



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) DE 698 30 661 T2 2006.05.04

(12)

Übersetzung der europäischen Patentschrift

(97) EP 1 013 089 B1

(21) Deutsches Aktenzeichen: 698 30 661.9

(86) PCT-Aktenzeichen: PCT/US98/21069

(96) Europäisches Aktenzeichen: 98 953 277.5

(87) PCT-Veröffentlichungs-Nr.: WO 99/018727

(86) PCT-Anmeldetag: 06.10.1998

(87) Veröffentlichungstag

der PCT-Anmeldung: 15.04.1999

(97) Erstveröffentlichung durch das EPA: 28.06.2000

(97) Veröffentlichungstag

der Patenterteilung beim EPA: 22.06.2005

(47) Veröffentlichungstag im Patentblatt: 04.05.2006

(51) Int Cl.⁸: H04N 7/01 (2006.01)

H04N 11/20 (2006.01)

H04N 5/64 (2006.01)

H04N 5/44 (2006.01)

(30) Unionspriorität:

60974 P	06.10.1997	US
93815 P	23.07.1998	US
94390 P	28.07.1998	US
95164 P	03.08.1998	US
96144 P	11.08.1998	US
100401 P	15.09.1998	US
102946 P	02.10.1998	US
166606	05.10.1998	US

(74) Vertreter:

Weickmann & Weickmann, 81679 München

(84) Benannte Vertragsstaaten:

DE, NL

(72) Erfinder:

ADAMS, R., Dale, San Jose, US; THOMPSON, A.,
Laurence, Saratoga, US; BANKS, D., Jano,
Cupertino, US; BUUCK, C., David, Santa Clara, US;
CHEE, Hwee, Cheng, Saratoga, US

(73) Patentinhaber:

Silicon Image, Inc., Sunnyvale, Calif., US

(54) Bezeichnung: DIGITALES VIDEOSYSTEM UND VERFAHREN ZUR VERFUEGUNGSTELLUNG DESSELBEN

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 eingeleitet, 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

Hintergrund der Erfindung

1. Gebiet der Erfindung

[0001] Die vorliegende Erfindung betrifft allgemein die Technologie von tragbaren Video-Spielern und die Verarbeitung von Videobildern und insbesondere Techniken für das De-Interlacing und Verbessern von Videobildern.

2. Beschreibung des Standes der Technik

[0002] Wegen der Vorteile, die digitales Video zu bieten hat, kam es zu einer raschen Entwicklung von der analogen Video-Technologie hin zur digitalen Video-Technologie. Digitales Video kann mit geringeren Kosten als analoges Video gespeichert und vertrieben werden, da digitales Video auf Direktzugriffs-Medien wie beispielsweise Magnetplattenlaufwerke (Festplatten) und Laserplatten-Medien, auch als Compact Disc (CD) bekannt, gespeichert werden kann. Sobald die Speicherung auf einem Medium mit direktem Zugriff erfolgt ist, kann digitales Video interaktiv werden und daher in Spielen, Katalogen, beim Training, im Unterricht und anderweitig Anwendung finden.

[0003] Eines der neuesten Produkte, das auf digitaler Video-Technologie basiert, ist die digitale Videoplatte, manchmal "Digital Versatile Disc" oder einfach "DVD" genannt. Diese Platten haben die Größe einer Audio-CD und können dennoch bis zu 17 Billionen Datenbytes speichern, d.h. 26 mal die Daten einer Audio-CD. Die Speicherkapazität von DVDs (17 Gbyte) ist sehr viel höher als die einer CD-ROM (600 Mbyte), und eine DVD kann die Daten mit einer höheren Rate liefern als eine CD-ROM. Aus diesem Grund stellt die DVD-Technologie gegenüber traditionellen Systemen wie Fernseher, VCR und CD-ROM eine enorme Verbesserung der Bild- und Tonqualität dar.

[0004] Jedoch ist ein wesentliches Problem, das sich bei der Nutzung von DVD und anderen digitalen Video-Technologien für die Anzeige vom Bewegt-Bildern stellt, dass die Bewegt-Bilder in unterschiedlichen Frame-Geschwindigkeiten vorliegen. So wird ein Standardfilm zum Beispiel mit einer Rate von 24 Hz aufgenommen, während eine nach der US-Fernsehnorm National Television System Committee (NTSC) ausgestrahlte Sendung die Bewegung mit 60 Hz fortschreibt. Die Konvertierung eines Films in digitales Video durch ein Verfahren, das De-Interlacing genannt wird, produziert häufig eine deutlich wahrnehmbare Reduzierung der Auflösung sowie Verzerrungen, die als Bewegungs-Artefakte bekannt sind.

[0005] Ein weiteres Problem bei der Nutzung der digitalen Video-Technologie sind die unterschiedlichen Formate der Filme. Beispielsweise sind Kinofilme für eine Breitwand formatiert, während Video-Anzeigen häufig viel schmäler sind. Eine Videobildskalierung konvertiert ein digitales oder digitalisiertes Bild von einem Format in ein anderes. So kann es beispielsweise notwendig sein, ein digitales Bild mit einer räumlichen Auflösung von 720 horizontalen mal 480 vertikalen Pixel in eine andere Auflösung zu konvertieren, damit das Bild auf einem bestimmten Anzeigegerät, zum Beispiel auf einer Flüssigkristallanzeige mit einer festen Auflösung von 640×480 , angezeigt werden kann. Jedoch liefern die meisten Videoskalierverfahren nach dem Stand der Technik eine schlechte Auflösung in dem konvertierten Bild. Skalierverfahren, die hinsichtlich der Auflösung Qualität bieten, sind extrem teuer in der Implementierung.

[0006] Ein noch weiteres Problem bei der Videoskalierung ist der gegebenenfalls notwendige Datentransfer zwischen zwei asynchronen und/oder unterschiedlichen Datenratenbereichen. Nimmt man das im vorhergehenden Absatz genannte Beispiel, so erfordert eine Skalierung von 720 horizontalen Pixel auf 640 Pixel (Verhältnis 9:8) eine Schnittstelle zwischen einem 54 Mhz-Bereich und einem 48 Mhz-Bereich (Verhältnis 9:8). Im Stand der Technik wurden für die Übertragung von Daten zwischen zwei asynchronen und/oder unterschiedlichen Datenratenbereichen große, teure Puffer verwendet.

[0007] Ein weiterer Grund für die schwierig zu nutzende digitale Video-Technologie ist, dass DVD-Spieler normalerweise große, unhandliche und daher schwer zu transportierende Geräte sind. Hinzu kommt, dass DVD-Spieler für ihren Betrieb an einem Video-Anzeigegerät wie zum Beispiel einem Fernsehgerät oder TV-Bildschirm angeschlossen sein müssen, wodurch es praktisch unmöglich ist, solche Geräte unter eingeschränkten Platzverhältnissen zu verwenden. Mobile Videosysteme nach dem Stand der Technik sind ferner mit vielen unangenehmen Problemen behaftet. Tragbare und mobile Fernsehgeräte haben die typischen Empfangskrankheiten wie Frequenzüberlagerungen und eine schwache oder sprunghafte Signalstärke in ländli-

chen Gegenden. Integrierte TV/VCR-Kombi-Einheiten begegnen diesem Problem, indem sie für die Möglichkeit einer Band-Wiedergabe sorgen, doch sind die Abmessungen dieser Geräte tendenziell groß, und die Geräte sind nicht wirklich tragbar.

[0008] Angesichts der vorstehenden Ausführungen besteht der Wunsch nach einem tragbaren Video-Spieler, der leicht transportiert werden und unter vielfältigen Bedingungen benutzt werden kann. Ferner sind ein Verfahren und eine Vorrichtung für ein De-Interlacing und für eine Umformatierung wünschenswert, dergestalt, dass bei einer Eliminierung von Bewegungs-Artefakten die volle Auflösung eines Bildes erhalten bleibt. Weiterhin sind ein Verfahren und eine Vorrichtung wünschenswert, das/die für eine Videoskalierung und vertikale Verarbeitung mit hoher Qualität sorgt. Schließlich werden ein Verfahren und eine Vorrichtung gewünscht, das/die bei minimalen Kosten eine asynchrone Datenschnittstelle vorsieht, wobei sich große und teuere Puffer gleichzeitig erübrigen.

[0009] Die Dokumente US 5,661,525 und US 4,937,667 beschreiben das De-Interlacing einer interlaced Videobildsequenz unter Verwendung von Interpolationsschätzungen wie beispielsweise räumlichen und zeitlichen Interpolationen.

Übersicht der Erfindung

[0010] Die vorliegende Erfindung sorgt für einen digitalen Bildverbesserer gemäß Anspruch 1 und für ein Verfahren zur Verarbeitung von digitalem Video gemäß Anspruch 6.

[0011] Es versteht sich, dass die vorliegende Erfindung auf viele Weisen implementiert sein kann, unter anderem als Prozess, als Vorrichtung, als System, als Gerät oder als Verfahren. Verschiedene Ausführungsformen der vorliegenden Erfindung werden nachstehend beschrieben.

[0012] In einer Ausführungsform der vorliegenden Erfindung ist ein digitaler Bildverbesserer offenbart. Der digitale Bildverbesserer umfasst einen De-Interlace-Prozessor, der für einen interlaced Videostrom empfänglich ist. Der De-Interlace-Prozessor umfasst einen ersten De-Interlacer und einen zweiten De-Interlacer und liefert einen de-interlaced Videostrom. Der digitale Bildverbesserer umfasst auch einen Videoausgabeprozessor, der für die Ausgabe des de-interlaced Videostroms empfänglich ist, für die Bereitstellung eines skalierten de-interlaced Videostroms.

[0013] In einer weiteren Ausführungsform der vorliegenden Erfindung ist ein digitaler Bildverbesserer offenbart, der einen De-Interlace-Prozessor umfasst, der für einen interlaced Videostrom empfänglich ist und der wirksam ist, für einen de-interlaced Videostrom zu sorgen. Der digitale Bildverbesserer umfasst auch einen Videoausgabeprozessor, der für die Ausgabe des De-Interlace-Prozessors empfänglich ist. Der De-Interlace-Prozessor verarbeitet den interlaced Videostrom in vertikalen Slices, um einen skalierten de-interlaced Videostrom bereitzustellen.

[0014] In einer noch weiteren Ausführungsform der vorliegenden Erfindung ist ein tragbarer DVD-Spieler offenbart. Der tragbare DVD-Spieler umfasst ein allgemein dünnes prismatisches Gehäuse mit einer ersten großen Oberfläche, einer von der ersten großen Oberfläche getrennten zweiten großen Oberfläche und mit Seitenflächen, die die erste große Oberfläche mit der zweiten großen Oberfläche verbinden. Zumindest ein Abschnitt der ersten großen Oberfläche weist ein Video-Display auf, und das Gehäuse hat eine DVD-Einschuböffnung, so dass eine DVD in das Gehäuse eingelegt werden kann.

[0015] Der tragbare DVD-Spieler umfasst auch ein digitales Verarbeitungssystem mit einem Dekodierer, einer Bildverbesserungs-Engine und einer Display-Steuereinheit. Der Dekodierer empfängt Signale von einer in das Gehäuse eingelegten DVD, um ein decodiertes interlaced Videosignal bereitzustellen. Die Bildverbesserungs-Engine konvertiert das interlaced Videosignal in ein de-interlaced Videosignal. Die Display-Steuereinheit nutzt das de-interlaced Videosignal, um ein progressiv abgetastetes Video auf dem Video-Display bereitzustellen.

[0016] In einer noch weiteren Ausführungsform der vorliegenden Erfindung ist ein Verfahren für die Verarbeitung von digitalem Video offenbart. Das Verfahren umfasst das De-Interlacing eines interlaced Videostroms durch zumindest eines von einer Anzahl von De-Interlace-Verfahren, um einen de-interlaced Videostrom zu liefern. Das Verfahren umfasst auch das Skalieren des de-interlaced Videostroms.

[0017] Weitere Aspekte und Vorteile der vorliegenden Erfindung ergeben sich aus der nachfolgenden Detail-

beschreibung im Zusammenhang mit den anliegenden Zeichnungen, in denen die Grundgedanken der Erfindung anhand von Beispielen dargestellt sind.

Kurze Beschreibung der Zeichnungen

[0018] Die vorliegende Erfindung ist durch die folgende Detailbeschreibung in Verbindung mit den anliegenden Zeichnungen ohne weiteres verständlich. Zur Vereinfachung der Beschreibung sind gleiche Bauteile mit gleichen Bezugszeichen versehen.

[0019] [Fig. 1A](#) und [Fig. 1B](#) zeigen einen tragbaren DVD-Spieler gemäß einer Ausführungsform der vorliegenden Erfindung.

[0020] [Fig. 2A](#), [Fig. 2B](#) und [Fig. 2C](#) zeigen mehrere unterschiedliche Anwendungen für den DVD-Spieler gemäß einer Ausführungsform der vorliegenden Erfindung.

[0021] [Fig. 2D](#) zeigt eine Andockstation und einen damit verbundenen Video-Monitor für den DVD-Spieler gemäß einer Ausführungsform der vorliegenden Erfindung.

[0022] [Fig. 2E](#) zeigt einen "DeskTablet®"-DVD-Spieler gemäß einer Ausführungsform der vorliegenden Erfindung.

[0023] [Fig. 3](#) ist ein Blockdiagramm des elektronischen Schaltkreises des DVD-Spielers gemäß einer Ausführungsform der vorliegenden Erfindung.

[0024] [Fig. 4](#) zeigt ein Systemdiagramm der Bildverbesserungs-Engine (IE²) gemäß einer Ausführungsform der vorliegenden Erfindung.

[0025] [Fig. 5](#) zeigt ein Verfahren zum Kombinieren von Fields zu Frames in einem De-Interlace-Verfahren gemäß einer Ausführungsform der vorliegenden Erfindung.

[0026] [Fig. 6](#) ist ein Blockdiagramm eines Video-De-Interlacers gemäß einer Ausführungsform der vorliegenden Erfindung.

[0027] [Fig. 7](#) ist ein Systemdiagramm eines progressiven Frame-Detektors gemäß einer Ausführungsform der vorliegenden Erfindung.

[0028] [Fig. 8](#) ist ein Flussdiagramm der Verarbeitungsschritte in einem Field-Differenzierungsmodul gemäß einer Ausführungsform der vorliegenden Erfindung.

[0029] [Fig. 9](#) stellt Details des Frequenz-Detektionsmoduls gemäß einer Ausführungsform der vorliegenden Erfindung dar.

[0030] [Fig. 10](#) ist ein Systemdiagramm des PFPD-Moduls gemäß einer Ausführungsform der vorliegenden Erfindung.

[0031] [Fig. 11](#) ist eine Darstellung eines De-Interlace-Prozesses gemäß einer Ausführungsform der vorliegenden Erfindung.

[0032] [Fig. 12](#) zeigt ein zweidimensionales Array von Pixelwerten, das zur Beschreibung der vorliegenden Erfindung verwendet wird.

[0033] [Fig. 13](#) ist ein Diagramm zur Darstellung eines Verfahrens zur Gewinnung eines Ausgabe-Pixel aus dem zweidimensionalen Array von [Fig. 12](#) gemäß einer Ausführungsform der vorliegenden Erfindung.

[0034] [Fig. 14A](#) ist eine für die Beschreibung des Verfahrens gemäß einer Ausführungsform der vorliegenden Erfindung verwendete Illustration.

[0035] [Fig. 14B](#) ist ein Graph einer Gruppe von Samples von der Abtastzeile von [Fig. 14A](#).

[0036] [Fig. 14C](#) ist ein Graph einer abgetasteten Cosinuswelle.

[0037] [Fig. 15](#) ist eine Darstellung, die verwendet wird für die Beschreibung eines Schwellwertverfahrens in Anwendung auf einen Detektionswert gemäß einer Ausführungsform der vorliegenden Erfindung.

[0038] [Fig. 16](#) ist ein Blockdiagramm einer Mischschaltung gemäß einer Ausführungsform der vorliegenden Erfindung.

[0039] [Fig. 17](#) ist ein Diagramm eines exemplarischen Betriebs der Mischschaltung gemäß einer Ausführungsform der vorliegenden Erfindung, wenn der DV größer als "0", jedoch kleiner als "1" ist.

[0040] [Fig. 18](#) ist eine Darstellung eines Verfahrens für die Detektion von diagonalen Merkmalen gemäß einer bevorzugten Ausführungsform der vorliegenden Erfindung.

[0041] [Fig. 19](#) ist ein Blockdiagramm einer Diagonal-Mischschaltung gemäß einer Ausführungsform der vorliegenden Erfindung.

[0042] [Fig. 20](#) ist ein Diagramm, das die Pixel eines zweiten Arrays zeigt, das für die Berechnung der Ausgabe der Diagonal-Mischschaltung von [Fig. 16](#) verwendet wird.

[0043] [Fig. 21](#) ist ein Flussdiagramm eines Diagonal-Detektionsverfahrens gemäß einer Ausführungsform der vorliegenden Erfindung.

[0044] [Fig. 22](#) ist ein Beispiel eines FIR-Filters mit variabler Skalierung gemäß einer Ausführungsform der vorliegenden Erfindung.

[0045] [Fig. 23](#) ist ein Graph von Tiefpass-Filterkoeffizienten in dem Zeitbereich.

[0046] [Fig. 24](#) ist eine Tabelle von Koeffizienten, die in L Gruppen von "mults" pro Gruppe organisiert ist.

[0047] [Fig. 25](#) ist ein Flussdiagramm eines Verfahrens für eine Quantisierung gemäß einer Ausführungsform der vorliegenden Erfindung.

[0048] [Fig. 26](#) ist ein Flussdiagramm eines Verfahrens zum Ändern von Koeffizienten gemäß einer Ausführungsform der vorliegenden Erfindung.

[0049] [Fig. 27](#) stellt einen Video-Frame gemäß einer Ausführungsform der vorliegenden Erfindung dar, welcher in eine Anzahl von vertikalen Slices unterteilt ist, für eine Slice-Abtastsequenz, die durch eine entsprechende Anzahl von Abtastzeilen veranschaulicht ist.

[0050] [Fig. 28](#) zeigt ein Beispiel eines initialen Slice-Kerns, bei dem ein Problem mit nicht verfügbaren Daten an seinem linken und rechten Rand besteht.

[0051] [Fig. 29](#) zeigt einen Slice, bei dem entlang des linken und rechten Rands des initialen Slice-Kerns Wings angefügt sind.

[0052] [Fig. 30](#) zeigt eine Gesamtstruktur von überlappenden Slice/Wing-Kombinationen.

[0053] [Fig. 31](#) ist ein Flussdiagramm zur Darstellung eines Verfahrens zur Verarbeitung von Video gemäß einer Ausführungsform der vorliegenden Erfindung.

[0054] [Fig. 32](#) zeigt ein Systemdiagramm für einen Videoprozessor auf Slice-Basis gemäß einer Ausführungsform der vorliegenden Erfindung.

[0055] [Fig. 33](#) zeigt ein Systemdiagramm einer Videoverarbeitungs-Chiparchitektur gemäß einer Ausführungsform der vorliegenden Erfindung.

[0056] [Fig. 34](#) ist ein Diagramm einer asynchronen Schnittstelle gemäß einer Ausführungsform der vorliegenden Erfindung.

[0057] [Fig. 35](#) ist ein Diagramm einer alternativen asynchronen Schnittstelle gemäß einer Ausführungsform der vorliegenden Erfindung.

[0058] [Fig. 36](#) ist ein Diagramm einer 3-Puffer-Synchronisierer-Sequenz zur Darstellung der Sequenzierung und der relativen Beziehungen von Schreib- und Lesevorgängen zu drei RAM-Puffern gemäß einer Ausführungsform der vorliegenden Erfindung.

[0059] [Fig. 37](#) ist ein Flussdiagramm eines Verfahrens für eine Sequenzierung durch RAM-Adressen und Module gemäß einer Ausführungsform der vorliegenden Erfindung.

[0060] [Fig. 38](#) ist ein Diagramm eines Synchronisierers der vorliegenden Erfindung, der für die Nutzung bei einer Video-Skalieranwendung gemäß einer Ausführungsform der vorliegenden Erfindung konfiguriert ist.

Detailbeschreibung der bevorzugten Ausführungsformen

[0061] Die nachstehende Beschreibung enthält zahlreiche spezielle Details, die dazu beitragen, dass die vorliegende Erfindung von Grund auf verstanden wird. Der Fachmann wird jedoch erkennen, dass die praktische Ausführung der Erfindung auch ohne diese Details oder ohne manche dieser Details möglich ist. Andererseits wurde aus Gründen der Übersichtlichkeit der Erfindung auf die detaillierte Beschreibung von hinlänglich bekannten Prozess-Operationen verzichtet.

[0062] Die [Fig. 1A](#) und [Fig. 1B](#) zeigen einen tragbaren DVD-Spieler **10** gemäß einer Ausführungsform der vorliegenden Erfindung. Der DVD-Spieler **10** umfasst ein Gehäuse **12**, das als Umhüllung oder Chassis für die Komponenten des DVD-Spielers **10** dient. An der Oberfläche des Gehäuses **12** befinden sich ein Bildschirm **14** für die Video-Betrachtung und Bedienknöpfe **16** zum Bedienen des DVD-Spielers **10**. Netzstrom- und Signal-Schnittstellen **18** befinden sich an einer der beiden Seitenflächen des Gehäuses **12**, während eine Infrarot-(IR)-Schnittstelle **20** und eine Medien-Schublade **22** an der anderen Seitenfläche vorgesehen sind. Eine DVD **24** kann in eine geeignet konfigurierte Vertiefung in der Medien-Schublade **22** eingelegt werden, und die Medien-Schublade **22** bewegt sich zurück in das Gehäuse **12**, um das Abspielen der DVD **24** zu ermöglichen.

[0063] [Fig. 2A](#) ist eine Darstellung der Benutzung in einem Fahrzeug, doch lässt sich die Vorstellung auf die meisten Arten von Transportmitteln ausdehnen. Die vorliegende Erfindung ist ideal geeignet für den Reisenden, der viele Stunden passiv in Verkehrsmitteln wie Flugzeug, Zug oder U-Bahn verbringt, wie das in [Fig. 2B](#) dargestellt ist. Darüber hinaus sind auch viele kommerzielle Anwendungen möglich, zum Beispiel Video-Werbung in Taxis oder an anderen Plätzen, an denen sich Publikum notgedrungen aufhält. Ebenso lässt sich die vorliegende Erfindung an dem Magazinhalter eines jeden Stairmaster®-Fitnessgeräts und Heimtrainers in einem Fitness-Center anbringen, wie das in [Fig. 2C](#) dargestellt ist.

[0064] Die vorliegende Erfindung ist überall dort einsetzbar, wo eine portable Video-Abspielung benötigt wird. Die vorliegende Erfindung ist die ideale mobile Lösung für sämtliche Orte, an denen eine großartige Ton-, DVD- oder CD Wiedergabe oder Fernsehen gewünscht wird. Sehr gute Beispiele für die Nutzungsvielfalt der vorliegenden Erfindung sind: Unterhaltung in Flugzeugen, Filme, Videos, individuelles Lernen und Studieren (jede Disk kann Tausende von Büchern speichern) und an Ort und Stelle einsetzbare Do-it-yourself-Videos beispielsweise für Autoreparatur, Haus und Garten und Kochen.

[0065] Ferner kann die vorliegende Erfindung als tragbares Videoanzeigegerät für Videospiele, Camcorder-Wiedergabe oder Bildbetrachtung bei einer Digital-Einzelbildkamera und für OEM-Anwendungen wie Video-Kiosks, kommerzielle Fluglinien und Fitness Center Verwendung finden. Der tragbare DVD-Spieler **10** kann auch für die Präsentation von Verkaufsinhalten verwendet werden: Videos in Bussen, Aufzügen, Taxis oder eine Führung durch Immobilien etc. Er lässt sich sogar als Fernseh-Flachbildschirm oder als Anzeigevorrichtung für Kunst- und Architektur-Disks, konstruktive Kunst etc. an der Wand aufhängen.

[0066] [Fig. 2D](#) zeigt eine "Heimkino-Andockstation", die eine Verbindung mit nur einem Handgriff vorsieht sowie die Möglichkeit des Betriebs als einfacher DVD-Spieler in Verbindung mit einem normalen Fernsehgerät. Das Dock sieht eine verkabelte elektrische Schnittstelle zu einem Fernsehgerät oder anderen Heimkinolelementen – zum Beispiel zu einem Audiosystem – vor, die als permanente Einrichtung am Ort verbleibt, wenn die vorliegende Erfindung in einem externen System verwendet wird. Das Dock hat vorzugsweise die gleiche Form wie ein typischer VCR und ist so gestaltet, dass es sich in die restlichen Systemkomponenten, die der Benutzer besitzt, einfügt und dass sofort erkennbar ist, für welche Funktion es vorgesehen ist.

[0067] Der Andockstation ist vorzugsweise eine Infrarot-Fernbedienung beigelegt. Die Fernbedienung ist serienmäßig mit einem umfangreichen Bedienfeld ausgestattet, das optimiert ist für den Zugriff auf sämtliche Erweiterungsfunktionen, die mit dem DVD-Format verfügbar sind. Zentraler Punkt bei der Gestaltung dieser

Fernbedienung ist ein äußerst attraktives Industriedesign, das die sonst übliche unübersichtliche und kompliziert zu bedienende "Flut von Knöpfen", wie man sie bei Fernbedienungen von Verbraucherprodukten häufig findet, sorgfältig vermeidet.

[0068] [Fig. 2E](#) zeigt eine Ausführungsform der vorliegenden Erfindung für den Desktop-Markt, die im folgenden als "DeskTablet®"-DVD-Spieler bezeichnet wird. Der DeskTablet findet zum Beispiel Anwendung im Schlafzimmer, im Arbeitszimmer oder in der Küche, d.h. an Orten, an denen die Einheit sozusagen "fest" aufgestellt werden muss. Dieses Produkt ist im wesentlichen in einer Marketingkategorie mit konventionellen TV/VCR-Kombinationseinheiten. In der Form ähnlich wie "Monorail"-Personal Computer kann der schlanke DeskTablet freistehend oder an der Wand hängend betrieben werden. Befreit von vielen portabilitätsbedingten Einschränkungen, die von der mobilen Maschine der vorliegenden Erfindung gefordert werden, hat der DeskTablet ein integriertes Lautsprechersystem hoher Qualität.

[0069] [Fig. 3](#) ist ein Blockdiagramm des Antriebsmoduls **26** des DVD-Spielers **10** der [Fig. 1](#) und [2](#). Das Antriebsmodul **26** enthält die Medien-Schublade **22**, die die DVD liest. Die Videodaten von der DVD werden dann zu einem MPEG/Dolby Digital (oder "MPEG/AC-3")-Dekodierer **28** übertragen. Nach der Dekodierung de-interlaced eine Bildverbesserungs-, d.h. Image Enhancement Engine® (IE²) **30** das digitale Video, um ein progressiv abgetastetes Videosignal zu liefern. Das Video wird schließlich durch ein Display **36** angezeigt.

[0070] Neben anderen dargestellten Komponenten, Bussen und Subsystemen umfasst das DVD-Antriebsmodul **26** auch eine Audio/Infrarot-Verbindung **32**, eine System-Steuereinheit **34**, ein Batteriepaket **38**, ein Netzteil **40**, einen Video-Datenpuffer **42** und eine Benutzer-Schnittstelle. Die Komponenten des DVD-Antriebsmoduls **26** befinden sich hauptsächlich in dem Gehäuse **12** und werden nachstehend jeweils im Detail erläutert.

[0071] Die DVD-Schublade **22** arbeitet vorzugsweise mit einem handelsüblichen Antriebsmodul **26**, das für tragbare Computer entwickelt ist. Das Antriebsmodul **26** ist vorzugsweise siebzehn oder weniger als siebzehn Millimeter dick und ermöglicht daher eine sehr schlanke Bauform des Systems. Ein geeignetes DVD-Antriebsmodul **26** ist bei Toshiba America Information Systems, Inc., Tokyo, Japan erhältlich. Das Antriebsmodul **26** der vorliegenden Erfindung ist vorzugsweise physisch und elektronisch gegen mechanische Stöße und Schläge isoliert. Insbesondere ist das Antriebsmodul **26** in dem Gehäuse **12** fest gegen mechanische Stöße montiert, und Daten aus dem Antriebsmodul **26** werden vor der Dekodierung eine Weile zwischengespeichert, um ein nochmaliges Lesen von Sektoren zu ermöglichen, falls der Datenstrom bedingt durch einen Stoß bzw. eine Erschütterung unterbrochen wurde.

[0072] Der MPEG/Dolby Digital-Dekodierer **28** ist vorzugsweise ein Ein-Chip-Dekodierer, der für handelsübliche DVD-Spieler gebaut ist. Vorzugsweise führt er eine MPEG-1 und MPEG-2-Dekodierung, Dolby Digital ("AC-3"), MPEG und LPCM Audio-Dekodierung durch, hat eine A/V-Synchronisation und verfügt über die Charakteristik, dass für die Datenspeicherung und Pufferung nur eine einzige Speichereinrichtung notwendig ist. Derartige Dekodierer gibt es von einer Vielzahl von Herstellern, unter anderem von C-Cube Microsystems of Milpitas, Kalifornien.

[0073] Die IE² **30** de-interlaced vorzugsweise digitales Video von einer DVD oder einer beliebigen anderen Quelle, um progressiv abgetastete Videosignale zu liefern, wobei herstellerspezifische Hardware für Bewegungsdetektion und Kompensation enthalten ist. Ferner korrigiert sie vorzugsweise Helligkeit, Kontrast, Gamma und Farbtemperatur für eine klare und genaue Videoanzeige. Um das Bild weiter zu verbessern, wird eine hochwertige Scaling Engine der IE² **30** vorzugsweise für die Konvertierung eines Videobilds von rechteckigen in quadratische Pixel und zum Skalieren des Videos für eine Vollbilddarstellung auf anderen Flat Panel-Größen als 640 × 480 verwendet.

[0074] Die Audi/IR-Verbindung **32** (umfassend einen oder mehrere integrierte Schaltungsschips) dekodiert den Dolby Digital-Datenstrom, der von dem MPEG2/Dolby Digital-Dekodierer **28** bereitgestellt wird, mischt 5.1 Kanäle auf 2 Kanäle für eine konventionelle Stereoausgabe ab und kodiert/verarbeitet 5.1 Kanäle für eine Surround-Kopfhörerausgabe (optionales Modul). Stereo D/A-Konverter sind für die Kopfhörerausgabe vorgesehen.

[0075] D/A-Konverter für einen 5.1 Kanal Dolby-Digitalstrom sind an dem Andockmodul-Verbinder verfügbar. Ein nicht dekodierter Dolby-Digitalstrom für Outboard-Dekodierer steht ebenfalls an dem Andockmodul-Verbinder zur Verfügung. Ein optionaler IR-Sender für schnurlose Kopfhörer kann vorgesehen sein, ebenso Stereolautsprecher mit schwacher Stereoleistungsverstärkung für Darbietungen oder Wiedergabe ohne Kopfhörer.

[0076] Die System-Steuereinheit **34** ist vorzugsweise ein Ein-Chip-Mikroprozessor, der die meisten, wenn nicht alle Systemsteuerfunktionen übernimmt. Zum Beispiel ist der Mikroprozessor vorzugsweise zuständig für den Systemstart und die Konfiguration, für die Benutzer-Schnittstelle und Bedienelemente, für die Auswahl der Funktionsgruppen (z.B. Kindersicherung etc.), für die DVD-Antriebssteuerung und die Konfiguration der IE², des MPEG-Dekodierers, des Audio-Systems und der Display-Steuereinheit. Ein geeigneter Mikroprozessor kann unter der Teilenummer MC68HC16 von Motorola, Inc., Austin, Texas, bezogen werden.

[0077] Das Display **36** ist vorzugsweise ein 11.3-Zoll-LCD Display (wobei Modelle sowohl mit aktiver als auch passiver Matrix erhältlich sind) mit einer leistungsstarken Kaltkathoden-Fluoreszenz-Rückbeleuchtung. Das Display **36** hat vorzugsweise eine Auflösung von 640 × 480 Pixel und eine Farbtiefe von 18 Bit. Ein geeignetes Display ist bei Sharp Electronics Corp., Camas, Washington, erhältlich. Die Video-Steuereinheit für das Display **36** sorgt für eine hohe Auflösung, eine flexible On Screen-Grafik, ein Grafik-Overlay auf einer Vollbild-Video-wiedergabe und für LCD-Treiber für eine direkte Verbindung mit dem Display.

[0078] Das Gehäuse **12** hat vorzugsweise eine "Tablet"-Form (Schreibblock-Form), die einfach zu benutzen und mitzuführen ist. Das einstückige Design des Gehäuses sorgt für Einfachheit, leichte Herstellung, Robustheit, geringeres Gewicht und lange Lebensdauer. Eine optionale Andock-Station erlaubt den Anschluss an externe Videosysteme durch nur eine einzige Verbindung.

[0079] Das Batterie-Paket **38** ist vorzugsweise ein austauschbares, wiederaufladbares Modul auf Basis der NiMH-Technologie für eine hohe Leistungsdichte bei gleichzeitig niedrigen Kosten. Vorzugsweise arbeitet das Batterie-Paket **38** mit handelsüblichen Standard-Batteriezellen, die über eine Kapazität von 40 Wattstunden verfügen und die eine für einen ununterbrochenen Betrieb von mehr als 2,5 Stunden ausreichende Energie liefern. Diese Zeitspanne reicht normalerweise aus, um einen Spielfilm in voller Länge anzusehen.

[0080] Es sind auch Eingänge in und Ausgänge aus der Einheit dargestellt. Während Verbindungen von der System-Steuereinheit **34** zu anderen Komponenten als Bus dargestellt sind, können diese auch als Einzelverbindungen ausgeführt sein, sofern dies für die speziellen handelsüblichen Komponenten, die gewählt wurden, angebracht ist. Die Architektur des DVD-Antriebsmoduls **26** wurde für eine Modularität der Komponenten und für eine einfache Erweiterbarkeit konzipiert.

[0081] Die Systemerweiterung wird über eine Modulschnittstelle mit I/O-Option bewerkstelligt, die eine Interaktion des Antriebsmoduls **26** mit den Video- und Audio-Subsystemen sowie mit der System-Steuereinheit **34** ermöglicht. Diese Schnittstelle enthält Erweiterungsmodule mit Funktionen wie alternative Audio- und/oder Videoeingaben und spezialisierte Audiobearbeitung, wodurch der Benutzer die vorliegende Erfindung für eine Vielfalt von Anwendungen maßschneidern kann.

[0082] Alternative Produkt-Konfigurationen sind ohne weiteres unterzubringen, indem Komponenten des Tastensystems ausgetauscht werden. Da in vielen Bereichen des Systems (z.B. in dem DVD-Antriebsmodul **26**) Standardkomponenten verwendet werden, können diese einfach durch ähnliche Komponenten ersetzt werden. Eine kompaktere Einheit lässt sich zum Beispiel bauen, indem ein kleineres LCD und eine kleinere Batterie verwendet werden. Die Display-Steuereinheit bietet direkt Platz für variierende Display-Größen, während der Leistungswandler und die Batterie-Ladeeinrichtung für verschiedene Batteriegrößen und verschiedene Technologien konfigurierbar sind. Basiskonfigurationen, die mit mehr Funktionen ausgestattet sind, sind möglich, indem für eine zusätzliche Funktionalität einfach zusätzliche (optionale) Komponenten integriert werden. Zum Beispiel erfordert eine Einheit mit einem kompletten Video- und Audio-I/O-Zusatz lediglich die Nachrüstung von weiteren DIA-Konvertern, einem Video-Dekodierer und zusätzlichen Verbindern sowie Gehäusemodifikationen.

[0083] Die System-Software des tragbaren DVD-Spielers der vorliegenden Erfindung hat vorzugsweise eine Struktur mit zwei Hauptebenen: die Software der höheren Ebene für die Benutzer-Schnittstelle und die Software der niederen Ebene für die Gerätesteuerung. Die Software läuft auf der System-Mikrosteuereinheit **34** und ist bevorzugt in einem Lesespeicher (ROM) (nicht dargestellt) gespeichert. Die Sektion der niederen Ebene ist direkt mit den verschiedenen Hardware-Komponenten des Systems zusammengeschaltet und interagiert direkt mit diesen auf einer Registerebene. Sie sorgt für das Anschalten und Initialisieren, für das Konfigurieren der Hardware-Komponenten, für die Basis-Gerätesteuerung und Synchronisation und für die Verbindung zur Benutzer-Schnittstellensoftware.

[0084] Zwischen den beiden Ebenen sieht die vorliegende Erfindung eine Schnittstelle für ein Anwendungsprogramm ("API") "Steuerung/Status" vor. Hierbei handelt es sich um eine Schnittstelle einer mittleren Ebene,

wobei die API-Funktionen fast direkt mit den typischen Benutzer-Bedienelementen übereinstimmen – zum Beispiel: "Wiedergabe", "Sprung zur nächsten Spur", "Spielzeitanzeige" etc. Die Steuerfunktionen stellen für die Benutzerschnittstellen-Software ein Mittel zur Verfügung für die Erteilung von Befehlen zur Steuerung des Spielers, während die Benutzerschnittstellen-Software durch die Statusfunktion die Möglichkeit hat, Informationen über den Zustand der Hardware des Spielers zu erwerben.

[0085] Diese Architektur erlaubt eine kundenspezifische Benutzerschnittstellen-Software. Das "Steuerung/Status"-API sorgt für eine klare Trennung zwischen der Benutzerschnittstellen-Software und der Gerätesteuerungs-Software der niederen Ebene, wodurch es möglich ist, eine andere Benutzerschnittstelle, z.B. eine kundenspezifische Benutzerschnittstelle, in der Standard-Software des Spielers im Austausch zu verwenden. Es versteht sich daher, dass die Software-Architektur der vorliegenden Erfindung eine Vielfalt von Produkt-Konfigurationen unterstützen kann, was wiederum eine reichhaltige Produktfamilie mit ausgezeichneter Langlebigkeit und Skalierbarkeit ermöglicht.

[0086] Ein Netzteil **40** umfasst einen DC-DC-Wandler, um die Benutzung einer internen Batterie oder einer externen Gleichstromquelle zu ermöglichen. Der DVD-Spieler **10** entnimmt alle benötigten inneren Spannungen einer einzigen Gleichstromquelle. Ein Hochspannungs-Inverter liefert die für die Hintergrundbeleuchtung des Displays benötigte Energie. Eine (nicht gezeigte) Ladeeinrichtung wird zum Aufladen der aufladbaren internen Batterie verwendet. Für einen schnellen Zugriff auf die üblichen Funktionen umfasst die Benutzerschnittstelle eine unkomplizierte Bedienleiste an der Frontblende, und die Benutzerschnittstelle für die Software erlaubt eine Steuerung/Bedienung des Spielers, die dessen sämtliche Ausstattungsmerkmale umfasst, die aber dennoch einfach und intuitiv erfolgen kann. Software-Steuerungen, die über eine hochauflösende Grafik gemacht werden, können für die unmittelbare Rückmeldung an den Benutzer das Video überblenden.

[0087] [Fig. 4](#) zeigt ein Systemdiagramm der Image-Enhancement-Engine (IE²) **30** (siehe [Fig. 3](#)) gemäß einer Ausführungsform der vorliegenden Erfindung. Die IE² **30** umfasst eine zweidimensionale Videoverarbeitungs-Chiparchitektur **50** und einen Videoausgabeprozessor **60**. Die Chiparchitektur **50** enthält eine erste De-Interlacing-Stufe **70** und eine zweite De-Interlacing-Stufe **80** und eine Gruppe von Adressierungs- und Sequenzierungs-FIFOs **90**. Die erste De-Interlacing-Stufe **70** umfasst eine progressive Frame-Sequenz-Detektion und Field-Differenz-Verarbeitung. Die zweite De-Interlacing-Stufe **80** umfasst eine vertikale Frequenz-Detektion, Vorzeichenumkehr-Detektion und eine Diagonalmerkmal-Detektion. Der Video-Ausgabeprozessor **60** umfasst eine horizontale Skalierung, Farbraum-Konvertierung, Dithering von 8 auf 6 bbp und Gamma-, Kontrast- und Helligkeits-Korrekturen.

[0088] Die erste De-Interlacing-Stufe **70** von [Fig. 4](#) wird mit Bezug auf die [Fig. 5](#) bis [Fig. 10](#) näher erläutert. [Fig. 5](#) zeigt ein Verfahren **100** zum Kombinieren von Fields zu Frames in einem De-Interlacing-Prozess gemäß einer Ausführungsform der vorliegenden Erfindung. Eine Reihe von Film-Frames **102** mit 24 Frames pro Sekunde wird in Video-Fields **104** mit 60 Fields pro Sekunde konvertiert. Die Video-Fields **104** werden dann durch einen De-Intelacing-Prozess in de-interlaced Frames **106** konvertiert. Die de-interlaced Frames **106a** werden gebildet, indem die beiden aktuellsten Fields zu einem einzelnen Frame zusammengesetzt werden, jedoch erfolgt die Bildung der de-interlaced Frames **106b** durch das Kombinieren von verschiedenen Video-Fields, deren Unterscheidung durch fett dargestellte Pfeile und fett dargestellte Umrahmungen erfolgt.

[0089] Wenn bezugnehmend auf [Fig. 5](#) das Field 2 das "aktuelle" Field ist, dann wird der DI-Frame 1 durch das Kombinieren von Field 1 (das "letzte" Field) und Field 2 (das "aktuelle" Field) gebildet. Die Bildung des DI-Frame 2 erfolgt in der gleichen Weise durch das Kombinieren von Field 3 mit Field 2, wenn Field 3 das "aktuelle" Field ist. Wenn dagegen Field 4 das aktuelle Field ist, entsteht durch die Kombination von Field 4 und Field 5 der DI-Frame 3. In diesem Fall besteht die Kombination aus dem "aktueller" Field und dem "nächsten" Field. Wenn die Field-Kombination auf diese Weise erfolgt, werden alle de-interlaced Frames **106** durch das Kombinieren von Fields gebildet, die denselben Quellen-Frame entstammen. Da diese sich zusammenschließenden Fields denselben Quellen-Frame entstammen, sind sie zeitkorreliert. Deshalb existieren in den de-interlaced Frames **106** keine Bewegungs-Artefakte.

[0090] Das Kombinieren von Fields zu Frames, wie das in [Fig. 5](#) dargestellt ist, erfordert die Identifizierung des in der Originalquelle verwendeten Typs des Bewegt-Bildes. Zunächst muss eine Quelle identifiziert werden als eine solche, in der original progressive Frames vorhanden sind; zweitens muss die für die Bildung von Fields aus den Frames der Originalquelle verwendete Sequenzierung bestimmt werden, um Video-Fields zu Frames zusammenzusetzen und Artefakte zu vermeiden.

[0091] [Fig. 6](#) ist ein Blockdiagramm eines Video-De-Interlacers **130** gemäß einer Ausführungsform der vor-

liegenden Erfindung. Ein digitaler Videostrom läuft durch ein FIFO-Speichermodul **132** in den De-Interlacer **130** ein, bevor er in eine digitale Speichereinheit **134** geschrieben wird. Die digitale Speichereinheit **134** hat die Kapazität zum Speichern von vier kompletten Video-Fields in einer Gruppe von Field-Puffern **134a-d**. Das eingehende Field wird der Reihe nach in den jeweiligen der Field-Puffer **134a-d** geschrieben. Also wird das erste eingehende Video-Field in den Puffer **134a**, das zweite eingehende Video-Field in den Puffer **134b** geschrieben etc. Nachdem der Field-Puffer **134d** voll ist, wird das nächste eingehende Video-Field wieder in den Field-Puffer **134a** geschrieben.

[0092] Während der Zeitperiode eines Fields wird das eingehende Field in einen Field-Puffer **134** geschrieben, und die drei vorhergehenden Fields werden aus den anderen Field-Puffern **134** ausgelesen. Wenn, wie in [Fig. 6](#) gezeigt ist, zum Beispiel das eingehende Field in **134a** geschrieben wird, dann werden die Field-Puffer **134b-d** an eine Gruppe von FIFO-Speichern **136, 138** und **140** ausgelesen. Die FIFO-Speicher **136, 138** und **140** sind vorgesehen, um das Lesen und Schreiben der vier Video-Fields in einer einzigen physikalischen Speichereinheit **134** unterzubringen und auch, um die Taktungsbereiche des eingehenden Videos, der Speicher-Einheit **134** und der anschließenden De-Interlacing-Verarbeitungsstufen zu entkoppeln.

[0093] Die Bezeichnung der Field-Puffer **134a-d** als "eingehendes", "1.", "2." und "3." vorhergehendes Field erfolgt vom Gesichtspunkt der eingehenden Videoströme. Die Ausgaben der drei gelesenen FIFOs **136, 138** und **140** werden jeweils als Nächstes Field, Aktuelles Field und Letztes Field bezeichnet. Diese Bezeichnung erfolgt vom Gesichtspunkt des De-Interlace-Prozessors **130** und impliziert, dass die De-Interlacing-Verarbeitung dem Videostrom eine Verzögerung von circa zwei Field-Zeitperioden aufprägt.

[0094] Die Ausgaben der FIFOs **136, 138** und **140** sind synchronisiert, so dass die den anschließenden Verarbeitungsstufen präsentierten drei Fields räumlich parallel sind. Die drei Fields werden dann als Eingaben einem Inter-Field-progressive-Frame-Detektor **142** präsentiert, der ein letztes/nächstes (L/N)-Signal **144**, ein progressives Frame-Dektions-(PFD)-Signal **146** und einen Stage 1-Dektionswert (Stage1DV) **148** erzeugt. Das L/N-Signal **144** ist ein Steuersignal, das eine Field-Assembly-Stufe **150** anweist, das Aktuelle Field mit dem Letzten Field oder dem Nächsten Field zusammenzusetzen, die drei sequenzielle Fields aus dem Eingabe-Videostrom sind.

[0095] Mit den drei Fields produziert die Field-Assembly-Stufe **150** einen progressiv formatierten Ausgabe-Frame **152**. Wenn das Aktuelle Field gerade ist, dann müssen das Nächste Field und das Letzte Field ungerade sein und umgekehrt. Deshalb ist der progressiv formatierte Frame **152**, der von der Field-Assembly-Stufe **150** ausgegeben wird, stets eine Kombination eines geraden und eines ungeraden Fields. Dies ist wichtig, da die korrekte Zusammensetzung von Fields zu progressiven Frames ein gerades Field und ein ungerades Field für die Eingabe erfordert.

[0096] Der optionale Intra-Frame-De-Interlacer **154** kann für eine zusätzliche Verarbeitung sorgen, um Artefakte zu beseitigen, die unter bestimmten Bedingungen in dem Ausgabe-Frame **152** auftreten, beispielsweise wenn das PFD-Signal **146** negativ ist. Wenn der Frame-Detektor **142** detektiert, dass die eingehenden Videosignale ursprünglich aus einer Quelle kamen, die progressive Frames enthält, wird das PFD-Signal **146** positiv. Es werden drei Originaltypen eines Quellenvideos detektiert: Film mit 24 Frames pro Sekunde, Computer-Animationen oder Frames mit 30 Frames pro Sekunde oder Einzelbilder, bei denen über eine Periode von mehreren Fields keine Bewegung in dem Bild stattfindet. Wenn das PFD-Signal **146** positiv wird, wird der optionale Prozessor **154** ausgeschaltet.

[0097] Wenn aber der Frame-Detektor **142** keine progressive Frame-Sequenz aus den eingehenden Video-Fields ermitteln kann, setzt er das L/N-Signal **114** so, dass dieses stets Aktuelle und Letzte Fields in der Field-Assembly-Stufe **150** zusammensetzt. Dann setzt der Frame-Detektor **142** das PFD-Signal **146** auf "0", wodurch der optionale De-Interlacer **154** darüber informiert wird, dass Artefakte in dem Ausgabe-Frame **152** vorhanden sein können und dass eine weitere Verarbeitung nötig sein kann, um die Artefakte zu entfernen und einen endgültigen progressiven Ausgabe-Frame **152'** zu bilden.

[0098] Die Detektion einer progressiven Frame-Sequenz erfordert eine Dokumentierung des Verlaufs des vorhergehenden Fields. Jedoch kann der progressive Frame-Detektor **142** nur ein Field vorausblicken. In Fällen, in denen ein bedeutender Übergang in dem Bild stattfindet, ist es möglich, dass sich die progressive Frame-Sequenz wegen Bearbeitungen in der Original-Videoquelle ändert. Diese Änderungen könnten Änderungen der Art und Weise erfordern, in welcher aus den eingehenden Fields Frames gebildet werden, jedoch ist es nicht möglich, von der Ankunft der eingehenden Fields im voraus zu erfahren.

[0099] Aus diesem Grund muss der progressive Frame-Detektor **142** in der Lage sein, Übergänge im Bewegt-Bild zu detektieren, weil Übergänge auf Bearbeitungen hinweisen können, die die progressive Field-Sequenzierung ändern können. Wenn ein Übergang detektiert wird, sorgt der progressive Frame-Detektor **142** dafür, dass das PFD-Signal **146** für die Zeitperiode, die für die Bestimmung der neuen progressiven Frame-Sequenz notwendig ist, negativ wird. Dadurch können Bewegungsartefakte während der Rückgewinnung der progressiven Frame-Sequenz notwendigen Zeitperiode durch den optionalen De-Interlacer **154** entfernt werden. In Fällen, in denen eine progressive Frame-Sequenz nicht detektiert wird und die PFD-Ausgabe nicht positiv ist, enthält die Ausgabe des Stage1DV **148** eine Größeninformation, die sich auf den Betrag der in dem Bild vorhandenen Bewegung bezieht. Diese Information kann in dem optionalen De-Interlacer **154** für die weitere Verarbeitung des Bildes verwendet werden.

[0100] [Fig. 7](#) ist ein Systemdiagramm eines progressiven Frame-Detektors **142** gemäß einer Ausführungsform der vorliegenden Erfindung. Der Frame-Detektor **142** umfasst ein Field-Differenzierungsmodul **154**, ein Frequenz-Detektionsmodul **156** und ein progressives Frame-Pattern-Detektions-Modul (PFPD) **158**. Das Field-Differenzierungs-Modul **154** berechnet die Differenz zwischen einem Nächsten Field **160** und einem Letzten Field **162**, verarbeitet die Differenzen zu dem Stage1DV **148**, einem Transitions-Detektion 3:2-Wert **166** und einer Vielzahl von Equal-Field-History-Bits **168**.

[0101] Das Frequenz-Detektionsmodul **156** kombiniert ein Aktuelles Field **164** und das Letzte Field **162** zu einem Frame und detektiert dann die vertikalen hohen Frequenzen, die das Ergebnis von Bewegungsartefakten sind. Dann gibt das Frequenz-Detektionsmodul **156** eine Anzahl von Frequenz-Detektions-History-Bits **170** und einen Transitions-Detektion 2:2-Wert **172** aus. Schließlich nimmt das PFPD **158** als Eingabe den Transitions-Detektion 3:2-Wert **166**, die Vielzahl von Equal-Field-History-Bits **168**, die Frequenz-Detektions-History-Bits **170** und den Transitions-Detektion-2:2 Wert **172**, um das L/N-Signal **144** und das PFD-Signal **146** zu erzeugen.

[0102] [Fig. 8](#) ist ein Flussdiagramm der Verarbeitungsschritte in dem Field-Differenzierungs-Modul **154** gemäß einer Ausführungsform der vorliegenden Erfindung. Ein Nächstes Array von Pixeln **174**, welches eine Untergruppe des Nächsten Fields **160** ist, und ein Letztes Array von Pixeln **176**, welches eine Untergruppe des Letzten Fields **162** ist, sind die Eingaben in einen Differenzierer **178**. Das Nächste und das Letzte Pixel Array **174** und **176** können als Fenster betrachtet werden, die sich über ihre jeweiligen Fields bewegen. Das "Fenster" wird von links nach rechts und von oben nach unten bewegt. Jedes Mal, wenn die Fenster bewegt werden, wird eine neue Differenz berechnet. Das Ergebnis dieser Differenz-Operation **178** ist ein Array von Differenzen **180**.

[0103] Die Berechnung des Stage1DV **148** erfolgt, indem ein gewichtetes Mittel des Arrays von Differenzen **180** verwendet wird. Die Gewichtung ist so, dass die Differenzwerte in der Nähe der Mitte des Arrays von Differenzen den größten Einfluss auf das gewichtete Mittel haben. Das Array von Differenzen **180** wird auch in einen Field-Differenz-Akkumulator **182** eingegeben, der die Differenzen über das gesamte Field summiert, um einen Field-Differenz-Wert **184** zu erzeugen. Die vorhergehenden Field-Differenz-Werte werden in einem Speichermodul **186** gespeichert und werden in einer Operation **188** summiert.

[0104] Die Summe der vorhergehenden fünf Field-Differenz-Werte wird in einer Operation **190** mit dem aktuellen Field-Differenz-Wert verglichen, und das Ergebnis ist das Transition 3:2-Ausgabesignal **192**. Der aktuelle Field-Differenz-Wert **184** wird in einer Schwellenwert-Operation **194** mit dem Wert in einem programmierbaren Field-Differenz-Register **196** verglichen. Das Ergebnis der Schwellenwert-Operation **194** ist ein Equal-Field-Bit **198**, welches ein einzelnes Bit ist, das anzeigt, dass das Nächste Field **160** und das Letzte Field **162** das gleiche sind. Die vorhergehenden Equal-Field-History-Bits **168** werden in einem Speichermodul **200** gespeichert und in dem PFPD **158** von [Fig. 7](#) verwendet.

[0105] [Fig. 9](#) zeigt Details des Frequenz-Detektions-Moduls **156** gemäß einer Ausführungsform der vorliegenden Erfindung. Vertikal benachbarte Pixels **206** aus dem Aktuellen Field **164** und dem Letzten Field **162** werden so assembliert, wie sie räumlich auf einem Display erscheinen würden. Ein Frequenz-Detektions-Wert wird in einer Operation **208** berechnet. Diese Berechnung erfolgt, um die Frequenzen zu detektieren, die mit den de-interlaced Bewegungsartefakten verbunden sind. In einer Operation **210** wird die Ausgabe der Frequenz-Detektion mit einem programmierbaren Schwellenwert **212** verglichen. Die Ergebnisse von fünf benachbarten Frequenz-Detektions-Werten werden in einem Speichermodul **214** gespeichert und in einer Operation **216** summiert.

[0106] Die Ausgabe der Operation **216** wird über die gesamte Field-Periode in einem Field-Frequenz-Detek-

tions-Akkumulator **218** akkumuliert, um einen Field-Frequenz-Detektions-Wert **220** zu erzeugen. Der Field-Frequenz-Detektions-Wert **220** wird in einem Speichermodul **222** gespeichert, das die vorhergehenden fünf Field-Frequenz-Detektions-Werte enthält. Die fünf vorhergehenden Field-Frequenz-Detektions-Werte werden in einer Operation **224** summiert, und die Ergebnisse werden in einer Operation **226** mit dem aktuellen Frequenz-Detektions-Wert **220** verglichen. Das Ergebnis dieses Vergleichs in Operation **226** ist das Transition 2:2-Bit **228**, welches anzeigen, dass eine Transition in der 2:2 Sequenz stattgefunden hat.

[0107] Als Teil eines parallelen Prozesses wird ein erster vorhergehender Field-Detektions-Wert **230**, der in dem Speichermodul **222** gespeichert wurde, zu einer Multipliziereinheit **232** übertragen und mit einem Wert multipliziert, der in einem programmierbaren Field-Frequenz-Schwellwertregister **234** gespeichert ist. Das Ergebnis der Multiplikation wird in einer Operation **236** mit dem aktuellen Field-Frequenz-Detektions-Wert **220** verglichen. Das Ergebnis ist ein Relativ-Frequenz-Differenz-Bit **238**, welches dann in einem Speichermodul **240** gespeichert wird. Die vorhergehenden zehn Relativ-Frequenz-Differenz-Bits **242** werden an das PFPD-Modul **158** ausgegeben.

[0108] [Fig. 10](#) ist ein System-Diagramm des PFPD-Moduls **158** gemäß einer Ausführungsform der vorliegenden Erfindung. Das PFPD-Modul **158** führt logische Operationen an einem Satz von Field-Differenz-History-Bits **244**, an den Frequenz-Detektions-History-Bits **242**, dem Transition 3:2-Wert **192** (siehe [Fig. 8](#)) und dem Transition 2:2-Wert **228** (siehe [Fig. 9](#)) durch. Nach Eingabe der Field-Differenz-History-Bit **244** bestimmt eine logische Operation **246** das 3:2 Pulldown-Detektions-Bit, indem nach Mustern gesucht wird, in denen jedes fünfte Field gleich ist. Dann detektiert eine logische Operation **248** Standbilder durch Einstellen des STILL-Bit, wenn die letzten vier Field-Differenzen Null sind.

[0109] Der Zustand des L/N-Steuersignals wird durch eine logische Operation **250** eingestellt.

[0110] Aus der Eingabe der Frequenz-Detektions-History-Bit **242** detektiert eine logische Operation **252** ein 2:2 Pulldown-Detektions-Bit durch ein Suchen nach alternierenden Mustern hoher Frequenz und niedriger Frequenzen in sukzessiven Field-Zeiten und durch Überwachen der Frequenz-Detektions-History-Bits **242**. Dann bestimmt eine logische Operation **254** das L/N-Steuersignal für den Fall des 2:2 Pulldown. Das PFD-Signal **146** wird aus dem 3:2 Pulldown-Detektions-Bit, dem Transition 3:2-Wert **192**, dem 2:2 Pulldown-Bit und dem Transition 2:2-Wert **228** bestimmt.

[0111] Es werden drei Arten von Quellen progressiver Frames detektiert. Film, der unter Verwendung des 3:2 Pulldown in Video konvertiert wird, wird detektiert durch die Berechnung der Differenz zwischen Paaren von geraden Fields oder Paaren von ungeraden Fields und durch Suchen nach dem Zustand, in dem jede fünfte Differenz Null ist. Dieser Zustand wird durch das 3:2 Pulldown-Signal angezeigt. Computergenerierte Animationen unter Verwendung des 2:2 Pulldown werden detektiert, indem ein Frequenz-Detektionsverfahren angewandt wird, das in jedem zweiten kombinierten Frame nach Frequenzen sucht, die mit Bewegungsartefakten verbunden sind, was durch das 2:2 Pulldown-Signal angezeigt wird. Standbilder werden detektiert, wenn die Field-Differenzen für mehrere konsekutive Fields Null sind, was durch das STILL-Signal angezeigt wird.

[0112] Progressive Frames sind einfach das logische ODER dieser drei Signale, wie das durch ein logisches ODER-Gatter **256** gezeigt ist. Jedoch müssen auch Transitionen berücksichtigt werden. Wie vorstehend erwähnt, ist eine Transition eine bedeutende Änderung der Field-Sequenz, die das Ergebnis von Bearbeitungen oder einer kurzzeitigen Unterbrechung des Video-Signals ist. Wird eine große Änderung detektiert, so kann eine progressive Frame-Detektion, die von einer History von mehreren Fields abhängt, unzuverlässig sein für die Periode, die notwendig ist, um ein progressives Frame-Muster zu erstellen. In den dargestellten Beispielen beträgt diese Periode zehn Field-Zeiten oder etwa ein Sechstel einer Sekunde für 60 Hz-Fields.

[0113] In einer Gruppe von logischen Operationen **258** wird ein Impuls unter zwei Bedingungen erzeugt. Die eine ist, dass eine 3:2 Pulldown-Sequenz detektiert wird und eine Transition stattfindet; die zweite ist, dass eine 2:2 Pulldown-Sequenz detektiert wird und in der Sequenz eine Transition detektiert wird. Jeder dieser Fälle setzt einen Timer **260** in Gang, der einen Impuls einer Dauer von zehn Fields erzeugt, da zehn Fields die Zeitperiode sind, die benötigt wird, um eine neue progressive Sequenz zu erstellen. Während dieser Zeit deaktiviert ein logisches UND-Gatter **262** das PFD-Bit **146**. Wie vorstehend erwähnt, kann der optionale Intra-Frame-De-Interlace-Prozessor **154** (in [Fig. 6](#) dargestellt), wenn das PFD-Bit **146** nicht positiv ist, während dieser Auszeit Bewegungsartefakte beseitigen.

[0114] Es versteht sich deshalb, dass das progressive Frame-Detektionsverfahren der vorliegenden Erfindung für eine Eliminierung von Artefakten an Kanten in einem Videobild sorgt. Dies wird bewerkstelligt durch

die Identifizierung des Typs des Originalfilms und die Verwendung dieser Information als Hilfe für das Kombinieren von Video-Fields im De-Interlacing-Verfahren. Die Kombination dieser Techniken liefert ein artefaktarmes de-interlaced Bild mit hoher Auflösung.

[0115] Die zweite De-Interlacing-Stufe **80** von [Fig. 4](#) wird mit Bezug auf die [Fig. 11](#) bis [Fig. 21](#) näher erläutert. [Fig. 11](#) zeigt ein Verfahren **300** für eine De-Interlacing-Verarbeitung. Ein Video-Field **302**, das Abtastzeilen **306** enthält, und ein vorhergehendes Video-Field **304**, das Abtastzeilen **308** enthält, werden in einen Field-Kombinations-Deinterlace-Prozessor **310** eingegeben. Jedes der Video-Fields hat 720 Pixel × 280 Pixel. Das Ergebnis ist ein kombinierter Frame **312** mit 720 Pixel × 480 Pixel, wobei die Abtastzeilen **314** auf das Video-Field **302** und die Abtastzeilen **316** auf das Video-Field **304** zurückgehen.

[0116] Wenn dieses einfache De-Interlacing nach dem Stand der Technik durchgeführt wird und ein für eine Interlace-Darstellung formatierter Film in ein progressives Format konvertiert wird, entsteht ein deutlich wahrnehmbares "Artefakt" bzw. ein Fehler, da der Bildinhalt der vertikal benachbarten Zeilen um 1/60 Sekunde zeitverschoben ist, wie das vorstehend erwähnt wurde. Dieser Fehler ist am stärksten rund um die Kanten/Ränder von Objekten sichtbar, die in Bewegung sind.

[0117] [Fig. 12](#) zeigt ein zweidimensionales Array von Pixel-Werten **318**, das eine Teilmenge des kombinierten Frame **312** von [Fig. 11](#) ist, die verwendet wird, um die vorliegende Erfindung anhand eines Beispiels zu beschreiben. Das Array von Pixeln **318** hat in der Darstellung eine Breite von 5 und eine Höhe von 7 Pixeln. Das Array **318** ist in Querrichtung oben mit C0 bis C4 bezeichnet, wodurch Spalten gekennzeichnet sind, und auf der linken Seite vertikal von oben nach unten mit R0 bis R6, wodurch Reihen gekennzeichnet sind. Die in dem Array **318** enthaltenen Pixel werden verwendet, um einen Frequenz-Detektionswert zu berechnen. Darüber hinaus wird das Array **318** auch verwendet, um diagonale Merkmale zu detektieren, und schließlich, um das resultierende Pixel zu berechnen.

[0118] Das Array **318** ist so positioniert, dass eine Gruppe von geradzahligen Reihen **320** Pixel aus dem jüngsten oder "aktuellen" Field der Originalquelle und eine Gruppe von ungeradzahligen Reihen **322** Pixel aus dem vorhergehenden Field enthält. Das Array **318** wird dann horizontal von links nach rechts schrittweise über den kombinierten Frame **312** (siehe [Fig. 11](#)) bewegt. Jeder Schritt bewirkt, dass sich die Pixel in jeder der Spalten C1, C2 und C3 und C4 in die unmittelbar links daneben liegende Spalte bewegen. Die Pixel in Spalte C0 verschieben sich nach außerhalb des Arrays **318**, und eine neue Spalte von Pixeln schiebt sich in die Spalte C4.

[0119] Nachdem das Array **318** schrittweise über alle horizontalen Positionen des kombinierten Frame **312** bewegt wurde, wird es um zwei Pixel schrittweise vertikal nach unten und zurück zur linken Seite des kombinierten Frame **312** bewegt. Deshalb enthalten die mit geraden Zahlen nummerierten Reihen **320** Pixel aus dem letzten Field, und die mit ungeraden Zahlen nummerierten Zeilen **322** enthalten Pixel aus dem vorhergehenden Field. Der Vorgang wiederholt sich dann selbst, während das Array **318** erneut horizontal von links nach rechts schrittweise über den kombinierten Frame **312** bewegt wird. An jeder Position in dem zweidimensionalen Array wird ein Detektionswert (DV) berechnet.

[0120] [Fig. 13](#) zeigt ein Verfahren **326** zur Gewinnung eines Ausgabe-Pixel **338** aus dem zweidimensionalen Array **318**. In einer Operation **328** wird ein Frequenz-Detektionswert ermittelt, wofür die sieben Pixel einer jeden Spalte des zweidimensionalen Arrays verwendet werden. Da fünf Spalten vorhanden sind, werden fünf Frequenzdetektions-Operationen durchgeführt, die einen Satz von Detektionswerten fd0, fd1, fd2, fd3 und fd4 liefern. Dann vergleicht eine Operation **330** den Satz der Detektionswerte fd0-fd4 mit einem Schwellenwert. Danach wird in einer Operation **332** der Satz von Detektionswerten fd0-fd4 zu einem gewichteten Mittel kombiniert, um einen ultimativen Detektionswert (UDV) **334** zu erhalten.

[0121] Die Gewichtungsfaktoren können Variablen sein. Ein Beispiel für eine Gewichtung ist das folgende: UDV = (fd0 + (2·fd1) + (8·fd2) + (2·fd3) + fd4)/14. Die Gewichtung bewirkt, dass der der Mitte des Arrays **318** am nächsten liegende Frequenz-Detektionswert den größten Einfluss auf den UDV **334** hat. Auf diese Weise führt die Verwendung von fünf horizontal benachbarten Frequenz-Detektionswerten zu einem Tiefpaß-Filtrationsvorgang, der für glattere Übergänge zwischen Bereichen innerhalb des kombinierten Frame **312** sorgt, in denen Bewegungsartefakte existieren und nicht existieren.

[0122] Der in der Operation **332** berechnete UDV wird zur Steuerung eines Abmischvorgangs **336** verwendet, der vorzugsweise die folgende Gleichung implementiert: pixelout = (UDV·(pR2C2 + pR4C2)/2) + ((1 - UDV)·pR3C2), wobei pixelout das neue Ausgabepixel der De-Interlacing-Operation, pR2C2 ein Pixel in dem

Array **318** an dem Ort in Reihe 2, Spalte 2, pR4C2 ein Pixel in dem Array **318** an dem Ort in Reihe 4, Spalte 2 und pR3C2 ein Pixel in dem Array **318** an dem Ort in Reihe 3, Spalte 2 ist.

[0123] Das Ergebnis des Abmischvorgangs ist, dass der neue Wert von Pixel pR3C2 des Arrays **318** von dem UDV **334** abhängt. Wenn durch die Berechnung des UDV **334** keine Bewegung festgestellt wird, dann ist das Pixel bei pR3C2 der unmodifizierte Wert des Pixel an dieser Position in dem vorhergehenden Field. Ist das Ergebnis ein hoher UDV **334**, d.h. es ergibt sich ein Wert von 1, so wurde ein starkes Bewegungsartefakt detektiert, und es wird der Wert von pR3C2 berechnet, indem ein Durchschnittswert der Werte von pR2C3 und pR4C3 des Arrays **318** ermittelt wird. Das Ergebnis dieser Durchschnittsermittlung zeigt keine Bewegungsartefakte, da es aus Werten des letzten Fields gebildet wird, die mit dem letzten Fields zeitkorreliert sind. Detektionswerte zwischen 0 und 1 bewirken, dass das Pixel bei pR3C eine Mischung aus pR3C2 und dem Durchschnitt von pR2C3 und pR4C3 ist.

[0124] In [Fig. 14A](#) ist ein Bild **340** dargestellt, das die Operation **328** in größerem Detail zeigt. Das Bild **340** zeigt die Berechnung eines einzelnen Frequenz-Detektionswerts für eine Spalte des Arrays **318**. Das Bild **340** enthält ein verzerrtes Objekt **342**, das durch ein Interlace-Bewegungsartefakt hervorgerufen wird. Das Bild wird entlang einer Linie **344** abgetastet, die zu Beispielzwecken dargestellt ist. Diese Abtastung entspricht einer der Spalten in dem zweidimensionalen Array **318**. In diesem Beispiel durchläuft die Linie **344** einen Bereich, in dem Artefakte vorhanden sind, aber im Allgemeinen kann eine Abtastung von vertikal benachbarten Pixeln Artefakte enthalten oder nicht.

[0125] [Fig. 14B](#) ist ein Graph **346** einer Gruppe von Samples **348**, die durch die Abtastung entlang der Linie **344** von [Fig. 14A](#) erhalten wurde. Die Gruppe von Samples **348** ist graphisch derart dargestellt, dass die Reihennummern entlang der horizontalen Achse und die Helligkeit oder Intensität des Pixel entlang der vertikalen Achse angegeben sind. Aus dem Graph **346** ist offensichtlich, dass die Bereiche, in denen Bewegungsartefakte vorhanden sind, wie beispielsweise die Gruppe von Samples **348**, eine charakteristische Frequenz zeigen. Statt der zeitlichen Frequenz handelt es sich hier vielmehr um die räumliche Frequenz, und diese wird höchst vorteilhaft in Zyklen pro Zeile und nicht in Zyklen pro Sekunde oder in Hertz angegeben. Die charakteristische Frequenz beträgt 1 Zyklus/2 Zeilen oder 0,5 Zyklen/Zeile.

[0126] [Fig. 14C](#) ist ein Graph einer abgetasteten Cosinuswelle **350**. Die durch das Bewegungsartefakt hervorgerufene charakteristische Frequenz wird durch Multiplizieren der Gruppe von Samples mit der abgetasteten Cosinuswelle **350** detektiert. Die abgetastete Cosinuswelle **350** hat eine Frequenz gleich der charakteristischen Frequenz des Bewegungsartefakts. Dann wird das Ergebnis unter Anwendung der folgenden Gleichung integriert:

$$fd = \sum_{R=0}^{R=6} Y(R) \cos(2R\pi * 0.5 \text{cycle / line})$$

wobei fd der Frequenz-Detektionswert für eine Spalte des Arrays **318** ist, R ein Zeilenindex ist, der dem R0 ... R6 des Arrays **318** entspricht und die Einheiten "line" hat, und wobei Y(R) der Satz von vertikal benachbarten Samples **348** ist.

[0127] Der Ausdruck $\cos(2\pi R \cdot 0.5 \text{ cycle/line})$ wird vereinfacht zu 1 für R = 0, 2, 4 und 6 und -1 für R = 1, 3 und 5. Wenn 1 und -1 für R0 R6 eingesetzt werden, wird die Frequenz-Detektionsgleichung zu: $fd = (Y6/2 + Y4 + Y2 + Y0/2) - (Y5 + Y3 + Y1)$. Es ist zu beachten, dass Y6 und Y0 durch 2 geteilt werden, da die Integration über den Limits 0 bis 6 ist. Der endgültige fd ist der Absolutwert: $fd = \text{Abs}(fd)$. Das Verfahren **326** von [Fig. 13](#) wird für jede Spalte in dem Array **318** wiederholt und liefert die Gruppe von Frequenz-Detektionswerten **330**.

[0128] [Fig. 15](#) ist ein Graph **352**, der die Thresholding/Binarisierungs-Operation **330** in größerem Detail zeigt. Jeder fd ist eine Zahl in dem Bereich 0 bis 1. Der Graph **352** umfasst eine non-thresholded/nichtbinarisierte Skala **354**, aus der Werte durch ein Schwellwertverfahren in die thresholded/binarisierte Skala **356** übertragen werden. Das Schwellwertverfahren setzt alle Werte über dem oberen Schwellenpunkt **358** auf den Wert von 1. Alle Werte unterhalb des unteren Schwellenpunkts **360** werden auf den Wert von 0 gesetzt. Die Werte zwischen dem oberen und dem unteren Schwellenwert werden auf den Bereich 0 bis 1 ausgedehnt. Das Schwellwertverfahren kann mit der folgenden Gleichung beschrieben werden: $tfd = (ptfd - LTH)/UTH$, wobei tdf der thresholded/binarisierte Frequenz-Detektionswert ist, pthfd der pre-thresholded/vorbinarisierte Frequenz-Detektionswert (die Ausgabe von Operation **328**) ist, LTH der untere Schwellenwert und UTH der obere Schwellenwert. Wenn $tfd > 1,0$, dann $tfd = 0,1$. Andernfalls gilt: Wenn $tfd < 0$, dann $tfd = 0$.

[0129] Es versteht sich daher, dass das erfindungsgemäße De-Interlacing-Verfahren für eine gute vertikale Auflösung ohne Bildung von Kanten-Artefakten in bewegten Objekten in einem Videobild sorgt. Dies wird bewerkstelligt durch Anwenden eines Two-Field-Interlacing, wenn das Bild relativ statisch ist, und eines One-Field-Linedoubling, wenn sich das Bild rasch ändert. Die Kombination dieser Techniken liefert ein artefaktares de-interlaced Bild mit hoher Auflösung.

[0130] Der UDV **334** wird dann vorzugsweise in einer Mischschaltung **400** verwendet, die in [Fig. 16](#) dargestellt ist. Die Mischschaltung **400** berechnet einen neuen Wert für das Pixel an dem Ort R3C2 des Arrays **318**. Wenn keine Bewegungsartefakte detektiert werden, ist der Wert des UDV **334** "0", und die Mischschaltung **400** gibt das Original-Pixel R3C2 aus. Ist der Wert des UDV **334** "1", dann gibt die Mischschaltung **400** den Durchschnitt der Pixel über und unter R3C2 aus, so dass die Ausgabe der Mischschaltung **400** der Durchschnitt von R2C2 und R4C2 ist.

[0131] [Fig. 17](#) ist ein Diagramm einer exemplarischen Operation der Mischschaltung **400**, wenn der UDV **334** größer als "0", jedoch kleiner als "1" ist. Die Mischschaltung **400** verwendet Informationen aus dem Drei-Pixel-Array **402** durch Mischen von R3C2 und dem Durchschnitt von R2C2 und R4C2, um ein neues Ausgabe-Pixel **406** an dem Ort R3C2 zu bilden.

[0132] [Fig. 18](#) ist eine Darstellung eines Verfahrens **408** zum Detektieren von Diagonal-Merkmalen. Ein zweites Array **410**, das eine Untermenge von Array **318** ist, wird in eine Diagonal-Detektionsschaltung **412** eingegeben, die parallel zu dem Verfahren **326** von [Fig. 13](#) arbeitet. Wenn kein Diagonal-Merkmal detektiert wird, dann liefert die Diagonal-Detektionsschaltung **412** keine Ausgabe. Wenn aber ein Diagonal-Merkmal detektiert wird, erzeugt die Diagonal-Detektionsschaltung **412** zwei Ausgaben: ein Einzelbit-Sign-Signal **414** und ein Mehrfachbit-SlopeFade-Signal **416**. Das spezielle Verfahren zum Berechnen des Sign- und des SlopeFade-Signals **414** und **416** ist in [Fig. 21](#) gezeigt und in der entsprechenden Beschreibung dargelegt.

[0133] Das Sign-Signal **414** wird verwendet, um zu bestimmen, welches Paar von Pixeln diagonal an R3C2 angrenzt. Das SlopeFade-Signal **416** ist ein Maß der Größe eines Diagonal-Merkmales. Die Größe wird bestimmt durch die Kontrastmenge entlang des Diagonal-Merkmales. Ein starker Kontrast, wie zum Beispiel eine diagonale weiße Linie über einem schwarzen Hintergrund, ergibt die höchsten Werte des SlopeFade-Signals **416**. Ein schwächerer Kontrast resultiert in einem niedrigeren Wert des SlopeFade-Signals **416**.

[0134] [Fig. 19](#) ist ein Blockdiagramm einer Diagonal-Mischschaltung **418** gemäß einer Ausführungsform der vorliegenden Erfindung. Die Diagonal-Mischschaltung **418** umfasst einen Multiplexer **420**, einen ersten Mischer **422** und einen zweiten Mischer **424**. Um zu bestimmen, welches Paar von diagonal benachbarten Pixeln verwendet wird, ist der Multiplexer **420** abhängig von dem Sign-Signal **414**. Nachdem ein Paar von diagonal benachbarten Pixeln gewählt wurde, mischt der erste Mischer **422** die Pixelwerte, die vertikal an R3C2 angrenzen, mit jenen, die diagonal an R3C2 angrenzen. Der Mischbetrag wird durch das Slope-Fade-Signal **416** bestimmt, welches proportional zur Größe des Diagonal-Merkmales ist, das detektiert wird.

[0135] Der zweite Mischer **424** ist die endgültige Mischstufe und ist identisch mit der in [Fig. 16](#) gezeigten Mischschaltung **400**. Der zweite Mischer **424** produziert eine Ausgabe, die durch das Eingabepixel R3C2 und die Ausgabe des ersten Mixers **422** bestimmt wird. Der UDV **334** ist die Steuereingabe für den zweiten Mischer **424**. Kurz zusammengefasst wird der neue Pixelwert bei R3C2 aus den Pixelwerten von dem zweiten Array **410** berechnet. Die Steuersignale zum Bestimmen des endgültigen Pixelwerts sind der UDV **334**, das Sign-Signal **414** und das SlopeFade-Signal **416**.

[0136] [Fig. 20](#) ist ein Diagramm, das die Pixel des zweiten Arrays **410** zeigt, das für die Berechnung der Ausgabe der Diagonal-Mischschaltung **418** verwendet werden. Wenn keine Diagonal-Merkmale in dem zweiten Array **410** detektiert werden, dann wird die Ausgabe der Mischschaltung aus den Pixeln entlang einer Linie **426** bestimmt. Wird ein Diagonal-Merkmal in der Diagonal-Detektionsschaltung **412** detektiert, dann werden die diagonal an R3C2 angrenzenden Pixel entlang einer Linie **428** oder einer Linie **430** für die Berechnung des Ausgabepixel verwendet. Das Sign-Signal **414** wird verwendet, um zu bestimmen, welche Linie, nämlich **428** oder **430**, verwendet wird.

[0137] [Fig. 21](#) ist ein Flussdiagramm eines Diagonal-Detektionsverfahrens **432** gemäß einer Ausführungsform der vorliegenden Erfindung. Das Verfahren **432** zeigt den Ablauf von logischen und mathematischen Operationen, die für die Berechnung des SlopeFade-Signals **416** und des Sign-Signals **414** aus dem zweiten Array **410** angewendet werden. Die Eckpixel werden durch eine Operation **434** in zwei horizontale Paare und zwei vertikale Paare unterteilt. Die horizontalen Paare sind mit hv2 und hv4 und die beiden vertikalen Paare mit vv2

und vv4 bezeichnet. Differenzen für jedes Paar von Eckpixelwerten werden durch Subtraktion berechnet, wodurch ein Paar von horizontalen Differenzen und ein Paar von vertikalen Differenzen erzeugt wird.

[0138] In einer Operation **436** werden die beiden horizontalen und vertikalen Differenzen summiert, um einen Vektor der horizontalen und vertikalen Differenz für das zweite Array **410** zu bilden. Eine Operation **438** berechnet den Absolutwert der Vektoren der horizontalen und vertikalen Differenz. Ein Schwellenwert wird verwendet, um die Größe der SlopeFade-Ausgabe **416** in einer Operation **440** zu berechnen. Die Ausgabe der Operation **440** ist ein unqualifiziertes Slope-Fade-Signal (unQualSlopeFade), das noch einer "Ausradierung" durch das DiagDetect-Signal und das SlopeDisQual-Signal, die durch parallele Operationen des Verfahrens **432** erzeugt werden, unterzogen wird.

[0139] Die Signs (Vorzeichen) der horizontalen und vertikalen Differenzen aus Operation **434** werden aufgezeichnet und in einer Operation **442** gespeichert. Die Signs geben an, ob das Ergebnis der Operation **434** positive oder negative Zahlen ergeben hat. Dann sucht eine Operation **444** nach Fällen, in denen die Signs der horizontalen und vertikalen Differenz-Operationen entgegengesetzt sind. Wenn solche Fälle gefunden werden, dann wird SlopeDisQual auf "1" gesetzt. Bei nicht entgegengesetzten Signs der Differenz-Operationen ist SlopeDisQual "0".

[0140] In Operation **444** sucht der Diagonal-Detektor nach Diagonal-Merkmalen, die relativ bedeutend sind; insbesondere muss das Merkmal größer sein als das neun Pixel umfassende zweite Array **410**, das als Eingabe in den Diagonal-Prozessor verwendet wird. Bildmerkmale, die kleiner sind als das zweite Array **410** können bewirken, dass die Diagonal-Verarbeitung ein Diagonal-Merkmal unkorrekt detektiert. Diese kleinen Merkmale lassen sich detektieren, indem die Vorzeichen observiert werden und SlopeDisQual entsprechend gesetzt wird.

[0141] Eine Operation **446** vergleicht die Größe der in Operation **438** berechneten horizontalen und vertikalen Vektoren, um ein Diagonal-Merkmal zu detektieren. Dann wird unter Verwendung der horizontalen und vertikalen Vektoren in Operation **446** das DiagDetect-Signal erzeugt. Das Verhältnis der Differenz der horizontalen und vertikalen Vektoren und der Summe der horizontalen und vertikalen Vektoren wird durch einen Schwellwert, diagDetThresh, eingestellt.

[0142] Eine endgültige SlopeFade-Ausgabe wird in einer Operation **448** berechnet, in der zwei Einzelbit-Eingaben verwendet werden, um die SlopeFade-Ausgabe zu qualifizieren. Das erste Bit ist ein DiagonalDetect-Bit, und das zweite Bit ist ein SlopeDisQual-Bit. SlopeFade ist Null, wenn DiagDetect 0 ist oder wenn SlopeDisQual eine 1 ist, andernfalls nimmt SlopeFade den Wert von unQualSlopeFade an. Das SlopeDisQual-Signal ändert das SlopeFade-Signal in Null für Fälle, in denen die Steilheit des Diagonal-Merkmales nicht zuverlässig berechnet werden kann.

[0143] Schließlich wird unter Verwendung der Sign-Bits, die durch die in Operation **436** gebildeten Summen der horizontalen und vertikalen Vektoren erzeugt wurden, in einer Operation **450** das Sign-Signal berechnet. Das Sign-Signal **414** wird berechnet, indem diese logische Operation angewendet wird, um die Steilheit des Diagonal-Merkmales zu bestimmen.

[0144] Der Videoausgabe-Prozessor **60** von [Fig. 4](#) wird im Einzelnen mit Bezug auf die [Fig. 22–Fig. 27](#) erläutert. [Fig. 22](#) zeigt ein Beispiel eines variablen Skalier-FIR-Filters **500**. Das variable Skalier-FIR-Filter **500** enthält ein Verschieberegister **502** mit einer Reihe von Registern **504**, deren jedes mit einem Taktgeber **506** verbunden ist. Jedes Register **504** ist durch eine Reihe von Filter-Taps **514** mit einem von einer Gruppe von Multiplizierern **508**, **510** und **512** verbunden. Die Multiplizierer **508**, **510** und **512** akzeptieren zwei zu multiplizierende Eingaben. Die erste Eingabe ist ein Acht-Bit-Datentyp, und die zweite Eingabe ist ein Koeffizient. Die Multiplizierer **508**, **510** und **512** unterscheiden sich voneinander insofern, als sie Koeffizienten akzeptieren, die auf eine unterschiedliche Anzahl von Bit quantisiert sind. Die Multiplizierer **508** verwenden die kleinste Anzahl von Bit pro Koeffizient, und die Multiplizierer **512** verwenden die meisten Bit pro Koeffizient. Die Multiplizierer **508**, **510** und **512** sind mit einer Koeffizienten-Speichereinheit **516** und mit einem Addierer **518** verbunden.

[0145] Acht Bit von Daten werden durch das Verschieberegister **502** in das variable Skalier-FIR-Filter **500** eingegeben. Die Ausgabe eines jeden Registers **504** ist durch eine Reihe von Filter-Taps **514** mit einem von einer Gruppe von Multiplizierern **508**, **510** und **512** verbunden, um mit einem Koeffizienten multipliziert zu werden, der von der Koeffizienten-Speichereinheit **516** erzeugt wird. Eine neue Gruppe von Koeffizienten wird bei jedem Zyklus des Taktgebers **506** durch die Koeffizienten-Speichereinheit **516** in die Multiplizierer **508**, **510**

und **512** eingegeben. Die Ergebnisse aus jedem Multiplizierer **508**, **510** und **512** werden durch den Addierer **518** summiert, um ein gefiltertes Ausgabe-Sample zu liefern.

[0146] [Fig. 23](#) ist ein Graph von Tiefpass-Filterkoeffizienten **520** in dem Zeitbereich, die in der Koeffizienten-Speichereinheit **516** gespeichert sind, um Koeffizienten zu erzeugen. Die Tiefpass-Filterkoeffizienten **520** werden durch nachstehende Gleichung dargestellt:

$$(1) \quad \sum_{i=1}^m c(i) = \frac{2Lfc}{2fc\pi(i-1/2)} * \sin[2fc\pi(i-1/2)] * \{0.54 + 0.46 \cos[2\pi(i-1/2)/taps]\}$$

[0147] Die nichtquantisierte und kontinuierliche Welle ist durch eine Kurve **522** dargestellt. Filterkoeffizienten **524** sind an oder in der Nähe der Kurve **522** aufgetragen. Manche Koeffizienten **524** scheinen geringfügig abseits der Kurve zu liegen, was auf den Fehler zurückzuführen ist, der durch die Quantisierung jedes Koeffizienten auf eine begrenzte Anzahl von Bit eingeführt wurde.

[0148] [Fig. 24](#) zeigt die Koeffizienten **524** organisiert in $L = 8$ Gruppen von $\text{mults} = 6$ Koeffizienten pro Gruppe. Die Summe aller Koeffizienten in jeder Gruppe i , in der $i = 1$ bis L ist, ist durch die nachstehende Gleichung angegeben.

$$(2) \quad s(i) = \sum_{j=1}^{j=\text{mults}} c(L(j-1)+i)$$

[0149] [Fig. 25](#) ist ein Flussdiagramm eines Quantisierungsverfahrens **526** gemäß einer Ausführungsform der vorliegenden Erfindung. Das Verfahren **526** initialisiert sich mit einer gegebenen Gruppe von Parametern **528**, die für die Berechnung der Koeffizienten benötigt wird, wobei L der Zähler des Skalierverhältnisses L/M ist, mults die Anzahl der in dem FIR-Filter verwendeten Multipliziereinheiten und n die Anzahl von Bit ist, auf welche die Koeffizienten quantisiert werden. Eine Operation **530** berechnet die FIR-Filterkoeffizienten durch Anwendung von Gleichung 1. In einer Operation **532** werden die Koeffizienten von links nach rechts organisiert und mit $c(1), c(2), c(3), \dots, c(L \cdot \text{mults})$ bezeichnet.

[0150] In einer Operation **534** wird jeder Koeffizient durch Runden auf eine Anzahl n von Bit quantisiert. Dann startet eine Operation **536** eine Schleife, die L Male abgearbeitet wird, einmal für jede Koeffizientengruppe, wobei alle Koeffizienten in jeder Gruppe summiert werden. Eine Operation **538** summiert die Koeffizienten für Gruppe(i), die in [Fig. 24](#) dargestellt sind. Dann testet eine Operation **540** das Ergebnis der Summieroperation **538** auf ein 1,0 Ergebnis. Wenn die Operation **540** ein wahres Ergebnis liefert, erfolgt keine weitere Verarbeitung.

[0151] Der Schleifen-Iterator wird in einer Operation **542** inkrementiert, und die Steuerung/Regelung führt zurück zu Operation **536**. Wenn die Operation **540** ein falsches Ergebnis liefert, dann wird ein Fudge-Wert F in einer Operation **544** berechnet, indem die in der Operation **538** gebildete Summe von 1,0 subtrahiert wird. Die Verarbeitung führt dann weiter zu einer Operation **546**, in der einer der Koeffizienten von Gruppe(i) derart geändert wird, dass die Summe aller Koeffizienten in der Gruppe(i) = 1,0 ist.

[0152] [Fig. 26](#) ist ein Flussdiagramm der Operation **546** aus [Fig. 25](#) in wesentlich größerem Detail. Eine Operation **548** ist eine Schleife, die so eingerichtet ist, dass sie in einer bestimmten Reihenfolge durch die Koeffizienten von Gruppe(i) führt. Die Reihenfolge beginnt mit dem äußersten Koeffizienten der Gruppe(i) und führt dann weiter in Richtung auf die Mitte der Gruppe. Die Operation **548** wird mehrmals (mults times) durchgeführt, da pro Gruppe eine Mehrzahl (mults number) von Koeffizienten vorhanden ist. Als nächstes wird in einer Operation **550** ein Index k berechnet, der für die Verarbeitung der Koeffizienten in der vorgenannten Reihenfolge verwendet wird.

[0153] Die Koeffizienten an dem linken oder dem rechten Rand der Gruppe von Koeffizienten müssen als Spezialfall behandelt werden. Deshalb wird an dem Index k eine Operation **552** durchgeführt, um zu bestimmen, ob der zu verarbeitende Koeffizient entweder der erste Koeffizient, $c(1)$, oder der letzte Koeffizient, $c(L \cdot \text{mults})$, ist. Wenn die Operation **552** bestimmt, dass der einzustellende Koeffizient derjenige ganz links ist, d.h. $c(1)$, dann wird eine Operation **554** durchgeführt.

[0154] Die Operation **554** evaluiert, ob der Absolutwert der Summe von $c(1)$ und F kleiner oder gleich dem Absolutwert des Koeffizienten rechts von $c(1)$ ist. Dies bedeutet, dass $c(k+1) \leq c(2)$ ist. Wenn das Ergebnis

wahr ist, dann kann c(1) eingestellt werden durch Addieren von F, ohne eine Diskontinuität oder Divergenz von der Nullachse zu erzeugen. Der Koeffizient wird in einer Operation **564** eingestellt, und die Operation **546** wird erfolgreich beendet. Ist das Ergebnis falsch, dann führt die Operation **560** eine Schleifeniteration durch.

[0155] Wenn die Operation **552** bestimmt, dass der einzustellende Koeffizient der äußerst rechte ist, das heißt $c(L \cdot \text{mults})$, dann führt das Verfahren weiter zu einer Operation **556**. Die Operation **556** evaluiert, ob der Absolutwert der Summe von $c(L \cdot \text{mults})$ und F kleiner oder gleich dem Absolutwert des Koeffizienten links von $c(L \cdot \text{mults})$ ist, das heißt $c(L \cdot \text{mults} - 1)$. Wenn das Ergebnis wahr ist, dann kann $c(L \cdot \text{mults})$ eingestellt werden durch Addieren von F, ohne eine Diskontinuität oder Divergenz von der Nullachse zu erzeugen. Der Koeffizient wird in **564** eingestellt, und die Operation **546** wird erfolgreich beendet. Wenn das Ergebnis der Operation **556** falsch ist, wird in Operation **560** eine Schleifeniteration durchgeführt.

[0156] Wenn die Operation **552** bestimmt, dass der einzustellende Koeffizient weder der äußerst linke noch der äußerst rechte ist, dann wird eine Operation **558** durchgeführt. Die Operation **558** evaluiert, ob die Summe von $c(k)$ und F außerhalb der Grenzen der Koeffizienten auf der linken und auf der rechten Seite, d.h. $c(k - 1)$ und $c(k + 1)$ ist, indem die Gleichungen $c(k - 1) \leq c(k) \leq c(k + 1)$ und $c(k - 1) \geq c(k) \geq c(k + 1)$ evaluiert werden. Wenn eine der Gleichungen wahr ist, dann wird der Koeffizient $c(k)$ in Operation **564** mit $c(k) + F$ gleichgesetzt, und es wird keine Diskontinuität eingeführt. Deshalb wird die Operation **546** erfolgreich beendet. Wenn eine der Gleichungen falsch ist, dann wird in Operation **560** eine Schleifeniteration durchgeführt.

[0157] Die Operation **560** inkrementiert die Variable des Schleifeniterators, so dass der nächste Koeffizient evaluiert werden kann. Ein Operation **562** fragt, ob alle Koeffizienten in der Gruppe(i) evaluiert wurden. Wenn nicht alle Koeffizienten in der Gruppe(i) evaluiert wurden, führt die Steuerung zur Spitze von Operation **548**, und die Prozedur wird für den nächsten Koeffizienten wiederholt. Wenn alle Koeffizienten in der Gruppe(i) evaluiert wurden, kann die Koeffizientengruppe nicht auf n Bit quantisiert werden, ohne eine inakzeptable Diskontinuität in die Koeffizientengruppe einzuführen. Deshalb wird n in einer Operation **566** inkrementiert, und Operation **546** wird beendet. Die Steuerung führt dann zurück zu dem Verfahren **526** bei Operation **534** (in [Fig. 25](#)), wo die Original-Koeffizienten auf den neuen Wert von n quantisiert werden und der Prozess wiederholt wird. In den Fällen, in denen die Operation **546** erfolgreich beendet wird, kehrt die Steuerung bei Operation **538** zu dem Verfahren **526** (in [Fig. 25](#)) zurück, und es wird die nächste Koeffizienten-Gruppe(i) evaluiert.

[0158] Die zweidimensionale Chip-Architektur **50** und die Adressierungs- und Sequenzierungs-FIFOs **90** von [Fig. 4](#) werden detaillierter mit Bezug auf die [Fig. 27](#) bis [Fig. 33](#) beschrieben. [Fig. 27](#) zeigt einen Video-Frame **600** gemäß einer Ausführungsform der vorliegenden Erfindung, der unterteilt ist in eine Anzahl von vertikalen Slices **602** für eine Slice-Abastsequenz, die beispielhaft durch eine entsprechende Anzahl von Abtastzeilen **604** dargestellt ist. Jeder Slice **602** wird einem Format ähnlich jenem abgetastet, das bei einer konventionellen Zeilenabtastsequenz verwendet wird, wobei die Abtastsequenz zum nächsten Slice übergeht, wenn das Ende eines gegebenen Slice erreicht ist. Der Vorteil dieses Formats ist, dass die Länge der Zeilenspeicher um einen Faktor reduziert ist, der annähernd gleich der Anzahl von verwendeten vertikalen Slices ist. Zeilenspeicher sind immer noch notwendig, doch sind sie jetzt viel kürzer als vorher, weshalb der Speicherbedarf auf dem Chip wesentlich reduziert ist. Wenn die Anzahl von Slices beispielsweise auf 10 festgelegt werden würde, so würde die Reduzierung des Speichers auf dem Chip eine komplette Größenordnung betragen.

[0159] Die Verwendung dieser "Slice"-Abtastorganisation führt jedoch zu Problemen. Erstens ist es häufig der Fall, dass die Verarbeitung sowohl in der horizontalen als auch vertikalen Richtung simultan erfolgen muss. Dies wiederum ergibt ein Problem an dem linken und rechten Slice-Grenzbereich, in dem horizontale Pixeldaten außerhalb des Slice möglicherweise nicht verfügbar sind. Zweitens wurde die konventionelle Zeilenabtastsequenz geändert, mit dem Ergebnis einer potenziellen Inkompatibilität mit üblichen Videoquellen und Anzeige-/Speichereinrichtungen. Diese beiden Probleme und deren Lösung durch die vorliegende Erfindung werden im Folgenden diskutiert.

[0160] [Fig. 28](#) zeigt ein Beispiel eines Slice-Kerns **606**, bei dem das Problem mit nicht verfügbaren Daten an seinem linken Rand **608** und seinem rechten Rand **610** besteht. Zu Darstellungszwecken sind nicht verfügbare Daten nur an dem linken Rand in [Fig. 28](#) gezeigt. Die Video-Verarbeitung erfordert, dass die ein gegebenes Pixel umgebenden Daten sowohl in der horizontalen als auch in der vertikalen Richtung verfügbar sind (in diesem Fall 5×5 Matrizen **612** und **614**, die an dem Pixel zentriert sind).

[0161] Die Verarbeitungsmatrix **612** liegt in der Mitte des Slice-Kerns **606**, so dass kein Problem mit der Verfügbarkeit von Daten besteht, weil diese sowohl in der horizontalen als auch in der vertikalen Richtung an allen Seiten der Verarbeitungsmatrix **612** verfügbar sind. Die Situation an dem oberen Rand **618** und unteren Rand

620 des Slice-Kerns **606**, wo Daten oberhalb des obersten Pixel und unterhalb des untersten Pixel nicht verfügbar sind, ist identisch mit der Situation bei einem konventionellen Zeilenabtast-Format. Dies kann auf vielfältige Weise gelöst werden, nämlich zum Beispiel, indem die nichtexistenten oberen/unteren Pixeldaten durch Null-Daten substituiert werden. Deshalb bereiten der obere und untere Rand **618** und **620** des Slice-Kerns keine Probleme mit Daten, die nicht verfügbar sind.

[0162] Andererseits liegt die Verarbeitungsmatrix **614** an dem linken Rand **608** der dem Slice-Kern **606** benachbarten Daten. Daher fehlen zwei Spalten von Pixeldaten **616**, weil sie außerhalb des linken Rands **608** des Slice-Kerns **606** liegen. Als Lösung für diese Situation werden Daten für diese Spalten von dem Slice unmittelbar links neben dem Slice, der gerade verarbeitet wird, zur Verfügung gestellt.

[0163] [Fig. 29](#) zeigt einen Slice **622**, der ein Paar schmaler vertikaler Slices oder "Wings" (Flügel) **624** und **626** entlang des linken und rechten Rands **608** und **610** aufweist. Der Wing **624** wurde an den Slice-Kern **606** angefügt, um die für die Verarbeitungsmatrix benötigten Pixeldaten zu liefern. Der Wing **626** wurde an den rechten Rand **610** des Slice-Kerns **606** angefügt. Durch den an den Slice **622** angefügten Wing **624** leidet die Matrix **614** nicht mehr an dem Datenmangel außerhalb des linken Rands **608** des Slice **622**.

[0164] [Fig. 30](#) zeigt eine Gesamtstruktur von überlappenden Slice/Wing-Kombinationen **628**. Der Slice **622** von [Fig. 29](#) ist als exemplarischer Slice gezeigt. Die Wings **624** und **626** des Slice **622** sind aus Daten von einem Paar benachbarter Slices jeweils links und rechts neben dem Slice **622** zusammengesetzt. Insbesondere werden die fehlenden zwei Pixelspalten in Wing **624** von den beiden äußerst rechten Spalten **630** eines Slice **632** unmittelbar links neben dem Slice **622** geliefert. Also überlappt in einer Sequenz von Slices **634** der äußerst linke Wing von Slice N den Kern von Slice N – 1, während der äußerst rechte Wing von Slice N – 1 den Kern von Slice N überlappt.

[0165] [Fig. 31](#) ist ein Flussdiagramm, das ein Verfahren **636** zur Verarbeitung von Video gemäß einer Ausführungsform der vorliegenden Erfindung zeigt. Die Eingabe in einen Video-Verarbeitungsblock ist deshalb der Slice **622** mit dem Slice-Kern **606**, dem linken Wing **624** und dem rechten Wing **626**. Der linke Wing **624** ist unterteilt in einen linken äußeren Wing **638** und einen linken inneren Wing **640**. Der rechte Wing **626** ist unterteilt in einen rechten äußeren Wing **644** und einen rechten inneren Wing **642**. In diesem Beispiel hat der Video-Verarbeitungsblock mehrere Verarbeitungsstufen, deren jede ihren eigenen Bedarf an horizontalen Pixeln beiderseits der Mitte hat.

[0166] Das Verfahren **636** nutzt eine erste Verarbeitungsstufe **646** und eine zweite Verarbeitungsstufe **650**. Die erste Verarbeitungsstufe **646** verwendet die äußeren Wings **638** und **644** und entfernt sie dann und hinterlässt einen Ausgabe-Slice **648**, der aus dem Slice-Kern **606** und den inneren Wings **640** und **642** besteht. Die zweite Verarbeitungsstufe verwendet die inneren Wings **640** und **642** und entfernt sie dann. Deshalb werden die Wings **624** und **626** bei der Verarbeitung effektiv entfernt, und die Ausgabe des Verarbeitungsblocks ist ein Slice **652**, dessen Breite gleich der Breite des Kerns **606** des Original-Slice ist.

[0167] Ein Effekt der Wings **624** und **626** ist es, den Slice-Zeilenspeicherbedarf auf dem Chip um die Breite der Wings **624** und **626** zu erhöhen. Doch ist die Wingbreite relativ zur Gesamtbreite des Slice üblicherweise gering. Die tatsächliche Slice- und Wingbreite ist abhängig von der Implementierung und von den Verarbeitungsanforderungen und der verfügbaren Bandbreite externer Speicher.

[0168] Eine bevorzugte Ausführungsform der vorliegenden Erfindung verwendet drei vertikale Video-Verarbeitungsblocks. Die erste Verarbeitungsstufe **646** erfordert ein Paar von äußeren Wings **638** und **644**, die eine Breite von 2 Pixel haben; die zweite Verarbeitungsstufe **650** erfordert ein Paar von inneren Wings **640** und **642** mit einer Breite von 4 Pixel, und die dritte Verarbeitungsstufe **652** erfordert keine Wings, da der angewandte spezielle Verarbeitungsalgorithmus keine Daten benötigt, die horizontal zu den in Verarbeitung befindlichen vertikalen Daten sind. Die gewählte Breite des Slice-Kerns betrug 36 Pixel, was eine initiale Breite des Eingabe-Slice von 48 Pixel ergab. (Kern + linker innerer Wing + rechter innerer Wing + linker äußerer Wing + rechter äußerer Wing = $36 + 4 + 4 + 2 + 2 = 48$)

[0169] Leider sind die Eingangsdaten und die Ausgangsdaten der vertikalen Verarbeitungsblöcke nicht in dem Zeilenabtastungs-Videoformat, das praktisch für alle Video-Eingabequellen und Video-Ausgabeanlagen und Speichereinrichtungen die Norm ist. Die vorliegende Erfindung umfasst eine standardisierte Eingabe/Ausgabe-Formatkonvertierung, die über die Verwendung eines externen Speichers der Video-Verarbeitungsvorrichtung erfolgt. Aus Kostengründen und wegen der Erhältlichkeit wird eine handelsübliche DRAM-Speicheranrichtung verwendet.

[0170] Je nach Art der Video-Verarbeitung, die durchzuführen ist, dienen ein oder mehrere Puffer in Field- oder Frame-Größe zusätzlich zur Konvertierung zwischen dem vollen Field/Frame-Format der Zeilenabtastung und dem Format der Slice-Abtastung auch noch anderen notwendigen Zwecken. Zum Beispiel erfordert das De-Interlacing-Verfahren üblicherweise einen (und manchmal mehrere) Field-Puffer, um mehrere Fields von Videodaten für die temporale Verarbeitung zu speichern. Puffer werden auch bei der Konvertierung der Frame-Rate benötigt, wenn die Ausgabe-Frame-Rate eine andere ist als die Eingabe-Rate. In diesem Fall können für das Verfahren zur Konvertierung der Frame-Rate mehrere Puffer für das Ausgabe-Field oder den Ausgabe-Frame erforderlich sein.

[0171] [Fig. 32](#) zeigt ein Beispiel eines Systemdiagramms für einen Videoprozessor **654** auf Slice-Basis. Ein erster Eingabe-Puffer **656**, ein zweiter Eingabe-Puffer **658**, ein erster Ausgabe-Puffer **660** und ein zweiter Ausgabe-Puffer **662** sind für das Slice-Konvertierungsverfahren notwendig. Weil Videoanwendungen normalerweise Echtzeit-Eingaben und Ausgaben erfordern und weil die Abtastverfahren für eine Zeilenabtastung und eine Slice-Abtastung verschieden sind, wird der erste Eingabe-Puffer **656** zum Speichern des Video-Eingabedatenstroms aus dem Eingangsdaten-Formatierer **664** verwendet. Der (in der vorhergehenden Field/Frame-Periode gefüllte) zweite Eingabe-Puffer **658** wird verwendet, um Daten im Slice-Abtastformat für den vertikalen Video-Verarbeitungsabschnitt **666** bereitzustellen.

[0172] Ein ähnlicher Prozess wird für die Ausgabe verwendet. Der zweite Ausgabe-Puffer **662** empfängt verarbeitete Daten im Slice-Format aus dem vertikalen Video-Verarbeitungsabschnitt **666**, während der erste Ausgabe-Puffer **660** (der in der vorhergehenden Field/Frame-Periode gefüllt wurde), für die Ausgabe von Daten in dem konventionellen Zeilenabtastformat an den Ausgangsdaten-Formatierer **668** verwendet wird. Der Ausgabedatenstrom kann übrigens Daten an zusätzliche Video-Verarbeitungsstufen liefern, die Daten nur in der horizontalen Richtung verarbeiten (z.B. horizontale Skalierung und Farbraumkonvertierung).

[0173] [Fig. 33](#) zeigt ein Systemdiagramm eines Beispiels einer Videoverarbeitungs-Chiparchitektur **670**. Die Videoverarbeitungs-Chiparchitektur **670** umfasst einen Video-Prozessor **672** und eine externe Speicherquelle **674**. Bei dieser besonderen Implementierung der Video-Verarbeitung ist ein Speicher für mehrere Eingabe-Fields (für temporale Verarbeitung) erforderlich. Videodaten werden an eine Eingabestufe **676** in dem Video-Prozessor **672** geliefert, der die redundanten Wingdaten direkt in den Video-Datenstrom addiert. Die Daten werden dann (einschließlich Wings) durch die Speicher-Steuereinheit **680**, die sich in dem Video-Prozessor **672** befindet, in einer Zeilenabtast-Sequenz in einen ersten Puffer **678** des Field-Speichers **678** in der externen Speicherquelle **674** geschrieben.

[0174] In anschließenden Field-Perioden werden Daten der Reihe nach in einen zweiten Puffer **682** des Field-Speichers, einen dritten Puffer **684** des Field-Speichers und einen vierten Puffer **686** des Field-Speichers geschrieben. Als nächstes werden die Daten aus dem zweiten, dritten und vierten Puffer **682**, **684** und **686** des Field-Speichers gelesen, wobei sich sämtliche Puffer in der externen Speicherquelle **674** befinden und parallelgeschaltet sind. Die Field-Puffer **682**, **684** und **686** speisen den vertikalen Video-Verarbeitungsabschnitt **688**, der sich in dem Video-Prozessor **672** befindet. Die Daten werden in dem vertikalen Video-Verarbeitungsabschnitt **688**, der die Wings entfernt, verarbeitet.

[0175] Die Daten werden von dem vertikalen Video-Verarbeitungsabschnitt **688** in einem Slice-Abtastformat in einen ersten Frame-Pufferbereich **690** in der externen Speicherquelle **674** zurückgeschrieben. Daten werden in einer konventionellen Zeilenabtastsequenz aus einem zweiten Frame-Pufferbereich **692** in die externe Speicherquelle **674** eingelesen, für eine Eingabe in einen horizontalen Verarbeitungsblock **694**, der sich in dem Video-Prozessor **672** befindet. Die Ausgabe des horizontalen Verarbeitungsblocks **694** ist im Zeilenabtastformat und ist die Ausgabe des Video-Prozessors **672**.

[0176] In einer bevorzugten Ausführungsform der vorliegenden Erfindung werden Video-Eingabedaten als interlaced Fields von Daten in einem 720×240 Pixel Field-Format bereitgestellt. Jedes Video-Field ist konzeptuell in 20 Slices mit einer Breite von 36 Pixel unterteilt, wobei jeder Slice einen linken und einen rechten Wing mit jeweils 6 Pixel (äußere Wings mit jeweils 2 Pixel und innere Wings mit jeweils 4 Pixel) hat. Die Wings werden an den geeigneten Punkten in dem Video-Eingabedatenstrom addiert, und der resultierende Datenstrom wird in einer Zeilenabtastsequenz in einen ersten Field-Puffer in einem externen SDRAM geschrieben.

[0177] Drei Fields von Daten werden simultan aus dem SDRAM ausgelesen. Die Daten für diese Fields kommen aus dem zweiten, dritten und vierten Field-Puffer und werden in vertikalen Slices gelesen, die 48 Pixel (Slice-Kern und Wings) in der Breite mal 240 Reihen betragen. Die Daten werden durch eine erste vertikale Verarbeitungsstufe verarbeitet, die Daten im Slice-Abtastformat mit der zweifachen Eingaberate eines einzel-

nen Fields an eine zweite Stufe liefert. Die Slice-Daten, die in die zweite Stufe eingegeben werden, werden in Slices von 44 Pixel Breite mal 480 Reihen (wegen des Raten-Verdopplungsvorgangs der ersten Stufe) formatiert. Die zweite vertikale Verarbeitungsstufe verarbeitet die Daten und liefert Daten in einem 36 Pixel breiten Slice-Abtastformat mit der gleichen Rate wie die der Eingabe in diese Stufe an eine dritte vertikale Verarbeitungsstufe.

[0178] Die dritte Stufe ist ein vertikaler Skalierer und führt keine horizontale Verarbeitung durch und erfordert daher keine Wings an den Slice-Formatdaten. Die Daten werden von der dritten Verarbeitungsstufe in einem 36 Pixel breiten Slice-Abtastformat an einen ersten Frame-Pufferbereich in dem SDRAM ausgegeben. Die Anzahl von Reihen in jedem Slice ist abhängig von dem gewählten speziellen vertikalen Skalierverhältnis. Daten werden in eine nur horizontale Verarbeitungsstufe in einem üblichen Zeilenabtastformat von $720 \times 480 \cdot M$ Pixel eingegeben, wobei M der vertikale Skalierungsfaktor in der dritten vertikalen Verarbeitungsstufe ist. Diese Daten werden durch den Horizontal-Prozessor (der einen Horizontal-Skalierer umfasst) verarbeitet und in einem üblichen Zeilenabtastformat mit einer Auflösung von $720 \cdot N \times 480 \cdot N$ ausgegeben, wobei N der horizontale Skalierungsfaktor ist.

[0179] Insgesamt resultiert diese Implementierung aufgrund der Slice-Abtast-Architektur in einer Reduzierung des Speicherbedarfs auf dem Chip, die größer als $10 \times$ ist. Diese Kosteneinsparung durch die Reduzierung des Speicherbedarfs auf dem Chip gleicht den zusätzlich benötigten externen Speicher mehr als aus und schafft eine Vielfalt von Prototyp- und Produktionsoptionen.

[0180] Der Video-Ausgabeprozessor **60** von [Fig. 4](#) wird detaillierter mit Bezug auf die [Fig. 34](#) bis [Fig. 38](#) erläutert. [Fig. 34](#) ist ein Diagramm einer asynchronen Schnittstelle **700** gemäß einer Ausführungsform der vorliegenden Erfindung. Eine Datenquelle **702** liefert Daten, die mit dem Takt C1 in einem Eingabe-Datenstrom **704** synchron sind, an einen Synchronisierer **705**. Sobald die Daten von dem Synchronisierer **705** empfangen wurden, organisiert eine Schreibsteuer-Logik (WCL) **706** das Schreiben dieser Daten in einen von einem ersten RAM-Puffer **708**, einem zweiten RAM-Puffer **710** und einem dritten RAM-Puffer **712**, indem sie Steuer- und Adresssignale an jeden dieser RAM-Puffer **708**, **710** und **712** liefert.

[0181] Sobald die Daten in die RAM-Puffer **708**, **710** und **712** geschrieben sind, organisiert eine Lesesteuer-Logik (RCL) **714** das Lesen von Daten aus den RAM-Puffern **708**, **710** und **712**, indem sie Adress- und Steuersignale an die RAM-Puffer **708**, **710** und **712** liefert. Die wieder aufgefundenen Daten werden dann in einem kontinuierlichen Datenausgabestrom **716** mit der Takt-Rate C2 von dem Synchronisierer **705** an einen Datenzielort **718** ausgegeben.

[0182] Das Auslesen von Daten aus einem gegebenen RAM-Puffer **708**, **710** oder **712** findet nicht statt, ehe der bestimmte RAM-Puffer **708**, **710** oder **712** von der WCL-Einheit **706** gefüllt wurde. Nachdem einer der RAM-Puffer **708**, **710** oder **712** gefüllt wurde, wird der nächste RAM-Puffer **708**, **710** oder **712** beschrieben, und es können Daten aus dem(den) vorher aufgefüllten RAM-Puffer(n) **708**, **710** und/oder **712** gelesen werden. Wenngleich diese besondere Ausführungsform mit drei RAM-Puffern arbeitet, versteht sich, dass abhängig von den Systemanforderungen auch mehr RAM-Puffer verwendet werden können.

[0183] Eine Synchronisationslogik-Einheit **720** außerhalb des Synchronisierers **705** koordiniert das Lesen und Schreiben von Daten. Optional könnte die Synchronisationslogik **720** Teil des Synchronisierers **705** selbst sein. Es können mehrere Synchronisationsschemen implementiert sein. Zum Beispiel könnte die Synchronisationslogik **720** der WCL-Einheit **706** und der Datenquelle **702** signalisieren, wann mit einem Datentransfer zu beginnen ist. Eine vorgegebene Periode später, wenn der erste RAM-Puffer **708** aufgefüllt ist und der zweite RAM-Puffer **710** gerade aufgefüllt wird, weist die Synchronisationslogik **720** die RCL-Einheit **714** an, mit dem Lesen von Daten aus dem ersten RAM-Puffer **708** zu beginnen und die Daten an den Datenzielort **718** zu liefern.

[0184] Ein alternatives Schema könnte einen mehr oder weniger freien Betrieb des Datentransfers vorsehen. Die WCL-Einheit **706** würde Daten aus der Datenquelle **702** nach Verfügbarkeit akzeptieren, würde diese Daten in den nächsten verfügbaren RAM-Puffer **708**, **710** oder **712** schreiben und der RCL-Einheit **714** signalisieren, dass der betreffende Puffer voll ist. Das Signal "Puffer voll" muss von dem Takt-Bereich C1 auf den Takt-Bereich C2 synchronisiert werden, was üblicherweise durch einen Zweistufen-Synchronisierer bewerkstelligt wird. Die RCL-Einheit **714** könnte den Zustand des Signals "Puffer voll" entweder halten oder ein Quittungssignal (das von dem Takt-Bereich C1 auf den Takt-Bereich C2 synchronisiert sein muss) zurück zur WCL-Einheit **706** leiten und anzeigen, dass entweder das Signal "Puffer voll" empfangen wurde oder dass der RAM-Puffer **708**, **710** oder **712** geleert wurde.

[0185] [Fig. 35](#) ist ein Diagramm einer alternativen asynchronen Schnittstelle **722**. Der Datentransfer wird über ein externes Signal an die WCL-Einheit **724** signalisiert, die anzeigt, dass ein Datentransfer zu beginnen hat. Die mit dem Takt C1 synchrone WCL **724** erzeugt Schreibfreigaben und Adressen für einen ersten RAM-Puffer **726** mit einem einzigen Port, einen zweiten RAM-Puffer **728** mit einem einzigen Port und einen dritten RAM-Puffer **730** mit einem einzigen Port. Die jeweils mit einem einzigen Port versehenen RAM-Puffer **726**, **728** und **730** haben eine synchrone Lese- und eine asynchrone Schreibfähigkeit.

[0186] Es ist zu einer Zeit immer nur eine Schreibfreigabe aktiv, wobei die Schreibfreigabe zuerst für den RAM-Puffer **726** erteilt wird, gefolgt von jener für den RAM-Puffer **728**, dann den RAM-Puffer **730**, und schließlich wieder für den RAM-Puffer **726** usw. An dem Punkt, an dem der RAM-Puffer **728** voll ist und der RAM-Puffer **730** etwa halbvoll ist, wird ein externes Signal an die RCL-Einheit **732** positiv, um das Lesen der Daten aus den RAM-Puffern **726**, **728** und **730** zu initiieren. Die mit dem Takt C2 synchrone RCL-Einheit **732** erzeugt Adressen und Datenauswahlleitungen für die RAM-Puffer **726**, **728** und **730** für Leseoperation.

[0187] Individuelle RAM-Puffer **726**, **728** und **730** erhalten eine Lese- oder Schreibadresse auf Grundlage dessen, in welchen der RAM-Puffer **726**, **728** oder **730** aktuell eingelesen wird, d.h. ein RAM-Puffer **726**, **728** oder **730**, in den gerade eingelesen wird, erhält seine Adresse von der WCL-Einheit **724**, während die anderen beiden RAM-Puffer **726**, **728** und **730** ihre Adressen von der RCL-Einheit **732** erhalten. Ein Daten-MUX **734**, der durch die RCL-Einheit **732** gesteuert wird, wählt den Ausgabedatenstrom aus einem der drei RAMs, die gerade ausgelesen werden. Obwohl in diesem Beispiel externe Signale den Lese- und Schreibvorgang des Datentransfers starten, kann eine optionale Synchronisationslogik-Einheit **736** vorhanden sein, sollten eines oder mehrere der externen "Start"-Signale fehlen.

[0188] [Fig. 36](#) ist ein Diagramm einer 3-Puffer-Synchronisierer-Sequenz **738**, die die Sequenzierung und die Beziehungsverhältnisse von Lese- und Schreiboperation zu den drei RAM-Puffern darstellt. Potenzielle Takt-synchronisationsprobleme, die zur Echtzeit-Puffersequenzierung für den kontinuierlichen Datenausgabestrom gehören, werden durch die Tatsache gemildert, dass Lese- und Schreiboperationen durch einen Skew/eine Divergenz von etwa 1½ RAM-Puffern voneinander getrennt sind.

[0189] Der Skew **739** sorgt für eine Pufferregion von etwa einer Hälfte eines RAM-Puffers, die potenzielle Kollisionen zwischen Lese- und Schreiboperationen in demselben RAM-Puffer zur selben Zeit separiert. Diese Pufferregion ist vorhanden, um Ungenauigkeiten in den Befehlen "starte Schreiben" und "starte Lesen" Variationen in Perioden einer Datenpause in dem Eingabedatenstrom, und Abweichungen in der Zeitsteuerung aufgrund von Synchronisationsverzögerungen und/oder Verschiebungen zwischen zwei Takt-Bereichen zu absorbieren.

[0190] [Fig. 37](#) ist ein Flussdiagramm eines Verfahrens **740** für eine Sequenzierung durch RAM-Adressen und Module gemäß einer Ausführungsform der vorliegenden Erfindung. Das Verfahren **740** beginnt bei einer Operation **742**, in der die RAM-Adresse sowohl für die Schreib- als auch Leseoperation auf Null gesetzt wird, ebenso wie der ausgewählte RAM-Puffer. Als nächstes fragt eine Operation **744**, ob die Daten gültig sind. Wenn die Antwort Nein lautet, wiederholt sich die Operation **744** selbst, bis Daten gültig sind. Lautet die Antwort Ja, wurde das letzte RAM-Modul erreicht, und eine Operation **748** inkrementiert, um das nächste RAM-Modul auszuwählen, bevor eine Operation **750** durchgeführt wird. Wenn die Antwort aus Operation **746** Nein lautet, inkrementiert die Operation **750** die RAM-Adresse.

[0191] Als nächstes führt das Verfahren **740** weiter zu einer Operation **752**, die bestimmt, ob die letzte RAM-Adresse erreicht wurde. Wenn ja, wird EndCnt in einer Operation **754** gleich 1 gesetzt. Wurde die letzte RAM-Adresse nicht erreicht, wird EndCnt in einer Operation **756** gleich 0 gesetzt. Von beiden Operationen **754** und **756** kehrt das Verfahren **740** in einer Schleife zurück zu Operation **744**, wo der gesamte Ablauf erneut startet. Hier sei bemerkt, dass die Lese- und Schreiboperationen zwar der gleichen Basis-Sequenz folgen, dass sie aber in der Zeit um etwa 1½ RAM-Puffer divergieren, wie vorstehend erwähnt wurde.

[0192] [Fig. 38](#) ist ein Diagramm eines Synchronisierers **758** gemäß einer Ausführungsform der vorliegenden Erfindung zur Verwendung bei einer Video-Skalieranwendung. Die Eingabequelle erzeugt einen 8 Bit breiten Eingabedatenstrom **760** für den Synchronisierer **758**. Der Eingabedatenstrom **760** läuft bei einer Taktrate von 732 MHz (C1) mit einer durchschnittlichen Datenrate von 48 MHz. Es werden RAM-Puffer **762**, **764** und **766** verwendet, die 8 Bit breit und 16 Elemente lang sind. Eine WCL-Einheit **768** erzeugt drei RAM-Schreiberlaubnis-Signale und eine 4-Bit-RAM-Adresse.

[0193] Ein Video-Skalierer produziert ein Signal "Daten gültig", das anzeigt, wenn gültige Pixeldaten an dem

Eingabedaten-Port verfügbar sind. Immer dann, wenn die Daten an diesem Port gültig sind, werden sie in die RAM-Puffer **762**, **764** und **766** geschrieben. Die WCL-Einheit **768** führt diese Schreiboperation wie vorstehend beschrieben durch. Die WCL-Einheit **768** ist aus einem ersten Zähler **770**, der von 0 bis 15 zählt und die RAM-Adresse erzeugt, und einem zweiten Zähler **772** (ein Ringzähler), der von 0 bis 2 zählt und die RAM-Auswahl/Schreibfreigabe-Leitungen herstellt, zusammengesetzt.

[0194] Etwa an dem Punkt, an dem eineinhalb RAM-Puffer gefüllt wurden, wird ein externes Signal empfangen, das anzeigen, dass die Datenausgabe beginnen sollte. Eine RCL-Einheit **774** beginnt dann, eine Reihe von Adressen an die RAM-Puffer **762**, **764** und **766** zu generieren, um die vorher unter der Steuerung der WCL-Einheit **768** eingelesenen Daten zu extrahieren. Die RCL-Einheit **774** ist ebenfalls aus einem dritten Zähler **776**, der von 0 bis 15 zählt und eine RAM-Adresse erzeugt, und einem vierten Zähler **778**, der von 0 bis 2 zählt und die Datenausgabe durch einen MUX **780** herstellt, zusammengesetzt.

[0195] Der MUX **780** produziert einen Ausgabedatenstrom **782**, der kontinuierlich sein muss, da er den Echtzeit-Datenstrom für eine Anzeige mit Zeilenabtastung liefert. Jede Abtastzeile von Videodaten hat eine Länge von 1280 8-Bit-Daten, und es muss ein kontinuierlicher Datenstrom dieser Länge für jede Ausgabe-Abtastzeile erzeugt werden. Eine Logik in der Synchronisierer-Schaltung erkennt das Ende einer horizontalen Abtastzeile und setzt die WCL-Einheit **768** und die RCL-Einheit **774** am Ende einer Zeile in Vorbereitung auf die nächste Zeile von Videodaten zurück.

[0196] Während die Erfindung anhand mehrerer bevorzugter Ausführungsformen beschrieben wurde, wird der Fachmann nach Lektüre der vorstehenden Beschreibung und dem Studium der Zeichnungen erkennen, dass verschiedene Änderungen, Hinzufügungen, Permutationen und Äquivalente davon möglich sind, ohne den Rahmen der Erfindung zu verlassen.

Patentansprüche

1. Digitaler Bildverbesserer (**30**) umfassend:

einen De-Interlace-Prozessor (**50**), der für mehrere Formate eines interlaced Videostroms empfänglich ist und der wirksam ist, einen de-interlaced Videostrom bereitzustellen, umfassend:

- einen ersten De-Interlacer (**70**), der wirksam ist, um progressive Frames des interlaced Videostroms zu analysieren, um einen für den interlaced Videostrom verwendeten Originalquellen-Typ und eine verwendete Originalquellen-Sequenzierung zu bestimmen, und der weiterhin wirksam ist, unter Anwendung eines Konvertierungsverfahrens, das von der Detektion des Originalquellen-Typs und der Originalquellen-Sequenzierung abhängig ist, den interlaced Videostrom in einen de-interlaced Videostrom zu konvertieren; und
- einen zweiten De-Interlacer (**80**), der wirksam ist, um Bewegungs-Artefakte, die durch eine Frequenzanalyse des interlaced Videostroms detektiert werden, zu reduzieren; und
- eine Videoausgabeprozessor (**60**)-Einrichtung, die für den de-interlaced Videostrom empfänglich ist und die wirksam ist, einen skalierten de-interlaced Videostrom bereitzustellen.

2. Digitaler Bildverbesserer nach Anspruch 1, wobei der zweite De-Interlacer wirksam ist, um diagonale Merkmale/Strukturen zu detektieren und um die detektierten diagonalen Merkmale/Strukturen zu glätten.

3. Digitaler Bildverbesserer nach Anspruch 1, wobei der De-Interlace-Prozessor den de-interlaced Videostrom in vertikalen Slices verarbeitet.

4. Digitaler Bildverbesserer nach Anspruch 1, wobei der Videoausgabe-Prozessor wirksam ist, um den de-interlaced Videostrom zu skalieren, um ein Videoanzeige-Ausgabeformat eines Videoausgabestroms zu modifizieren.

5. Digitaler Bildverbesserer nach Anspruch 1, wobei der Videoausgabe-Prozessor einen Datenraten-Synchronisierer zwischen einer ersten Datenrate des de-interlaced Videostroms und einer zweiten Datenrate eines Videoausgabestroms aufweist.

6. Verfahren zur Verarbeitung von digitalem Video, umfassend das De-Interlacing von mehreren Formaten eines interlaced Videostroms mit einem Videoprozessor mittels der folgenden Schritte:

- Analysieren von progressiven Frames des interlaced Videostroms, um einen für den interlaced Videostrom verwendeten Originalquellen-Typ und eine verwendete Originalquellen-Sequenzierung zu bestimmen,
- Konvertieren des interlaced Videostroms in einen de-interlaced Videostrom unter Anwendung eines Konvertierungsverfahrens, das von der Detektion des Originalquellen-Typs und der Originalquellen-Sequenzierung

abhängig ist;

- Reduzieren von Bewegungs-Artefakten, die durch eine Frequenzanalyse des interlaced Videostroms detektiert werden; und
- Empfangen des de-interlaced Videostroms, um einen skalierten de-interlaced Videostrom bereitzustellen.

7. Verfahren zur Verarbeitung von digitalem Video nach Anspruch 6, wobei die De-Interlace-Verfahren die Verarbeitung des interlaced Videostroms in vertikalen Slices umfassen.

8. Verfahren zur Verarbeitung von digitalem Video nach Anspruch 6, wobei das Skalieren ein horizontales Skalieren des interlaced Videostroms umfasst.

9. Verfahren zur Verarbeitung von digitalem Video nach Anspruch 6, wobei das Skalieren einen Datenraten-Synchronisierer zwischen einer ersten Datenrate des de-interlaced Videostroms und einer zweiten Datenrate eines Videoausgabestroms umfasst.

Es folgen 40 Blatt Zeichnungen

Anhängende Zeichnungen

10 ↗

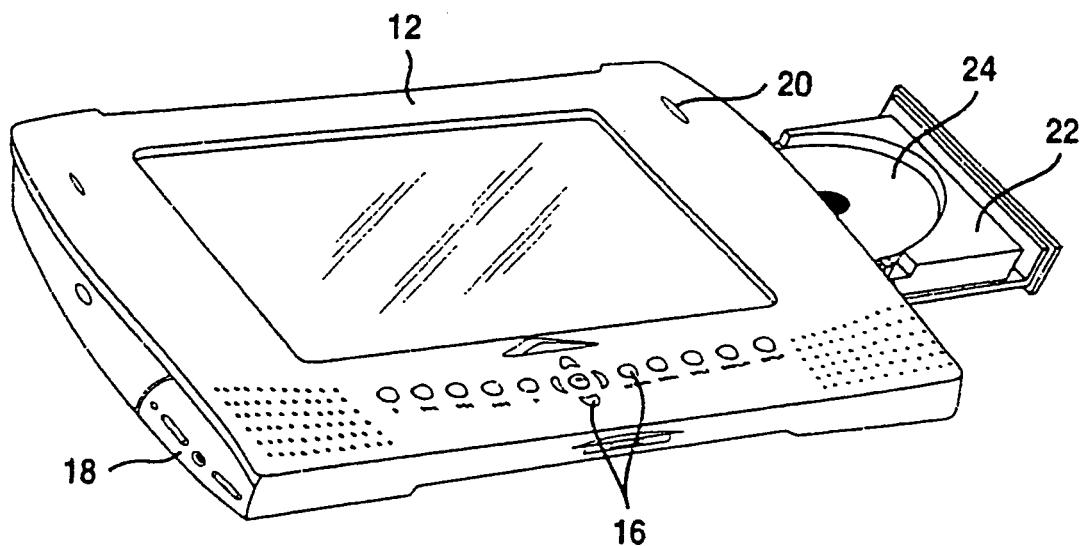


Fig. 1A

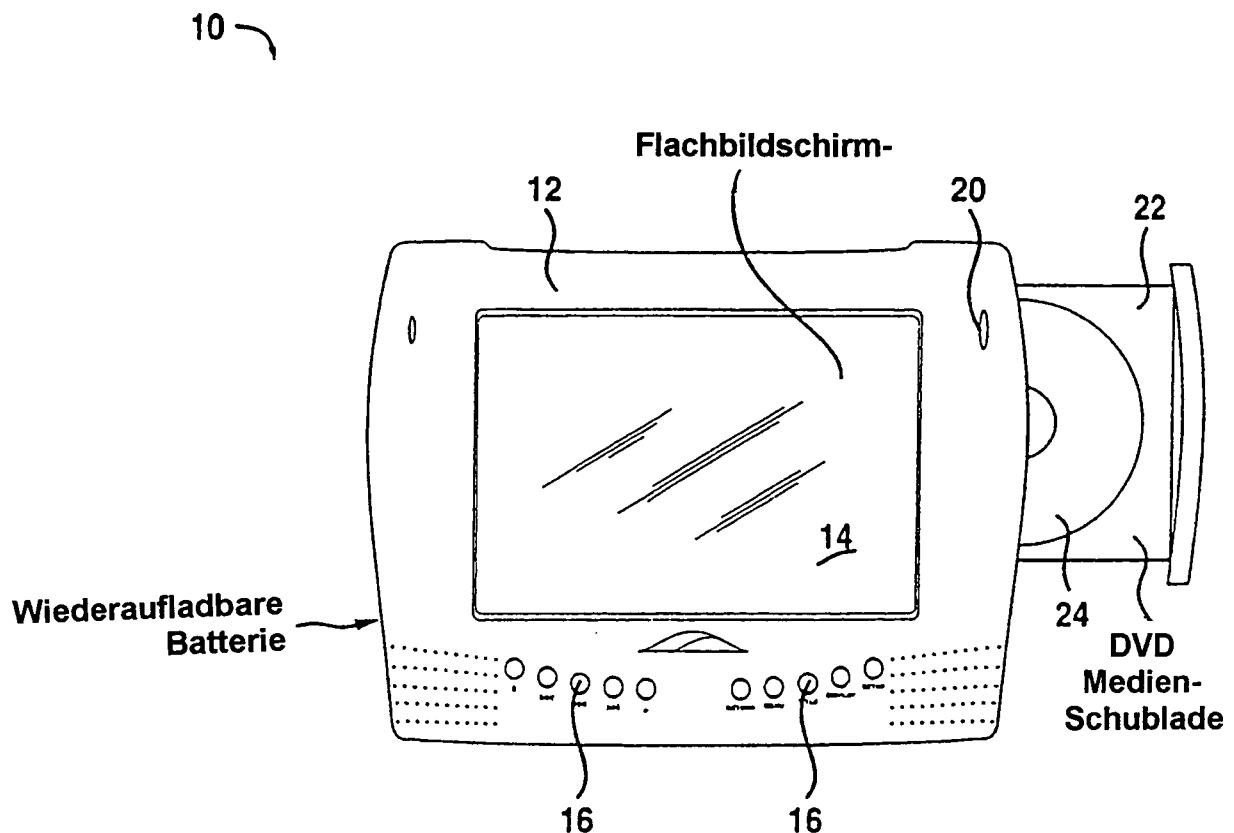


Fig. 1B

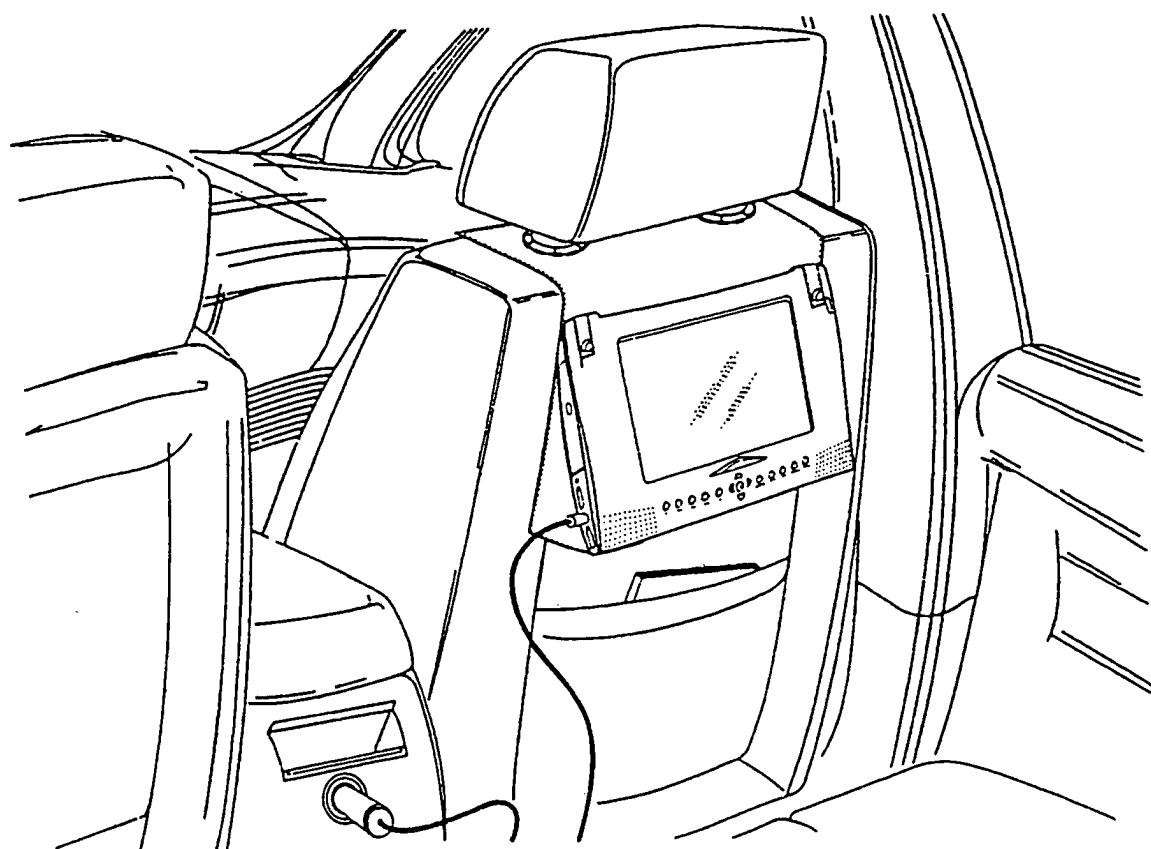


Fig. 2A

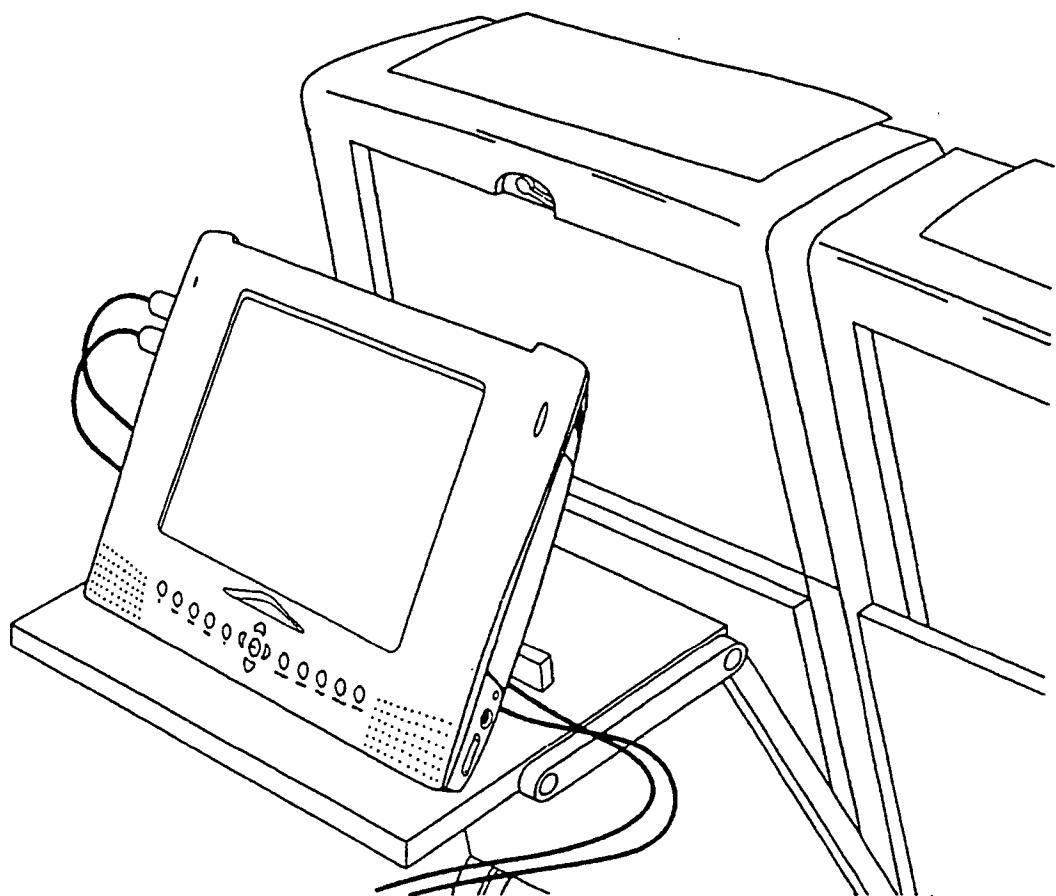


Fig. 2B



Fig. 2C

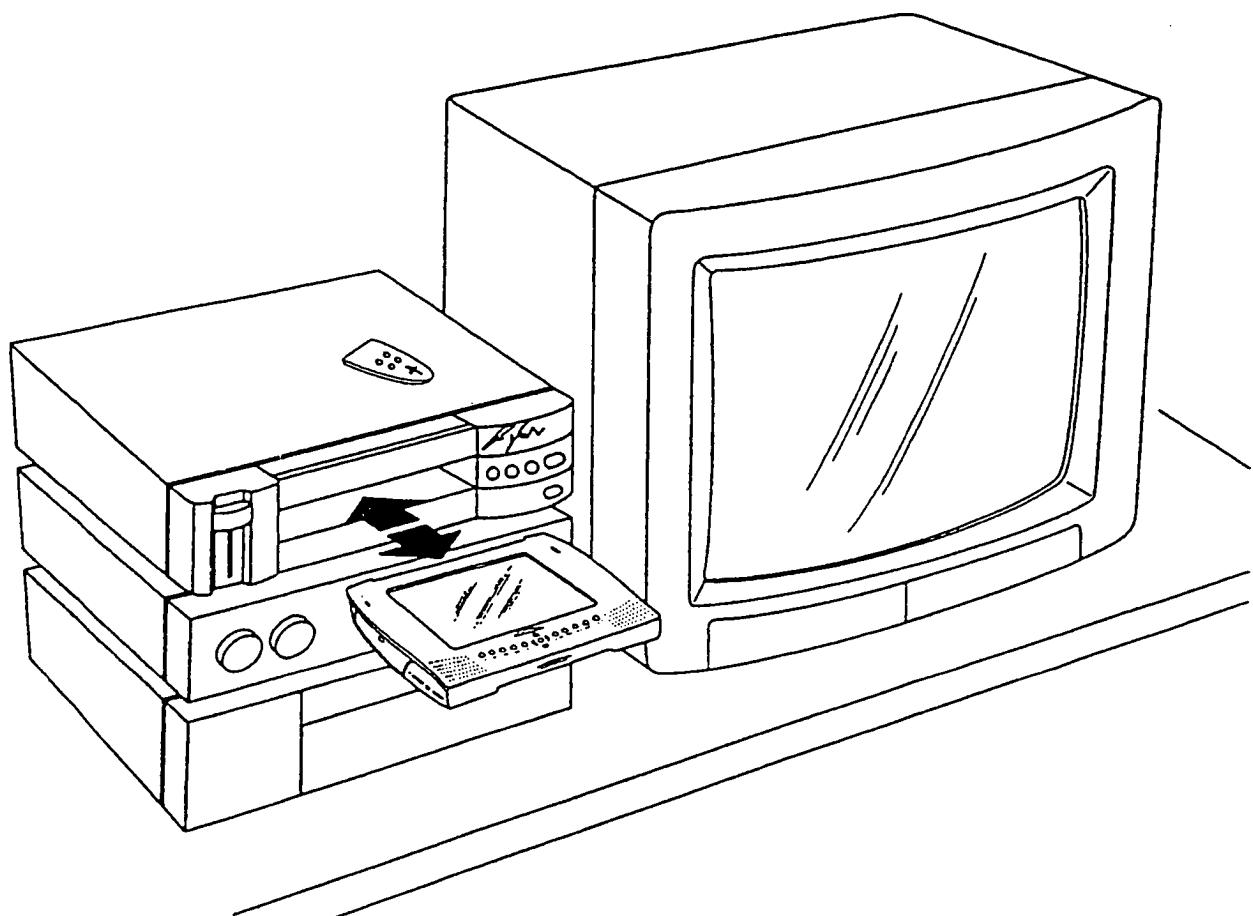


Fig. 2D

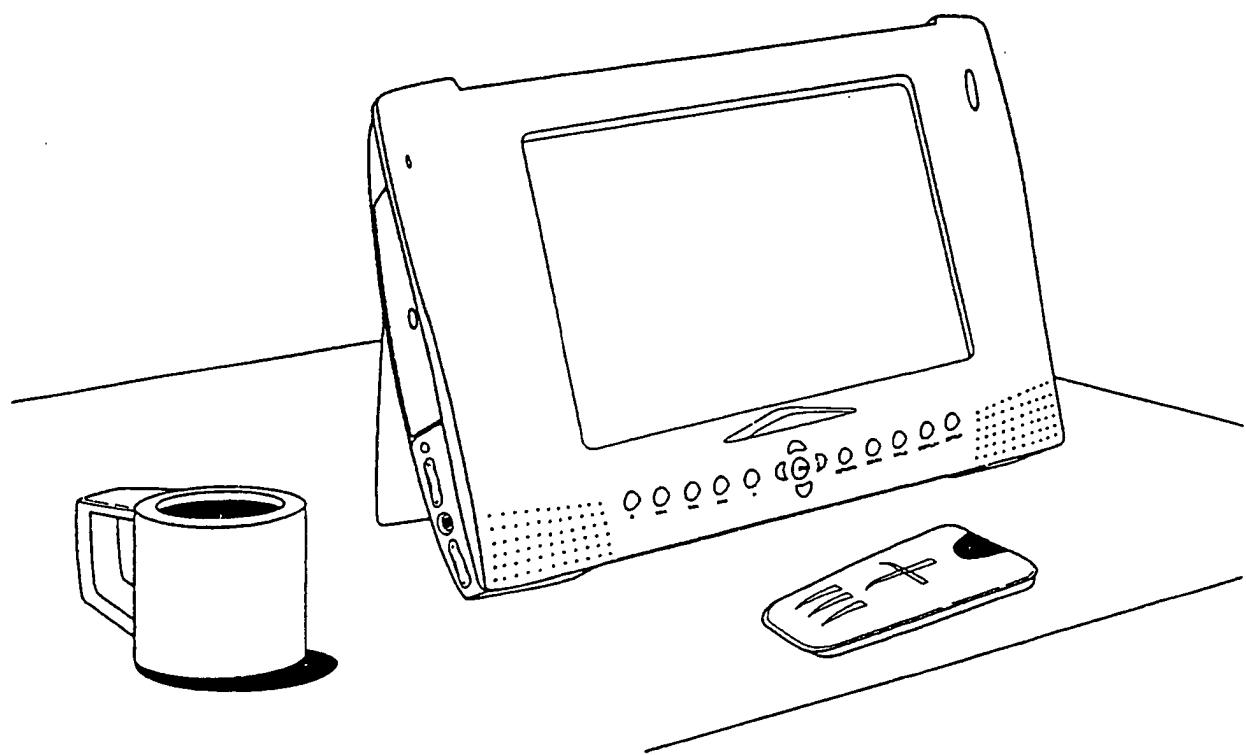
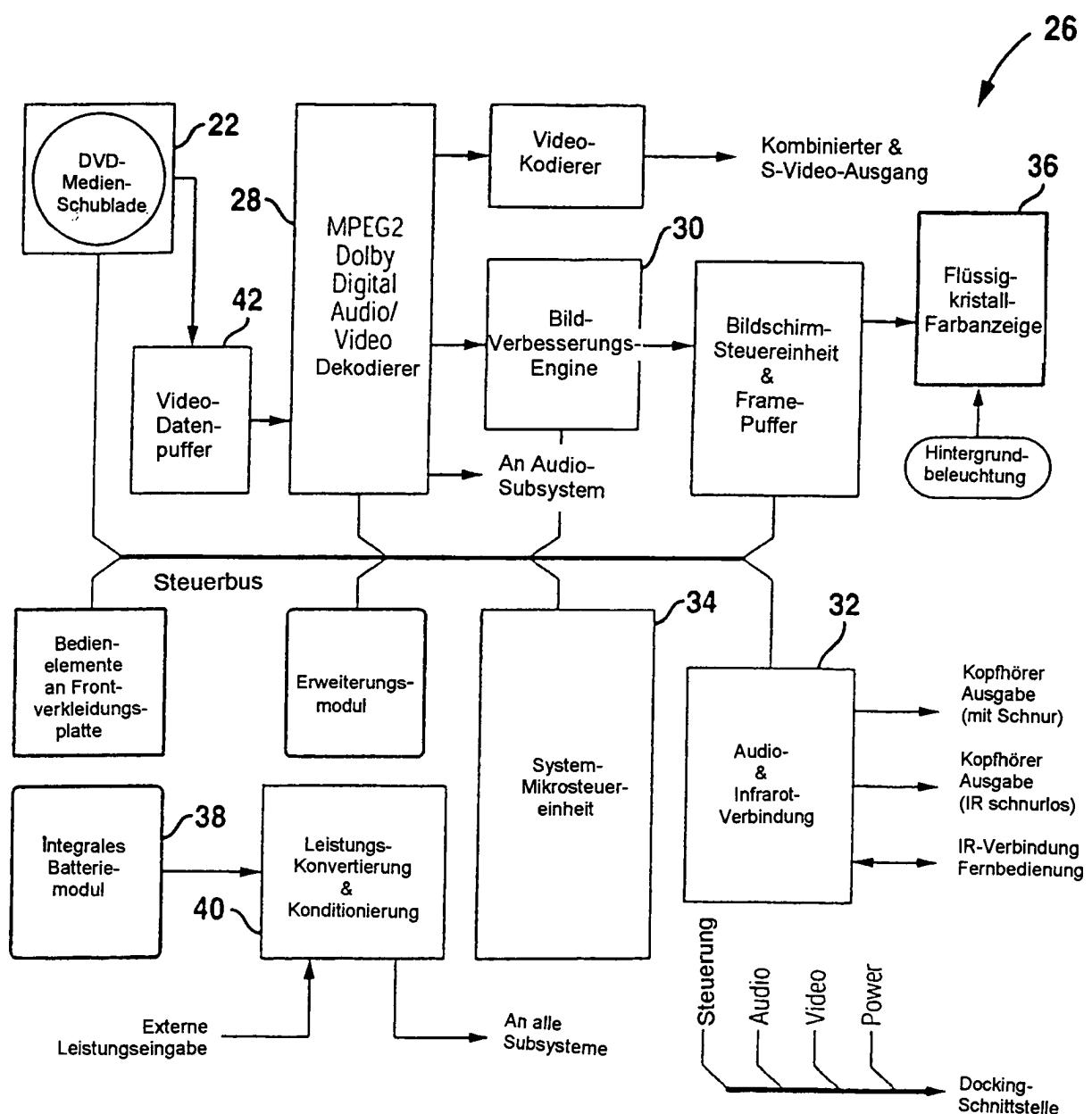


Fig. 2E

**Fig. 3**

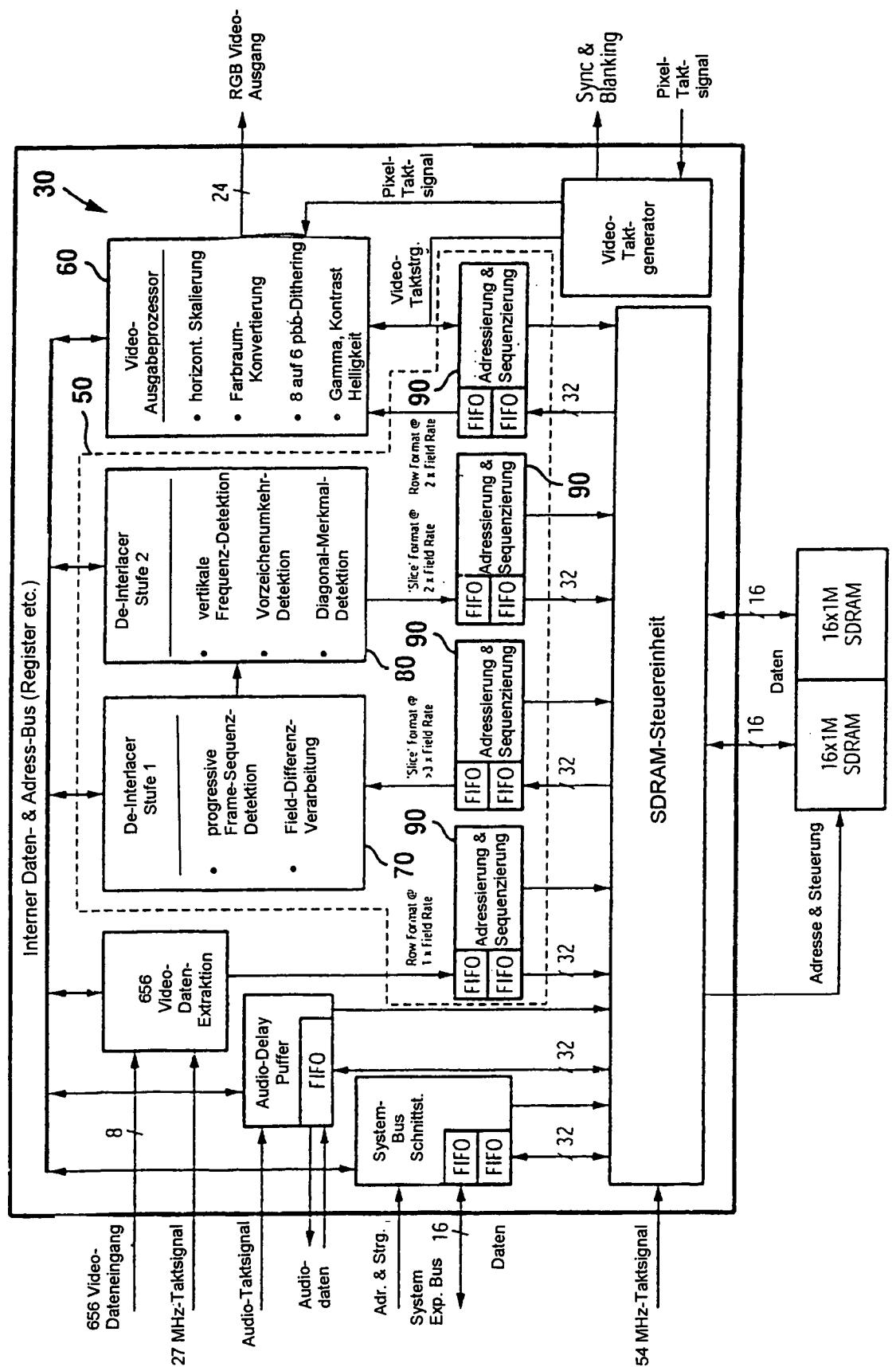
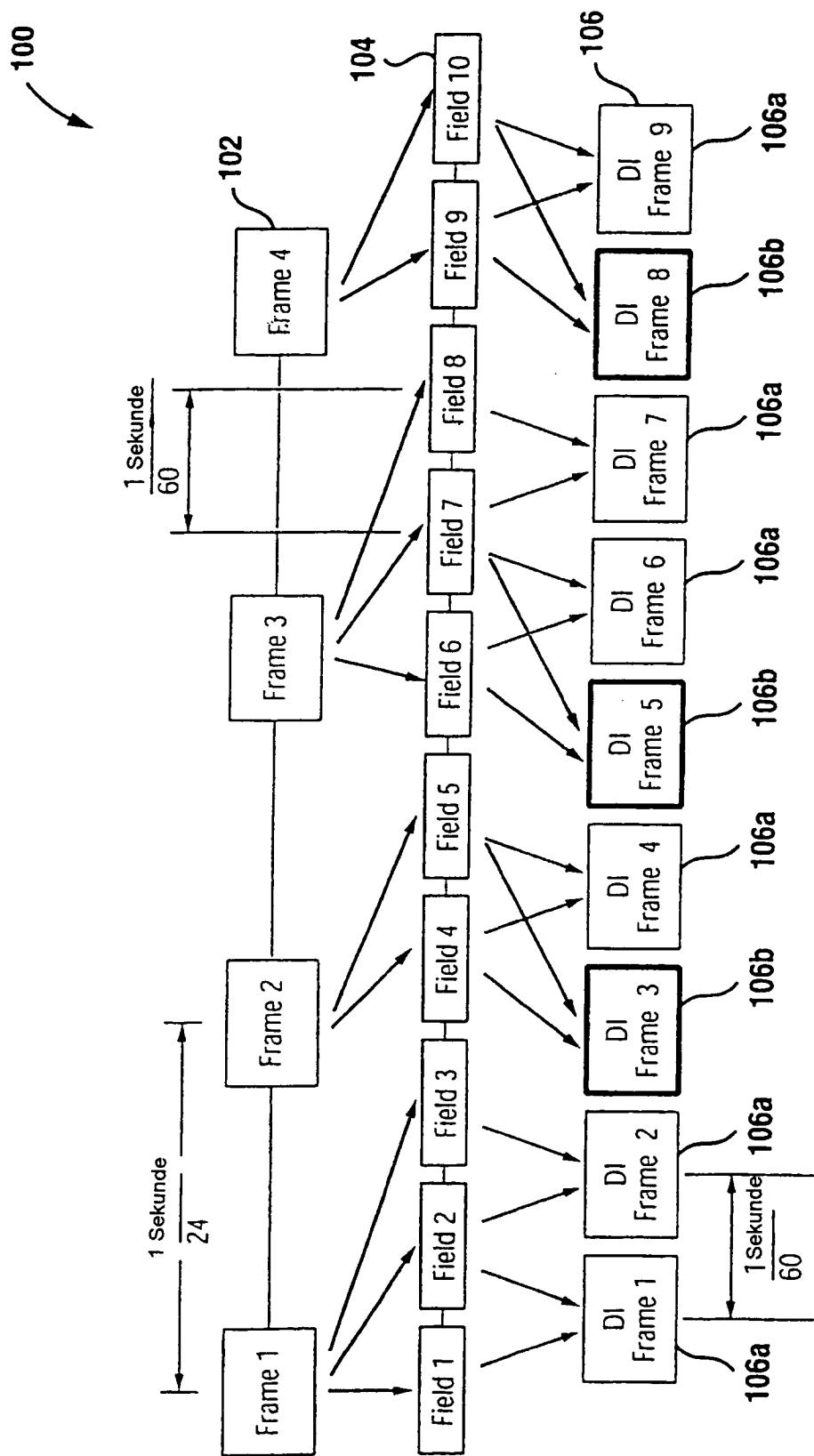


Fig. 4

**Fig. 5**

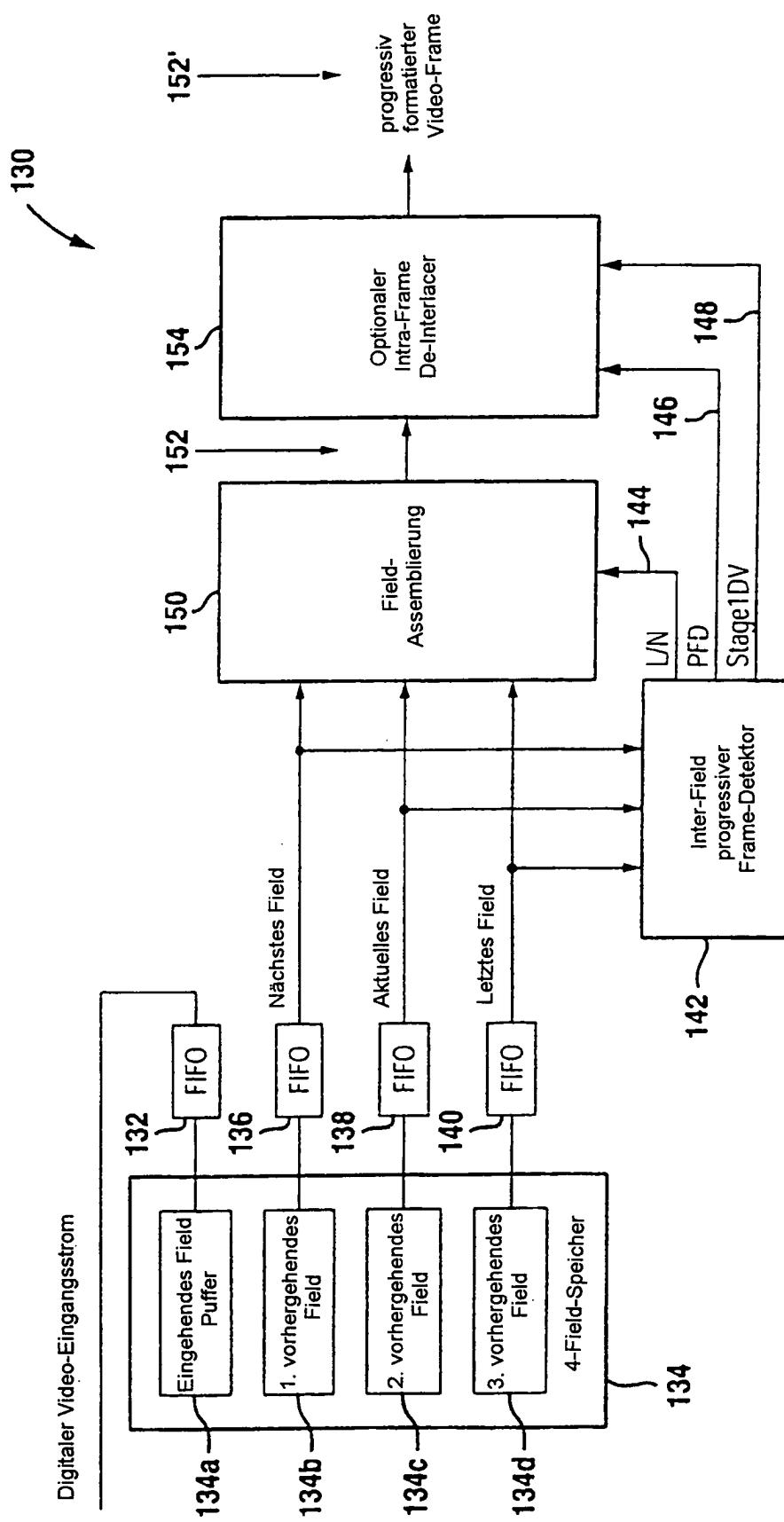


Fig. 6

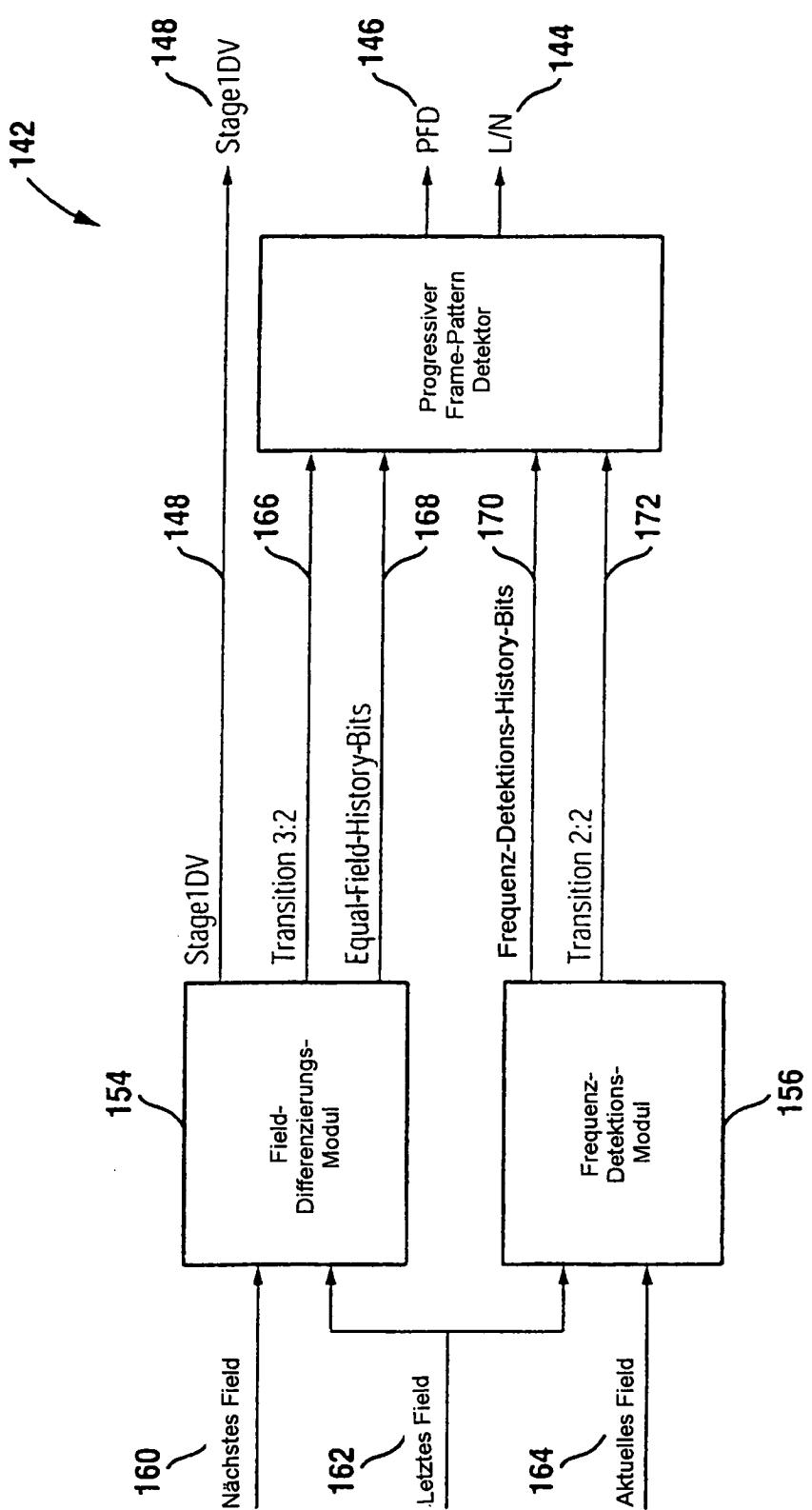
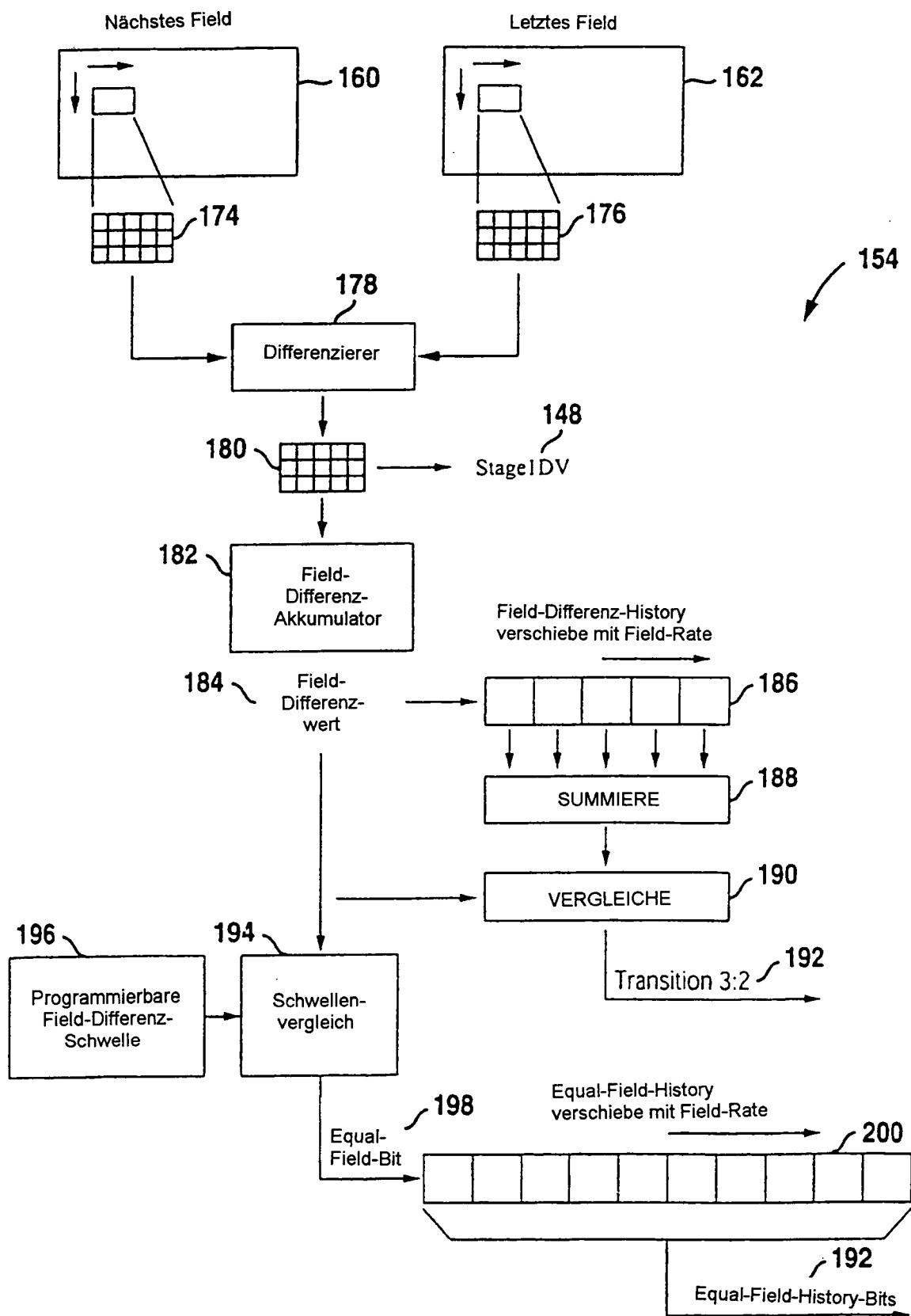
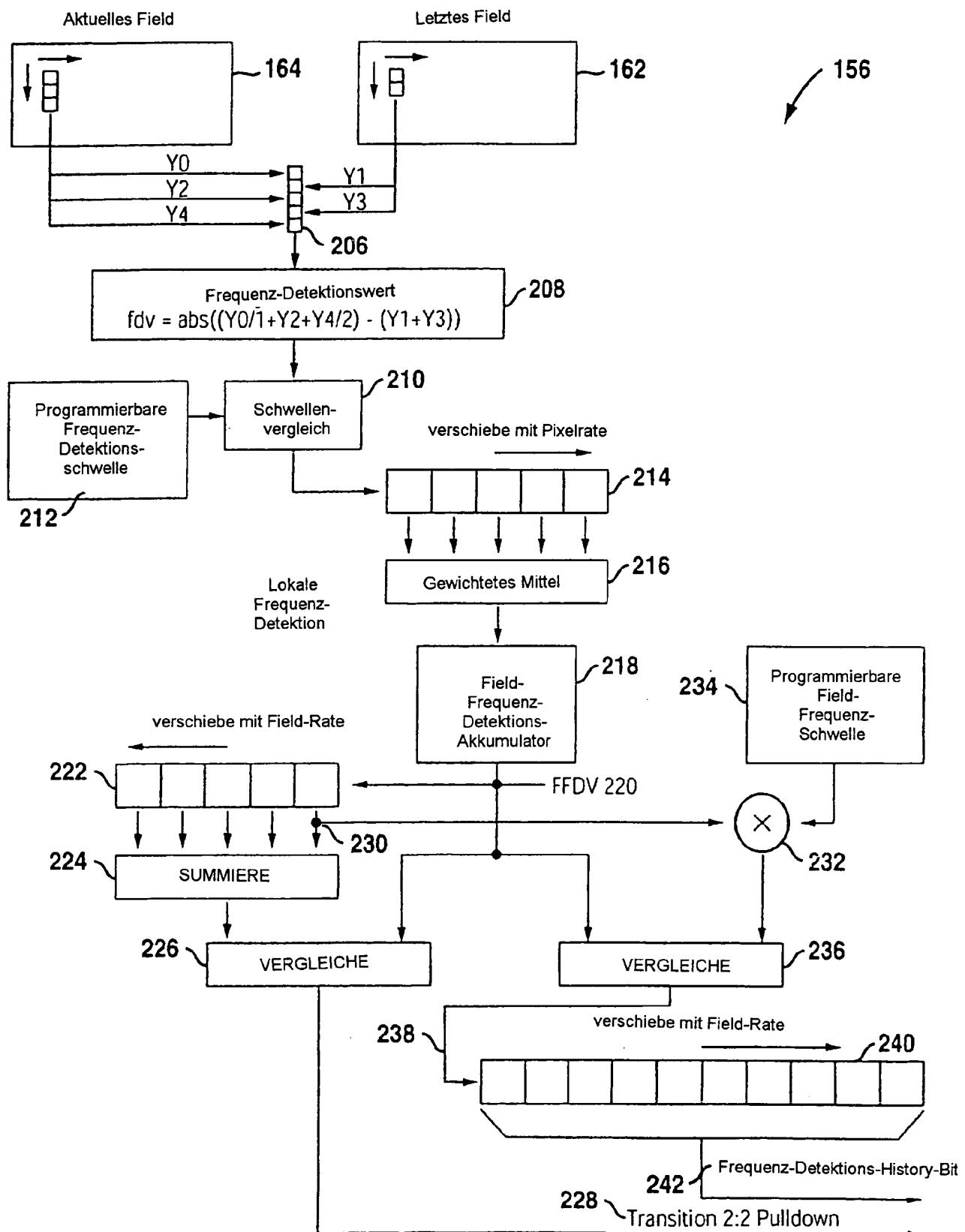


Fig. 7

***Fig. 8***

**Fig. 9**

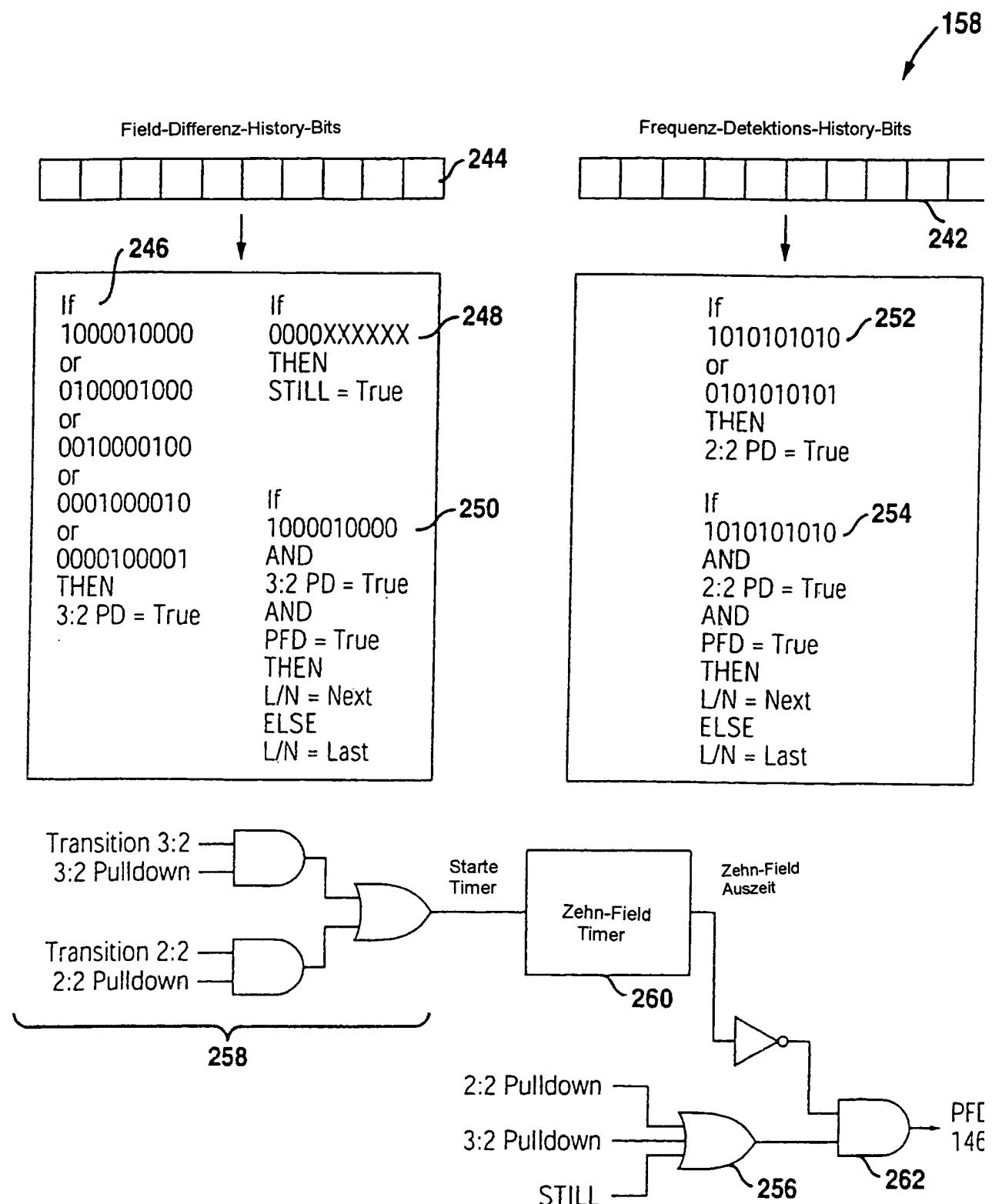


Fig. 10

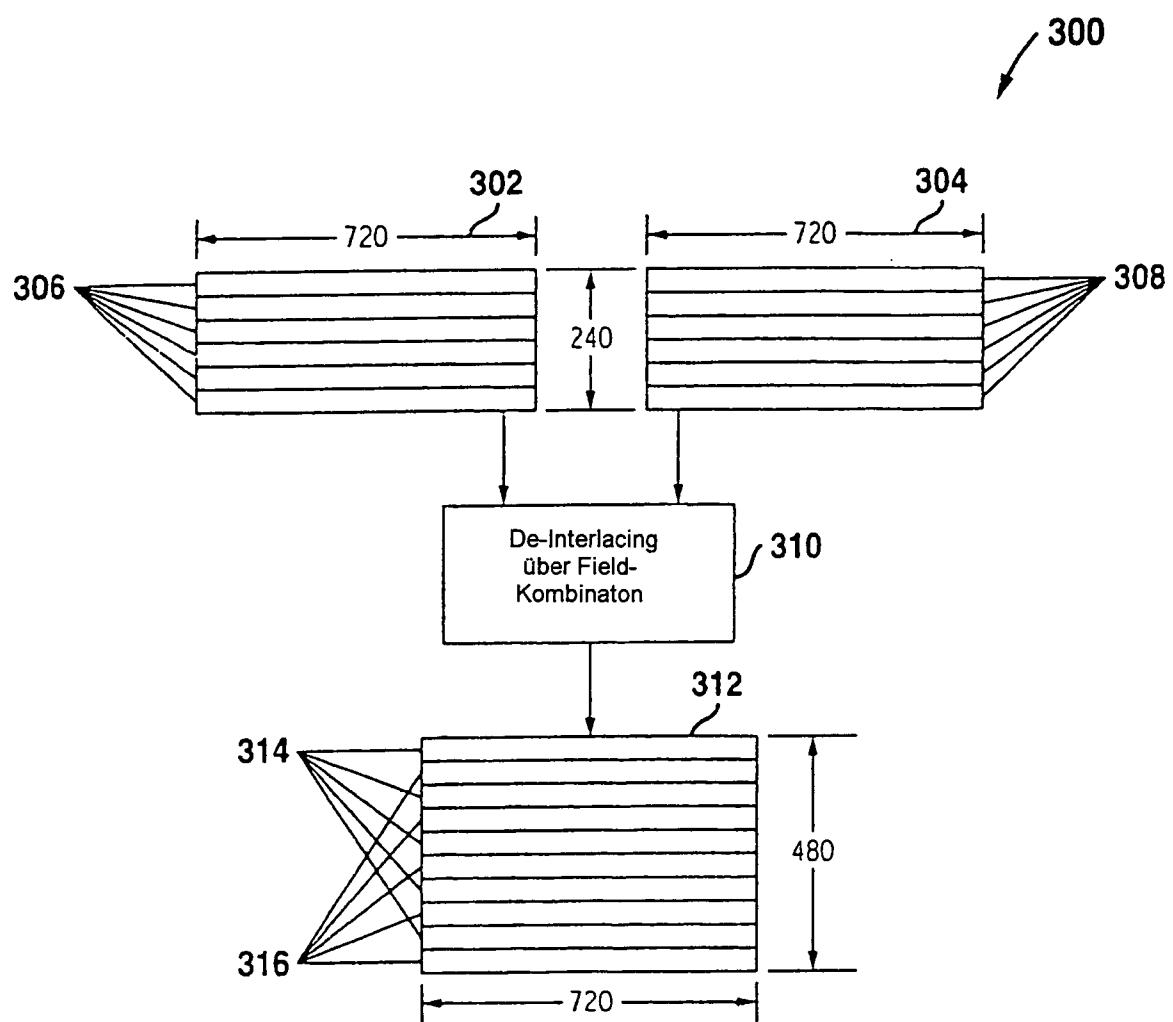


Fig. 11

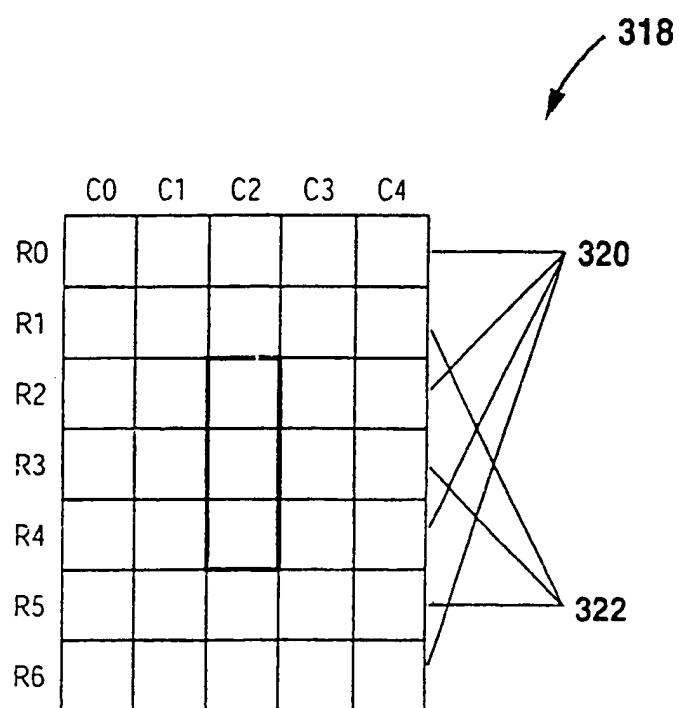
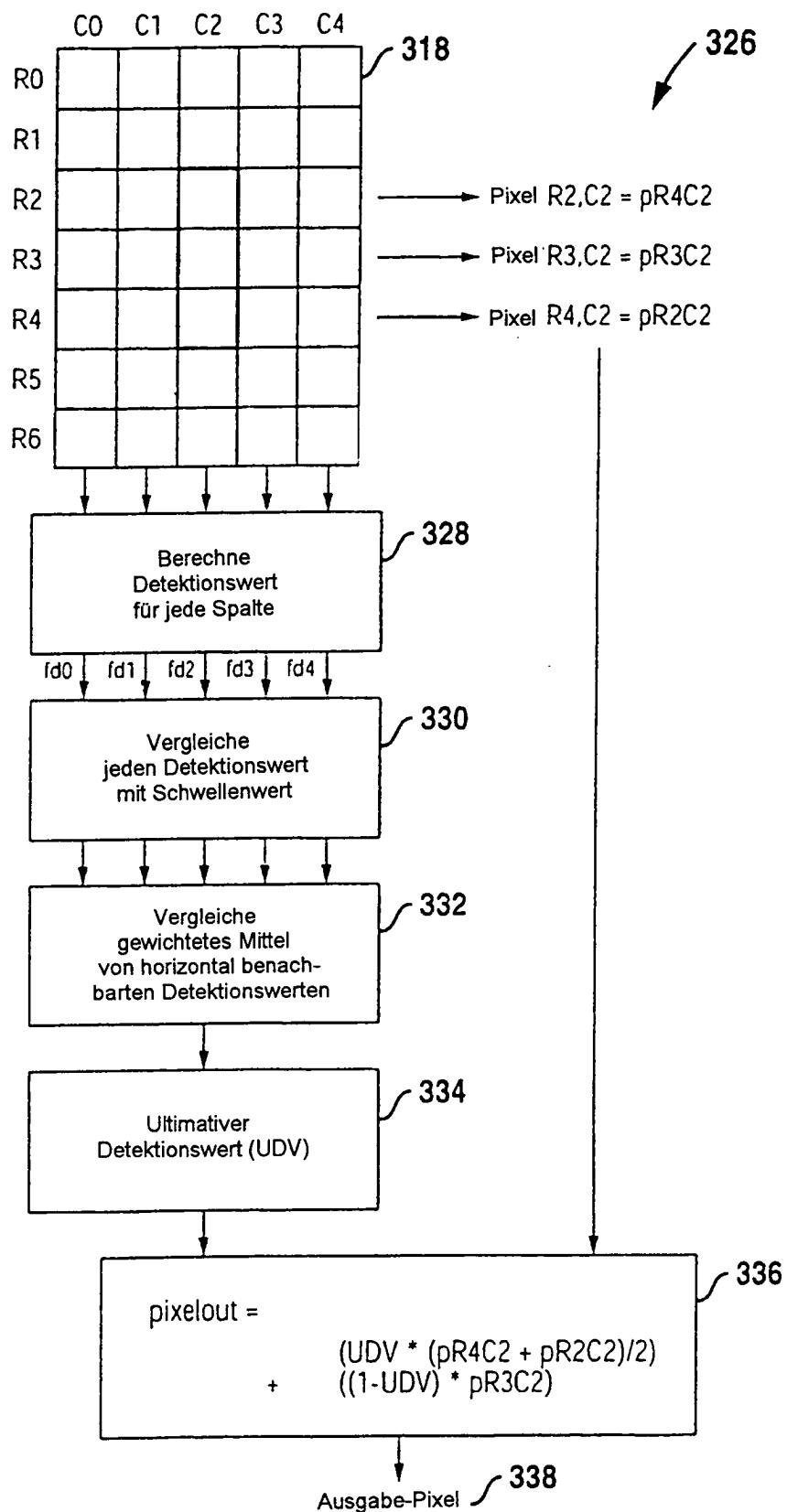


Fig. 12

**Fig. 13**

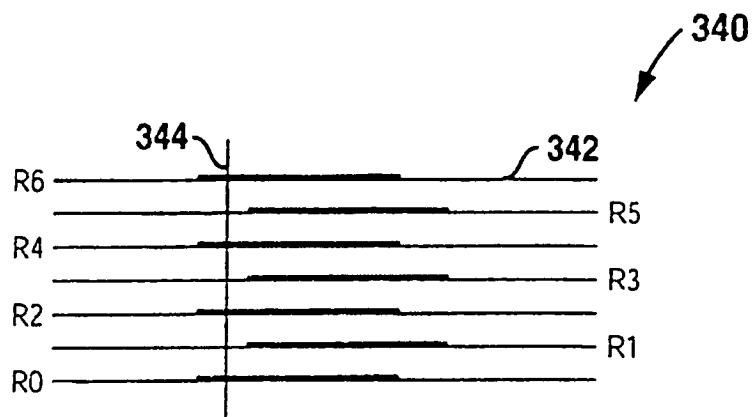


Fig. 14A

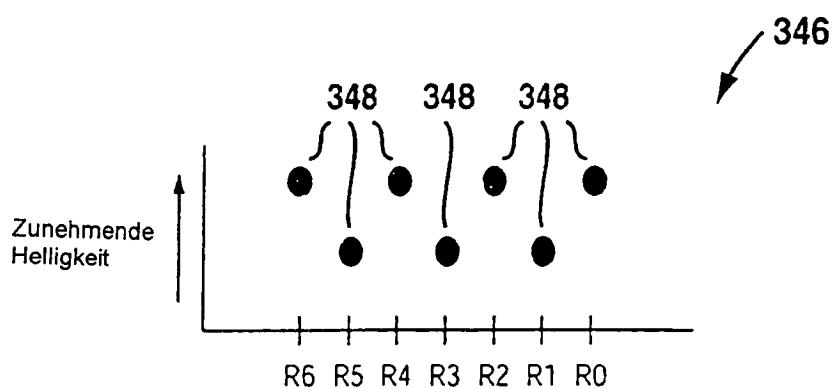


Fig. 14B

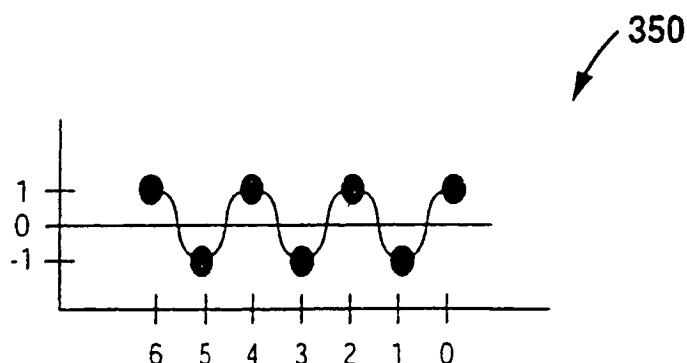


Fig. 14C

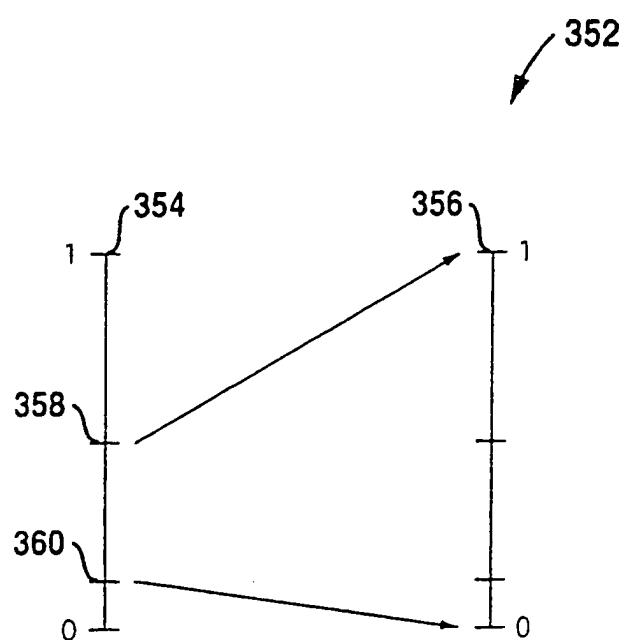


Fig. 15

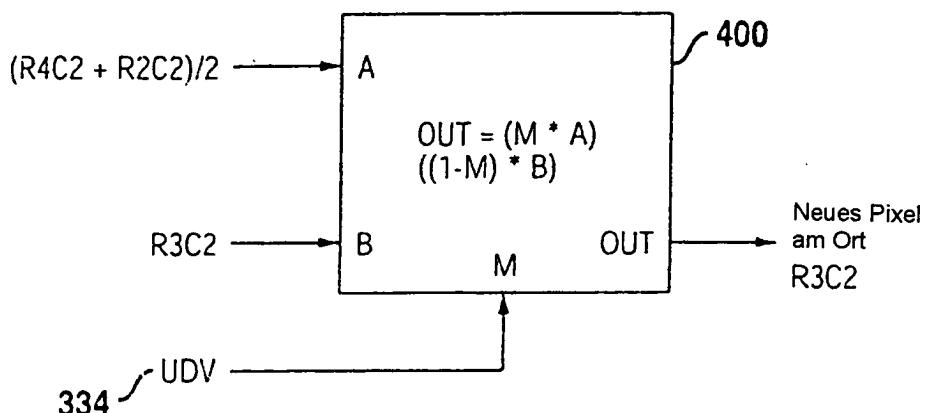


Fig. 16



Fig. 17

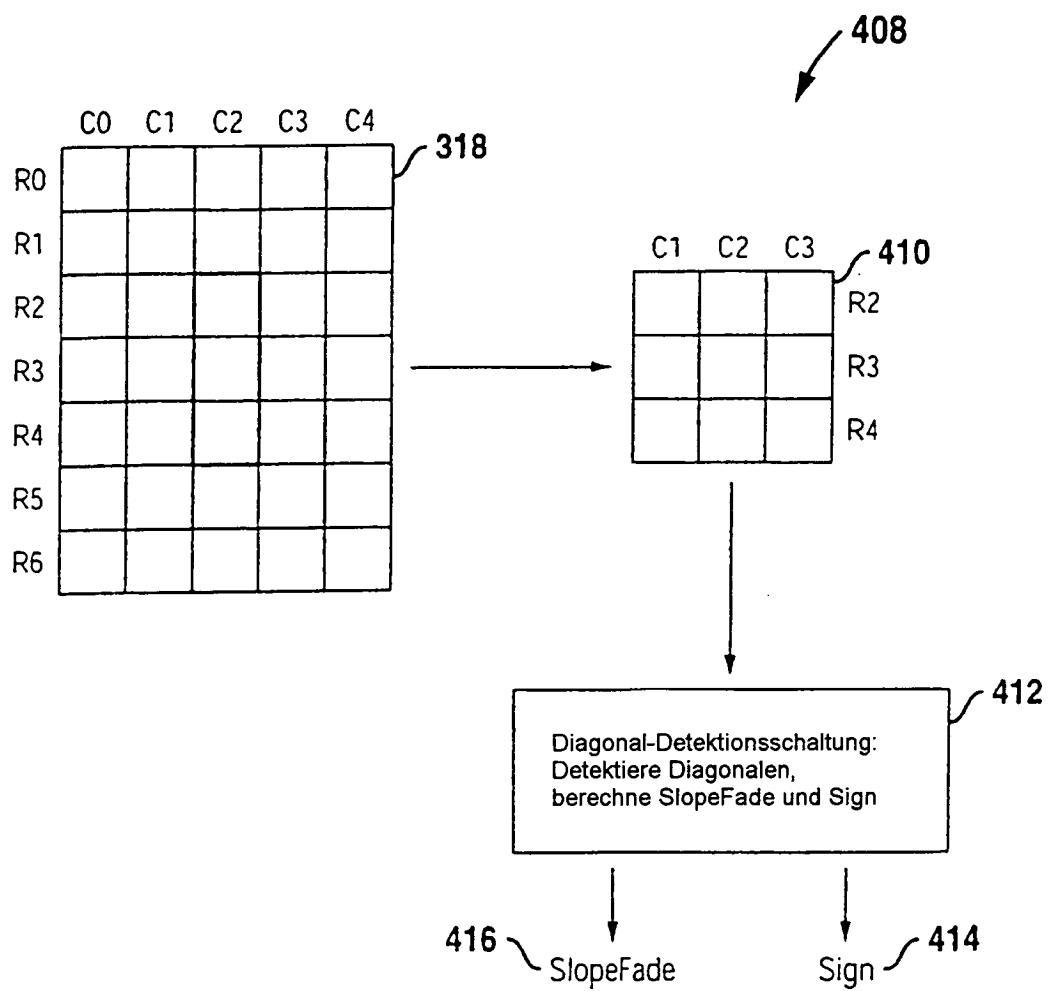
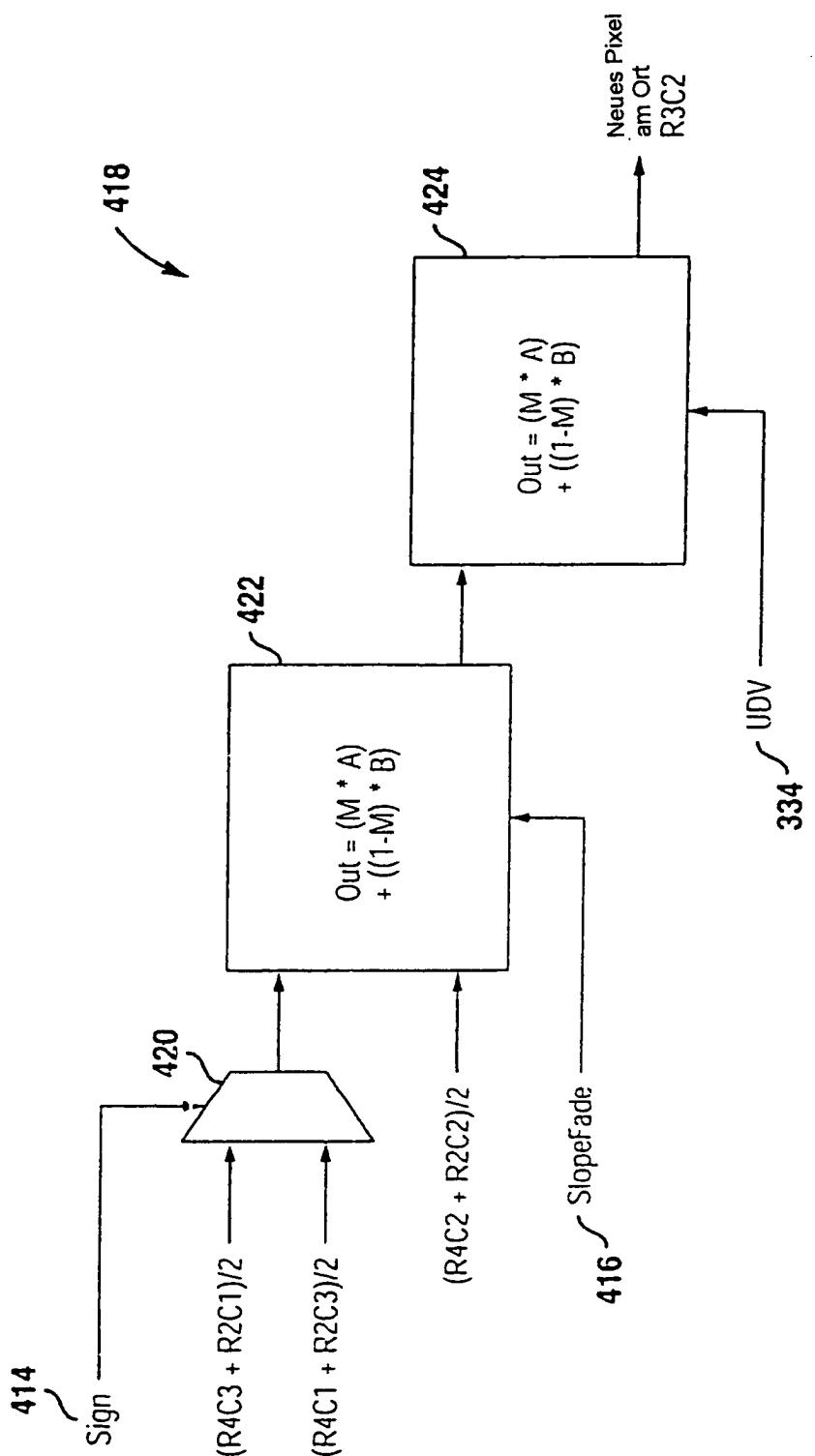


Fig. 18

**Fig. 19**

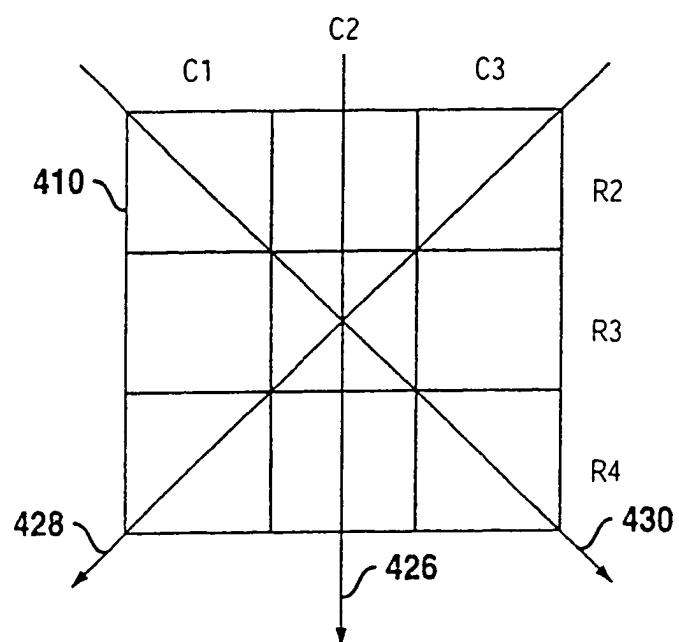
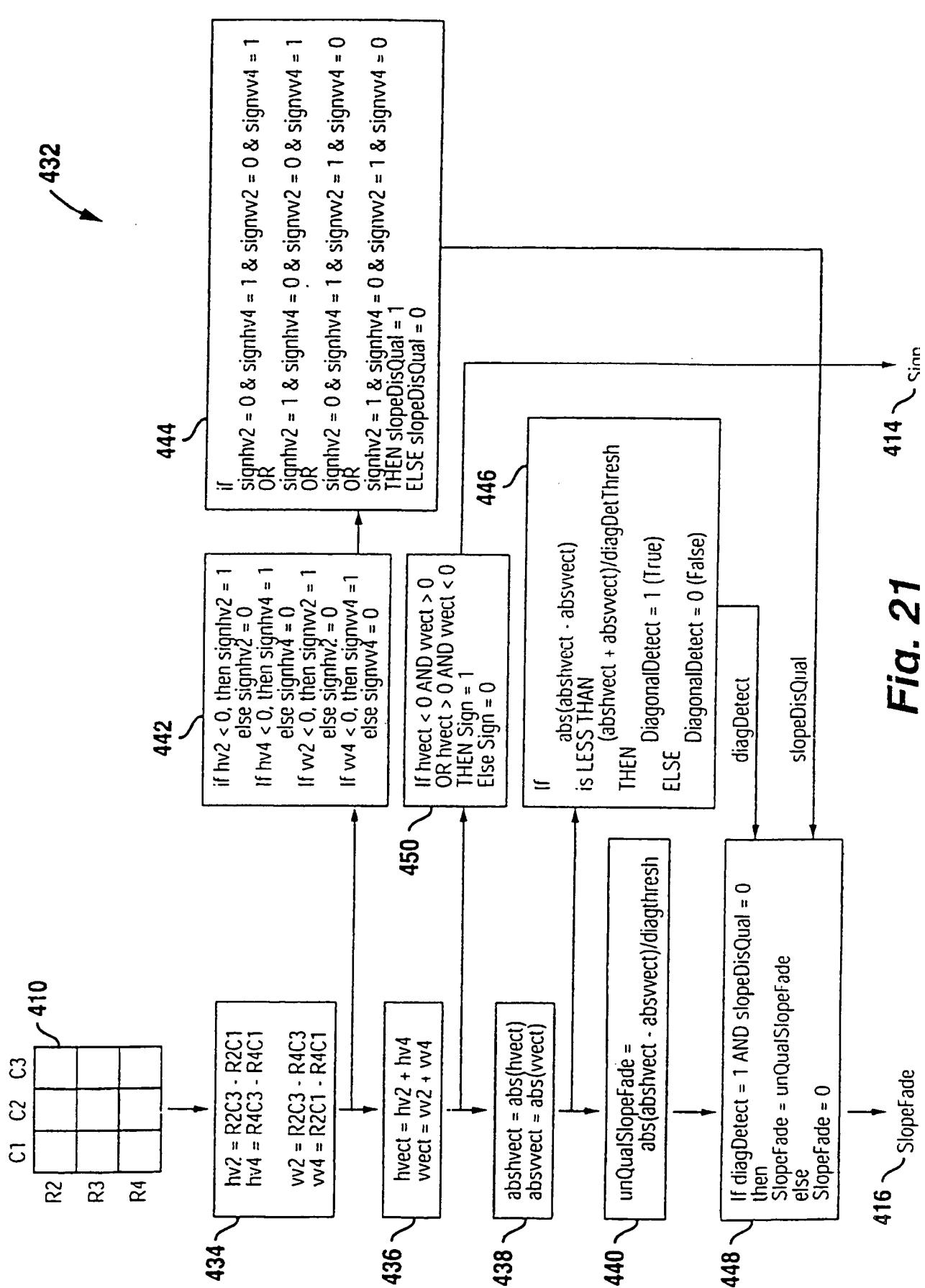
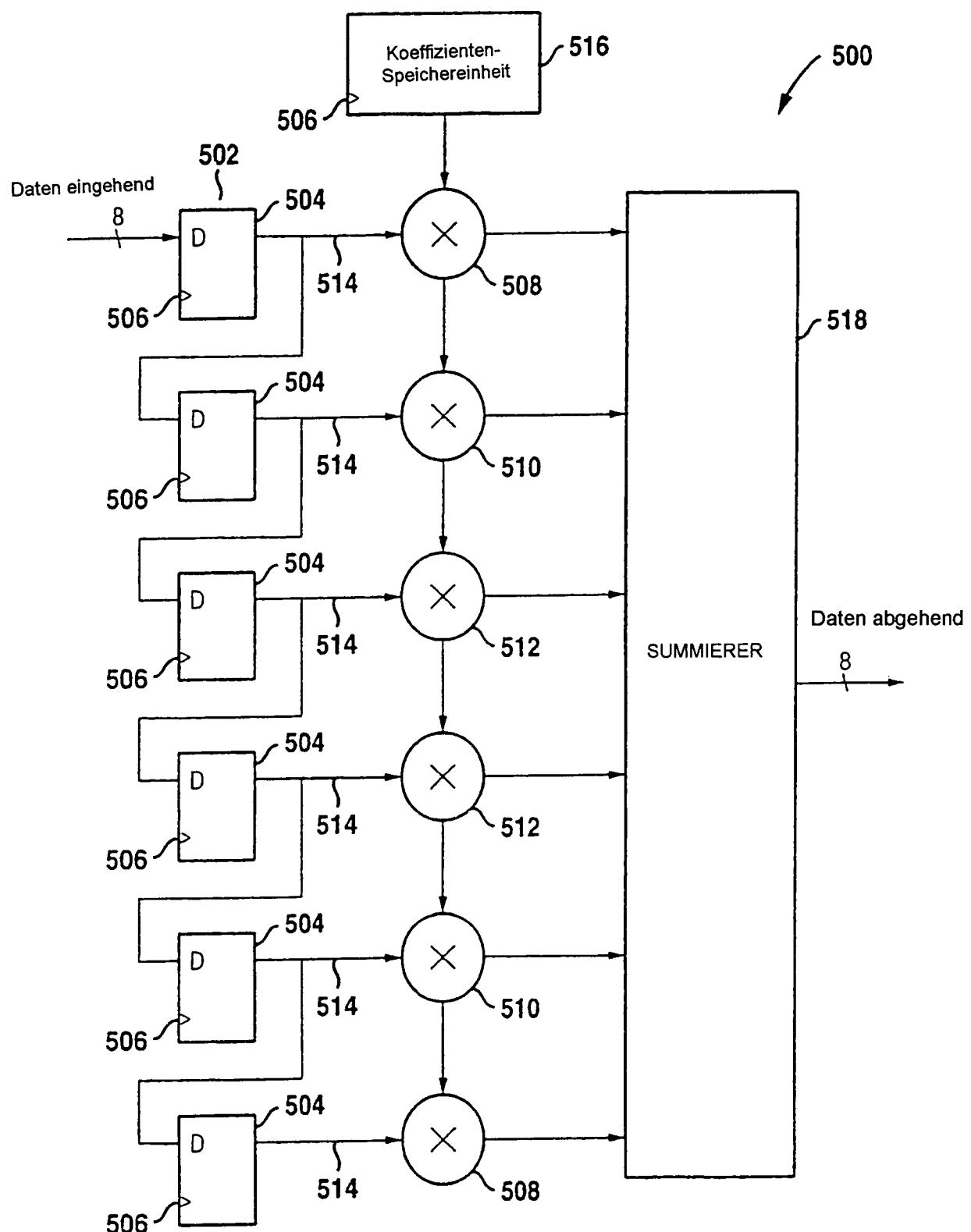
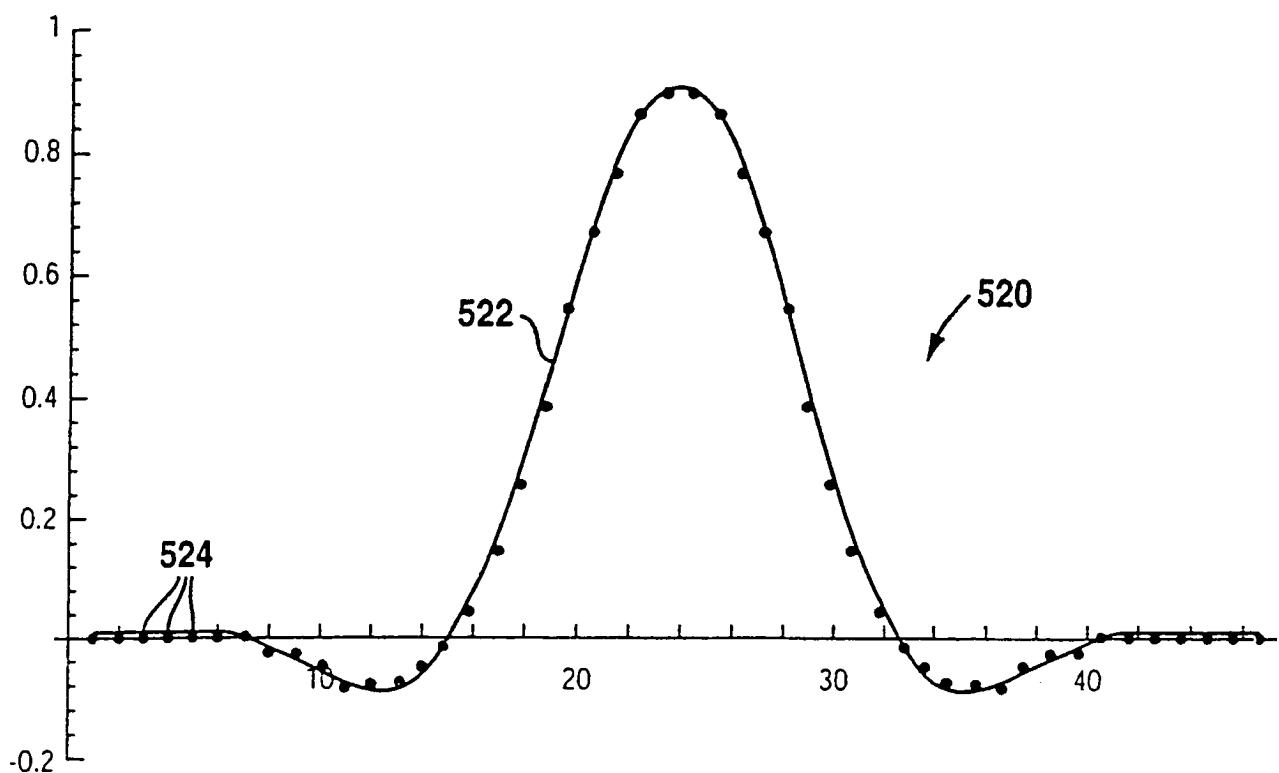


Fig. 20

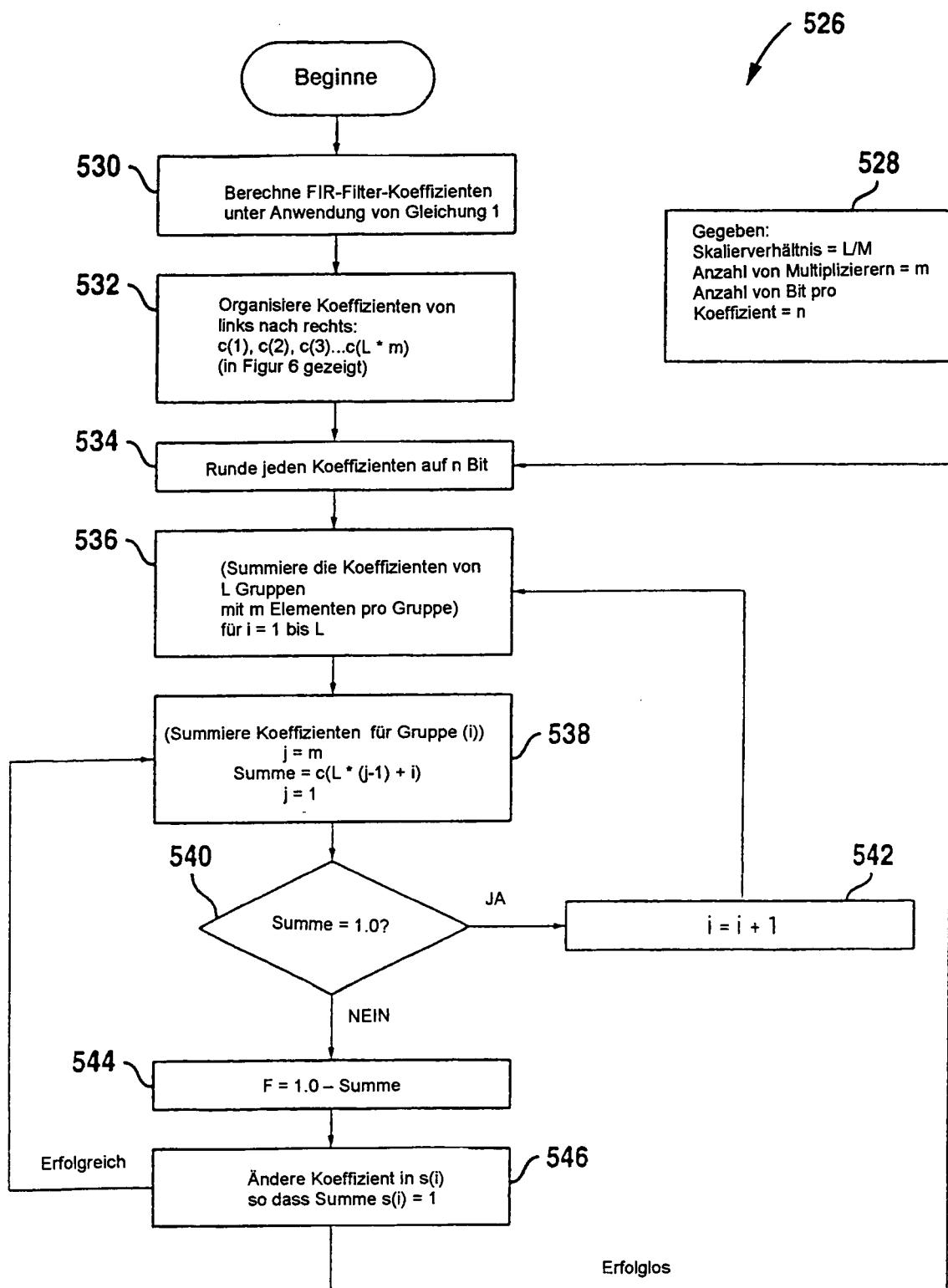


***Fig. 22***

***Fig. 23***

Gruppe(1)	c(1), c(9), c(17), c(25), c(33), c(41)
Gruppe(2)	c(2), c(10), c(18), c(26), c(34); c(42)
Gruppe(3)	c(3), c(11), c(19), c(27), c(35), c(43)
Gruppe(4)	c(4), c(12), c(20), c(28), c(36), c(44)
Gruppe(5)	c(5), c(13), c(21), c(29), c(37), c(45)
Gruppe(6)	c(6), c(14), c(22), c(30), c(38), c(46)
Gruppe(7)	c(7), c(15), c(23), c(31), c(39), c(47)
Gruppe(8)	c(8), c(16), c(24), c(32), c(40), c(48)

Fig. 24

**Fig. 25**

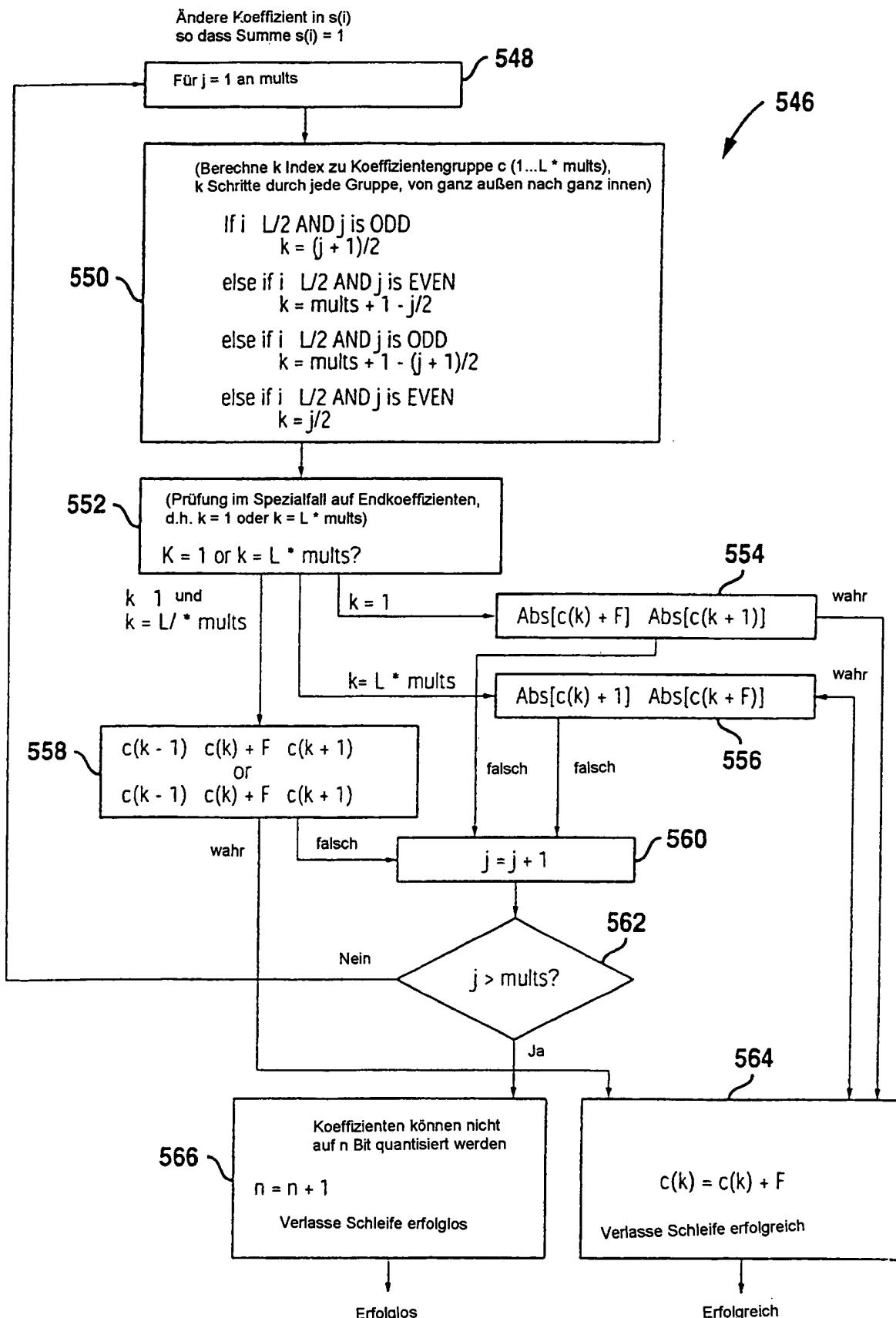


Fig. 26

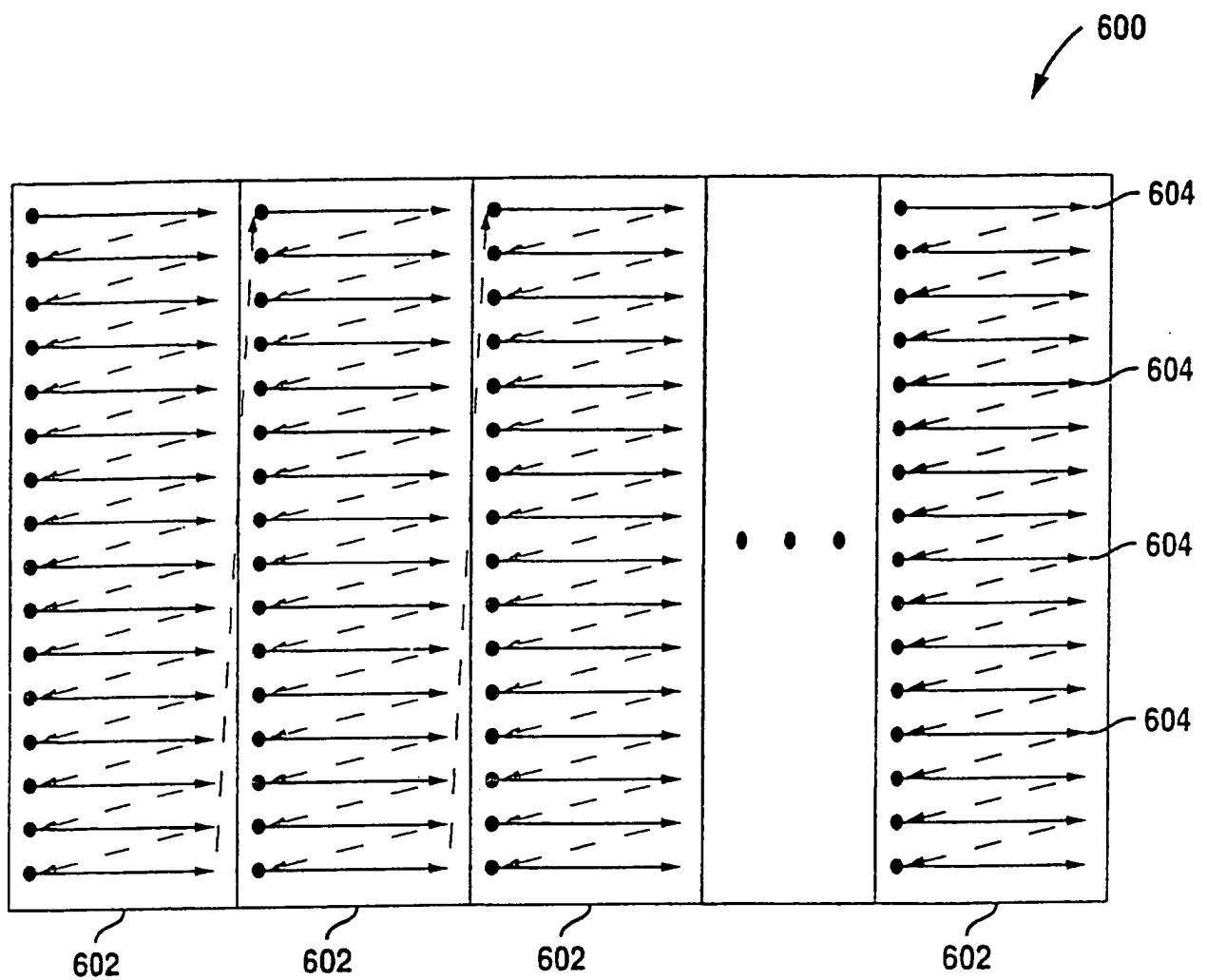


Fig. 27

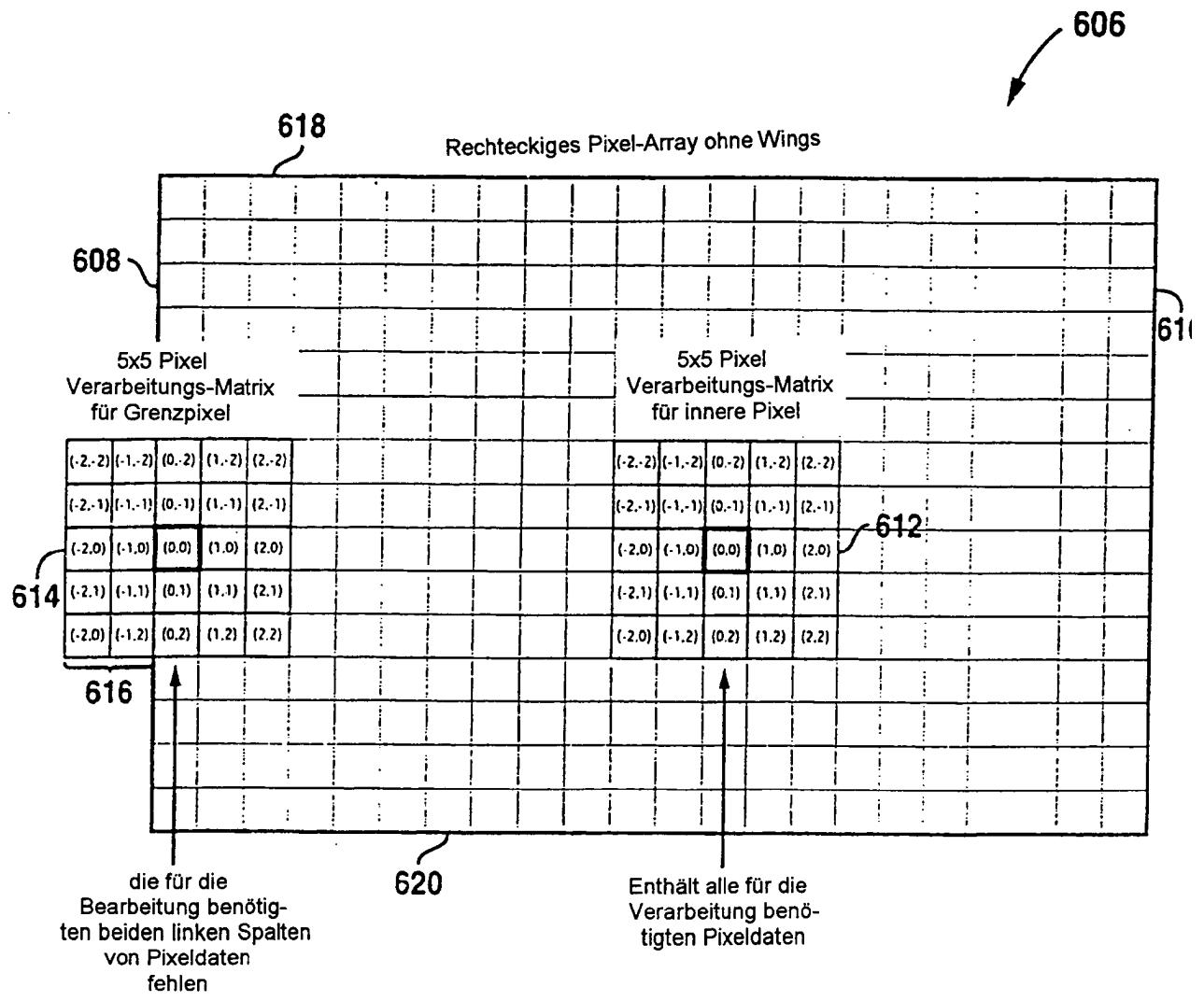
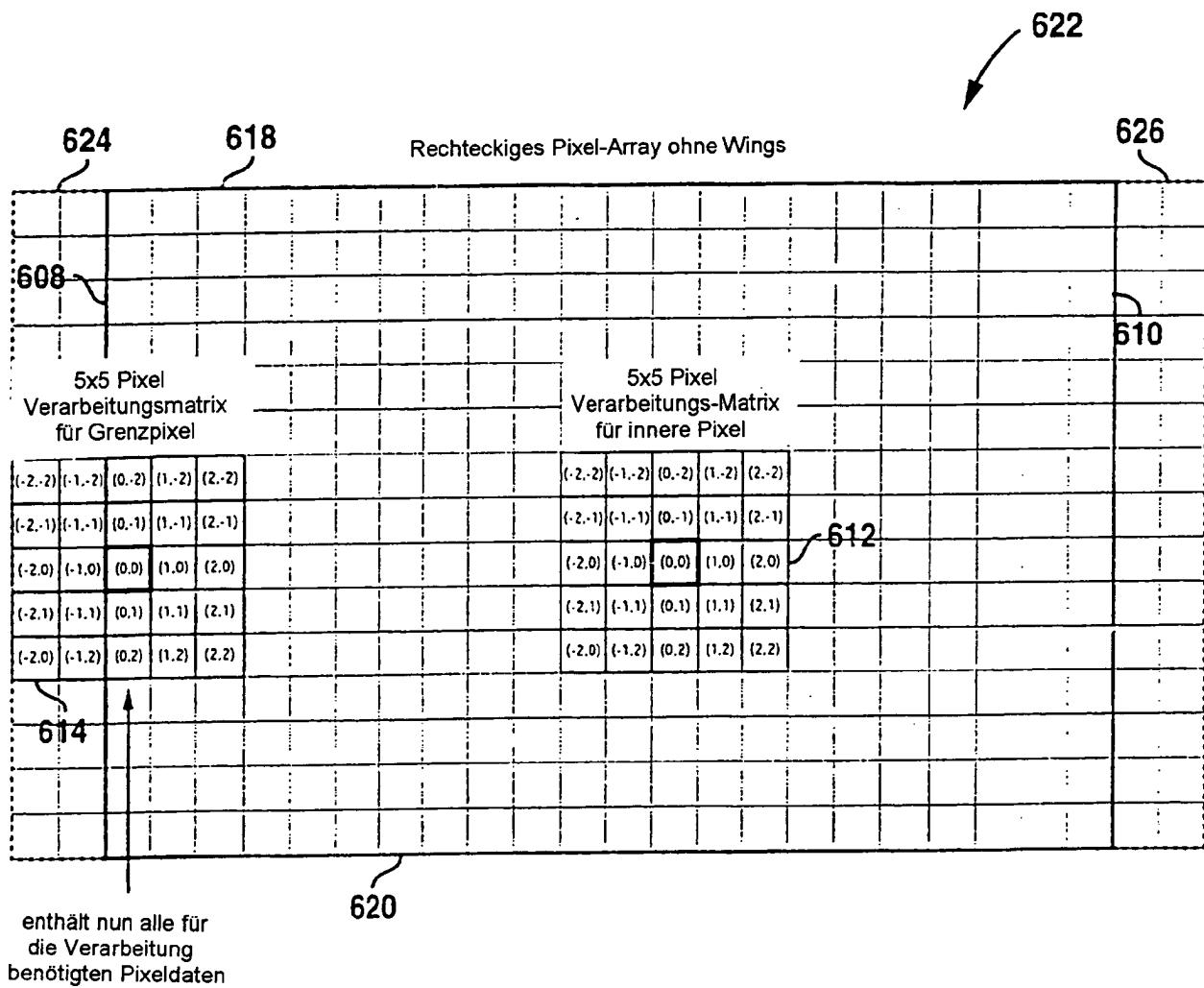


Fig. 28

*Fig. 29*

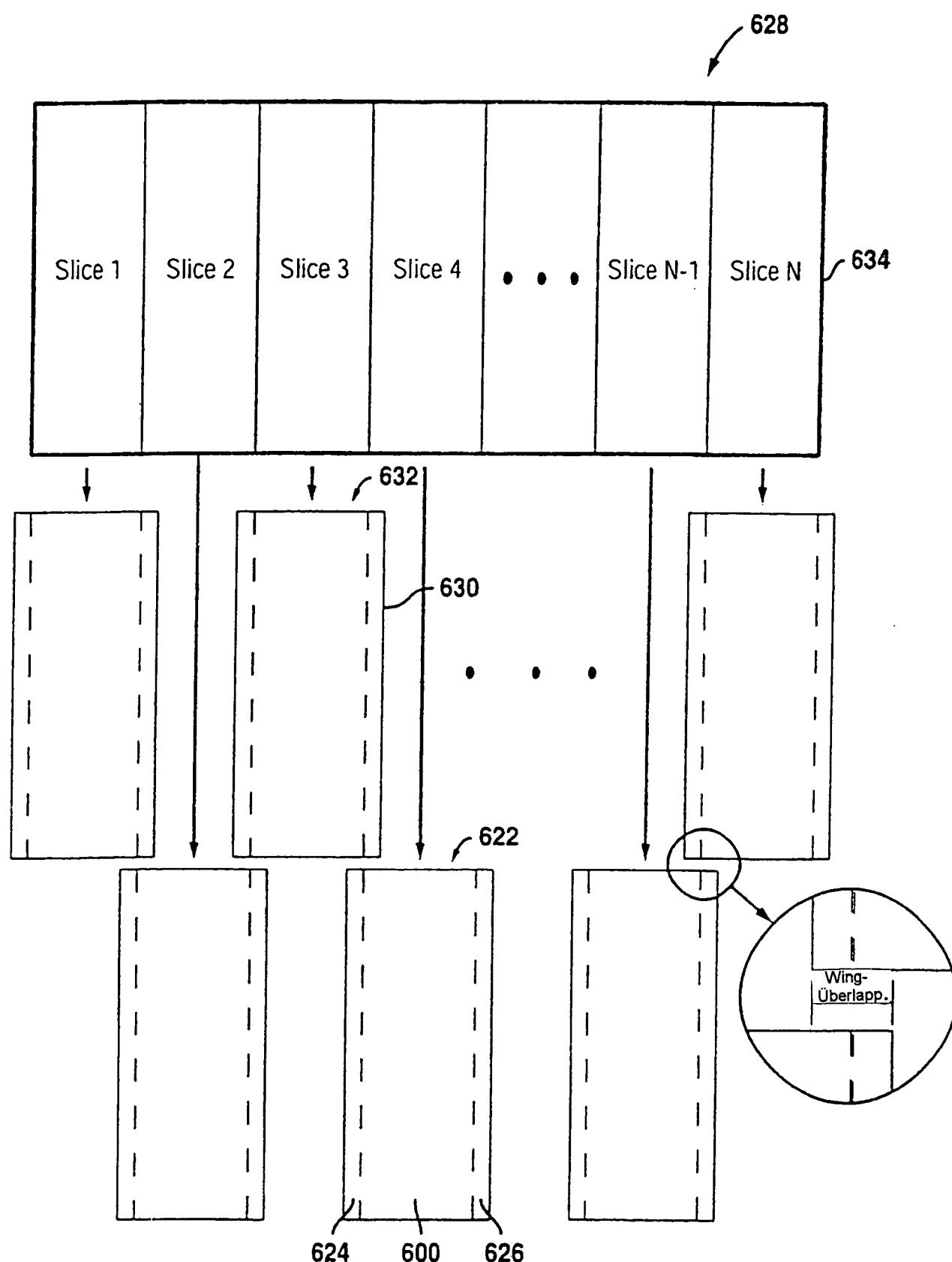


Fig. 30

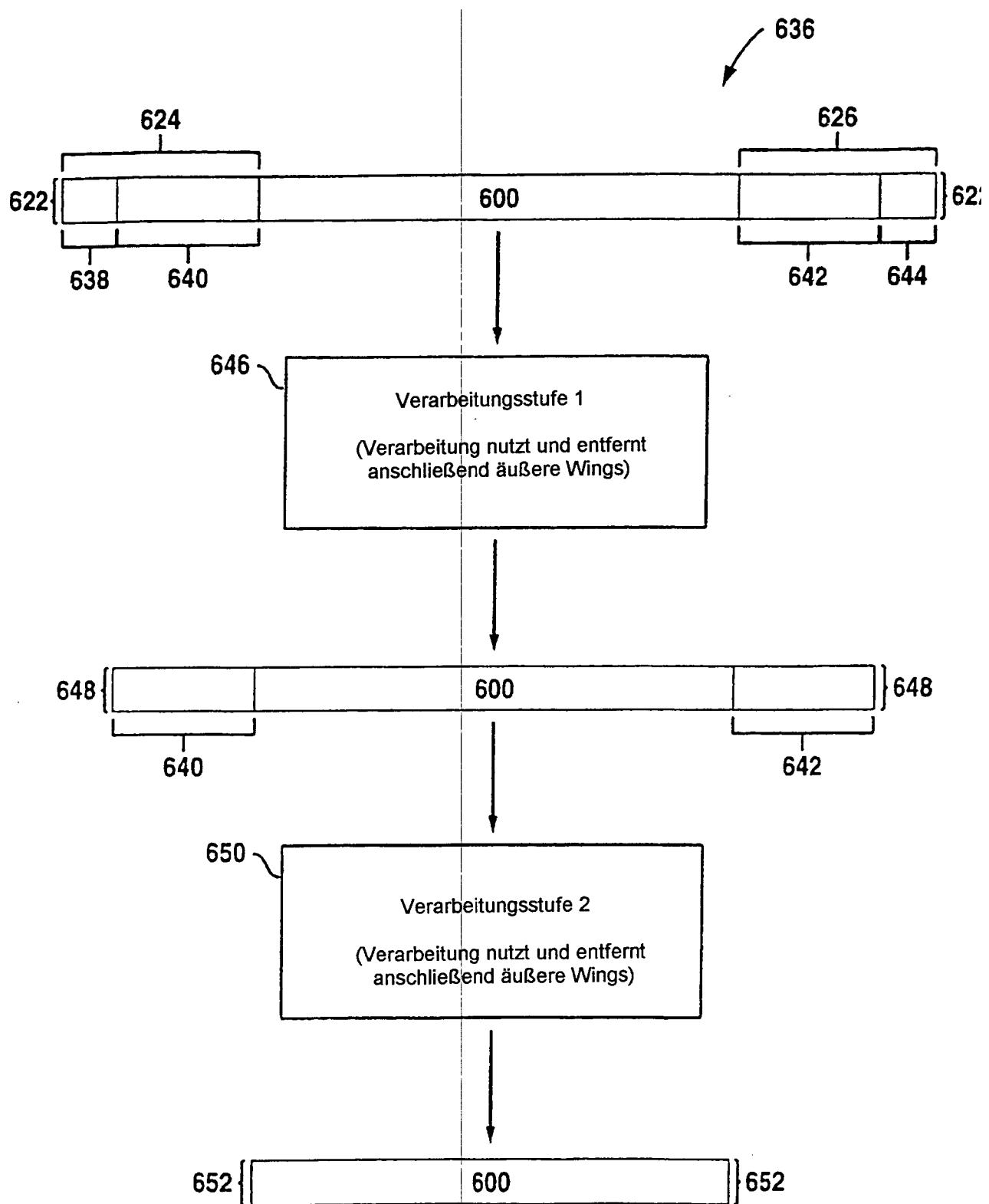
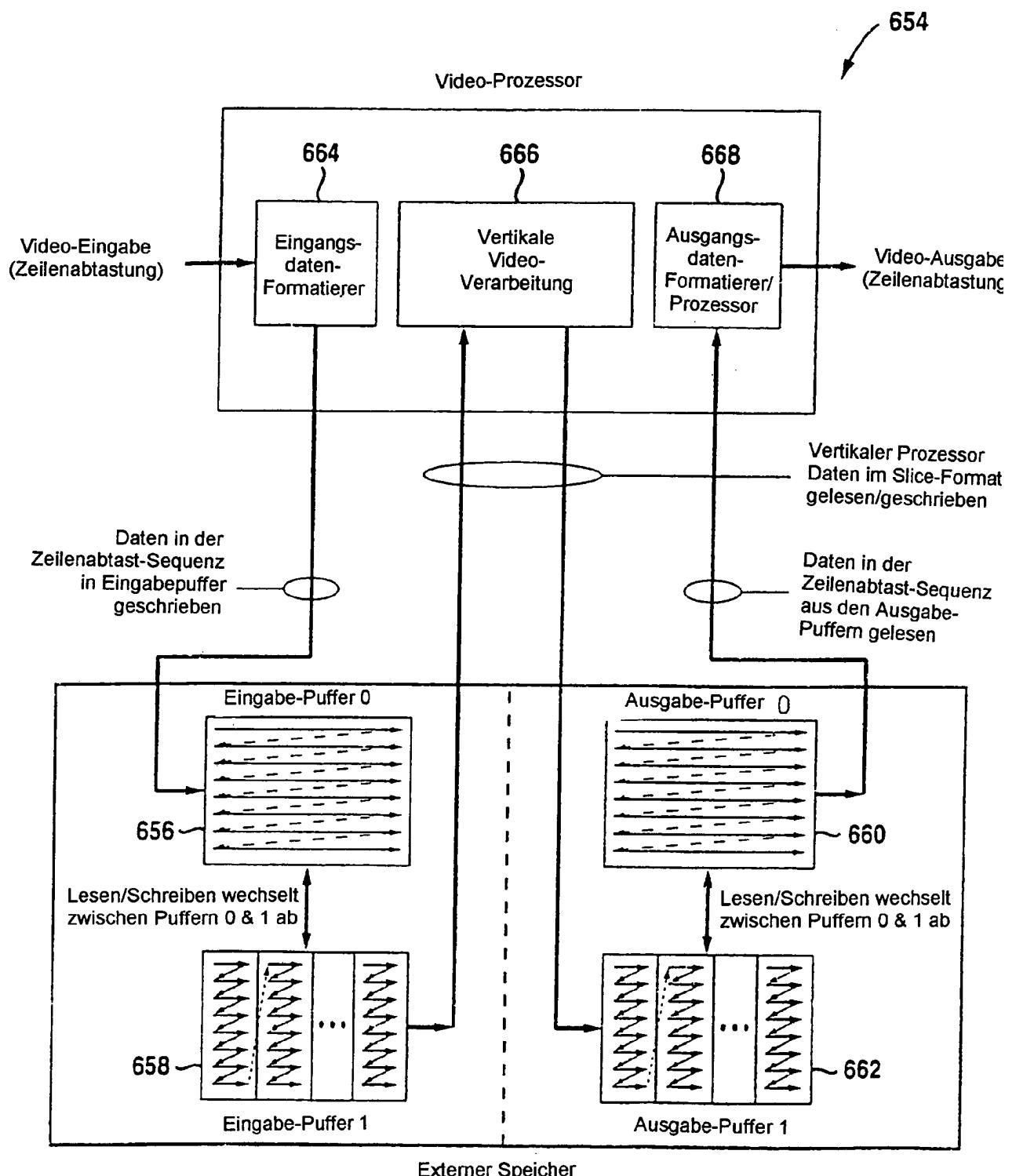
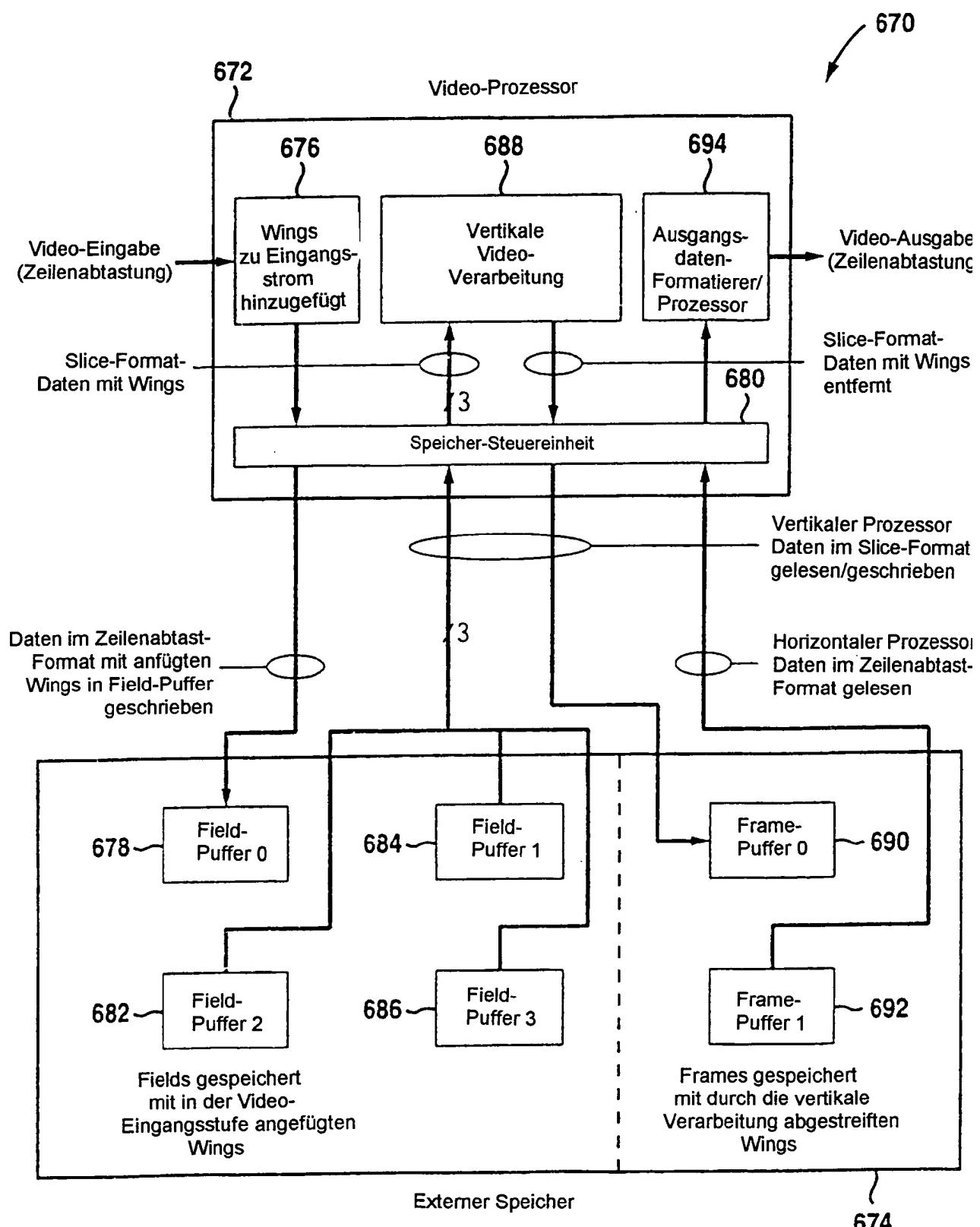
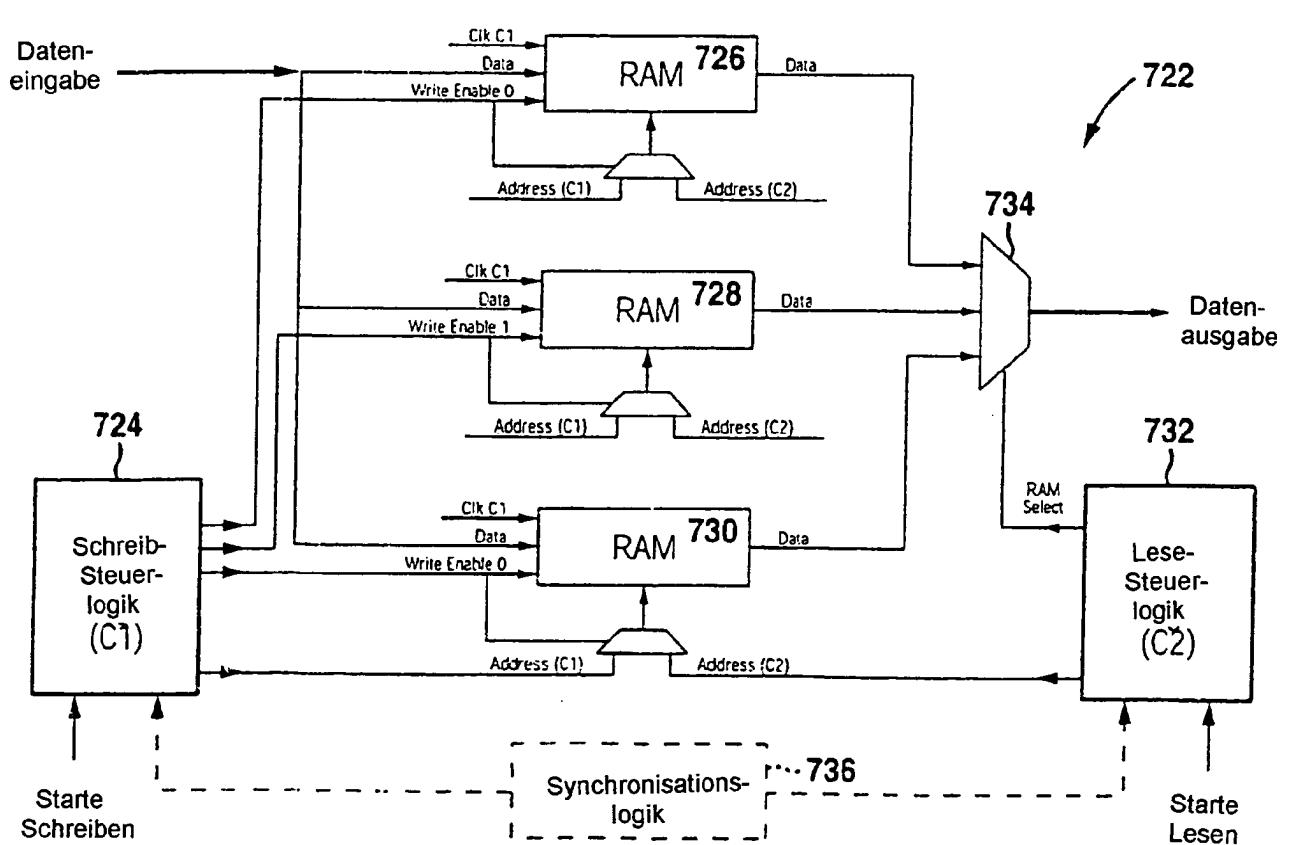
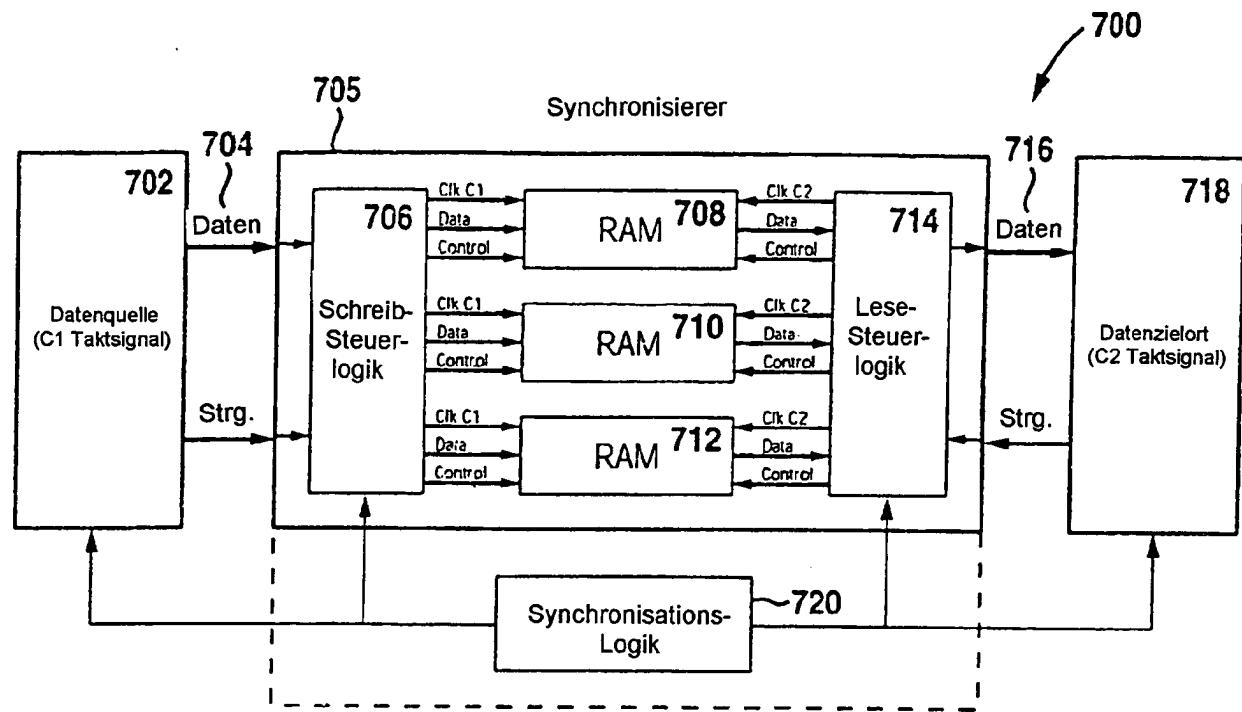
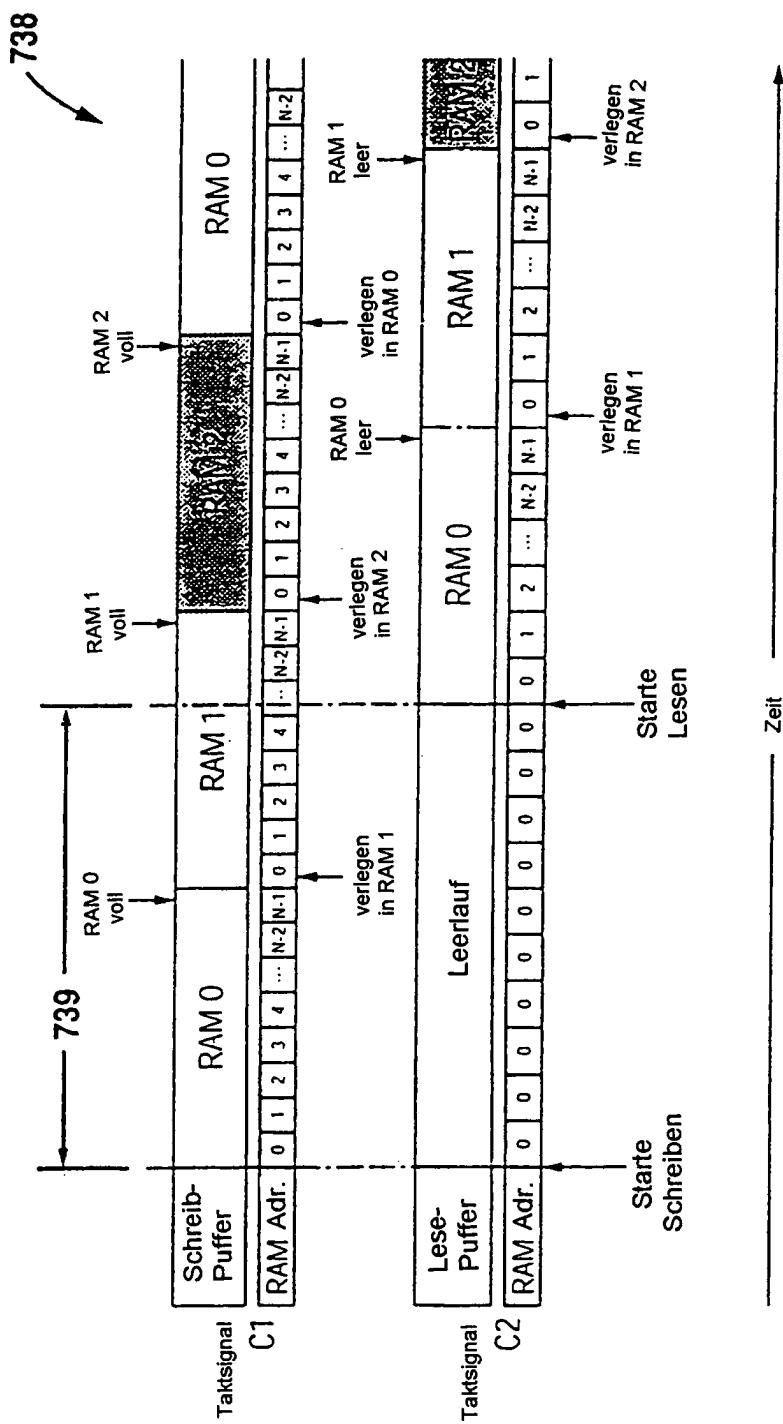


Fig. 31

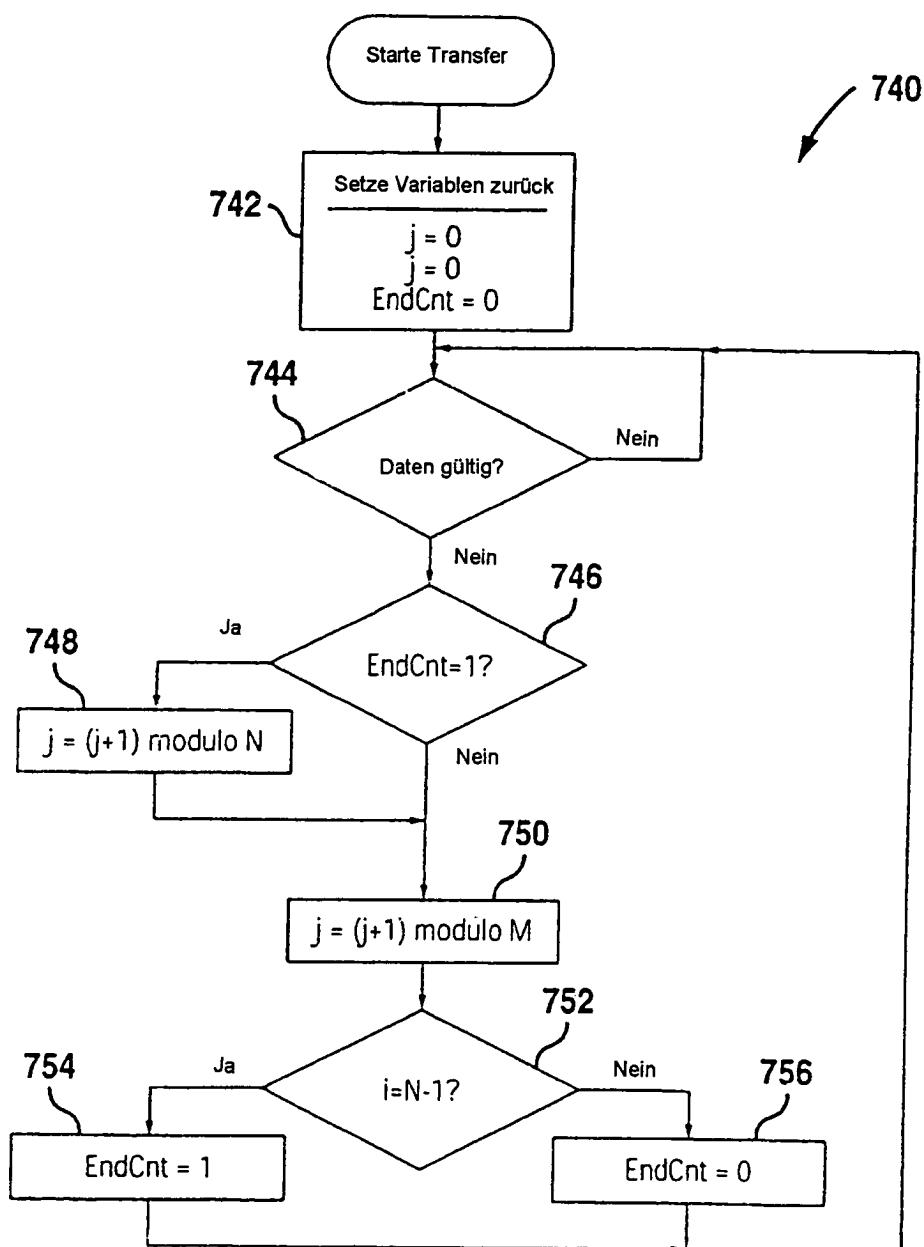
**Fig. 32**

**Fig. 33**

**Fig. 35**

**Fig. 36**

Anmerkung: Jedes RAM hat n adressierbare Orte



Anmerkungen:

1. "i" ist die RAM-Adresse
2. "j" bezeichnet das gewählte RAM-Modul
die Adressen-MUX-Steuerung und RAM-Schreibfreigabe
3. "EndCnt" zeigt an, dass die RAM-Adresse auf den letzten
Ort in einem RAM-Modul zeigt
4. "M" ist die Anzahl von adressierbaren Orten in einem
RAM-Modul
5. "N" ist die Anzahl von RAM-Modulen

Fig. 37

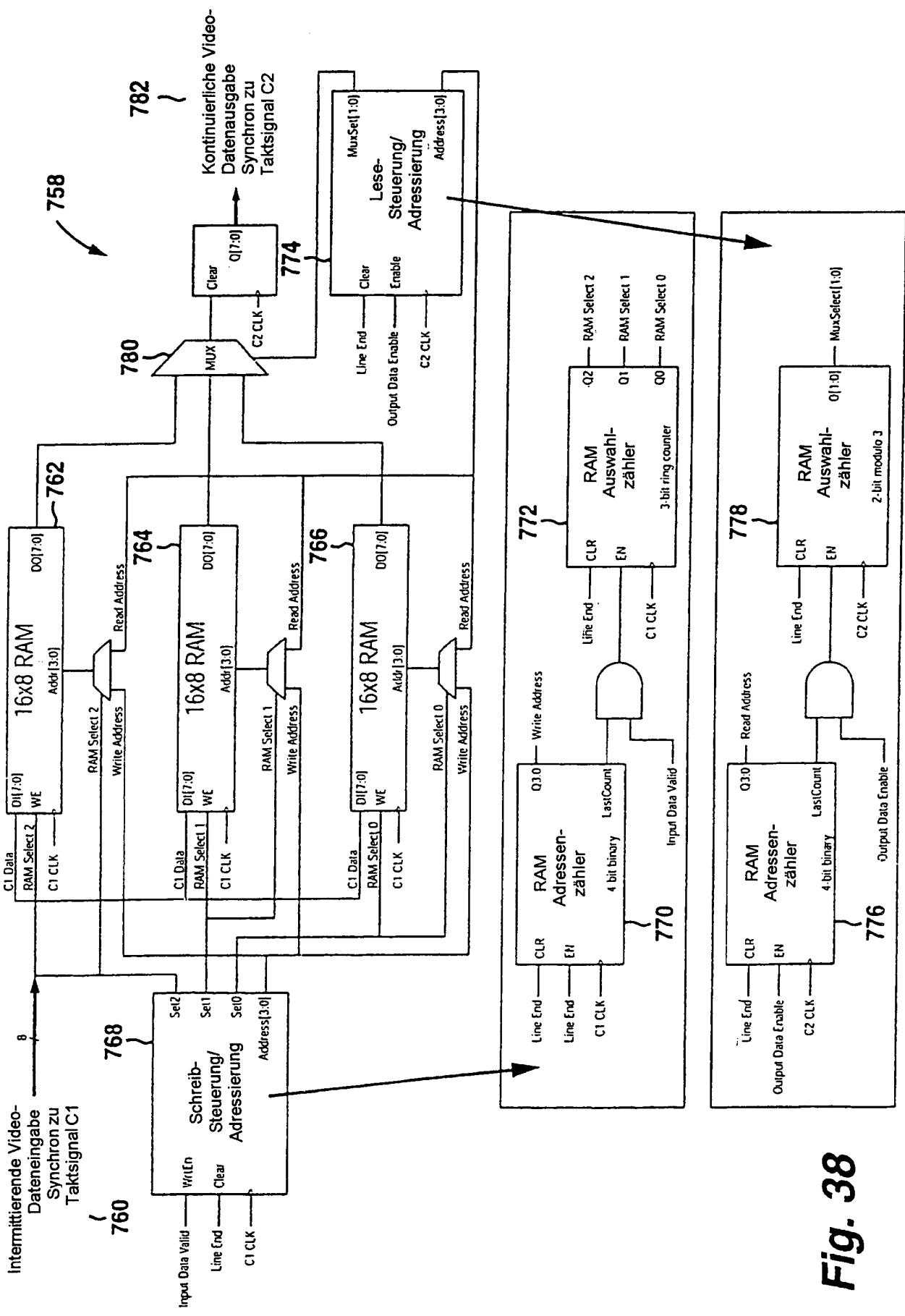


Fig. 38