



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 699 25 987 T2** 2006.04.20

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 044 119 B1**

(51) Int Cl.⁸: **B60R 16/02** (2006.01)

(21) Deutsches Aktenzeichen: **699 25 987.8**

(86) PCT-Aktenzeichen: **PCT/US99/00356**

(96) Europäisches Aktenzeichen: **99 902 108.2**

(87) PCT-Veröffentlichungs-Nr.: **WO 99/035009**

(86) PCT-Anmeldetag: **07.01.1999**

(87) Veröffentlichungstag
der PCT-Anmeldung: **15.07.1999**

(97) Erstveröffentlichung durch das EPA: **18.10.2000**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **29.06.2005**

(47) Veröffentlichungstag im Patentblatt: **20.04.2006**

(30) Unionspriorität:
4076 07.01.1998 US

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

(73) Patentinhaber:
Microsoft Corp., Redmond, Wash., US

(72) Erfinder:
**BECKERT, D., Richard, Lake Stevens, US;
MOELLER, M., Mark, Bellingham, US; LI, Hang,
Redmond, US**

(74) Vertreter:
Ullrich & Naumann, 69115 Heidelberg

(54) Bezeichnung: **KRAFTFAHRZEUGCOMPUTERSYSTEM MIT AUDIOUNTERHALTUNGSSYSTEM**

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

Technisches Gebiet

[0001] Die Erfindung betrifft ein Audio-Unterhaltungssystem für Fahrzeuge. Insbesondere betrifft die Erfindung ein Fahrzeug-Computersystem, das ein Audio-Unterhaltungssystem implementiert.

Hintergrund der Erfindung

[0002] Moderne Fahrzeuge sind üblicherweise mit mehreren unabhängigen elektronischen Systemen ausgerüstet. So besitzen beispielsweise die meisten modernen Fahrzeuge eine Audioanlage und ein Sicherheitssystem. Die meisten aktuellen Fahrzeugmodelle sind ebenso mit einem Diagnosesystem aufgebaut, das die Leistungen des Fahrzeugmotors, des Getriebes und der Treibstoffanlage und andere Komponenten analysiert (1996 oder später mit OBD II, 1993 oder später mit OBD 1). Bei einigen neuen Modellen sind die Fahrzeuge mit einem Navigationssystem ausgerüstet, das einen GPS(Global Positioning System)-Empfänger zum Empfangen von Signalen eines Satellitennetzwerks zur Berechnung von Koordinaten enthält, die das Fahrzeug auf der Erdoberfläche in Bezug auf das Längengrad, das Breitengrad und die Höhe lokalisiert. Zellulare Kommunikationssysteme wurden ebenso zu den Fahrzeugen hinzugefügt. Diese Kommunikationssysteme erlauben es dem Fahrzeugfahrer oder Fahrzeuginsassen, Telefonanrufe von ihrem Fahrzeug aus durchzuführen.

[0003] Obwohl diese verschiedenartigen elektronischen Systeme sich als nützlich für den Fahrzeugnutzer erwiesen haben, besteht ein Nachteil darin, dass die Systeme unverbunden und inkompatibel sind. Jedes System setzt separate proprietäre fest zugeordnete Prozessoren oder ASICs (Application Specific Integrated Circuits) ein, die inkompatible proprietäre Software ausführen. Wenn ein Fahrzeughalter ein Sicherheitssystem zu seinem/ihrer Fahrzeug hinzufügen möchte, so muss der Halter ein komplettes Sicherheitssystem von einem der Anbieter erwerben und muss es üblicherweise einbauen lassen. Es besteht keine Möglichkeit, Sicherheitsfunktionalitäten zu einem bestehenden elektronischen System wie beispielsweise dem Navigationssystem oder der Audioanlage hinzuzufügen.

[0004] Die US-Patentanmeldung 08/564,586 mit dem Titel „Vehicle Computer System“, die am 29. November 1995 im Namen von Richard D. Beckert, Mark M. Moeller und William Wong angemeldet wurde, beschreibt ein Fahrzeug-Computersystem, das in der Lage ist, diese unterschiedlichen und getrennten Systeme zu integrieren und ebenso eine Mehrzweck-Computer-Plattform anzubieten, die eine einfache Erweiterbarkeit erlaubt. Das Fahrzeug-Computersystem bietet eine offene Hardware-Architektur und unterstützt ein Betriebssystem für offene Plattformen. Das Betriebssystem für offene Plattformen unterstützt mehrere unterschiedliche Anwendungen, die von einem Softwareanbieter bereitgestellt werden können. So kann das Betriebssystem beispielsweise Anwendungen unterstützen, die Unterhaltung, Navigation, Kommunikation, Sicherheit, Diagnosen und anderes betreffen. In der bevorzugten Implementierung ist das Betriebssystem ein Multi-Tasking-Betriebssystem, das in der Lage ist, mehrere Anwendungen konkurrierend ablaufen zu lassen.

[0005] Diese Erfindung betrifft ein Audio-Unterhaltungssystem, dass durch das Fahrzeug-Computersystem implementiert wird.

[0006] Die US 5701347 offenbart ein Audiosystem für Multimedia Computer Systeme einschließlich eines Audiomoduls mit Lautsprechern und Mikrofon und zugeordnetem Schaltkreis. Das Audiomodul ist derart angepasst, dass es zwischen einem Monitor und einem Monitorfuß eingefügt wird. Der Schaltkreis ist zum Betreiben der Lautsprecher des Audiomoduls und zum Betreiben lediglich eines Lautsprechers und des Mikrofons in einem Lautsprechertelefonmodus angepasst. Das Audiomodul ist mit einem Personalcomputer über einen Standardanschluss mit der Soundkarte des PCs verbunden. Ebenso wie die Lautsprecher und das Mikrofon kann das Audiomodul externe Audioeingänge zur Verbindung von externen Audiosignalquellen wie beispielsweise einen CD-Spieler umfassen. Ein Lautsprecherverstärker verstärkt die Audiosignale auf einen zum Betreiben der Lautsprecher geeigneten Pegel. Die Soundkarte von der CD ist in der Lage, die Lautsprecher in dem Audiomodul unter Kontrolle des PCs zu betreiben.

Zusammenfassung der Erfindung

[0007] Die Erfindung betrifft ein Fahrzeug-Computersystem, insbesondere die Audio-Unterhaltungsaspekte des Systems. Das Fahrzeug-Computersystem umfasst drei Module: ein Unterstützungsmodul, ein Computer-Modul und ein Blendenmodul. Das Unterstützungsmodul ist als ein Teil einer stationären Basiseinheit des Gehäuses ausgebildet, die sich in dem Fahrzeugarmaturenbrett oder an einem anderen Standort befindet. Sie

weist ihre eigene Logikeinheit auf, die in einem FPGA (Field Programmable Gate Array), einem ASIC (Application Specific Integrated Circuit), einem kundenspezifischem Prozessor oder ähnlichem implementiert sein kann. Das Unterstützungsmodul weist zudem einen Audiosignalprozessor wie beispielsweise einen Audio-DSP (Digitaler Signalprozessor) auf, der die Signalverarbeitung von Audiodaten durchführt.

[0008] Das Computermodule kann eine separate Einheit sein, die an dem Unterstützungsmodul angeschlossen ist, oder seine Funktionalität ist in dem Unterstützungsmodul integriert. Das Computermodule weist einen Prozessor auf, der ein Betriebssystem ausführt. Das Computermodule und das Unterstützungsmodul sind mit einem mehrere Bit breiten Bus, wie beispielsweise einem PCI-Bus miteinander verbunden. Das Blendenmodul ist abnehmbar mit dem Unterstützungsmodul verbunden. Das Blendenmodul weist eine eigene Logikeinheit zur Kontrolle eines Displays und eines Tastenfelds auf, und optional einen RF-Transceiver wie beispielsweise ein Mobiltelefon oder ein Zweiwege-Funkempfänger. Das Blendenmodul und das Unterstützungsmodul sind über eine serielle Hochgeschwindigkeitsverbindung verbunden.

[0009] Das Audio-Unterhaltungssystem ist in erster Linie auf die Logikeinheit und den Audio-DSP des Unterstützungsmoduls gestützt. Das Unterstützungsmodul kann viele Audioquellen, wie beispielsweise einen CD-Player, CD-Wechsler, AM/FM-Empfänger, Auxiliary 1, Auxiliary 2, Computer-Wave-Ausgaben und ein Mikrofon aufnehmen. Das Unterstützungsmodul empfängt die Daten von den Quellen, bearbeitet sie mit dem Audio-DSP und gibt die Daten an das Lautsprechersystem oder an andere Ziele, wie beispielsweise ein USB-Peripheriegerät oder einen Speicher in dem Computermodule, aus.

[0010] Die Logikeinheit des Unterstützungsmoduls weist eine interne Adress-/Datenbusstruktur auf, die unabhängig und getrennt von dem internen Bus des Computermoduls ist. Dieser Bus der Logikeinheit arbeitet als peripherer Kommunikationspfad, so dass Daten von den Audioquellen ohne einen Eingriff des Computermoduls verarbeitet werden können. So kann beispielsweise die gesamte Audiotonerzeugung, -verarbeitung und -ausgabe durch das Unterstützungsmodul gehandhabt werden, während die Verarbeitungsressourcen des Computermoduls minimal belegt sind.

[0011] Gemäß einem ersten Aspekt der vorliegenden Erfindung umfasst ein Unterhaltungssystem einen Datenspeicher, eine zum Empfang von Audiodaten von einem oder mehreren peripheren Audioquellen verbundene Logikeinheit und einen Datenprozessor zum Verarbeiten von Audiodaten, wobei die Logikeinheit eine oder mehrere Audio-DMA(direct memory access)-Schaltungen aufweist, die jeweils entsprechenden peripheren Audioquellen zugeordnet sind, wobei jede Audioquellen-DMA-Schaltung die von der zugeordneten peripheren Audioquelle empfangenen Audiodaten in einen Speicherbereich innerhalb des Datenspeichers schreibt, und wobei die Logikeinheit auch eine Datenprozessor-DMA-Schaltung aufweist, die dem Datenprozessor zum Auslesen der Audiodaten aus dem Speicherbereich innerhalb des Datenspeichers zur Übermittlung an den Datenprozessor zugeordnet ist.

[0012] Vorzugsweise umfasst das System zusätzlich eine Synchronisationsschaltung, um eine erste Rate, mit der die Audioquellen-DMA-Schaltung die Audiodaten in den Datenspeicher schreibt, mit einer zweiten Rate zu synchronisieren, mit der die Datenprozessor-DMA-Schaltung die Audiodaten aus dem Datenspeicher liest. Die Logikeinheit kann Audiodaten von vielen verschiedenen Audioquellen an den Datenprozessor zur konkurrierenden Verarbeitung konkurrierend übermitteln.

[0013] In dem System umfassen die eine oder mehrere Audioquellen-DMA-Schaltungen vorzugsweise eine erste DMA-Schaltung, eine zweite DMA-Schaltung und eine dritte DMA-Schaltung, und die eine oder mehreren peripheren Audioquellen weisen eine erste periphere Audioquelle und eine zweite periphere Audioquelle auf, wobei die erste DMA-Schaltung der ersten peripheren Audioquelle zum Übermitteln erster Audiodaten von der ersten peripheren Audioquelle zu einer ersten Stelle in dem Datenspeicher zugeordnet ist, wobei die zweite DMA-Schaltung der zweiten peripheren Audioquelle zum Übermitteln zweiter Audiodaten von der zweiten peripheren Audioquelle zu einer zweiten Stelle in dem Datenspeicher zugeordnet ist, und wobei die dritte DMA-Schaltung dem Datenprozessor zum Übermitteln der ersten und zweiten Audiodaten von den ersten und zweiten Stellen in dem Datenspeicher an den Audiodatenprozessor zur konkurrierenden Verarbeitung und Ausgabe an ein oder mehrere Audioziele zugeordnet ist.

[0014] Es ist bevorzugt, dass die ersten Audiodaten zu einem ersten Satz von Lautsprechern und die zweiten Audiodaten gleichzeitig zu einem zweiten Satz von Lautsprechern ausgegeben werden.

[0015] Es ist bevorzugt, dass das Unterhaltungssystem zusätzlich ein Application Program Interface zur Definition einer Schnittstelle zwischen den auf dem System betriebenen Anwendungsprogrammen und einem Au-

diounterhaltungssystem umfasst, wobei das Application Program Interface auf dem Datenprozessor des Systems ausführbar ist und Funktionen ausweist, die von einem Anwendungsprogramm zur Steuerung des Audiounterhaltungssystems aufrufbar sind.

[0016] Das Application Program Interface umfasst vorzugsweise eine Funktion zum Auswählen einer Audioquelle und/oder eine Funktion zum Ein- und Ausschalten von Surround Sound und/oder eine Funktion zum Ändern von Equalizer-Einstellungen und/oder eine Funktion zum Ändern von Lautstärke, Balance und Überblendcharakteristiken.

[0017] Die Erfindung kann ein mit der internen Adress-/Datenbus-Struktur der Logikeinheit des Unterstützungsmoduls verbundenen Satz von Ping/Pong-Puffern in dem Datenspeicher verwenden, die vorübergehend die zwischen den unterschiedlichen Audiogeräten kommunizierten Daten halten. Die Logikeinheit umfasst DMA(Direct Memory Access)-Schaltungen, die mit jedem der Audiogeräte zum Kennzeichnen eines bestimmten Speicherbereichs des Eingabe-/Ausgabespeichers zum Halten von Daten zugeordnet sind, wobei die Daten von den zugeordneten Geräten empfangen oder zu den zugeordneten Geräten gesendet wurden. Audiodaten werden unter Verwendung einer Mapping-Overlay-Technik ausgetauscht, bei der die DMA-Schaltungen für zwei Audiogeräte von bzw. zu demselben Speicherpuffer lesen bzw. schreiben. Während eine DMA-Schaltung Audiodaten von einem Gerät (d.h. einem AM/FM-Empfänger, einem CD-Spieler, usw.) in dem Ping-Puffer ablegt, liest die andere DMA-Schaltung Daten von dem Pong-Puffer zur Ausgabe an ein anderes Audiogerät (beispielsweise ein Audiosignalprozessor).

[0018] Das Audio-Unterhaltungssystem verarbeitet die digitalen Audiodaten mit einer internen Abtastrate (beispielsweise 44,1 kHz), die durch eine Haupttaktfrequenz bestimmt und von einem 48 MHz Taktgeber und einem durch Software anpassbaren Teilerregister in dem Unterstützungsmodul abgeleitet wird. Daten, die von dem Puffer zu dem Audio-DSP zur Verarbeitung übermittelt werden, sind mit dieser Rate abgetastet. Dennoch kann ein mit seinem eigenen Taktsignal betriebenes Audioquellengerät Audiodaten in den Speicherpuffer mit einer Rate schreiben, die nicht exakt mit der Ausgabe-DMA-Taktrate übereinstimmt. Die Software des Audio-Unterhaltungssystems passt das Teilerregister zum Beeinflussen der Haupttaktfrequenz zu dem CODEC an, um die Datenschreibe- und Datenleseoperationen in den Ping/Pong-Puffer zu synchronisieren.

[0019] Das API ermöglicht, Anwendungen zum Kontrollieren von verschiedenen Audioquellen ohne Kenntnis der Hardware und Details der Implementierung der zugrunde liegenden Audioanlage auf dem Computer zu betreiben. Verschiedene Audiogeräte und ihre Treiber kontrollieren verschiedene Funktionalitäten des Audiosystems, wie beispielsweise Equalizer, Lautstärkekontrollen und Decodierung des Surround Sounds. Das Audio-Manager-API überträgt durch die Anwendungen getätigte Aufrufe an den/die passenden Gerätetreiber.

[0020] Die Erfindung kann zum Handhaben von Sprachdaten genutzt werden, die von einem Mikrofon zur Eingabe an ein Spracherkennungssystem empfangen wurden. Die Sprachäußerungen, die durch das Mikrofon erfasst wurden, werden mit einer durch den CODEC zur Verfügung gestellten Abtastrate von 44,1 kHz abgetastet.

[0021] Dennoch verwendet das Spracherkennungssystem eine unterschiedliche Abtastrate, wie beispielsweise eine erheblich langsamere Abtastrate von 11 kHz. Statt einen separaten Analog-/Digital-Wandler zum Abtasten von Mikrofondaten hinzuzufügen (was zusätzliche Kosten verursachen würde) konvertiert der Audio-DSP eher die Mikrofondaten von der höheren Abtastrate von 44,1 kHz zu den gewünschten 11 kHz. Das System benutzt ein SPI (Serial Peripheral Interface) des Audio-DSPs, um die Mikrofondaten huckepack mit dem Kommando-/Nachrichtenstrom zu transportieren. Der DSP führt eine normale Tiefpassfilterung und ein Down-Sampling des Datenstroms durch und nutzt dann das SPI, um die Mikrofondaten mit ihrer neuen Abtastrate von 11 kHz huckepack mit dem Nachrichtenstrom zurück auszusenden.

Kurze Beschreibung der Zeichnungen

[0022] In allen Zeichnungen werden dieselben Bezugszeichen verwendet, um gleiche Komponenten oder Merkmale zu bezeichnen.

[0023] [Fig. 1](#) ist eine schematische Darstellung eines Fahrzeug-Computersystems.

[0024] [Fig. 2](#) ist eine schematische Darstellung des mit mehreren externen Peripheriegeräten verbundenen Fahrzeug-Computersystems.

[0025] [Fig. 3](#) ist ein Blockdiagramm des Fahrzeug-Computersystems gemäß einer Implementierung mit einem Blendenmodul, einem Unterstützungsmodul und einem Computermodul.

[0026] [Fig. 4](#) ist ein Blockdiagramm einer Logikeinheit und eines Audio-DSPs, die in dem Unterstützungsmodul des Computersystems implementiert sind.

[0027] [Fig. 5](#) ist ein Blockdiagramm einer Speicherzugriffsschaltung, die Daten von Peripheriegeräten in einem Ping/Pong-Puffer abbildet.

[0028] [Fig. 6](#) ist ein Blockdiagramm einer Haupttakteinstellerschaltung.

[0029] [Fig. 7](#) ist ein Blockdiagramm einer Bus-Schlichter-Schaltung.

[0030] [Fig. 8](#) zeigt eine Audio-Software/Hardware-Interface-Architektur.

[0031] [Fig. 9](#) ist ein Funktionsdiagramm, das den Audiodatenfluss in einem Audio-Manager-API darstellt.

[0032] [Fig. 10](#) ist ein Blockdiagramm, das den Gebrauch eines SPI-Ports zum Austauschen von Sprachdaten zwischen dem Audio-DSP und der Logikeinheit darstellt.

Detaillierte Beschreibung der bevorzugten Ausführungsform

[0033] Die Erfindung bezieht sich auf eine Audioarchitektur für ein Audio-Unterhaltungssystem. Aspekte dieser Erfindung sind besonders gut für Fahrzeug-Unterhaltungssysteme geeignet. Zur Erörterung wird die Audioarchitektur im Zusammenhang eines Fahrzeug-Computersystems beschrieben.

[0034] [Fig. 1](#) zeigt ein Fahrzeug-Computersystem **20** gemäß einer Implementierung dieser Erfindung. Das Fahrzeug-Computersystem **20** weist einen zentralen Computer **22** auf, der mit verschiedenen externen Peripheriegeräten einschließlich eines optionalen Monitors **24**, Sicherheitssensoren **26**, eines Fahrzeugdiagnose-Interface **28**, Lautsprecher **30**, einer Fahrzeugbatterie **32**, einer Ersatzbatterie **33** und Antenne(n) **34** verbunden ist. Der Computer **22** ist in einem Gehäuse **36** eingebaut, das zur Montage in einem Fahrzeug-Armaturenbrett ähnlich einer konventionellen Autostereoanlage bemessen ist. Vorzugsweise weist das Gehäuse **36** einen Formfaktor eines einfachen DIN (Deutsche Industrie Normen) auf. Allerdings könnte es möglicherweise in einer 2 DIN Einheit oder anderen speziellen Formfaktoren für ein OEM untergebracht sein.

[0035] Auf den Computer **22** läuft ein Betriebssystem für offene Plattformen, das viele Anwendungen unterstützt. Durch die Verwendung eines Betriebssystems für offene Plattformen und einer offenen Computersystemarchitektur können verschiedene Softwareanwendungen und Hardwareperipherien von unabhängigen Anbietern hergestellt und anschließend durch den Fahrzeugnutzer nach dem Kauf des Fahrzeugs installiert werden. Dies ist dahingehend vorteilhaft, dass die Softwareanwendungen nicht eigens für einmalig entworfene Embedded Systems konfiguriert werden müssen. Auf der offenen Hardwarearchitektur läuft vorzugsweise ein Multitasking-Betriebssystem, das ein grafisches Nutzer-Interface verwendet. Ein bevorzugtes Betriebssystem ist das Windows CE-Betriebssystem, das durch die Microsoft Corporation verkauft wird. Ein Multitasking-Betriebssystem erlaubt die gleichzeitige Ausführung von mehreren Anwendungen.

[0036] Der Computer **22** umfasst mindestens ein Speicherlaufwerk, das dem Fahrzeugnutzer das Herunterladen von Programmen und Daten von einem Speichermedium erlaubt. In der dargestellten Implementierung weist der Computer **22** ein CD-ROM-Laufwerk **38** auf, das anwendungsbezogene CDs ebenso wie Musik-, Video-, Spiele- oder andere Arten von Unterhaltungs-CDs liest. Das CD-ROM-Laufwerk **38** übernimmt eine doppelte Aufgabe als Speicherlaufwerk und als Abspielgerät für die Unterhaltung. Der Computer **22** weist ein optionales 3,5"-Diskettenlaufwerk **40**, ein optionales SmartCard-Lesegerät **42** und ein doppelter PC-Karten-Sockel oder CompactFlash-Karten-Sockel **44** auf, der PC-Karten vom Typ I, II und III (ehemals PCMCIA-Karten) oder CompactFlash-Karten akzeptiert. Daneben kann ein Festplattenlaufwerk (nicht abgebildet) auf dem Computer **22** zur Speicherung von Anwendungsprogrammen und Nutzerdaten eingefügt sein. Ein DVD(Digital Videodisk)-Abspielgerät kann ebenso in dem Computer **22** enthalten sein.

[0037] Die Speicherlaufwerke sind in einer Grundeinheit **46** des Gehäuses **36** eingebaut. Die Grundeinheit **46** ist zum Einbau in dem Armaturenbrett konstruiert und dimensioniert. Optional kann die Grundeinheit in derselben Art und Weise wie ein Laptop von seiner ihm zugeordneten Docking Station abnehmbar ausgestaltet sein. Diese gehobene Variante erlaubt es dem Nutzer, seinen Fahrzeug-Computer nach Hause oder ins Büro zu

nehmen, damit er als sein tragbarer PC dient.

[0038] Das Gehäuse **36** weist eine abnehmbare Blende **48** auf, die zentral an der Vorderseite der Basiseinheit **46** befestigt ist. Die Blende kann gedreht werden, um einen einfachen und angenehmen Zugang zu den Speicherlaufwerken zu ermöglichen. Die komplette Blendeneinheit **48** verhält sich wie ein multifunktionales Peripheriegerät zu der Computerplattform.

[0039] Die Blende **48** weist ein Tastenfeld **52** und eine grafische Anzeige **54** auf. Die Anzeige **54** ist vorzugsweise ein hinterleuchtetes LCD mit einer rechteckigen Pixel-Matrix, die einzeln für eine Beleuchtung oder Anzeige auswählbar sind. Das Anzeigeelement ist ein Bitmap-Anzeigesystem mit mittlerer Auflösung, das mindestens 10.000 Pixel aufweist. In der beschriebenen Implementierung weist die Pixel-Matrix eine Größe von mindestens **64** Pixeln in vertikaler Richtung und 256 Pixel in horizontaler Richtung auf. Das Betriebssystem der Grundeinheit **46** wirkt mit dem Tastenfeld **52** der Blende und der Anzeige **54** der Blende als Peripheriegeräte zusammen, wenn die Blende **54** mit der Grundeinheit **52** verbunden ist.

[0040] Die Blende **48** weist ein Start-Knopf **56** auf, der gewohnten „Start“-Funktionen eines Betriebssystems der Marke Windows anbietet. Die Blende **48** weist ebenso einen Netzschalter **58**, ein Vier-Position-Bedienteil **60**, eine „Eingabe“-Taste **62**, eine „Abbruch“-Taste **64**, eine „Hilfe“-Taste **66**, eine „Menü“-Taste **68** und eine Lautstärkereglereingabe **70** auf.

[0041] Der Computer **22** weist ein Spracherkennungssystem auf, um dem Nutzer eine Spracheingabe von Kommandos in einer handfreien, blickkontaktsfreien Umgebung zu ermöglichen. Diese Sprachkommandos können zur Kontrolle der meisten Betriebszustände der Fahrzeugcomputerplattform ebenso wie zur Kontrolle von auf dem Computer ablaufenden Anwendungen genutzt werden. Ein mit dem Unterstützungsmodul verbundener Spracheingabe-Port **72** kann vorzugsweise an dem Blendschutz oder an anderen Stellen zum optionalen Erfassen von Spracheingabe zur Eingabe an das Spracherkennungssystem (siehe [Fig. 3](#)) befestigt sein. Der Computer **22** unterstützt ebenso ein auf der Blende **48** befestigter IrDA(Infrared Developers Association)-Transceiver-Port **74** zum Übertragen und Empfangen von Daten und Programmen unter der Verwendung von Infrarotsignalen.

[0042] Zum Laden einer Anwendung oder Daten auf den Computer **22** legt der Fahrzeugnutzer eine CD oder andere Medien (falls die Anwendung nicht bereits auf der Festplatte enthalten ist) in das entsprechende Laufwerk und das Betriebssystem lädt die Anwendung oder die Daten von diesem herunter. Der Installationsvorgang kann automatisch durch das Betriebssystem durchgeführt werden oder durch die Unterstützung von Kommandoeingaben des Nutzers in Form einer Tastenfolge auf dem Tastenfeld **52** oder durch Sprachanweisungen unter Verwendung des Spracherkennungsgeräts. Eine weitere Technologie zum Laden von Daten und Anwendungen oder zum Übertragen von Daten mit anderen EDV-Geräten besteht durch die Nutzung des IrDA-Transceiver-Ports **74** oder der drahtlosen Internetverbindung.

[0043] Im Allgemeinen kann das Fahrzeug-Computersystem **20** zum Integrieren von mehreren fahrzeugbezogenen Systemen auf einer offenen Plattform-Hardware- und Software-Architektur genutzt werden. Das Fahrzeug-Computersystem **20** kann beispielsweise als ein Multimedia-Unterhaltungssystem, ein Navigationssystem, ein Kommunikationssystem, ein Sicherheitssystem und ein Diagnosesystem dienen. Außerdem bietet das Fahrzeug-Computersystem **20** zusätzliche Funktionalität, die üblicherweise mit Desktop oder Laptop-Personalcomputern in Verbindung gebracht werden. So kann beispielsweise das Fahrzeug-Computersystem **20** Adressbuch-Anwendungen, E-Mail-Programme und Terminkalender-Anwendungen unterstützen. Darüber hinaus kann das Fahrzeug-Computersystem **20** zum Betreiben als ein Server für andere EDV-Einheiten in dem Fahrzeug zur Verteilung von Spielen, Videofilmen oder dergleichen an die Insassen konfiguriert werden.

[0044] [Fig. 2](#) zeigt den Computer **22** gemäß einer Implementierung der Erfindung. Der Computer **22** weist drei primäre Module auf: ein Blendenmodul **80**, ein Unterstützungsmodul **82** und ein Computermodul **84**. Das Computermodul **84** ist im betriebsbereiten Zustand mit dem Unterstützungsmodul **82** über einen mehrere Bit breiten Bus **86** verbunden. In der bevorzugten Implementierung umfasst der mehrere Bit breite Bus ein PCI(Peripheral Component Interconnect)-Bus. Das Unterstützungsmodul **82** und das Blendenmodul **80** sind über eine serielle Hochgeschwindigkeitsschnittstelle **88**, die serielle Hochgeschwindigkeitsdatenkommunikation unterstützt, untereinander verbunden.

[0045] Das Unterstützungsmodul **82** ist ebenso mit einem USB(Universal Serial Bus)-Hub **90** über ein USB-Seriell-I/F-Verbinder **92** verbunden. Der USB-Hub **90** bietet Verbindungen zu vielen Peripheriegeräten (zum Beispiel, bis zu 128 Geräten). Peripheriegeräte umfassen beispielsweise das OBD(On Board Diag-

nostic)-System **28**, das Sicherheitssystem **26**, Navigationssystem **94**, eine drahtlose Verbindung **96** zu dem Internet, einen CD-ROM-Wechsler **98**, einen TV-Empfänger **100** und Nutzer-Eingabe/Ausgabe-Geräte wie einen Joystick, Tastatur oder eine Maus **102**. Dieser USB-Hub **90** könnte wie in [Fig. 2](#) dargestellt getrennt oder in einem oder mehreren der USB-Peripheriegeräte integriert ausgestaltet sein.

[0046] [Fig. 3](#) zeigt die drei Module des Fahrzeug-Computers **22** etwas ausführlicher. Das Unterstützungsmodul **82** befindet sich in der typischerweise stationären Grundeinheit **46** ([Fig. 1](#)), die in das Fahrzeug-Armaturenbrett oder an anderer geeigneter Stelle befestigt ist. Das Unterstützungsmodul **82** umfasst eine Logikeinheit **110**, die zum Ermöglichen der Kommunikation zwischen Peripheriegeräten, dem Erzeugen eines quer über das Fahrzeug-Computersystem genutzten Hauptaudiotaktsignals und dem Koordinieren der Unterhaltungsfunktionalität des Computersystems verantwortlich ist. Die Logikeinheit **110** kann als ein FPGA (Field Programmable Gate Array) implementiert sein. Es sei bemerkt, dass der Logik-Code für einen FPGA unter Verwendung einer Hardware-Beschreibungssprache wie beispielsweise VHDL (IEEE Standard 1076-1993) entwickelt wird und für eine ASIC(Application Specific Integrated Circuit)-Architektur rekompiliert werden kann. Die Logikeinheit **110** kann zudem als ein Mikroprozessor, ein RISC(Reduced Instruction Set Computing)-Prozessor oder andere Verarbeitungsgeräte implementiert sein.

[0047] Das Unterstützungsmodul **82** weist mehrere Hardware-Interfaces auf. Ein USB-Interface **112** wird über den PCI-Bus **86** betrieben und stellt die Verbindung zu verschiedenen USB-Peripheriegeräten, wie in [Fig. 2](#) dargestellt, her. Ein separater USB-Hub kann – wie gezeigt – notwendig sein, falls einige der angebundenen Peripheriegeräte nicht ein und mehrere USB-Peripheriegeräte einschließen als direkt durch das Unterstützungsmodul unterstützt angebunden sind. Ein optionaler VGA- oder ein Controller **114** für andere hochwertige Grafik wird auf dem Unterstützungsmodul **82** zum Betreiben einer optionalen Anzeige **24** ([Fig. 1](#)) zur Verfügung gestellt. Der VGA-Controller **114** wird ebenso über den PCI-Bus betrieben. Ein IDE-Controller **115** wird ebenso in dem Unterstützungsmodul zur Verfügung gestellt.

[0048] Es sei angemerkt, dass das USB-Interface **112** und der VGA-Controller **114** in der Logikeinheit **110** enthalten sein könnten. Ein ATAPI-IDE-Interface (zum Betreiben von CD-ROM oder DVD-Abspielgerät benötigt) und ein PCI-Interface sind als Teil der Logikeinheit **110** implementiert. Zur Unterstützung von Videoanwendungen mit höherer Bandbreite wie beispielsweise mehrere Bildschirme, könnte ein P1394-Interface dem System hinzugefügt und über das PCI-Interface betrieben werden oder in der Logikeinheit **110** enthalten sein.

[0049] Das Unterstützungsmodul **82** umfasst ebenso einen Audio-Signalprozessor **116** zur Durchführung von Tonverarbeitungsalgorithmen, die enthalten können: Tonentzerrung, digitale Überblendung, Bass, Höhe, Lautstärke, Surround Sound, Dolby Pro-Logic, AC-3- und MPEG-Decodierung. Der Audio-Signalprozessor **116** betreibt ebenso einen Digital/Analog-Wandler für eine Sechskanalaudioausgabe (nicht dargestellt). Der Audio-Signalprozessor **116** ist vorzugsweise als ein DSP (digitaler Signalprozessor) wie beispielsweise ein Motorola DSP56009 implementiert. Das Unterstützungsmodul umfasst ebenso ein AM/FM-Empfängermodul **118**, ein GPS(Global Positioning System)-Chipsatz **120** zum Ermöglichen von Satellitennavigation zur Messung von Längengrad, Breitengrad und Höhe des Fahrzeugs und einen oder mehrere Audio-Analog/Digital- und Digital/Analog-Wandler („CODECs“) **122**.

[0050] Ein schneller Daten-Eingabe/Ausgabe-Speicher **130** arbeitet als ein Hochgeschwindigkeitsdatenkommunikationspuffer zwischen den seriellen Peripheriegeräten. Der schnelle Datenspeicher ist vorzugsweise als ein Hochgeschwindigkeits-SRAM(Static Random Access Memory) mit Hochgeschwindigkeits-Ping/Pong-Puffern **132** implementiert, um eine Hochgeschwindigkeitspufferung und eine Ping/Pong-Speicherung von Audio-daten oder USB-Daten, die die internen Aktionen des Prozessors minimiert, zu bieten. In der bevorzugten Implementierung agiert der Eingabe/Ausgabe-Speicher **130** als ein schneller Datenspeicherpuffer, der einen Datenaustausch zwischen vielen Geräten aufnimmt.

[0051] Das Blendenmodul **80** befindet sich auf der abnehmbaren Blende ([Fig. 1](#)). Das Blendenmodul **80** ist mit dem Unterstützungsmodul **82** über eine Steckverbindung verbunden, die die einfache Abnehmbarkeit der Blende **48** von dem Hauptgehäuse **36** erleichtert. Das Blendenmodul **80** übermittelt über die abnehmbare Steckverbindung bidirektionale Daten an das Unterstützungsmodul **82** mittels einer seriellen Hochgeschwindigkeitsverbindung **88**. Das Blendenmodul **80** umfasst eine Logikeinheit **140**, die als ein FPGA, ASIC oder andere Einheit implementiert sein kann. Das Blendenmodul **80** umfasst ebenso ein Tastenfeld **52**, eine Anzeige **54** und einen IrDA-Port. Zusätzlich enthält das Blendenmodul **80** einen Schacht für einen optional einsteckbaren RF-Transceiver **144** wie beispielsweise ein Mobiltelefon, Zweifach-Funkempfänger oder Punkt-zu-Punkt-Spreizspektrum-Tranceiver.

[0052] Typischerweise befindet sich das Computermodule **84** in der in dem Armaturenbrett befestigten Grundeinheit **46** und umfasst einen Prozessor in Form eines Mikroprozessors **150** wie beispielsweise einen Intel x86-artigen Mikroprozessor. Im eingesteckten Zustand ist das Computermodule **84** mit dem PCI-Bus **86** zur Kommunikation mit dem Unterstützungsmodul **82** verbunden.

[0053] Auf dem Mikroprozessor **150** läuft das Windows CE Betriebssystem der Microsoft Corporation. Das Betriebssystem ist in einem ROM **152** oder einem Flash-Speicher **154** auf dem Unterstützungsmodul gespeichert. Das Computermodule **84** unterstützt jede Variante von Anwendungen, die der Fahrzeugnutzer wünschen könnte. Diese Anwendungen können ebenso auf dem ROM **152**, Flash-Speicher **154**, DRAM **156** oder auf einem entfernbaren Speichermedium (nicht dargestellt) wie beispielsweise einer CD-ROM, Kassette, PC-Card-Flash-Speicher, PC-Card, Festplattenlaufwerk oder Diskette gespeichert sein. Zusätzlich können Nutzeranwendungen über eine optionale drahtlose Internetverbindung heruntergeladen werden.

[0054] Das Computermodule **84** weist ein PC-Card-Interface **158** auf, das einen PC-Card-Sockel oder ein Compact-Flash umfasst und PC-Cards vom Typ I, II oder III unterstützt (beispielsweise zusätzlichen Speicher, Festplattenlaufwerke, Modems, RF-Transceiver, Netzwerkkarten oder andere PC-Card-Peripheriegeräte). Das Computermodule **84** weist ebenso ein optionales SmartCard-Interface **160** auf, das SmartCards oder vergleichbare Karten mit integrierten Schaltkreisen akzeptiert.

[0055] Eine detailliertere Erklärung der drei Module in dem Fahrzeug-Computersystem befindet sich in der gleichzeitig anhängigen US-Patentanmeldung mit der Nummer 08/564,586 und dem Titel „Vehicle Computer System“, die am 29. November 1995 im Namen von Richard D. Beckert, Mark M. Moeller und William Wong angemeldet wurde. Diese Anmeldung wurde auf die Microsoft Corporation übertragen und ist hier durch Verweis eingearbeitet worden.

[0056] Die Logikeinheit **110** ist mit ihrer eigenen mehrere Bit breiten Busstruktur aufgebaut, die von dem Bus des Mikroprozessors **150** des Computermoduls **84** getrennt ist. Die Logikeinheit **110** und der Mikroprozessor **150** sind untereinander mit einem Bus wie beispielsweise einem PCI-Bus **86** verbunden. Durch Aufbauen der Logikeinheit **110** mit ihrem eigenen internen Bus ist die Logikeinheit **110** in der Lage, ihre Aufgaben unabhängig von den Eingriffen des Mikroprozessors **150** besser zu erfüllen. Ferner erleichtert der interne Bus der Logikeinheit **110** die Datenkommunikation zwischen den Audiokomponenten und anderen seriellen Geräten bei gleichzeitig minimaler Nutzung der Bearbeitungsressourcen des Mikroprozessors **130**.

[0057] Eine detaillierte Beschreibung einer Implementierung der Logikeinheit **110** befindet sich in der gleichzeitig anhängigen US-Patentanmeldung mit der Nummer 08/668,781 und dem Titel „Vehicle Computer System with High Speed Data Buffer and Serial Interconnect“, die am 24. Juni 1996 im Namen von Richard D. Beckert, Mark M. Moeller, Ron Randall und William Wong angemeldet wurde. Diese Anmeldung wurde auf die Microsoft Corporation übertragen und ist hier durch Verweis eingeflossen. Die in dieser Offenbarung beschriebene Logikschaltung stellt eine andere Implementierung der Logikeinheit **110** dar, die spezieller für die Implementierung des Audio-Unterhaltungssystems zugeschnitten wurde.

Audio-Architektur

[0058] Das Computersystem **20** unterstützt ein Audio-Unterhaltungssystem. Wie oben erwähnt übernimmt die Logikeinheit **110** in dem Unterstützungsmodul **82** viele der Funktionen für das Audio-Unterhaltungssystem. Diese Erfindung bezieht sich auf die Audio-Architektur für ein Audio-Unterhaltungssystem wie beispielsweise das in dem Computersystem **20** implementierte.

[0059] [Fig. 4](#) zeigt die Verbindung der Logikeinheit **110** und dem Audiosignalprozessor **116** und zusätzlich zwischen der Logikeinheit **110** und anderen durch einen 12C-Bus kontrollierten Audiokomponenten einschließlich des CODECs **122**, des AM/FM-Empfängers **118**, des RBDS-Decodierers **170**. In diesem Referenzaufbau ist der Audio-Signalprozessor **116** als ein DSP in dem SPI(Serial Peripheral Interface)-Slave-Modus konfiguriert.

[0060] Die Logikeinheit **110** gibt zwei serielle digitale I²S Audioströme DSPin0 und DSPin1 an den Audio-DSP **116** aus. Eine identische Parallel-zu-seriell-Wandlerschaltung erzeugt jeden Datenstrom. Die Schaltung liest ein paralleles Wort aus einem Halteregeister und ordnet es in einem Ausgabe-Schieberegeister an. Ein I²S-Takt (64 × 44,1 kHz) für die serielle Schiebeoperation, der extern durch den CODEC **122** erzeugt wird, schiebt die Daten aus dem Schieberegister. Die Übertragung der Daten von dem Halteregeister zu dem Schieberegister ist mit einem Haupt-SCLK-Signal (linker, rechter Taktgeber) synchronisiert, das extern durch den CODEC **122** ge-

neriert wird. Das LRCK-Signal wird zur Signalisierung genutzt, ob die Daten mit dem Audiosignal des rechten oder linken Kanals verknüpft sind.

[0061] Die beiden Stereoaudioströme werden an dem DSP **116** über zwei getrennte I²S-Eingangskanäle empfangen und werden zum Aufrufen irgendeiner der zahlreichen DSP-Audioalgorithmen wie beispielsweise dem Surround Sound genutzt. Die verarbeiteten Audiodaten werden von dem DSP **116** zurück an die Logikeinheit **110** über drei separate I²S-Kanäle zu einem DSPout-Interface ausgesendet. Die drei DSP-Ausgabekanäle umfassen einen Kanal SD0 für den Lautsprecher vorne links und vorne rechts, einen Kanal SD1 für den Lautsprecher hinten links und hinten rechts und SD2 umfasst den Center und Subwoofer. Jeder Tonkanal weist einen 20-bit Ton auf, der in einem 32 Takte langen Rahmen zusammengefasst ist, wobei das MSB (Most Significant Bit) dem MSB für I²S entspricht und zwölf Nullen dem LSB (Least Significant Bit) folgen.

[0062] Eine Tonquelle ist mit einem der DSPin-Kanäle in der Logikeinheit **110** durch einen DMA(Direct Memory Access)-Prozess der Überlagerung einer Ping/Pong-Pufferzone in den Eingabe/Ausgabe-Speicher **130** verbunden. Daten von einer Tonquelle werden an einem seriellen Interface empfangen und zu einem der Ping/Pong-DMA-Schaltungen **172** übergeben. Die DMA-Schaltungen **172** umfassen einen Seriell-zu-Parallel-Wandler zur Wandlung der Daten in ein paralleles Format für die Übergabe über den internen Bus der Logikeinheit zu dem Eingabe/Ausgabe-Speicher **110**.

[0063] Eine weitere Besonderheit dieser Hardwareimplementierung ist die Fähigkeit, dass vier Stereo-DMA-Prozesse parallel betrieben werden. Zwei Audioquellen wie beispielsweise eine Wave-Ausgabe und CD-ROM könnten in zwei Ziele wie beispielsweise DSPin0 und DSPin1 eingegeben werden. Die Vordergrundaudioquelle (wie beispielsweise die Anweisungen des Navigationssystems) können über die Frontlautsprecher des Autos abgespielt werden, während die Hintergrundaudioquelle (wie beispielsweise CD-Musik) über die Rücklautsprecher des Autos abgespielt werden können.

Allgemeine DMA-Struktur

[0064] [Fig. 5](#) zeigt ausführlicher eine Ping/Pong-DMA-Schaltung **172**. Alle der Ping/Pong-DMA-Schaltungen sind Abwandlungen desselben Grundaufbaus. Zur Erörterung wird eine DMA-Schaltung für einen Monomikrofonempfänger gezeigt.

[0065] Die DMA-Schaltung **172** weist einen Seriell-zu-Parallel-Wandler **200** zum Wandeln der seriellen von dem zugeordneten Tonquellenperipheriegerät empfangenen Daten in ein paralleles Format auf. Die parallelen Daten sind anfangs in einem Halterregister **202** angeordnet. Wenn das Halterregister **202** voll wird, sendet es ein Signal zu der DMA-Request-and-Grant-Schaltung **204**.

[0066] Ein DMA-Adresszeiger wird durch die Ausgabe von zwei logischen Komponenten gebildet: (1) einem Startadressregister **206**, das die oberen DMA-Adresszeilen zum Auffinden eines geeigneten Puffergebietes innerhalb des Eingabe-Ausgabe-Speichers **130** zum Halten der von dem Halterregister **202** empfangenen Daten enthält, und (2) einem Aufwärtszähler **208**, der die unteren DMA-Adresszeilen für das Puffergebiet erzeugt. Der Computerprozessor initialisiert den DMA-Prozess durch Schreiben eines Adresszeigers auf das Startadressregister **206**. Der Startadresszeiger wird in den unteren Adresszähler **208** und in das obere Adressregister **206** gleichzeitig geschrieben. Der obere Adressregisterzeiger definiert den Anfang des Puffergebiets in dem schnellen Datenspeicher **110**. In dieser Implementierung werden die unteren Adressbits in dem Startadressregister **208** zum Vergleichen mit dem zuletzt gültigen Adressregister genutzt. Die Startadresse des Puffergebiets ist auf einem Rand der Speicherseite und die Größe des Puffergebiets ist eine vorgegebene Größe entsprechend der Größe der Speicherseite.

[0067] Der untere Adresszähler **208** greift auf eine bestimmte Stelle innerhalb des bestimmten Pufferbereichs des Eingabe/Ausgabe-Speichers **130** zu. Die Anzahl der Bits in dem Zähler **208** entspricht der Größe des Puffergebietes. In dieser Implementierung verwendet der Zähler neun Adressbits A2 bis A10, wobei die Bits A0 und A1 zu Null gesetzt sind. Die Zahlausgabe durch den Zähler **208** wird als Zeiger auf eine bestimmte Stelle genutzt, in die die Daten in dem Halterregister **202** gespeichert werden müssen. Sobald die Daten an die Speicherstelle übermittelt sind, wird der Zähler **208** erhöht, um die nächste geeignete Stelle zu referenzieren.

[0068] Die DMA-Schaltung **172** definiert einen Puffer des Eingabe/Ausgabe-Speichers, der sich in zwei Hälften aufteilt: einem „Ping“-Puffer und einem „Pong“-Puffer. Eine Hälfte des Puffers repräsentiert einen Teil des Puffergebietes für ein bestimmtes Peripheriegerät wie beispielsweise die Monoeingabeschaltung, in die Daten von diesem Gerät geschrieben werden. Umgekehrt repräsentiert die andere Hälfte des Puffers einen anderen

Teil des Puffergebiets, von dem die Daten gelesen werden, der DSPin-DMA-Schaltung. Die Eingabe/Ausgabe-Pufferzonen für die Tonquelle weisen die gleichen Anfangs- und Endpunkte wie die Zone für die DSPin-Daten für den DSP auf, und daher kann über diese Zonen gesagt werden, dass sie sich „überlagern“. Während die Quelle den Ping-Speicherpuffer auffüllt, wird der Pong-Puffer zu dem DSPin-Ausgang ausgelesen. Nachdem die Quelle den Ping-Puffer gefüllt hat, beginnt sie, den Pong-Puffer zu füllen. Zur gleichen Zeit wird der Prozess zur seriellen DSPin-Ausgabe das Lesen des Pong-Puffers beenden und das Lesen des Ping-Puffers beginnen.

[0069] Das obere Adressbit A10 des unteren Adresszählers **208** unterteilt das Puffergebiet in den Ping- und den Pong-Puffer. Wenn ein Wechsel des oberen Adresszählerbits auftritt, kann ein Interrupt erzeugt werden, falls die entsprechende Interrupt-Maske aktiviert ist. Nach dem Schreiben des Startadresszeigers in das Register **206** und den Zähler **208** initialisiert der Prozessor ein Register **210** für die „letzte gültige Adresse“. Dieses Register ist ein Latch, das dieselbe Größe wie der untere Adresszähler **208** aufweist. Das Register **210** für die „letzte gültige Adresse“ enthält ebenso ein Bit zum Aktivieren der DMA, das zum Starten des DMA-Prozesses gesetzt wird.

[0070] Der DMA-Prozess beginnt mit dem Anfangsstartadresszeiger. Jeder DMA-Zyklus wird eine Erhöhung der DMA-Adresse um eine Einheit verursachen. Wenn der Wert in dem unteren Adresszähler **208** gleich dem Wert in dem Register **210** für die „letzte gültige Adresse“ ist, was durch den Komparator **212** bestimmt wird, wird der untere Adresszähler **208** auf Null zurückgesetzt und ein Interrupt erzeugt. Der DMA-Prozess fährt dann von dem Anfang des DMA-Puffers fort, der auf dem Rand einer Speicherseite liegt, wobei die Größe der Speicherseite gleich der Größe des Ping/Pong-Puffers ist. Wenn das MSB des unteren Adresszählers wechselt, wird ein Interrupt erzeugt werden (falls unmaskiert), um den Wechsel zwischen dem Ping- und dem Pong-Pufferraum zu definieren.

[0071] Der DMA-Prozess ist derart voreingestellt, dass er bei Datenpaketen in Byte-, Word- oder Double-Word-Größe vorkommen kann. Eine IrDA-DMA-Schaltung verwendet beispielsweise Datenpakete in Byte-Größe (d.h. 8 Bits), da diese die Grundgröße der empfangenen Daten ist. CD-ROM-Daten werden in Word-Größen (d.h. 16 Bits) übertragen, was für seine Busgröße angebracht ist. I²S-DMA-Daten werden auf einer Double-Word-Größe (d.h. 32 Bits) übertragen, was ein links/rechts 16 Bit-Datenpaar repräsentiert.

[0072] Die DMA-Adressbits A0 und A1 werden durch die geeignete DMA-Schaltung gesteuert. Diese Bits werden durch ein vier Byte breites Auswahlsignal decodiert und ersetzt. Die unteren beiden Bits (A0 und A1) des unteren Adresszählers werden genutzt, um eine der Byte-Auswahlleitungen auszuwählen, falls die Übertragung Byteweise geschieht. Das untere Bit (A1) des Zählers wird genutzt, um das obere Paar oder das untere Paar der Byte-Auswahlleitungen auszuwählen, falls die Übertragung Word-weise geschieht. In diesem Fall werden beide Byte-Auswahlleitungen gleichzeitig gesetzt. Alle vier Byte-Freigaben werden für Double-Word-Übertragungen gesetzt. Auf diese Weise bringt der DMA-Prozess Datenpakete in Byte-, Word- oder Double-Word-Größe in Einklang.

[0073] Im Falle eines seriellen Empfangsprozesses wird ein von einem Peripheriegerät empfangener serieller Datenstrom in einen Seriell-Parallel-Wandler **200** zum Bilden von Einheiten mit 8, 16 oder 32 Bit Länge geschoben. Nach einer geeigneten Anzahl von Schiebevorgängen werden die Daten an das Register **202** übergeben. Der Prozess des Übergabens von Daten an das Halterregister **202** setzt das REQ-Signal in der Ping/Pong-DMA-Schaltung.

[0074] Immer wenn das Halterregister **202** voll ist, wird die Ping/Pong-DMA-Schaltung **172** das nächste Audio-Wort von dem Halterregister **202** greifen und es in dem Eingabe/Ausgabe-Speicherpuffer **132** ablegen. Immer wenn ein DMA-Prozess eine Ping- oder Pong-Grenze überschreitet, wird ein Interrupt erzeugt werden (falls unmaskiert). Dies trifft sowohl für den DMA-Prozess der Audio-Quelle als auch für den DMA-Prozess des DSPin-Kanal zu. Der Umkehrschluss beinhaltet, dass im Falle eines Übertragungs-DMA-Prozesses wie beispielsweise dem DSPin0-Kanal bei einem leeren Halterregister die DMA-Ping/Pong-Schaltung das nächste Audio-Wort von dem Eingabe/Ausgabe-Speicher greifen und es in dem Halterregister ablegen wird.

Synchronisieren der DMA-Prozesse auf gemeinsam genutzte Ping/Pong-Puffer

[0075] Nochmals in Bezug auf [Fig. 4](#) benutzt die Audio-Architektur die durch die DMA-Schaltungen **172** erzeugten Interrupts als eine Möglichkeit zum Synchronisieren von zwei DMA-Prozessen, die denselben Ping/Pong-Puffer **132** teilen. Daten von einer DMA-Schaltung werden in den gemeinsam Ping/Pong-Puffer **132** geschrieben und anschließend von dem Puffer zu einer anderen DMA-Schaltung gelesen. In Abhängigkeit von

der Audio-Quelle können die DMA-Schaltungen mit unterschiedlichen Raten schreiben und lesen. In der beschriebenen Implementierung nutzt die Ausgabe-DMA-Schaltung eine Abtastrate von 44,1 kHz, die durch die Logikeinheit zur Verfügung gestellt wird. Jedoch kann die Eingabe-DMA-Schaltung eine geringfügig andere Datenrate nutzen. Ein Audio-Gerät beispielsweise, das mit seinem eigenen Taktsignal arbeitet und das an das USB angeschlossen ist, kann Audiodaten in den Puffer **132** mit einer Rate schreiben, die nicht genau 44,1 kHz beträgt. Demzufolge kann die Eingabe-DMA-Taktrate nicht genau mit der Ausgabe-DMA-Taktrate übereinstimmen, da sie von unterschiedlichen Taktschaltungen gesteuert werden.

[0076] In den bevorzugten Implementierungen steuern unterschiedliche Threads und Gerätetreiber die Lese- und Schreibe-DMA-Prozesse des gemeinsam genutzten Ping/Pong-Puffers. Ein Gestalter einer Audioquelle braucht lediglich einen Gerätetreiber zu entwickeln, der den gemeinsam genutzten Puffer mit seiner Taktrate füllt.

[0077] Es werden folgende Beispiele betrachtet. In einem ersten Szenario sind die Schreibe- und Lese-DMA-Prozesse auf dem 44,1 kHz Haupttakt basiert. Zur Erläuterung dieses Szenarios wird davon ausgegangen, dass der AM/FM-Empfänger **118** die Quelle der Audiodaten umfasst. Die Audio-Datenausgabe des AM/FM-Empfängers **118** wird durch den CODEC **122** digitalisiert und als ein I²S-Audiostrom zu der Logikeinheit **110** ausgegeben. Die Logikeinheit **110** wandelt den I²S-Strom in ein linkes und ein rechtes Paar von 16 Bit-Worten um und eine erste DMA-Schaltung für den AM/FM-Empfänger schreibt das Wortpaar in den „Ping“-Teil des durch den DMA-Zeiger angezeigten Puffers. Während die AM/FM-Empfängerdaten in den Ping-Speicher geschrieben werden, liest eine zweite DMA-Schaltung gleichzeitig Daten von dem entsprechenden „Pong“-Teil des Pufferspeicherprozesses und gibt diese Daten an den DSPin-Ausgang aus. Da sowohl der Schreibe-Prozess als auch der Lese-Prozess mit 44,1 kHz ablaufen, geschieht der Wechsel zwischen Ping und Pong gleichzeitig sowohl bei den einkommenden als auch bei den abgehenden Daten. Die Interrupts, die sowohl durch die einkommenden als auch abgehenden Wechsel erzeugt werden, können ignoriert werden, außer wenn der Computer ein Interesse an den Daten (beispielsweise beim Senden an USB oder an Diskette) hat.

[0078] In dem zweiten Szenario schreibt die Quelle Daten mit einer geringfügig von den internen 44,1 kHz abweichenden Rate. Angenommen ein CD-ROM-Spieler ist beispielsweise die Quelle der Daten. In diesem Fall wird der Ping/Pong-Speicherpuffer **132** derart konfiguriert, dass der CD-ROM-Speicherbereich mit dem Speicherbereich der DSPin-Ausgabe überlappt. Eine mit dem CD-ROM assoziierte DMA-Schaltung **172** füllt den „Ping“-Speicherpuffer mit Bytes der Audiodaten zuzüglich einiger Worte von Sub-Kanal-Daten. Gleichzeitig liest die DMA-Schaltung für die DSPin-Ausgabe Daten aus dem „Pong“-Speicherpuffer mit einer Rate von 44,1 kHz.

[0079] Als nächstes wird angenommen, dass der CD-ROM-Spieler Audiodaten von der CD-ROM mit einer Audiorate von 44,1 kHz liest. Mit dieser Rate erzeugt der CD-ROM-Spieler jede 1/75 Sekunde einen Block von 2,352 Bytes der Audiodaten. Wenn dieser Block bereit ist, aktiviert der CD-ROM-Spieler die DMA-Request-Leitung, überträgt rasch den Datenblock und deaktiviert die Leitung bis er den nächsten Block empfängt. Die DMA-Schaltung erzeugt ein Interrupt an dem Ende der Übertragung des Blocks. Aufgrund dieses Prozesses kann die Audiodatenrate des CD-ROM-Spielers geringfügig anders als der Haupttakt von 44,1 kHz sein, der den Audio-DSP **116** und den CODEC **122** ansteuert.

[0080] Zum Synchronisieren dieser beiden Datenströme benutzen die Software-Gerätetreiber die Logikeinheit **110**, um die Zeit zwischen dem Datentransfer der Blöcke des CD-ROM-Spielers zu messen, und passt den Hauptaudiotakt des Unterstützungsmoduls derart an, dass er mit der CD-ROM-Datenrate übereinstimmt.

[0081] [Fig. 6](#) zeigt eine Haupttakteinstellerschaltung **230**, die in der Logikeinheit **110** implementiert ist. Die Haupttakteinstellerschaltung **230** weist ein Referenzregister **232**, ein Zähler **234** mit eingebautem Teiler durch N, zwei Komparatoren **236**, zwei Latches **238** und eine Steuerlogik **240** auf. Die zwei Komparatoren **236** helfen bei der Bestimmung der Rate, mit der Daten in einen Ping/Pong-Puffer geschrieben werden. Der Host-Computer schreibt einen Zeiger auf die Latches **238**. Der Zeiger entspricht einer Adresse in dem Eingabe/Ausgabe-Speicher **130** eines Ping/Pong-Puffers **132**. Jede Operation, die Daten an diese Adresse schreibt, verursacht eine Erzeugung eines Interrupts an den Host (falls unmaskiert). Wenn der Interrupt auftritt, liest der Computer die Adressposition des DMA-Prozesses, während er Daten ausliest, und basierend auf diese Position bestimmt er, ob die ankommende Datenrate mit der ausgehenden Datenrate übereinstimmt.

[0082] Falls die Raten abweichen, sendet der Computer einen Wert an das Referenzregister **232**. Dieser Wert wird an den Zähler **234** mit eingebautem Teiler durch N zur Erzeugung der Referenz-Audiotaktrate für die di-

gitalen Audiokanäle auf dem Unterstützungsmodul eingegeben. Für gewöhnlich liegt der Referenzaudiotakt bei 44,1 kHz. Jedoch kann der Computer einen Wert übertragen, der den Referenzaudiotakt durch Beschleunigung oder Verlangsamung der Rate, mit der die Daten ausgelesen werden, derart anpasst, dass sie mit den Datenraten der ankommenden Audioströme synchronisiert wird.

[0083] In dem Software-Aufbau kann ein einziger gemeinsam genutzter Speicherpuffer für mehrere Eingabegeräte zum Schreiben von Daten und für lediglich ein Ausgabegerät zum Lesen der Daten genutzt werden. Der Lese- und Schreibe-DMA-Prozess des gemeinsam genutzten Ping/Pong-Puffers werden durch verschiedene Threads und Gerätetreiber gesteuert. Die Aufgabe hier ist das Schreiben eines Gerätetreibers zur Standardausgabe, um zu den meisten Audioquellen zu passen. Der Gerätetreiber der Audioquelle wird lediglich dazu benötigt, um den gemeinsam genutzten Puffer mit seiner eigenen Rate zu füllen.

[0084] Eine Alternative zur Benutzung eines einzigen gemeinsam genutzten Puffers zwischen verschiedenen Audioquellen ist der Einsatz von mehreren gemeinsam genutzten Speicherpuffern für sich entsprechende Audioquellen. Jede Audioquelle benutzt ihren eigenen Puffer und der Ausgabegerätetreiber schaltet zwischen den verschiedenen Eingangspuffern, falls die aktuelle Audioquelle wechselt. Der Nachteil bei dieser Implementierung besteht darin, dass sie mehr Speicherplatz benötigt. Zusätzlich ist es nicht notwendiger Weise einfach für eine neue Audioquelle, den Ausgabegerätetreiber von einem neuen gemeinsam genutzten Audiopuffer in Kenntnis zu setzen.

[0085] Wie oben erwähnt füllt der Schreibe-Thread (durch den Gerätetreiber der Audioquelle gesteuert) den gemeinsam genutzten Puffer mit einer durch das Gerät gesteuerten Taktrate, die ungefähr bei einer Abtastrate von 44,1 kHz liegt. Falls der Lese-Thread ebenso auf der gleichen Taktrate läuft (beispielsweise ein Radio) entstehen keine Probleme. Jedoch wird für Peripheriegeräte (beispielsweise CD und USB-Geräte) die Taktrate nicht exakt zusammenpassen. Früher oder später wird ein Ausgaberauschen entstehen, da die Eingabe-DMA und Ausgabe-DMA einander einholen können und mit einem Zugriff auf dieselbe Ping/Pong-Pufferregion beginnen. Damit der Lese-Thread seine Taktrate zu der des Schreibe-Threads anpassen kann, verwendet das System den Schreibe-Benachrichtigungs-Interrupt (der auslöst, wenn das mittlere DWORD des Ping/Pong-Puffers geschrieben ist). Durch Beobachten des gewöhnlichen Lesen-Erfolgt-Interrupts und dieses Schreibe-Benachrichtigungs-Interrupts kann der Lese-Thread seinen Takt derart anpassen, dass er mit dem des Schreibe-DMA übereinstimmt.

[0086] Wenn der Lese-Thread den ersten Schreibe-Interrupt detektiert, kann er mit dem Lesen beginnen. Für jeden folgenden Schreibe-Interrupt ist er in der Lage herauszufinden, wo er aktuell aus dem Puffer liest. Aus dieser Information kann er den Lesetakts zur Übereinstimmung mit dem des Schreibe-Threads erhöhen oder erniedrigen. Wenn er zwei Lese-Interrupts ohne einen Schreibe-Interrupt detektiert kann er automatisch den Vorgang beenden.

Bus-Schlichter

[0087] [Fig. 7](#) zeigt eine Bus-Schlichterschaltung **250**, die den Zugriff auf den internen Bus steuert. Dieser interne Bus wird zum Zugriff zwischen der PCI-Bus-DMA-Schaltung und dem Eingabe/Ausgabe-Speicher genutzt. Die Bus-Schlichterschaltung **250** schaltet einen 2:1-Multiplexer **252** zurück und vor zwischen den von den PCI-Bus **86** empfangenen Busanfragen und den von den DMA-Schaltungen **172** in der Logikeinheit **110** empfangenen DMA-Anfragen. Die Bus-Schlichterschaltung **250** gewährt dem PCI-Bus **86** die Hälfte der verfügbaren Zeit auf dem internen Bus und verteilt die verbleibende Hälfte der Zeit zwischen den acht DMA-Prozessen, von denen alle gleichzeitig aktiv sein können. In der Zukunft können mehr als acht DMA unterstützt werden.

[0088] In dem dargestellten Beispiel unterstützt die Schlichterschaltung **170** acht DMA-Schaltungen: Stereo, Mono, DSPout (d.h. der Audio-Signalprozessor), IrDA-Empfang, DSPin0, DSPin1, IrDA-Übertragung und Blende. Diese acht Schaltungen teilen sich die verbleibende Hälfte der Bandbreite des internen Busses und daher wird jeder DMA-Schaltung 1/16 der Busbandbreite garantiert. Man beachte, dass das CD-ROM auf den Eingabe/Ausgabe-Speicher unter Verwendung des PCI-DMA-Prozesses schreibt.

[0089] Falls der 2:1-Multiplexer **252** eine aktive Anfrage decodiert, beendet er den Wechsellvorgang und stößt eine Buszyklusstatusmaschine **254** für den Bus der Logikeinheit an. Die aktive Anfrage bewilligt ebenso den internen Bus der Logikeinheit zu entweder den PCI-Bus oder zu einem der internen DMA-Schaltungen, wie durch einen Decoder **256** ausgewählt.

[0090] Ein Schieberegister **258** lädt alle anstehenden Anfragen von den DMA-Schaltungen. Die Ausgabe des Schieberegisters **258** fungiert als Freigabe für einen Zähler **260**. Jeden Mal wenn das Schieberegister **258** schaltet, zählt der Zähler **260**. Wenn die Ausgabe aus dem Schieberegister **258** eine aktive DMA-Anfrage anzeigt, enden der Zähler **260** und das Schieberegister **258**. Wenn der 2:1-Multiplexer **252** zu den DMA-Schaltungen wechselt, hält die aktive Anfrage von dem Schieberegister **258** das Wechseln an und startet einen Buszyklus für eine DMA-Anfrage. Nach dem Abschluss des Buszyklus wird ein „Beendet“-Signal an die entsprechende DMA-Schaltung generiert, die die DMA-Anfrageleitung freigibt und den 2:1-Multiplexer **252**, das Schieberegister **258** und den Zähler **260** dazu veranlasst, mit dem Suchen der nächsten aktiven DMA-Anfrage fortzusetzen.

Audio-Manager-API

[0091] Das Computersystem **20** ist ein voll funktionsfähiger Mehrzweck-Computer, der ein Audio-Unterhaltungssystem unterstützt. Das System **20** ist dazu konzipiert, mehrere Audioquellen wie beispielsweise Radio, CD, Auxiliary-Eingänge und CD-Wechsler zu unterstützen. Der Audio-DSP **116** bearbeitet die Tonentzerrung, das Decodieren des Surround Sounds und die Steuerung der Lautstärke/Balance/Überblenden. Der Ton kann über ein Lautsprechersystem (beispielsweise sechs Lautsprecher) oder über irgendein USB-Gerät abgespielt werden.

[0092] Das Computersystem **20** implementiert ein Audio-Manager-API (Application Program Interface), um die auf dem Computer ablaufenden Anwendungen zum Steuern der verschiedenen Audioquellen ohne Kenntnis der Hardware und Details der Implementierung der zugrunde liegenden Audioanlage zu befähigen. Verschiedene Audiogeräte und ihre Treiber steuern unterschiedliche Funktionalität des Audiosystems wie beispielsweise Entzerrung, Lautstärkensteuerung und Decodierung von Surround Sound. Die Audio-Manager-API übermittelt die durch die Anwendungen getätigten Aufrufe an den/die entsprechenden Gerätetreiber.

[0093] [Fig. 8](#) zeigt die Anwendung-zu-Hardware-Architektur. Die Audio-Hardware **270** bildet die unterste Ebene in der Architektur. Eine Audio-Hardware-Abstraktionsschicht (HAL, hardware abstraction layer) **272** definiert eine grundlegende Interface-Schicht zwischen den audiobezogenen Treibern für die Hardware **270** und der Audio-Manager-API-Schicht **274**. Über der Audio-Manager-API **274** befinden sich die Anwendungen **276**. Die Audio-Manager-API **274** definiert die APIs zum Zugreifen und Steuern des zugrunde liegenden Audiosystems.

[0094] Die Audio-Manager-API **274** weist fünf Kernkomponenten auf: die Audio-Quellen-Steuerungs-API **278**, die Wave-in-und-Wave-out-API **280**, die Surround-Sound-Decodier-API **282**, die Entzerrer-API **284** und die Lautstärke/Balance/Überblende-API **286**. Verschiedene API steuern verschiedene Aspekte des Audiosystems. Zusammen bieten sie eine geeignete API für die Anwendungen **276**.

[0095] Die Audioquellen sind allgemein in zwei Kategorien unterteilt: Vordergrund-Audioquellen und Hintergrund-Audioquellen. Vordergrund-Audioquellen umfassen eine Computer-Wave-Dateiausgabe und eine Text-zu-Sprache-Ausgabe. Vordergrund-Audioquellen werden mit aktiven Hintergrund-Audioquellen gemischt. Die Vordergrund-Audioquellen werden durch die Wave-out-API gesteuert. Im Gegensatz dazu umfassen die Hintergrund-Audioquellen AM/FM-Empfänger, CD-Spieler, Auxiliary-Eingänge und andere Quellen von dem USB. Hintergrund-Quellen werden durch die Audioquellen-Steuerungs-API gesteuert.

[0096] Obwohl das Audiosystem mehr als ein Ziel, wie beispielsweise die Lautsprecher, USB-Geräte und den Computer aufweisen kann, ist das Hauptziel die Lautsprecher. Daher läuft jede der ausgewählten vielen Audioquellen letztendlich bei einem Audio-Ziel – dem Lautsprechersystem – zusammen.

[0097] [Fig. 9](#) zeigt wie die Module der Audio-Manager-API den Audiodurchflussweg steuern.

[0098] Die Audioquellen-Steuerungs-API **278** wird dazu verwendet, die Hintergrund-Audioquelle auszuwählen. Sie dient als der Koordinator zwischen den verschiedenen Audio-Anwendungen, die verschiedene Teile des Audiosystems steuern. Während beispielsweise der AM/FM-Empfänger spielt und der Bediener das Abspielen einer CD wählt, dient die Audioquellen-Steuerungs-API **278** dazu, das Anhalten des Empfängers und das Starten des CD-Spielers zu koordinieren. Tabelle 1 listet die Funktionen der Audioquellen-Steuerungs-API **278** auf.

Name	Funktion
GetNumSources	Gibt die gesamte Anzahl der registrierten Audioquellen zurück
GetSourceCaps	Stellt Informationen über eine bestimmte Audioquelle zur Verfügung
RegisterCallback	Registriert ein Rückfrage-Thread
SelectSource	Wählt eine bestimmte Audio-Quelle zum Abspielen aus.

Tabelle 1: Audioquellen-Steuerungs-API 278

[0099] Die Wave-in-und-Wave-out-API **280** sind kompatibel mit den Win32-Wave-APIs, die in einem Standard Windows SDK wie beispielsweise dem SDK für das Betriebssystem Windows CE dokumentiert sind. Für genauere Informationen über diese APIs wird der Leser auf die erwähnte SDK verwiesen.

[0100] Die Surround-Sound-API **282** ermöglicht das Decodieren von Surround Sound. Sie enthält Teilfunktionen, um SSD zu aktivieren und deaktivieren, die Verzögerungszeit des Surround Sounds zu setzen und eine Grenzfrequenz eines Subwoofer-Tiefpassfilter einzustellen. Tabelle 2 listet die Funktionen der Surround-Sound-API **282** auf.

Name	Funktion
EnableSSD	Aktiviert oder deaktiviert das Decodieren des Surround Sounds
SSDControl	Steuert den Decodieralgorithmus für den Surround Sound

Tabelle 2: Surround-Sound-API 282

[0101] Die Entzerrer-API **284** verändert die Entzerrer-Einstellungen. Tabelle 3 listet die Funktionen der Entzerrer-API **284** auf.

Name	Funktion
GetEQCaps	Empfängt verschiedene Eigenschaften des Audio-Equalizers
EQPreset	Wählt eine Equalizer-Voreinstellung: Pop, Country, Rock, Klassik, etc.
EQBandControl	Steuert die Verstärkung, Mittelfrequenz und Bandbreite jedes einzelnen Bandes
FilterControl	Steuert das Übergangsfiler

Tabelle 3: Entzerrer-API 284

[0102] Die Lautstärke/Balance/Überblende-API **286** verändert die Lautstärke, Balance und Überblenden. Tabelle 4 listet die Funktionen der Lautstärke/Balance/Überblende-API **286** auf.

Name	Funktion
VolumeControl	Steuert Lautstärke, Stummschalten, Balance, Überblenden, Mitte und Woofer

Tabelle 4: Lautstärke/Balance/Überblende-API 286

[0103] Wieder beziehend auf [Fig. 8](#) kommuniziert die Audio-Manager-API **274** mit den Audio-Gerätetreibern für bestimmte Geräte über das Audio-HAL-Interface **272**. Für die Zusammenarbeit mit Anwendungen **276** unterstützt jeder Gerätetreiber den folgenden Satz der in Tabelle 5 aufgelisteten Interfaces.

Name	Funktion
IUnknown	Standardinterface, wird zur Anfrage der Verfügbarkeit aller Interfaces genutzt
ISource	Interface zwischen der Audioquellen-Steuerungs-API und den Audioquellen-Treibern
IEQ	Interface für die Audio-Entzerrung
ISSD	Interface für das Decodieren des DSP von Surround Sound
IVolume	Interface für die Steuerung von Lautstärke/Balance/Überblenden

Tabelle 5: Interfaces für HAL 272

[0104] Zum Abspielen einer Audioquelle sind gewöhnlich mehr als ein Gerät beteiligt. Beispielsweise wird zum Abspielen einer CD zuerst ein gemeinsam genutztes Puffergerät vorbereitet bevor der CD-Treiber die Audiodaten zu dem gemeinsam genutzten Puffer sendet. Zum Aufbauen irgendeiner Abhängigkeit zwischen den Geräten registrieren sich alle audiobezogenen Geräte und spezifizieren, auf welchen Gerätetreiber sie angewiesen sind.

[0105] Wenn eine Anwendung eine Audioquelle abspielen möchte, ruft sie SelectSource mit der Audioquellen-ID als ihr Parameter auf. Die Audio-Manager-API deselektiert irgendeine aktuelle Audioquelle und selektiert die Audioquelle, die durch die Anwendung angefordert wurde. Zum Deselektieren der aktuellen Audioquelle durchläuft die Audio-Manager-API die Struktur der Verbindungsliste und informiert jede Quelle oder jedes abhängige Gerät über ihre Beendigung. Zum Auswählen der neuen Quelle informiert sie die Quelle und die abhängigen Geräte über ihre Inbetriebnahme.

DSP-SPI für das Mikrofon-Interface

[0106] Das Computersystem **20** unterstützt ein Spracherkennungssystem wie beispielsweise ein diskretes Steuerungs- und Befehl-Spracherkennungssystem. Alle Audiodaten einschließlich der durch die Mikrofon-Buchse **72** erfassten Sprachäußerungen werden mit einer Abtastrate von 44,1 kHz, die durch den CODEC zur Verfügung gestellt wird, abgetastet. Dennoch kann das Spracherkennungssystem eine andere Abtastrate verwenden. In der beschriebenen Implementierung kann das Spracherkennungssystem eine erheblich langsamere Abtastrate von 11 kHz verwenden. Anstatt einen zusätzlichen Analog/Digital-Wandler lediglich für die Mikrofondaten hinzuzufügen (was zusätzliche Kosten verursachen würde) konvertiert das System **20** die Mikrofondaten von der höheren Abtastrate von 44,1 kHz zu den gewünschten 11 kHz.

[0107] Eine Möglichkeit wäre, die Daten mit 44,1 kHz in dem Computer aufzunehmen und dem Host-Computer zu erlauben, eine Abtastratenumwandlung durchzuführen. Jedoch ist dieser Ansatz weniger erstrebenswert, da er Ressourcen des Computers verbraucht.

[0108] Ein eher bevorzugter Ansatz ist, eine Abtastratenumwandlung in dem Audio-DSP **116** oder in der Logikeinheit **110** durchzuführen. Da jedoch die I²S-Interfaces alle anderweitig zweckbestimmt sind, ist ein Aspekt der Erfindung, die Mikrofondaten huckepack mit den Befehls/Nachrichten-Strömen, die an der SPI (Serial Peripheral Interface) des Audio-DSPs empfangen werden, zu senden. Die Mikrofondaten werden huckepack mit den Kommandos der Host-CPU genommen und in den DSP über das SPI gesendet. Der DSP führt eine normale Tiefpassfilterung und ein Down-Sampling mit dem Datenstrom durch und benutzt dann das SPI, um die Mikrofondaten zurück auf ihrer neuen Abtastrate von 11 kHz huckepack mit dem Nachrichtenstrom auszusenden.

[0109] [Fig. 10](#) zeigt den Fluss der Mikrofondaten bis zum dem Punkt, an dem sie der Host-CPU zum Durch-

führen der Spracherkennung übergeben wird. Ein durch das Mikrofon **72** erzeugtes Analogsignal wird in einem Verstärker **290** verstärkt und in den Monokanal des CODEC **122** eingegeben. Der CODEC **122** digitalisiert das analoge Mikrofonsignal mit einer Auflösung von 20 Bit und einer Abtastrate von 44,1 kHz. Die 20 Bits werden in einem 32 Taktzyklen langen Rahmen zusammengefasst, wobei das MSB dem MSB des I²S entspricht und 12 Nullen dem LSB folgen. Der CODEC sendet die 20 Bits der Mikrofondaten zu der Logikeinheit **110** über den I²S.

[0110] Die Logikeinheit **110** empfängt die seriellen I²S-Daten und wandelt sie in parallele Daten in der Mono-DMA-Schaltung **172**, wie in [Fig. 5](#) dargestellt, um. Die Mono-DMA-Schaltung **172** überreicht die 20 Bit der Monodaten dem SPI-Modul **292**, das die Monodaten in einen 24 Bit SPI-Seriell-Vorgang für den Audio-DSP **116** umwandelt. Es sei bemerkt, dass der DSP **116** gleichzeitig zwei Kanäle I²S-Ton von den DSPin-TX-Ping/Pong-DMAs für insgesamt drei Audio-Eingangsströme mit dem Mono-Audio-Strom huckepack mit den Befehls-/Nachrichtenströmen empfangen kann.

[0111] Der Audio-DSP **116** wendet ein Bandpassfilter **294** auf die Monodaten an und dezimiert dann die Daten auf ein 11 kHz Strom durch Behalten von jedem vierten Abtastwert der Filterausgabe und Verwerfen der anderen drei. Nach jeder vierten SPI-Mono-Eingabetransaktion gibt der DSP die gültig gefilterten Monoausgabedaten an die Logikeinheit **110** zurück. Für jeden 44,1 kHz Zeitpunkt wird ein Mikrofondatenwort zu dem DSP gesendet. Jedoch kommt lediglich nach jedem vierten gesendeten Mikrofonwort ein gefiltertes Mikrofonwort von dem DSP zurück, das Mikrofondaten mit einer neuen Abtastrate von 11 kHz repräsentiert. Für jedes an den DSP gesendete Wort wird ein Wort zurückgesendet. Mit anderen Worten für jede zu dem DSP gesendete vier Worte gibt der DSP ein als gefilterte Daten markiertes Wort und drei als ungültige Daten markierte Worte zurück.

[0112] Das SPI-Modul **292** konvertiert die 16 MSBs der gefilterten Monodaten in parallele Daten und überreicht sie zurück zu der Mono-DMA-Schaltung **172**.

[0113] Die Mono-DMA-Schaltung **172** erzeugt ein DMA-Anfrage-Signal, nachdem sie zwei aufeinander folgende gefilterte 16 Bit Worte in ihrem Halterregister gespeichert hat. Die Daten werden über einen Multiplexer **256** in einem Ping/Pong-Puffer 132 übertragen. Wenn der Ping-Puffer gefüllt ist, wird ein Interrupt zur Host-CPU erzeugt. Eine PCI-DMA-Schaltung **296** liest die Monodaten aus dem Ping-Puffer in den Eingabe/Ausgabe-Speicher **130** als Reaktion auf die Anweisungen von dem PCI-Controller **298** und legt die Daten in einen Puffer **300** in dem lokalen Host-Speicher **152** ab. Die Spracherkennungssoftware **302** ruft Blöcke der Monodaten von dem Puffer **300** zur Verarbeitung ab.

[0114] Falls der Host einen Befehl in das SPI-Datenregister in der Logikeinheit **110** schreibt, wird die SPI-Interface-Logik dieses Befehlswort zu dem Audio-DSP **116** unmittelbar der nächsten Mikrofondatentransaktion folgend senden. Da die Mono-Audio-Daten nicht die volle Bandbreite des SPI-Busses benutzt, kann der Befehls-/Nachrichtenstrom mit den Monodaten verschachtelt werden. Nachdem der Audio-DSP zurückgesetzt wird, wird er das Laden von 512 (24 Bit) Worten von dem SPI-Port erwarten, die sein ausführbares Programm oder ein Programm zum Laden eines größeren ausführbaren Programms enthalten. Unter diesen Umständen enthält das SPI-Steuerregister Bits, die die Einspeisung von Mikrofondaten deaktivieren werden, und ermöglicht, in den SPI-Datenport geschriebene Daten direkt an den DSP zu senden. In diesem Modus werden alle von dem DSP empfangenen Daten ignoriert.

[0115] Obwohl die Erfindung in einer auf die strukturellen Merkmale und/oder methodologischen Schritte zugeschnittenen Sprache beschrieben wurde, ist es zu verstehen, dass die in den angehängten Patentansprüchen definierte Erfindung nicht notwendiger Weise auf die beschriebenen bestimmten Merkmale und Schritte beschränkt ist. Die besonderen Merkmale und Schritte sind eher als beispielhafte Form der Implementierung der beanspruchten Erfindung offenbart.

Patentansprüche

1. Unterhaltungssystem, das
 einen Datenspeicher (**130**),
 eine zum Empfang von Audiodaten von einer oder mehreren peripheren Audioquellen (**26, 28, 34, 94, 96, 98, 100, 102**) angekoppelte Logikeinheit (**110**) und
 einen Datenprozessor (**116**) zum Verarbeiten der Audiodaten umfasst,
 wobei die Logikeinheit (**110**) eine oder mehrere Audioquellen-DMA(direct memory access)-Schaltungen (**172**) aufweist, die jeweils entsprechenden peripheren Audioquellen zugeordnet sind, wobei jede Audioquel-

len-DMA-Schaltung (**172**) die von der zugeordneten peripheren Audioquelle empfangenen Audiodaten in einen Speicherbereich innerhalb des Datenspeichers (**130**) schreibt, und wobei die Logikeinheit (**110**) auch eine Datenprozessor-DMA-Schaltung aufweist, die dem Datenprozessor (**116**) zum Auslesen der Audiodaten aus dem Speicherbereich innerhalb des Datenspeichers (**130**) zur Übermittlung an den Datenprozessor (**116**) zugeordnet ist.

2. Unterhaltungssystem nach Anspruch 1, das zusätzlich eine Synchronisationsschaltung umfasst, um eine erste Rate, mit der die Audioquellen-DMA-Schaltung (**172**) die Audiodaten in den Datenspeicher (**130**) schreibt, mit einer zweiten Rate zu synchronisieren, mit der die Datenprozessor-DMA-Schaltung die Audiodaten aus dem Datenspeicher (**130**) liest.

3. Unterhaltungssystem nach Anspruch 1, worin die Logikeinheit (**110**) Audiodaten von den vielen verschiedenen Audioquellen an den Datenprozessor (**116**) zu einer konkurrierenden Verarbeitung konkurrierend übermittelt.

4. Fahrzeug-Computersystem, das ein Unterhaltungssystem nach Anspruch 1 umfasst.

5. Unterhaltungssystem nach Anspruch 1, worin die eine oder mehreren Audioquellen-DMA-Schaltungen (**172**) eine erste DMA-Schaltung (**172**), eine zweite DMA-Schaltung (**172**) und eine dritte DMA-Schaltung (**172**) umfassen, und die eine oder mehreren peripheren Audioquellen eine erste periphere Audioquelle und eine zweite periphere Audioquelle ausweisen, wobei die erste DMA-Schaltung (**172**) der ersten peripheren Audioquelle zum Übermitteln erster Audiodaten von der ersten peripheren Audioquelle zu einer ersten Stelle in dem Datenspeicher (**130**) zugeordnet ist, wobei die zweite DMA-Schaltung (**172**) der zweiten peripheren Audioquelle zum Übermitteln zweiter Audiodaten von der zweiten peripheren Audioquelle zu einer zweiten Stelle in dem Datenspeicher (**130**) zugeordnet ist, und wobei die dritte DMA-Schaltung (**172**) dem Datenprozessor (**116**) zum Übermitteln der ersten und zweiten Audiodaten von den ersten und zweiten Stellen in dem Datenspeicher (**130**) an den Audiodatenprozessor (**116**) zur konkurrierenden Verarbeitung und Ausgabe an ein oder mehrere Audioziele zugeordnet ist.

6. Unterhaltungssystem nach Anspruch 5, worin die ersten Audiodaten zu einem ersten Satz von Lautsprechern und die zweiten Audiodaten gleichzeitig zu einem zweiten Satz von Lautsprechern ausgegeben sind.

7. Unterhaltungssystem nach einem der Ansprüche 1 bis 6, das zusätzlich ein Application Program Interface (**274**) zur Definition einer Schnittstelle zwischen den auf dem System betriebenen Anwendungsprogrammen (**276**) und einem Audiounterhaltungssystem umfasst, wobei das Application Program Interface (**274**) auf dem Datenprozessor (**116**) des Systems ausführbar ist und Funktionen ausweist, die von einem Anwendungsprogramm zur Steuerung des Audiounterhaltungssystems aufrufbar sind.

8. Unterhaltungssystem nach Anspruch 7, worin das Application Program Interface zusätzlich eine Funktion (**278**) zur Auswahl der Audioquelle umfasst.

9. Unterhaltungssystem nach Anspruch 7 oder 8, worin das Application Program Interface zusätzlich eine Funktion (**282**) zum Ein- und Ausschalten einer Surround Sounds umfasst.

10. Unterhaltungssystem nach einem der Ansprüche 7 bis 9, worin das Application Program Interface zusätzlich eine Funktion (**284**) zum Ändern von Equalizer-Einstellungen umfasst.

11. Unterhaltungssystem nach einem der Ansprüche 7 bis 10, worin das Application Program Interface zusätzlich eine Funktion (**286**) zum Ändern von Lautstärke, Balance und der Überblendcharakteristiken umfasst.

12. Fahrzeug-Computersystem, das ein Unterhaltungssystem mit einem Application Program Interface nach einem der Ansprüche 7 bis 11 umfasst.

Es folgen 10 Blatt Zeichnungen

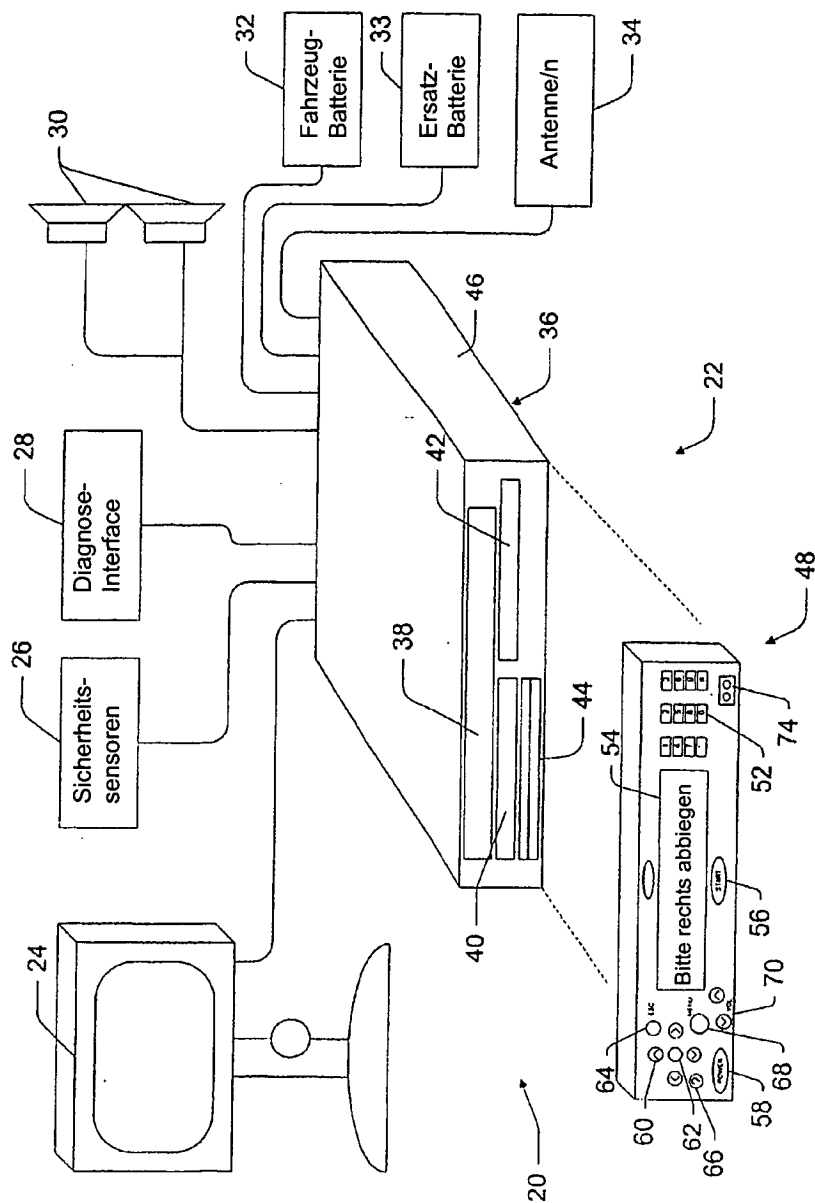


Fig. 1

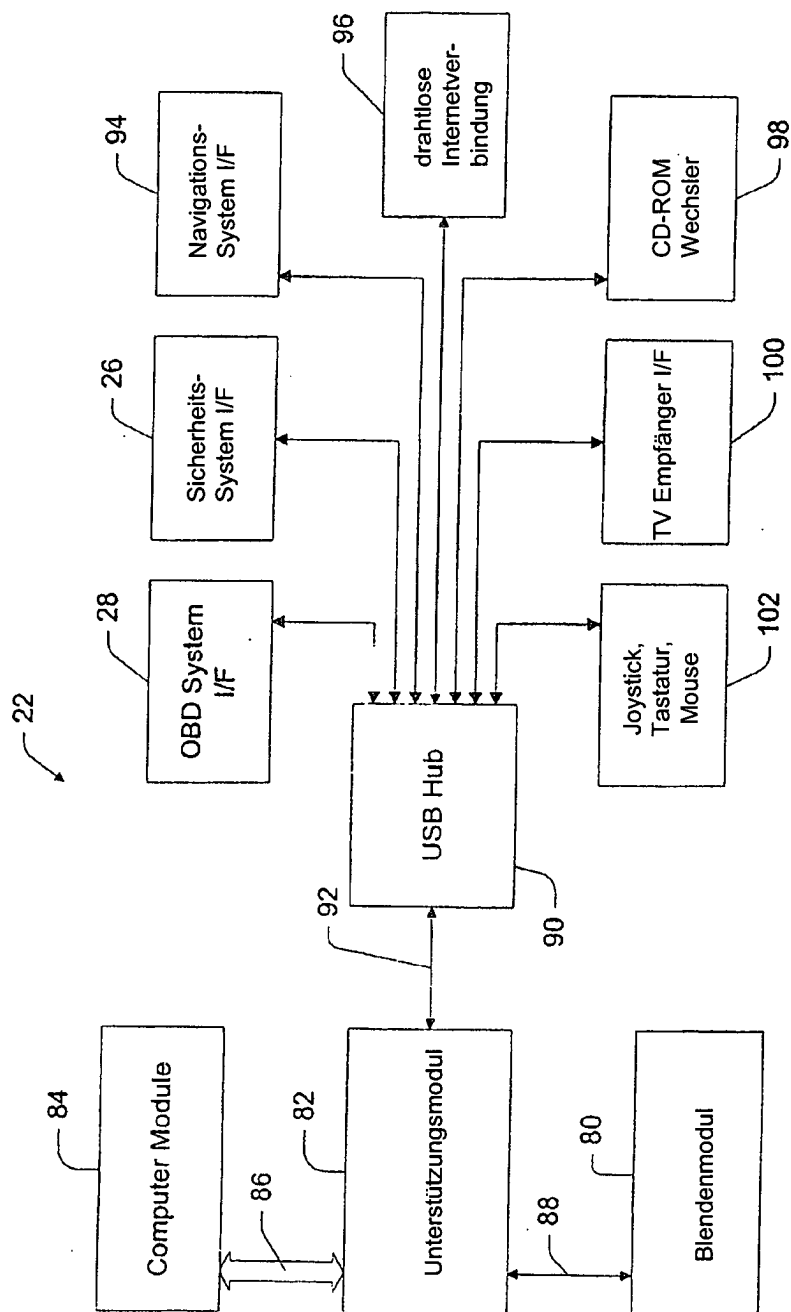


Fig. 2

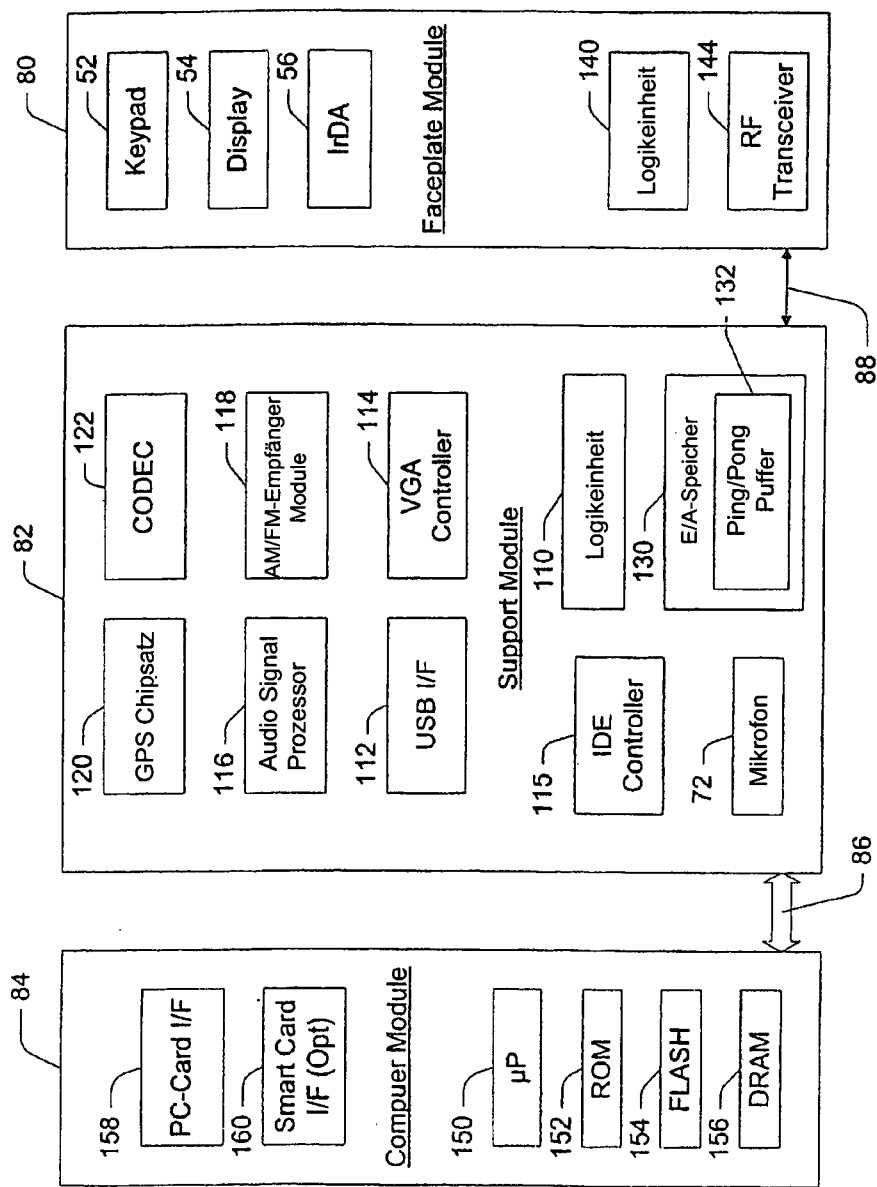
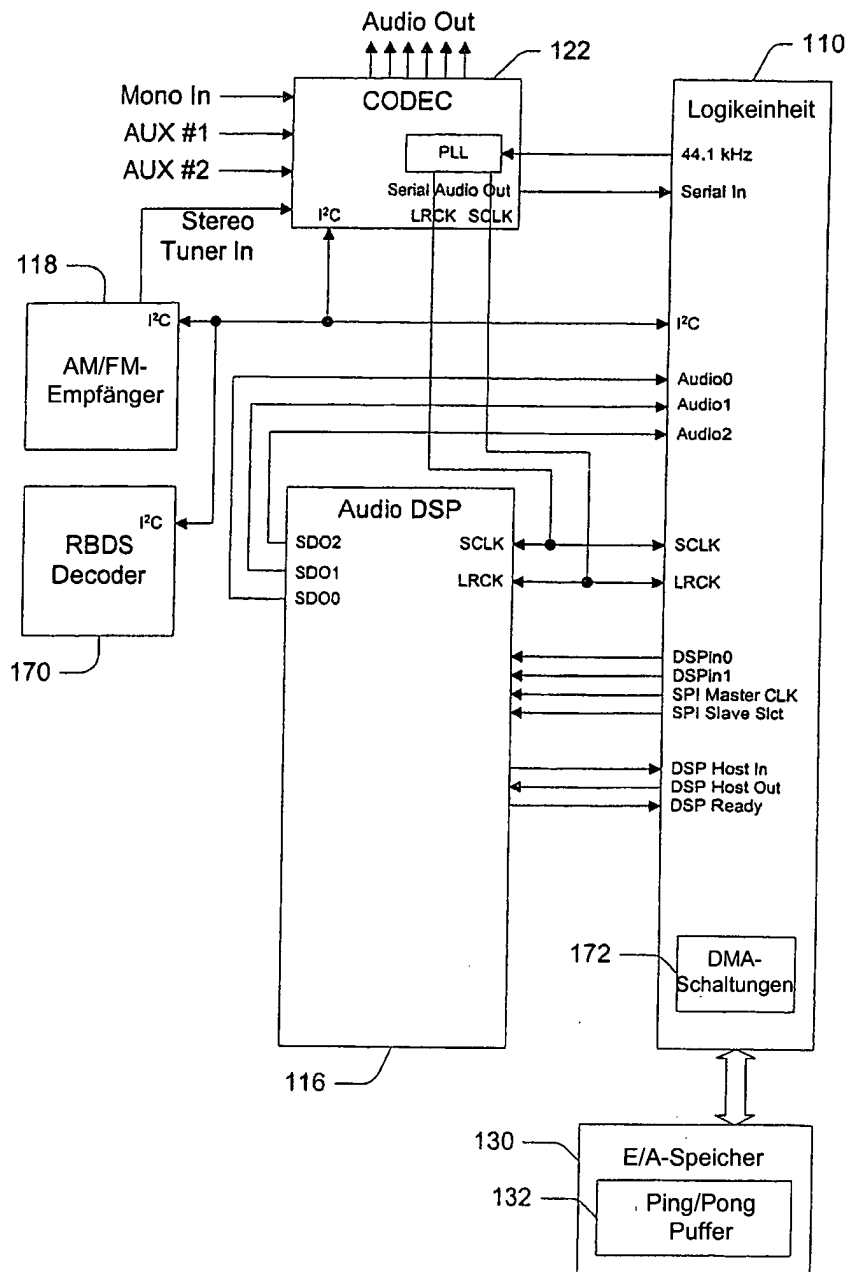
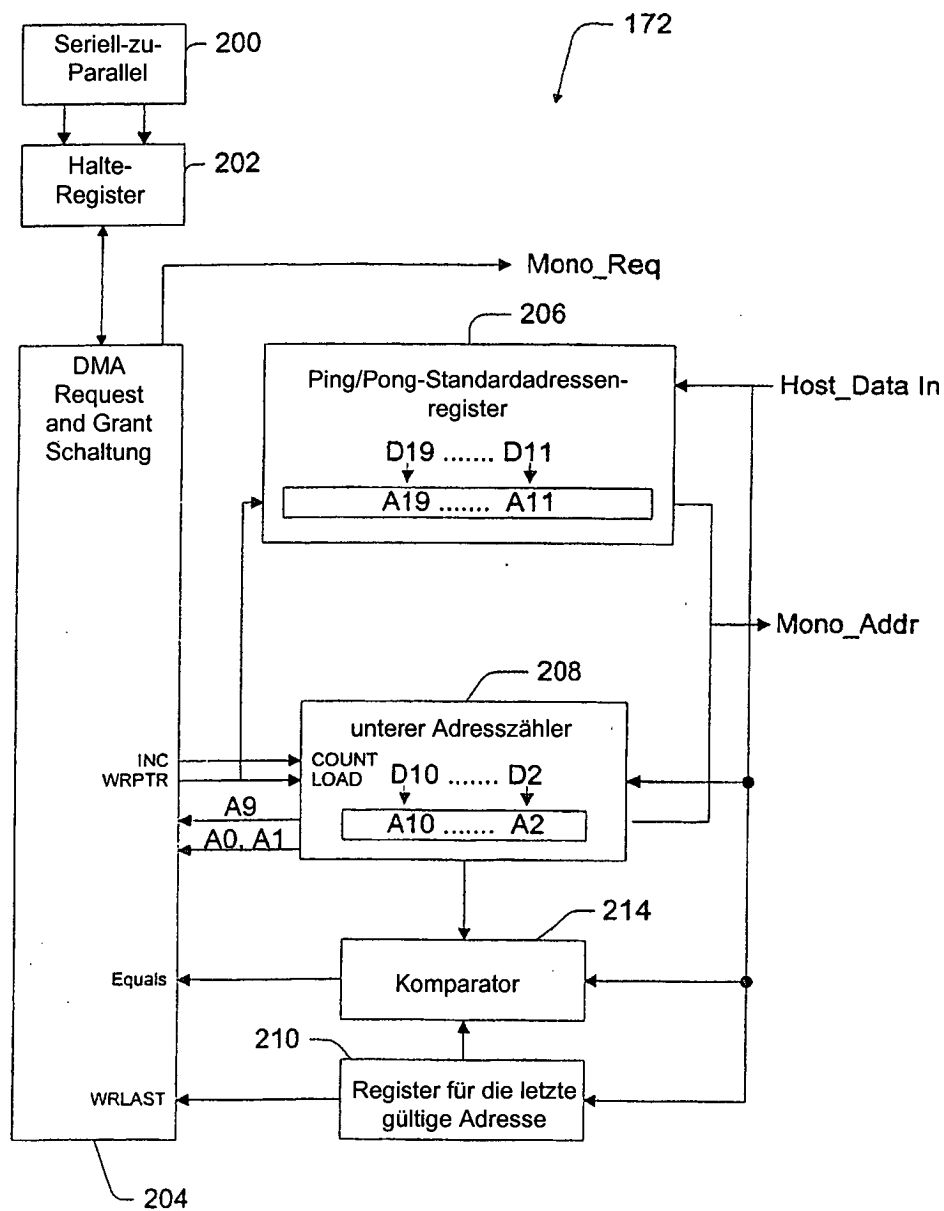


Fig. 3

**Fig. 4**

**Fig. 5**

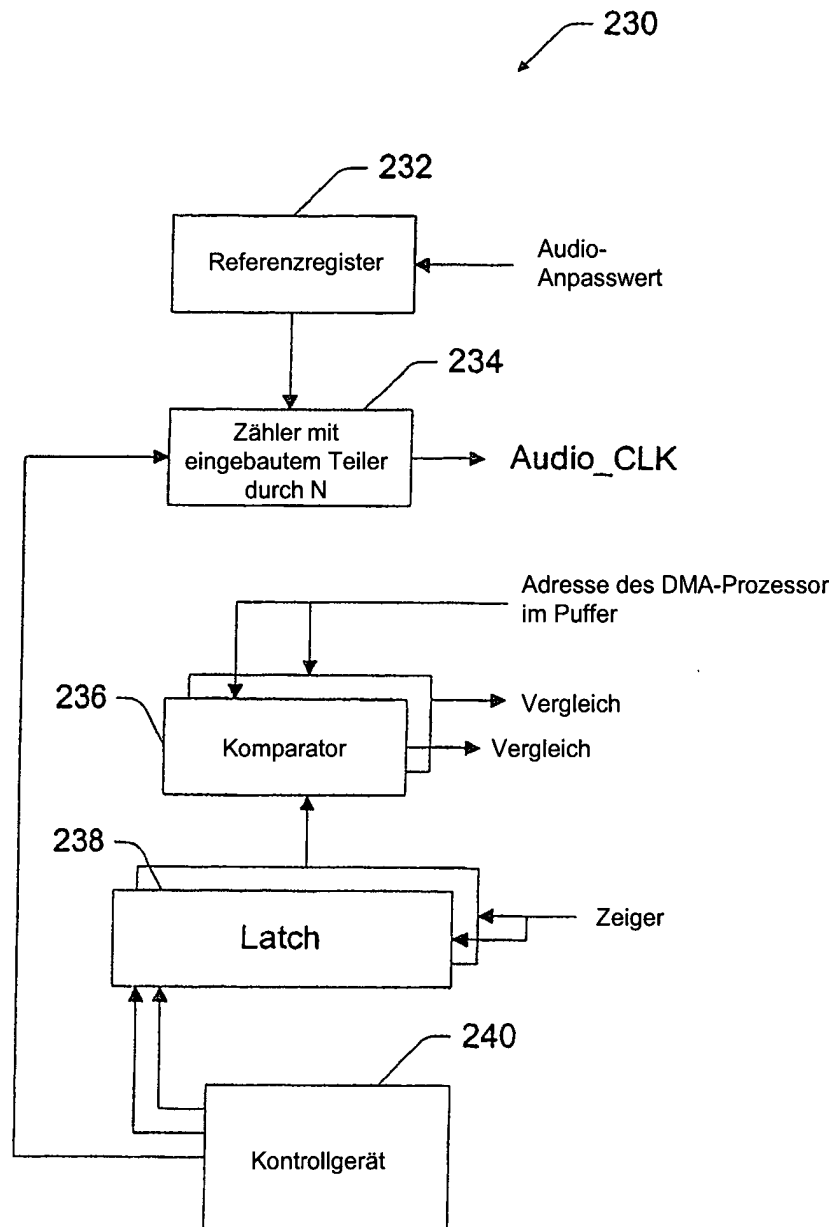
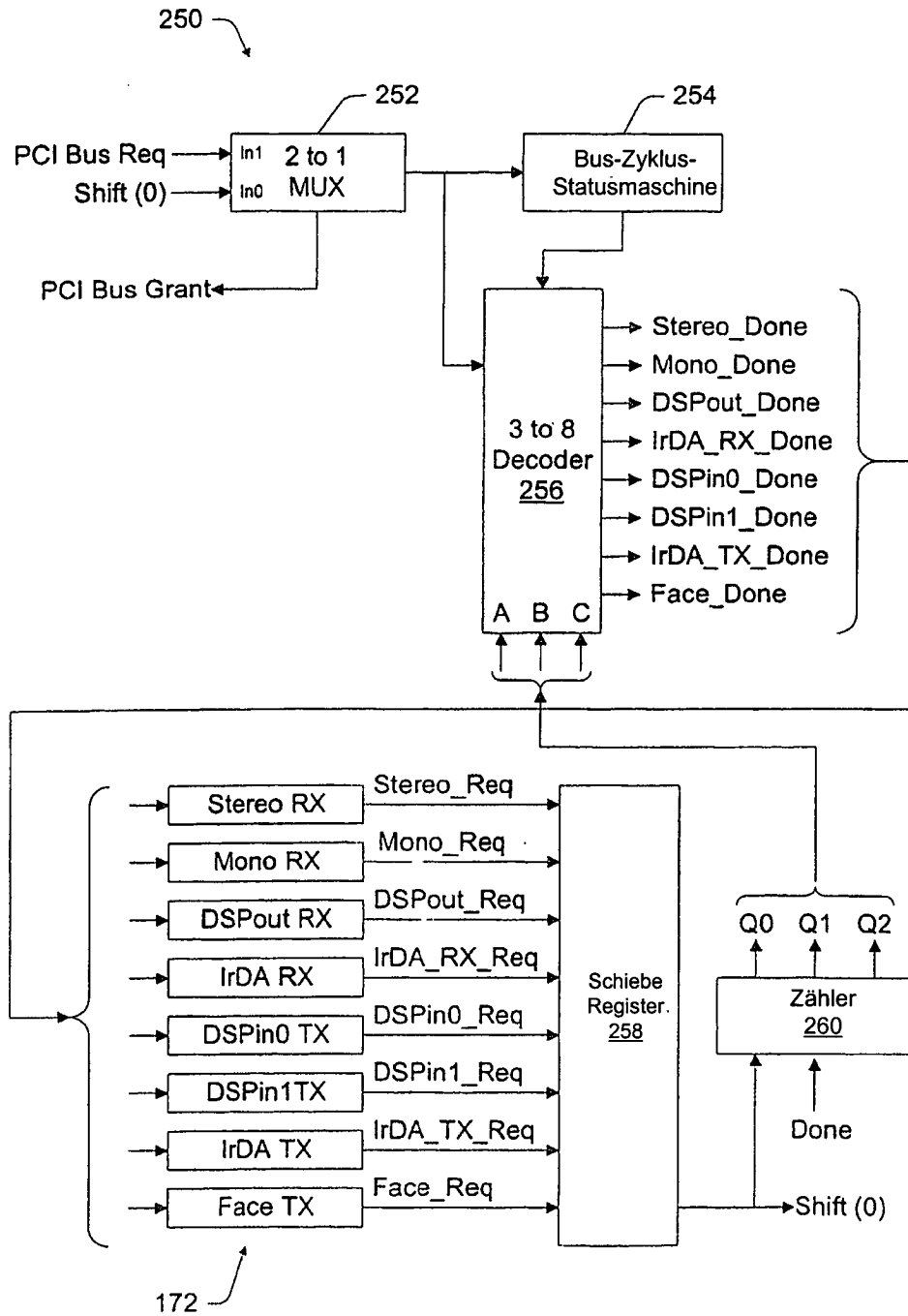


Fig. 6

**Fig. 7**

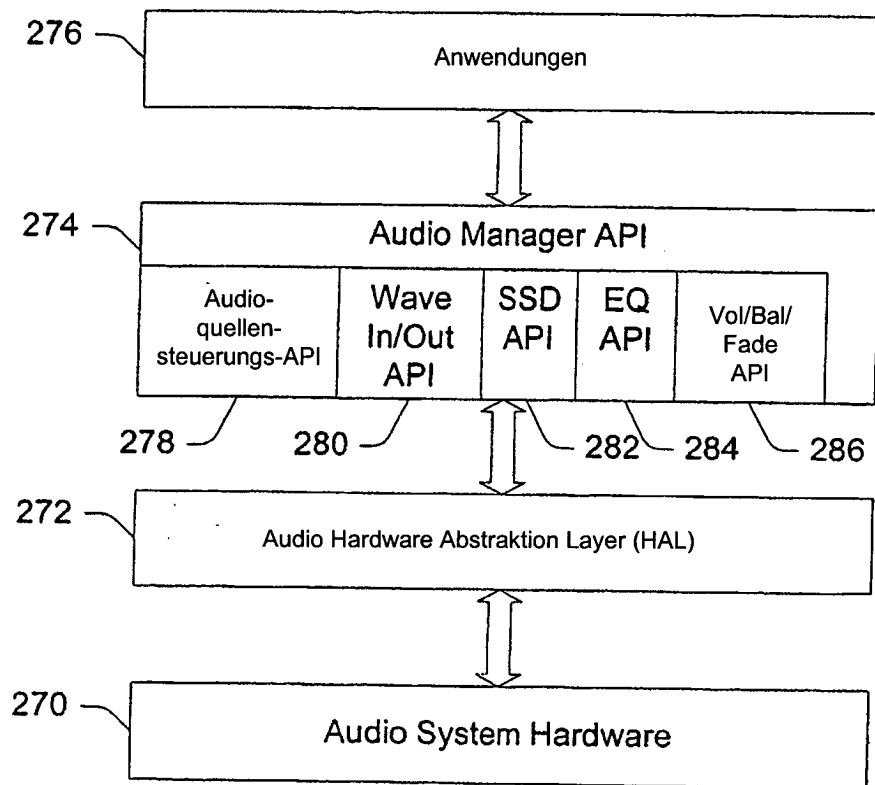


Fig. 8

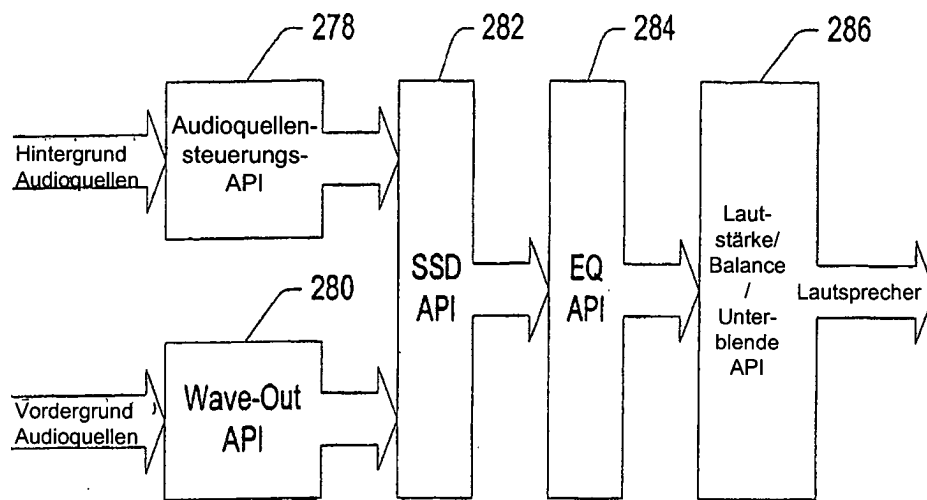
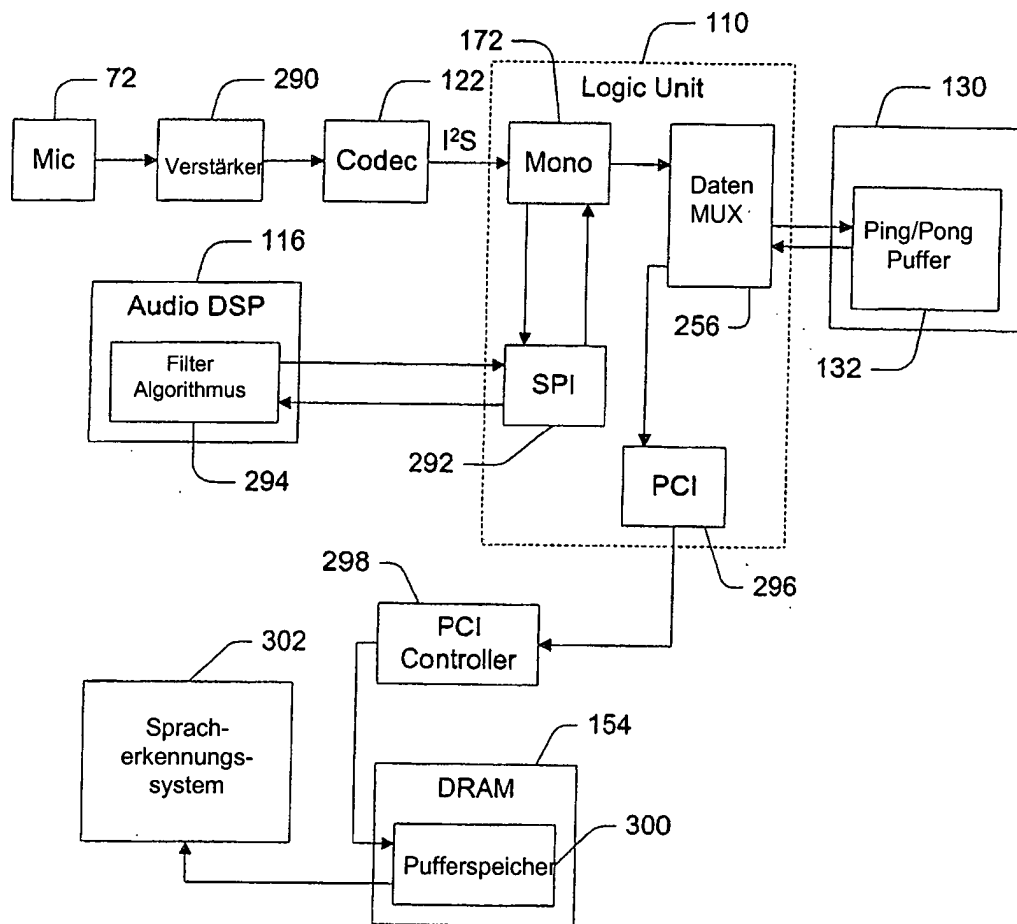


Fig. 9

**Fig. 10**