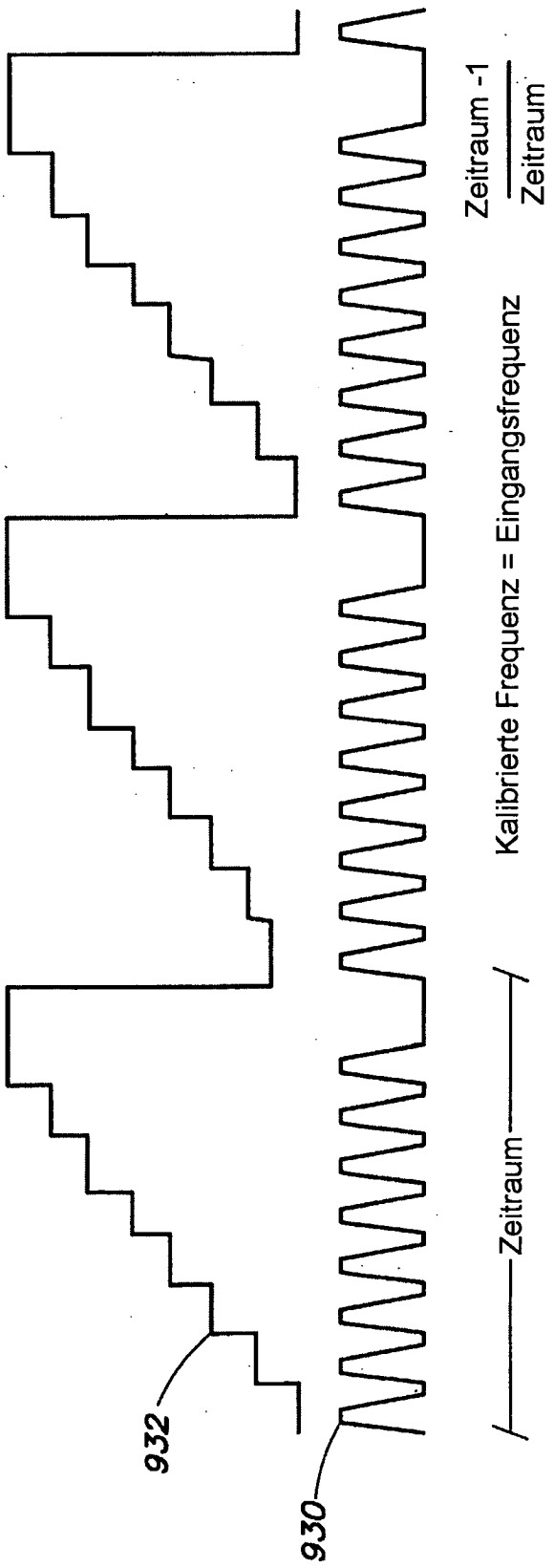

Zeitraum -1
Zeitraum
Kalibrierte Frequenz = Eingangsfrequenz
Zeitraum
932
930

FIG. 9B

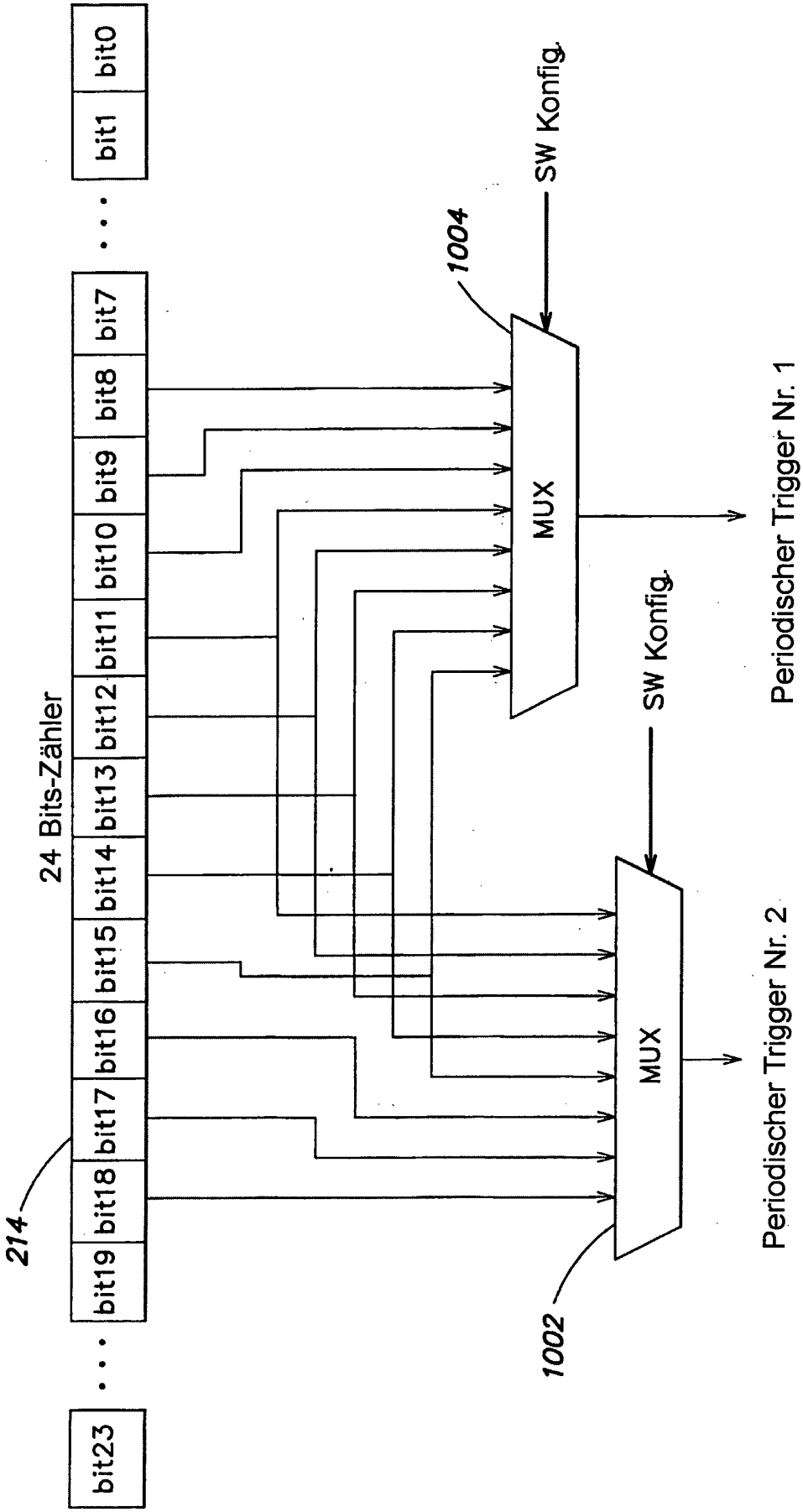

bit23
bit19
bit18
bit17
bit16
bit15
bit14
bit13
bit12
bit11
bit10
bit9
bit8
bit7
bit1
bit0
24 Bits-Zähler
214
1002
1004
MUX
MUX
SW Konfig.
SW Konfig.
Periodischer Trigger Nr. 2
Periodischer Trigger Nr. 1

FIG. 10

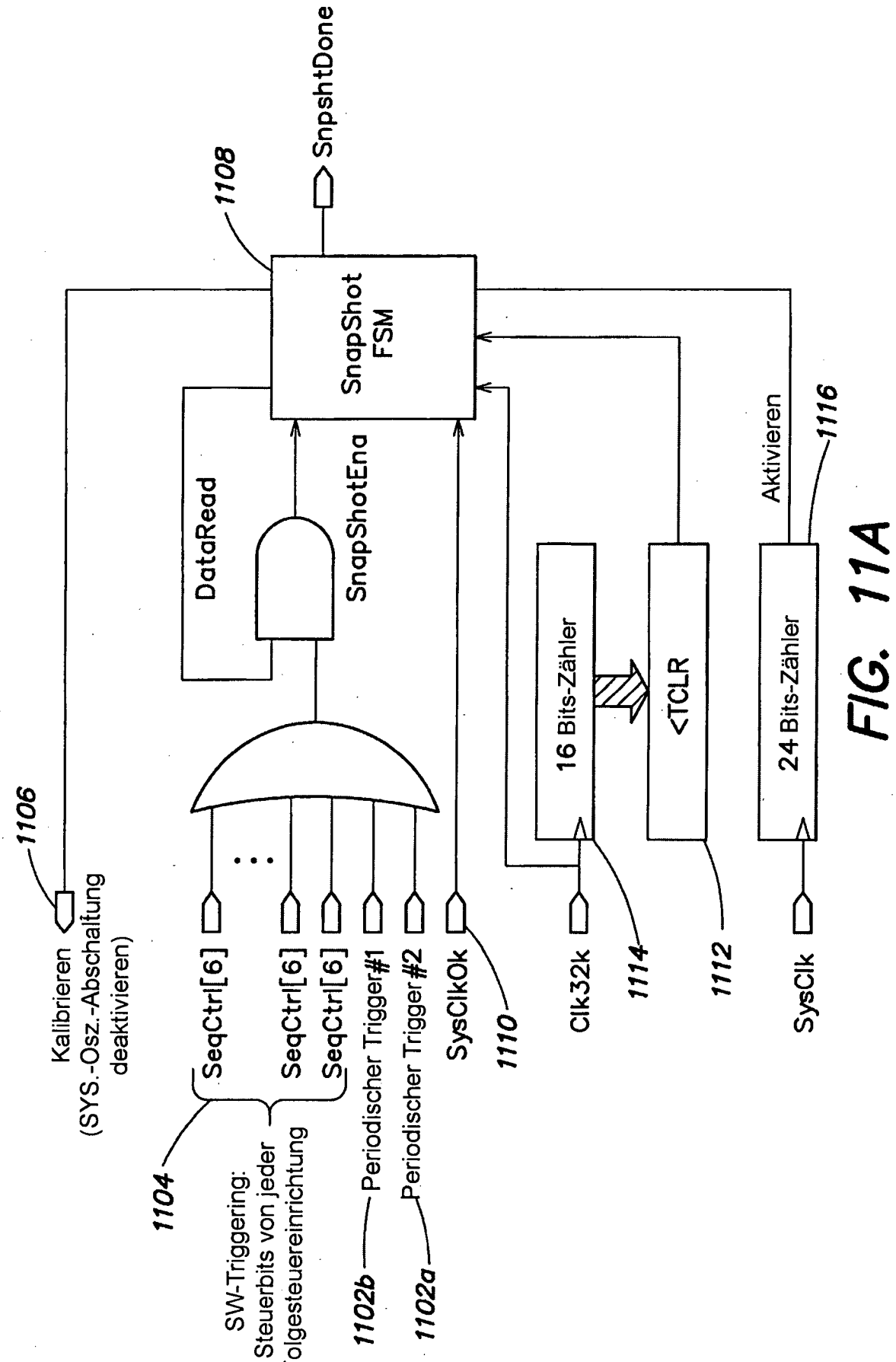

SnpshtDone
1108
SnapShot FSM
DataRead
SnapShotEna
Aktivieren
1116
16 Bits-Zähler
<TCLR
24 Bits-Zähler
1106
Kalibrieren (SYS.-Osz.-Abschaltung deaktivieren)
SeqCtrl[6]
SeqCtrl[6]
SeqCtrl[6]
Periodischer Trigger#1
Periodischer Trigger#2
SysClkOk
1110
Clk32k
1114
1112
SysClk
1104
SW-Triggering: Steuerbits von jeder Folgesteuereinrichtung
1102b
1102a

FIG. 11A

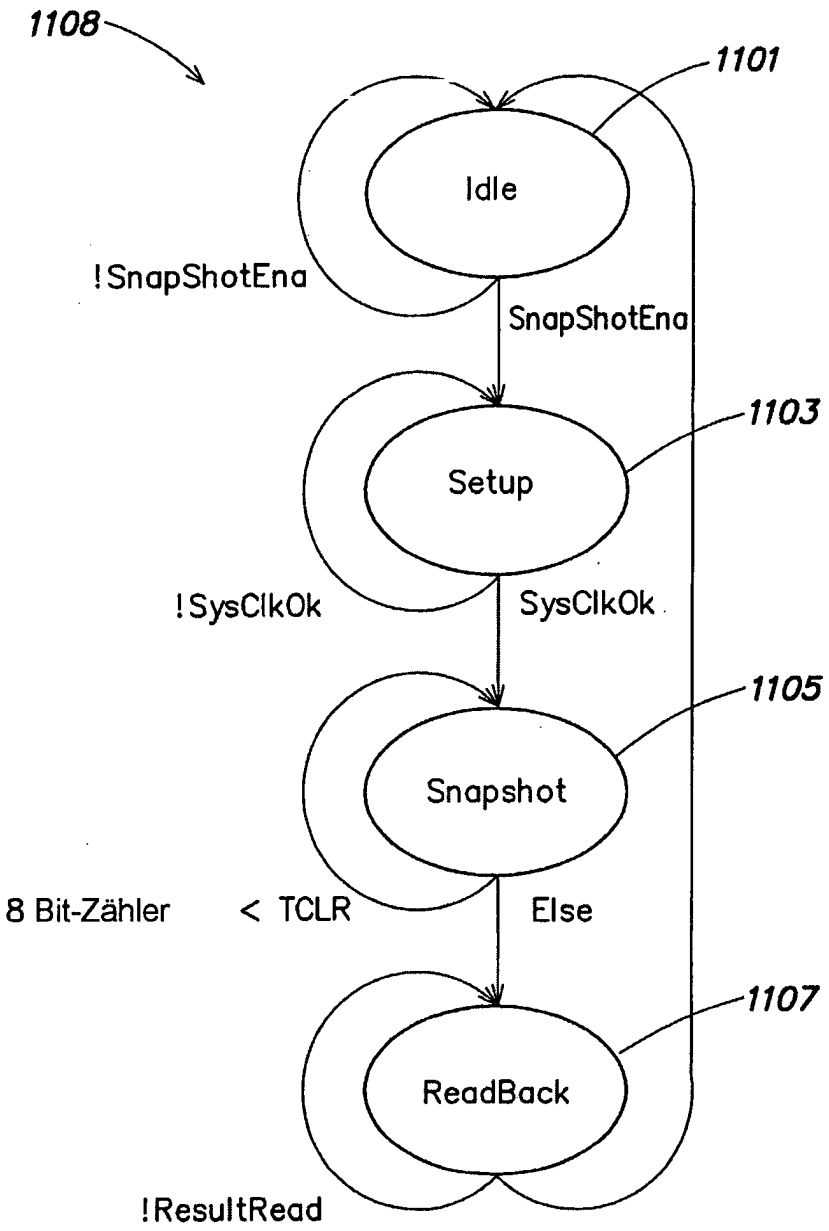

1108
1101
Idle
!SnapShotEna
SnapShotEna
1103
Setup
!SysClkOk
SysClkOk
1105
Snapshot
8 Bit-Zähler
< TCLR
Else
1107
ReadBack
!ResultRead

FIG. 11B

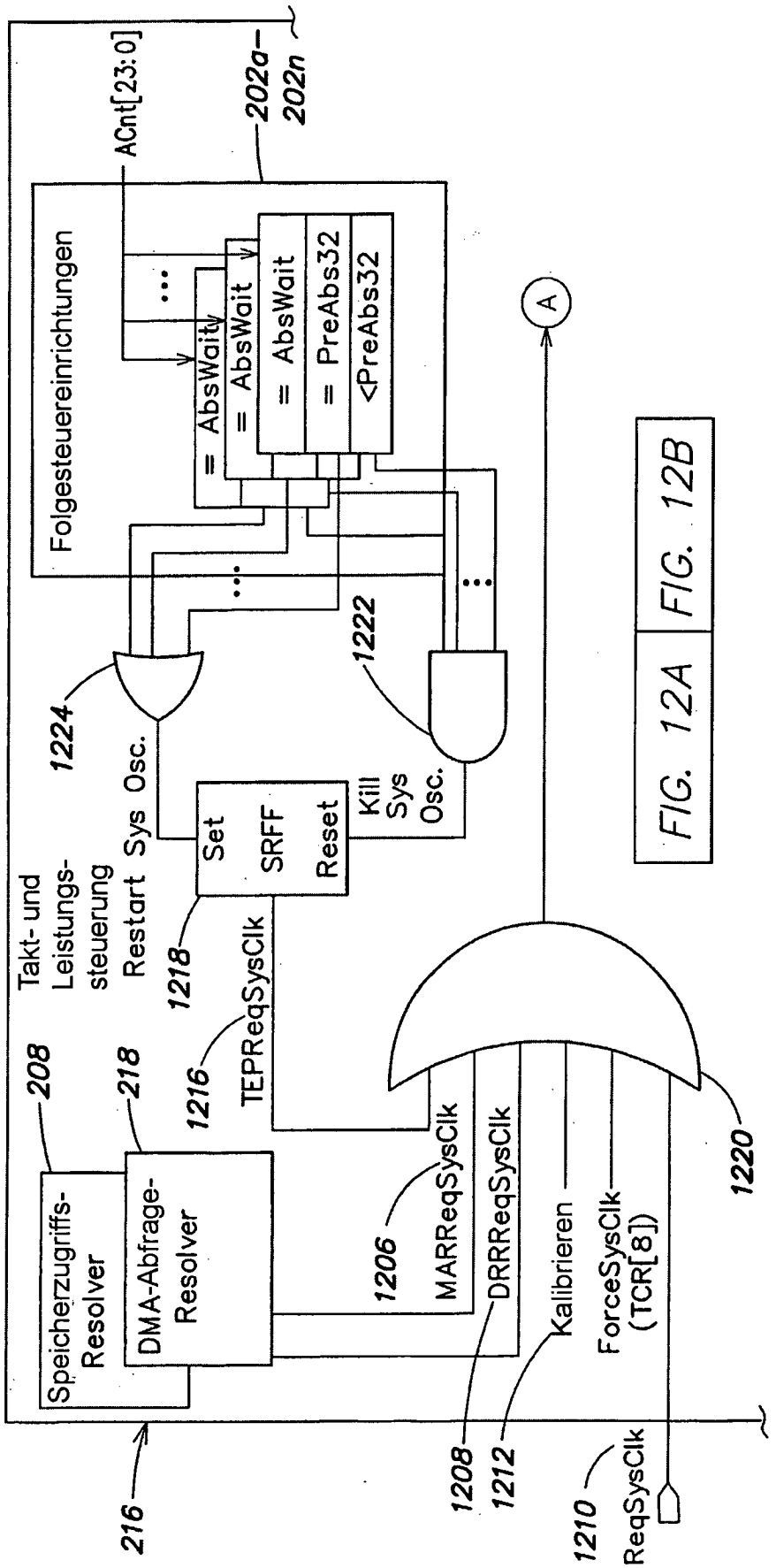


FIG. 12A

FIG. 12A FIG. 12B

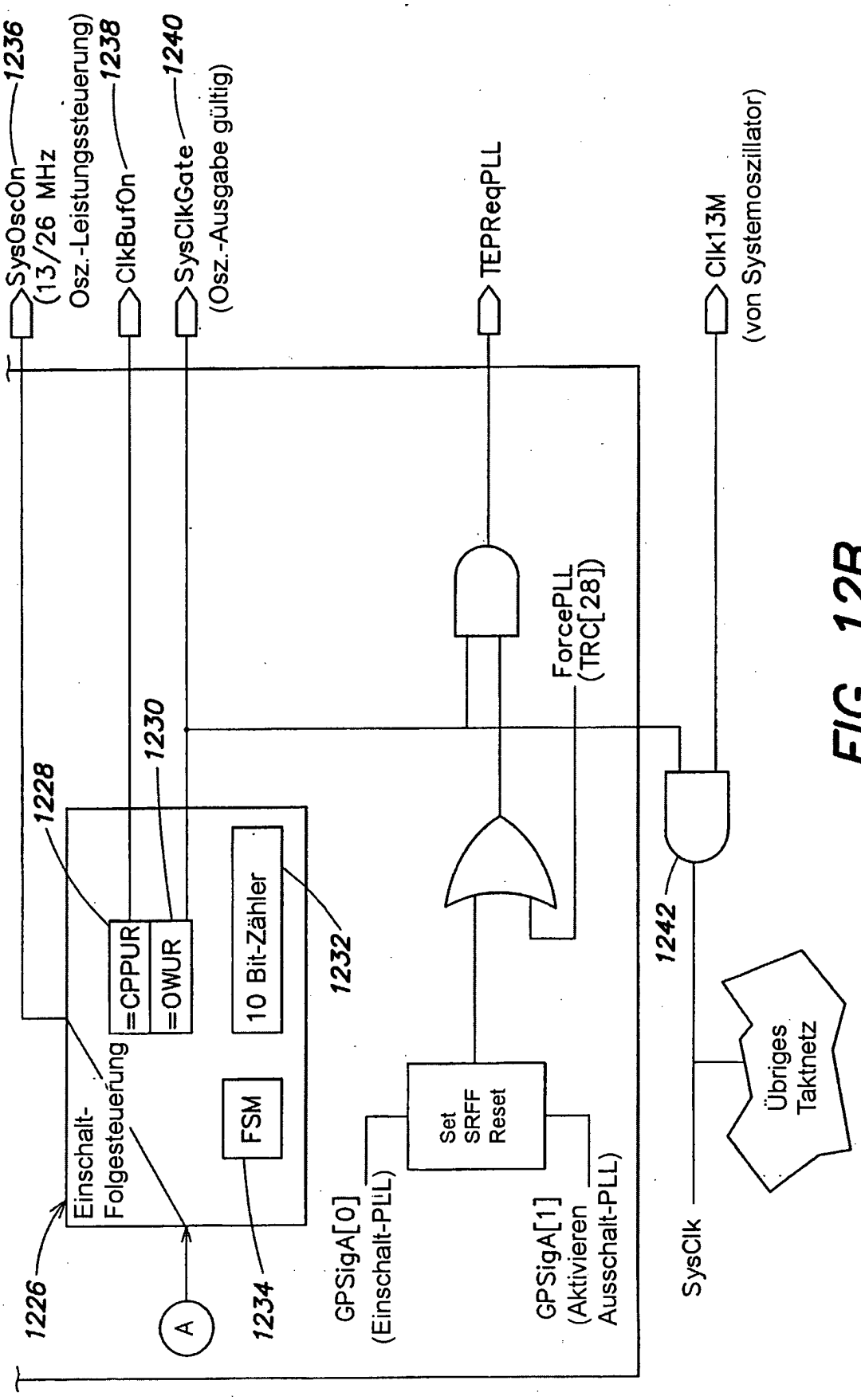

SysOscOn
(13/26 MHz
Osz.-Leistungssteuerung)
1236
ClkBufOn
1238
SysClkGate
1240
(Osz.-Ausgabe gültig)
TEPReqPLL
Clk13M
(von Systemoszillator)
1228
1230
=CPPUR
=OWUR
10 Bit-Zähler
1232
ForcePLL
(TRC[28])
Einschalt-
Folgesteuerung
FSM
1234
1226
A
GPSigA[0]
(Einschalt-PLL)
Set
SRFF
Reset
GPSigA[1]
(Aktivieren
Ausschalt-PLL)
1242
SysClk
Übriges
Taktnetz

FIG. 12B

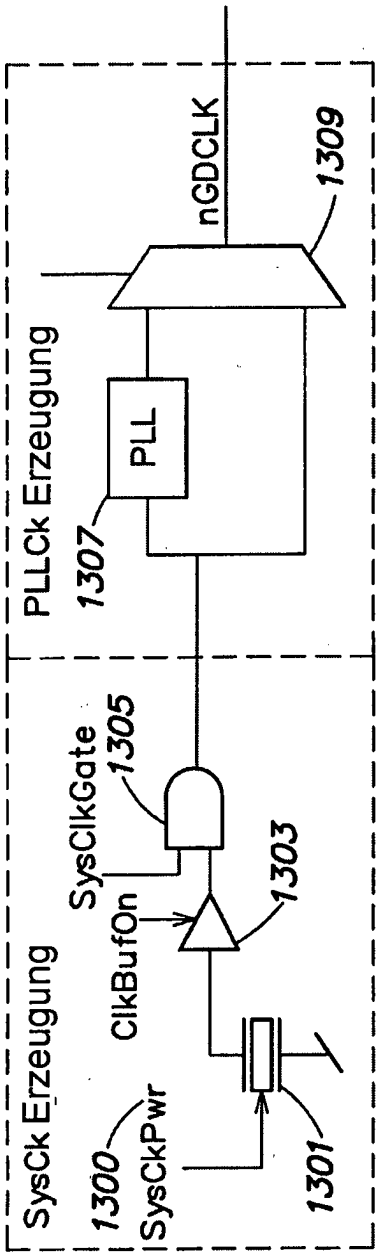


FIG. 13A

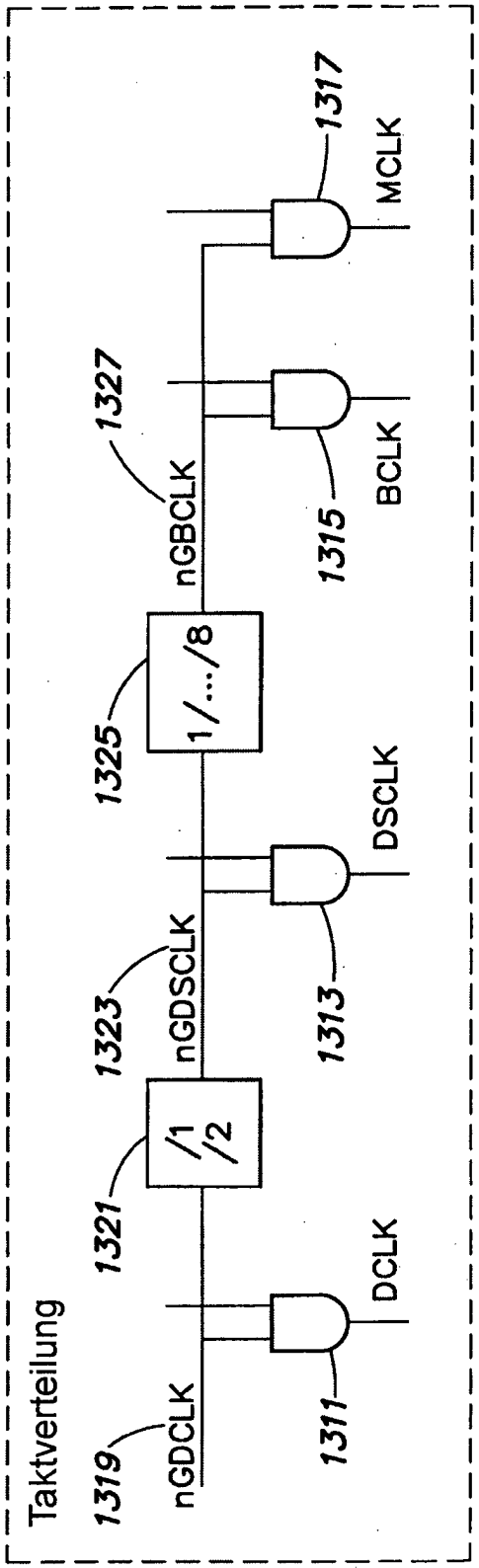


FIG. 13B

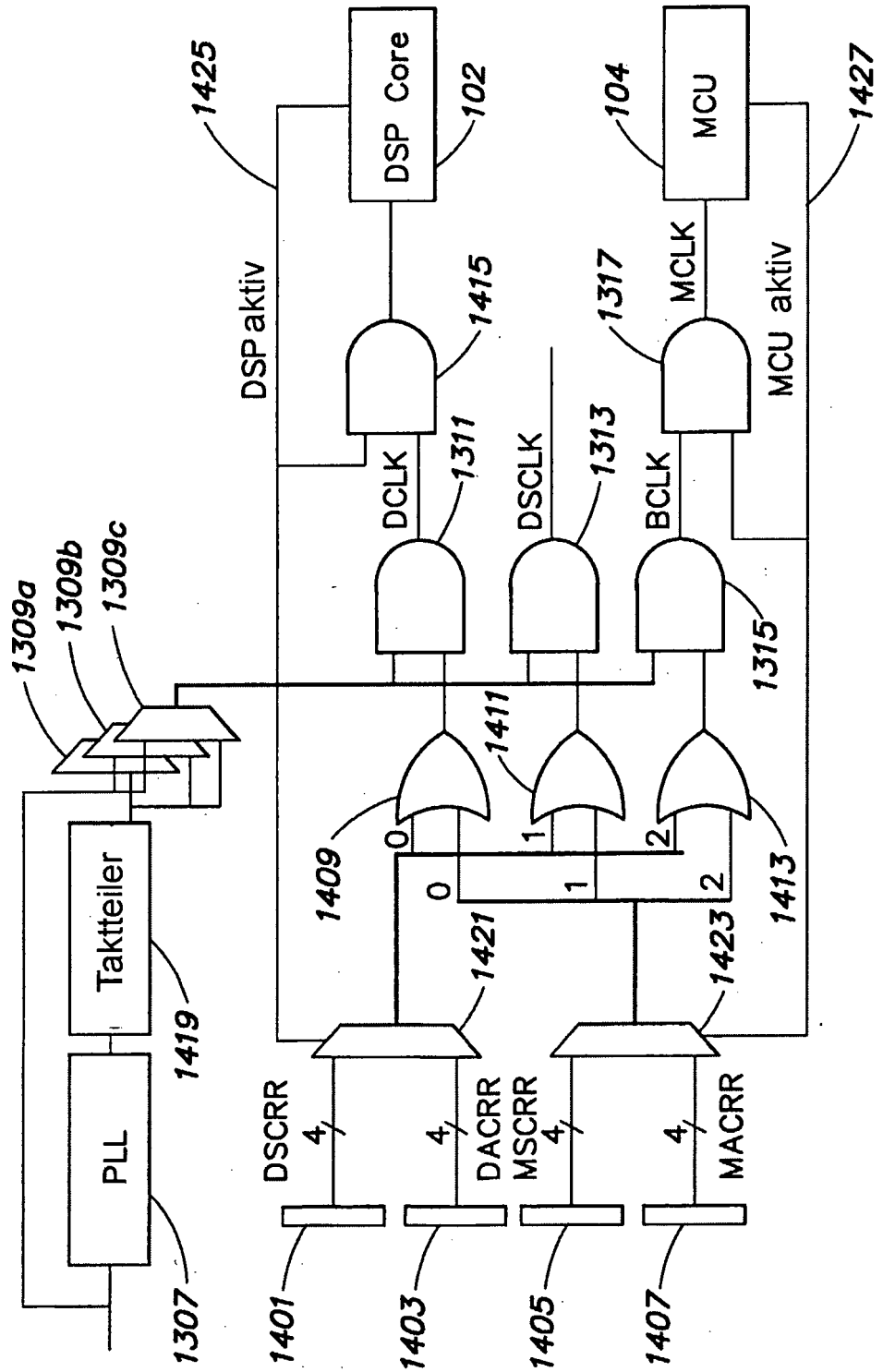

PLL
1307
Taktteiler
1419
1309a
1309b
1309c
1425
DSP aktiv
DSP Core
102
1415
DCLK
1311
DSCLK
1313
BCLK
1315
1317
MCLK
MCU aktiv
MCU
104
1427
1411
1409
1413
0
1
2
1421
1423
DSCRR
DACRR
MSCRR
MACRR
4
1401
1403
1405
1407

FIG. 14A

MCU Sleep Clock Requirements Register (MSCRR)

[0] –DCLK, 1 = Muss während MCU-Schlaf aktiv sein
[1] –DSCLK, 1 = Muss während MCU-Schlaf aktiv sein
[2] –BCLK, 1 = Muss während MCU-Schlaf aktiv sein
[3] – Erlaubt Umgehung der PLL während des Schlafes

MCU Active Clock Requirements Register (MACRR)

[0] –DCLK 1 = Aktiv, während MCU aktiv ist
[1] –DSCLK 1 = Muss aktiv sein, während MCU aktiv ist
[2] –BCLK Dieses Bit ist in dieses Register gezwungen
[3] – Erlaubt Umgehung der PLL. Dieses Bit ist eine erzwungene Null in diesem Register

DSP Sleep Clock Requirement Register (DSCRR)

[0] –DCLK, 1 = Muss während DSP-Schlaf aktiv sein
[1] –DSCLK, 1 = Muss während DSP-Schlaf aktiv sein
[2] –BCLK, 1 = Muss während DSP-Schlaf aktiv sein
[3] –, Erlaubt Umgehung während des Schlafes

DSP Active Clock Requirement Register (DACRR)

[0] –DCLK Dieses Bit ist in dieses Register gezwungen
[1] –DSCLK Dieses Bit ist in dieses Register gezwungen
[2] –BCLK Dieses Bit ist in dieses Register gezwungen
[3] – Erlaubt PLL-Umgehung. Dies ist eine erwzungene Null in diesem Register

FIG. 14B

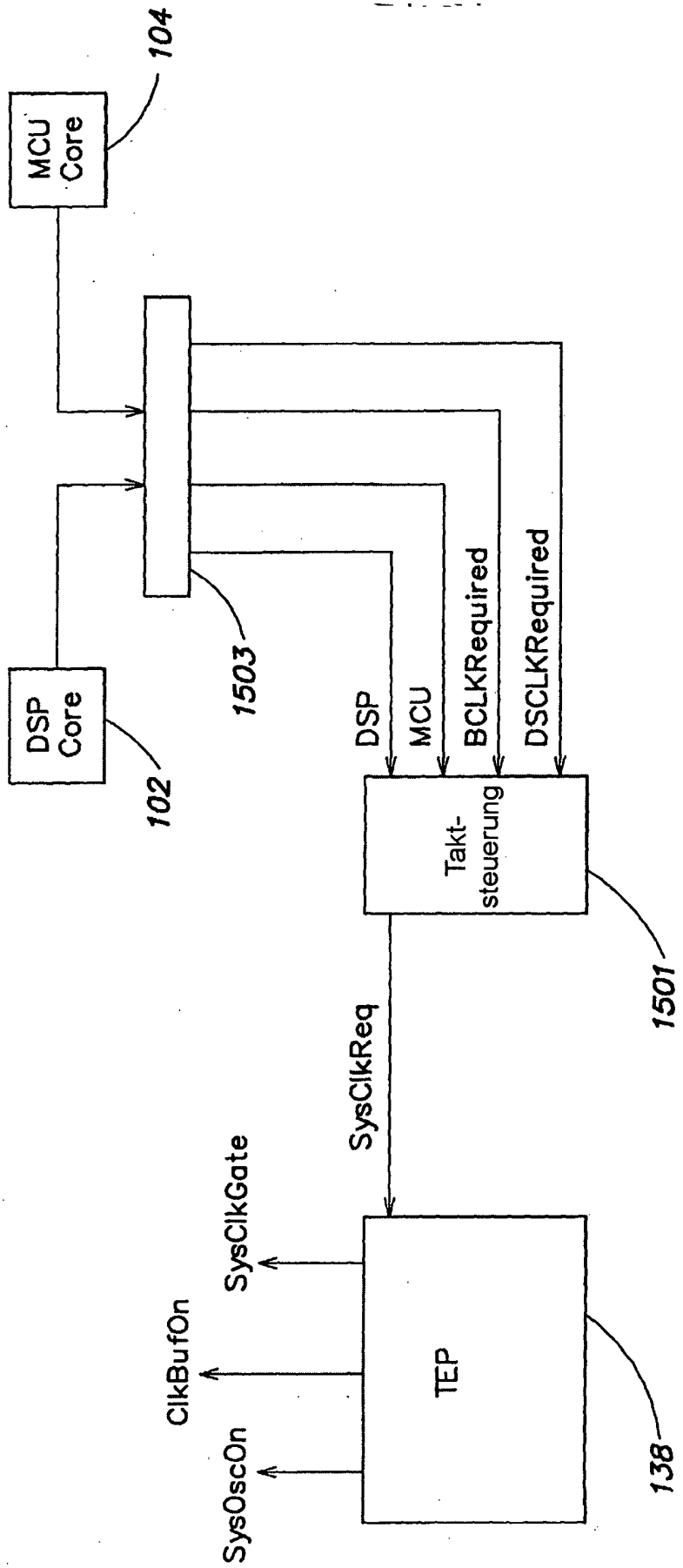

MCU Core
104
DSP Core
102
1503
DSP
MCU
BCLKRequired
DSCLKRequired
Takt-steuerung
1501
SysClkReq
SysClkGate
ClkBufOn
SysOscOn
TEP
138

FIG. 15