



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) **DE 695 32 318 T2 2004.10.21**

(12)

## Übersetzung der europäischen Patentschrift

(97) EP 0 712 249 B1

(21) Deutsches Aktenzeichen: 695 32 318.0

(96) Europäisches Aktenzeichen: 95 116 378.1

(96) Europäischer Anmeldetag: 18.10.1995

(97) Erstveröffentlichung durch das EPA: 15.05.1996

(97) Veröffentlichungstag  
der Patenterteilung beim EPA: 17.12.2003

(47) Veröffentlichungstag im Patentblatt: 21.10.2004

(51) Int Cl.<sup>7</sup>: **H04N 7/24**

H04N 7/26, H04N 7/50

(30) Unionspriorität:  
335651 08.11.1994 US

(84) Benannte Vertragsstaaten:  
**BE, CH, DE, DK, ES, FR, GB, IE, IT, LI, NL, SE**

(73) Patentinhaber:  
**General Instrument Corporation, Horsham, Pa., US**

(72) Erfinder:  
**Hoogenboom, Chris, Calabasas, California 91302, US; Vuong, Bao, San Diego, California 92115, US**

(74) Vertreter:  
**HOEGER, STELLRECHT & PARTNER  
Patentanwälte, 70182 Stuttgart**

(54) Bezeichnung: **Pixelinterpolationsfilter für einen Videodekomprimierungsprozessor**

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 eingelebt, 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**

**[0001]** Die vorliegende Erfindung betrifft einen Videodekompressionsprozessor und insbesondere ein effizientes Verfahren zur Bereitstellung einer horizontalen, vertikalen und/oder bidirektionalen Interpolation vorheriger Frame-Pixeldaten, die zur Rekonstruktion eines gegenwärtigen Video-Frames notwendig sind.

**[0002]** Die digitale Übertragung von Fernsehsignalen kann Video- und Audiodienste wesentlich höherer Qualität als bei analogen Techniken liefern. Digitale Übertragungsverfahren sind besonders für Signale von Vorteil, die über ein Kabelfernsehnetz oder durch Satellit zu Kabelfernsehstationen und/oder direkt zu privaten Satellitenfernsehempfängern rundgesendet werden. Es wird erwartet, dass digitale Fernsehsender- und -empfängersysteme existierende analoge Systeme ersetzen werden, so wie digitale Compact Discs analoge Schallplatten in der Audioindustrie ersetzt haben.

**[0003]** Jedes digitale Fernsehsystem muss eine erhebliche Menge digitaler Daten senden. In einem digitalen Fernsehsystem empfängt ein Teilnehmer den digitalen Datenstrom über einen Empfänger/Entwürfler, der dem Teilnehmer Video, Audio und Daten liefert. Um das verfügbare Funkfrequenzspektrum am effizientesten zu nutzen, ist es vorteilhaft, die digitalen Fernsehsignale zu komprimieren, um die Datenmenge, die übertragen werden muss, möglichst gering zu halten.

**[0004]** Der Videoteil eines Fernsehsignals umfasst eine Sequenz von Video-„Frames“, die zusammen ein bewegliches Bild bereitstellen. In digitalen Fernsehsystemen wird jede Zeile eines Video-Frames durch eine Sequenz digitaler Datenbit definiert, die als „Pixel“ bezeichnet werden. Es ist eine große Datenmenge erforderlich, um jeden Video-Frame eines Fernsehsignals zu definieren. Zum Beispiel sind 7,4 Megabit Daten erforderlich, um einen Video-Frame mit NTSC-Auflösung (National Television System Committee) bereitzustellen. Dabei wird angenommen, dass eine Anzeige mit 640 Pixeln mal 480 Zeilen mit acht Bit Intensitätswert für jede der Primärfarben Rot, Grün und Blau verwendet wird. Bei der Auflösung von PAL (Phase Alternating Line) sind 9,7 Megabit Daten erforderlich, um einen Video-Frame bereitzustellen. In diesem Fall wird eine Anzeige mit 704 Pixeln mal 576 Zeilen mit acht Bit Intensitätswert für jede der Primärfarben Rot, Grün und Blau verwendet. Um diese Informationsmenge zu verwalten, müssen die Daten komprimiert werden.

**[0005]** Videokompressionstechniken ermöglichen die effiziente Übertragung digitaler Videosignale über herkömmliche Kommunikationskanäle. Solche Tech-

niken verwenden Kompressionsalgorithmen, die die Korrelation zwischen angrenzenden Pixeln ausnutzen, um eine effizientere Darstellung der wichtigen Informationen in einem Videosignal abzuleiten. Die leistungsstärksten Kompressionssysteme nutzen nicht nur räumliche Korrelation aus, sondern können auch Ähnlichkeiten zwischen angrenzenden Frames verwenden, um die Daten weiter zu komprimieren. In solchen Systemen wird in der Regel Differenzkodierung verwendet, um nur die Differenz zwischen einem tatsächlichen Frame und einer Prädiktion des tatsächlichen Frames zu übertragen. Die Prädiktion basiert auf Informationen, die aus einem vorherigen Frame derselben Videosequenz abgeleitet werden.

**[0006]** Beispiele für Videokompressionssysteme, die Bewegungskompensation verwenden, finden sich bei Krause et al., US-Patente Nr. 5,057,916; 5,068,724; 5,091,782; 5,093,720 und 5,235,419. Solche Bewegungskompensationssysteme benutzen in der Regel einen Blockanpassungsbewegungsschätzalgorithmus. In diesem Fall wird für jeden Block in einem aktuellen Frame eines Bildes durch Identifizieren eines Blocks in einem vorangehenden Frame, der dem bestimmten gegenwärtigen Block am stärksten ähnelt, ein Bewegungsvektor bestimmt. Der gesamte gegenwärtige Frame kann dann in einem Dekodieren rekonstruiert werden, indem die Differenz zwischen den entsprechenden Blockpaaren zusammen mit den Bewegungsvektoren, die erforderlich sind, um die entsprechenden Paare zu identifizieren, gesendet wird. Häufig wird die Menge übertragener Daten weiter verringert, indem sowohl die verschobenen Blockdifferenzen als auch die Bewegungssektor-signale komprimiert werden. Blockanpassungsbewegungsschätzalgorithmen sind besonders effektiv, wenn sie mit blockgestützten räumlichen Kompressionstechniken, wie z. B. der diskreten Kosinustransformation (DCT) kombiniert werden.

**[0007]** Jeder einer Abfolge digitaler Video-Frames, die ein Videoprogramm bilden, kann als ein Innen-Frame (I-Frame), ein Prädiktions-Frame (P-Frame) oder ein bidirektionaler Frame (B-Frame) kategorisiert werden. Die Prädiktion basiert auf der zeitlichen Korrelation zwischen sukzessiven Frames. Teile von Frames unterscheiden sich über kurze Zeitspannen hinweg nicht voneinander. Die Kodierungs- und Dekodierungsverfahren unterscheiden sich für jeden Bildtyp. Die einfachsten Verfahren sind die für I-Frames verwendeten, gefolgt von denen für P-Frames und dann denen für B-Frames.

**[0008]** I-Frames beschreiben vollständig einen einzelnen Frame ohne Bezug auf jegliche andere Frames. Zur verbesserten Fehlerverbergung können Bewegungsvektoren in einen I-Frame aufgenommen werden. Ein Fehler in einem I-Frame hat das Potential für eine größere Auswirkung auf das angezeigte Videosignal, da sowohl P-Frames als auch B-Frames

aus einem I-Frame vorhergesagt werden.

**[0009]** P-Frames werden auf der Grundlage vorheriger I- oder P-Frames vorhergesagt. Der Bezug besteht von einem früheren I- oder P-Frame auf einen zukünftigen P-Frame und wird deshalb als „Vorwärtsprädiktion“ bezeichnet. B-Frames werden aus dem nächstfrüheren I- oder P-Frame und dem nächstspäteren I- oder P-Frame vorhergesagt. Der Bezug auf ein zukünftiges Bild (d. h. ein Bild, das noch nicht angezeigt worden ist) wird als „Rückwärtsprädiktion“ bezeichnet. Es gibt Fälle, in denen die Rückwärtsprädiktion sehr nützlich ist, um die Kompressionsrate zu erhöhen. Zum Beispiel kann in einer Szene, in der sich eine Tür öffnet, das aktuelle Bild auf der Grundlage eines zukünftigen Bildes, in dem die Tür bereits offen ist, vorhersagen, was sich hinter der Tür befindet.

**[0010]** B-Frames ergeben die größte Kompression, enthalten aber auch den größten Fehler. Um eine Fehlerausbreitung zu eliminieren, dürfen B-Frames niemals aus anderen B-Frames vorhergesagt werden. P-Frames führen zu weniger Fehlern und weniger Kompression. I-Frames ergeben die geringste Kompression, können aber Direktzugriffseintrittspunkte in eine Videosequenz bereitstellen.

**[0011]** Ein Standard, der für die Kodierung digitaler Videosignale angenommen wurde, ist der Standard der Motion Picture Experts Group (MPEG) und insbesondere der Standard MPEG-2. Dieser Standard spezifiziert keine konkrete Verteilung, die I-Frames, P-Frames und B-Frames in einer Sequenz annehmen müssen. Stattdessen lässt der Standard verschiedene Verteilungen zu, um verschiedene Grade von Kompression und Direktzugriffsfähigkeit bereitzustellen. Eine übliche Verteilung sind I-Frames etwa jede halbe Sekunde und zwei B-Frames zwischen sukzessiven I- oder P-Frames. Um P-Frames zu dekodieren, muss der vorherige I-Frame verfügbar sein. Ähnlich müssen, um B-Frames zu dekodieren, die vorherigen und zukünftigen P- oder I-Frames verfügbar sein. Folglich werden die Video-Frames in Abhängigkeitsreihenfolge kodiert, so dass alle für die Prädiktion verwendeten Bilder vor den daraus vorhergesagten Bildern kodiert werden. Weitere Einzelheiten über den Standard MPEG-2 (und des alternativen Standards DigiCipher®II) und seine Implementierung in einem Videodekompressionsprozessor finden sich in dem Dokument MC68VDP/D, einem vorläufigen Datenblatt mit dem Titel „MPEG-2/DCII Video Decompression Processor“, ©Motorola Microprocessor and Memory Technologies Group, 1994. Insbesondere gibt der Anhang A dieses Dokuments eine Übersicht über MPEG-2-Video.

**[0012]** Um die Videokompression in praktischen Systemen zu implementieren, ist für jeden digitalen Fernsehempfänger ein Videodekompressionspro-

zessor erforderlich. Die Entwicklung höchstintegrierter integrierter Schaltungsschips (VLSI) findet gerade statt, um solche Videodekompressionsprozessoren zu implementieren. Bei Verbraucherprodukten, wie zum Beispiel Fernsehapparaten, ist es imperativ, dass die Kosten der Systemkomponenten so niedrig wie möglich gehalten werden. Einer der signifikantesten einem Videodekompressionsprozessor zugeordneten Kostenfaktoren ist der Direktzugangsspeicher (RAM), der für folgendes erforderlich ist: (i) Puffern der komprimierten Daten vor der Dekomprimierung, (ii) Speichern der vorherigen Frame-Daten, die zur Vorhersage eines gegenwärtigen Frames unter Verwendung von Bewegungsschätztechniken notwendig sind, und (iii) Puffern der dekomprimierten Daten vor ihrer Ausgabe an ein Videogerät, wie zum Beispiel einen Fernsehapparat, einen Videorekorder oder der gleichen. Andere signifikante Kosten eines Dekompressionsprozessor liegen in der notwendigen Hardware zur Berechnung der Prädiktionen gegenwärtiger Frame-Daten aus vorherigen Frame-Daten, insbesondere wenn Interpolation zwischen angrenzenden Pixeln notwendig ist, um Subpixel-(„Subpel“-)Daten für die Prädiktion bereitzustellen.

**[0013]** Die effiziente Ausnutzung des oben erwähnten Direktzugriffsspeichers durch einen Videodekompressionsprozessor, die in der Regel in einem externen DRAM implementiert wird, erfordert ein Verfahren, das eine minimale Speichermenge benutzen kann, während die erforderlichen Datenzugriffsraten (d. h. Speicherbandbreite) aufrechterhalten werden. DRAMs sind in der Regel als ein Array aus Reihen (die auch als „Seiten“ bezeichnet werden) und Spalten organisiert. Eine der Regeln des DRAM-Betriebs lautet, dass eine Änderung einer Reihenadresse zu einem langsameren Zugriff für die ersten Daten der nächsten Reihe führt. Um die DRAM-E/A-Bandbreite zu maximieren, ist es deshalb wünschenswert, Daten so zu lesen, dass die kleinstmögliche Anzahl von Änderungen in der Reihenadresse verursacht wird. So mit ist es vorteilhaft, den Speicherplan so anzupassen, dass Reihenänderungen minimiert werden. Weiterhin ist es vorteilhaft, sequenziell auf die in dem Speicher gespeicherten Daten zuzugreifen. Solche sequenziellen Zugriffe sind schnell und deshalb wünschenswert. Direktzugriffe, die häufige Änderungen der Reihenadresse erfordern können, sind dagegen langsam und deshalb nicht wünschenswert.

**[0014]** Bei einem Videodekompressionsprozessor, wie zum Beispiel einem dem Standard MPEG (Motion Picture Experts Group) oder DigiCipher®II (DCII) entsprechenden, erfordern verschiedene Prozesse, darunter die Prädiktionsberechnung, DRAM-Zugriff. Wenn die Prädiktion eines gegenwärtigen Frame-Blocks aus einem vorherigen Frame gut ist, d. h., der Prädiktions-Frame dem zu übertragenden Frame sehr ähnlich ist, bleibt nur ein sehr kleiner Restfehler für die Übertragung. Dies führt zu einer höheren

Kompressionseffizienz. Wenn eine schlechte Prädiktion durchgeführt wird, dann kann der Restfehler so groß sein, dass die Kompressionseffizienz beeinträchtigt wird. Eine genaue Prädiktion der Bewegung von Frame zu Frame in einer Videosequenz ist also für die Erreichung eines hohen Kompressionsverhältnisses entscheidend.

**[0015]** Für eine typische Videosequenz kann die Szene viele Objekte enthalten, die sich mit verschiedenen Geschwindigkeiten und Richtungen unabhängig bewegen. Um die Hardwareimplementierung zu erleichtern und die notwendige Informationsmenge zur Darstellung jeder Bewegung zu begrenzen, wird ein Video-Frame häufig in rechteckige Blöcke segmentiert. Man nimmt dann an, dass sich nur die Blöcke mit unabhängigen Geschwindigkeiten und Richtungen bewegen. Um die Systemkomplexität zu verringern und die Geschwindigkeit zu erhöhen, kann der Bereich, der nach der besten Übereinstimmung zwischen einem aktuellen Frame-Block und dem vorherigen Frame durchsucht wird, auf die Umgebung des Zielblocks begrenzt werden. Diese Begrenzung des Suchbereichs ist gewöhnlich akzeptabel, da die Bewegung eines Objekts in den meisten typischen Videosequenzen selten schnell genug ist, um eine große Verschiebung von einem Frame zum nächsten zu erzeugen. Mit einem begrenzten Suchbereich ist es möglich, eine erschöpfende Suche effizient durchzuführen, um die beste Übereinstimmung zu finden. Nachdem die beste Übereinstimmung gefunden wurde, wird der Prädiktions-Frame durch Zusammenstellen aller am besten übereinstimmenden Blöcke konstruiert. Um dies in Hardware zu implementieren, wird der vorherige Frame in einem Direktzugriffsspeicher gespeichert und der Prädiktionsframe blockweise aus dem Speicher erzeugt, indem unter Verwendung des richtigen Verschiebungsvektors für diesen Block jeweils ein Pixel auf einmal gelesen wird.

**[0016]** Dieses Verfahren erzeugt einen guten Prädiktions-Frame, wenn Objekte in einer Videosequenz sowohl vertikal als auch horizontal um eine ganze Zahl von Pixeln verschoben sind. Bei einer typischen Videosequenz ist die Distanz bei Objektbewegungen gewöhnlich jedoch keine ganze Zahl von Pixeln. In den Fällen, in denen die Verschiebung zwischen zwei Pixel fällt, kann ein besserer Prädiktions-Frame erzeugt werden, indem Werte verwendet werden, die aus angrenzenden Pixeln interpoliert werden. Wenn man nur die Mittelpunkte zwischen Pixeln betrachtet, gibt es drei mögliche Interpolationsarten, d. h. horizontal, vertikal und diagonal. Die Horizontal-Interpolation besteht aus der Bildung des Mittelwerts zweier horizontal angrenzender Pixel. Die Vertikalinterpolation wird durch Berechnen des Mittelwerts zwischen zwei vertikal angrenzenden Pixeln erzeugt. Die Diagonalinterpolation erfordert die Bildung des Mittelwerts vier benachbarter Pixel. Ein Beispiel für einen Halbpixel-Interpolationsprozessor für ein bewe-

gungskompensierte digitale Videosystem findet sich in der eigenen US-Patentanmeldung 08/009,831, registriert am 27. 1. 1993 und erteilt als US-Patent 5,398,079.

**[0017]** In diesem Dokument wird ein Verfahren zur Verarbeitung vorheriger Frame-Daten in einen bewegungskompensierten digitalen Videosystem zur Interpolation von bei der Rekonstruktion eines gegenwärtigen Video-Frames verwendeten Pixeln beschrieben, mit den folgenden Schritten: Anordnen von Pixeldaten aus einem vorherigen Video-Frame zu Subblöcken, die eine Vielzahl N von Pixeln enthalten; Speichern der Subblöcke in einem Speicher, um das Abrufen mindestens eines Subblocks pro Speicherzugriffszyklus zu ermöglichen; Abrufen einer Vielzahl M der Subblöcke aus dem Speicher als Reaktion auf einen Verschiebungsvektor, der einem Teil eines gegenwärtigen Video-Frames zugeordnet ist, wobei  $M \leq N$  ist und die M Subblöcke in höchstens M Speicherzugriffszyklen abgerufen werden; Wählen einer Teilmenge von Pixeln aus der gewählten Vielzahl von Subblöcken zur Verwendung bei der Interpolation von Pixeln für den gegenwärtigen Video-Frame-Teil; und Verarbeiten der Teilmenge von Pixeln über höchstens N Speicherzugriffszyklen hinweg, um N interpolierte Pixel für den gegenwärtigen Video-Frame-Teil bereitzustellen.

**[0018]** Außerdem wird in diesem Dokument eine Vorrichtung zum Interpolieren von Pixeln, die bei der Rekonstruktion eines gegenwärtigen Video-Frames verwendet werden, aus Pixeln eines vorherigen Video-Frames in einem bewegungskompensierten digitalen Videosystem offen gelegt.

**[0019]** Die Prädiktionsberechnung, die bei einem Videodekompressionsprozessor, der Bewegungskompensation verwendet, erforderlich ist, ist eine der schwierigsten Dekodierungsaufgaben, insbesondere wenn Interpolation erforderlich ist. Im Idealfall ist ein VLSI-Design für einen Dekompressionsprozessor schnell, klein, einfach und speicherbandbreiteneffizient. Konzeptuell wäre es der einfachste Ansatz zur Implementierung einer Prädiktionsberechnungsfunktion, alle zur Berechnung der Prädiktion notwendigen Daten in einer einfachen sequenziellen Reihenfolge einzulesen und dann jegliche erforderliche Interpolationsfilterung durchzuführen. Ein solcher simplistischer Ansatz ist jedoch aus verschiedenen Gründen nachteilig. Wenn die Hardware wartet, bis alle Daten eingelesen worden sind, bevor sie mit der Filterungsfunktion beginnt, sind große Mengen Speicherung erforderlich. Außerdem ist nur eine feste Zeit verfügbar, in der die Prädiktion berechnet werden muss. Wenn es notwendig ist, alle Daten einzulesen, bevor die Filterung durchgeführt wird, ist nur ein kurzer Zeitraum übrig, um die Filterung selbst durchzuführen. Generell ist mehr Hardware notwendig, wenn weniger Zeit zur Durchführung einer Berechnung verfügbar ist.

Wenn Daten in einer sequenziellen Reihenfolge gelesen werden, kommt es weiterhin zu vielen Reihenänderungen und schlechter DRAM-E/A-Bandbreite.

**[0020]** Aus EP-A-0 572 263 sind eine Vorrichtung und ein Verfahren zum Dekodieren eines Codestroms von Codes variabler Länge gemäß einem Befehl aus einer Zentralverarbeitungseinheit mit einem Speichersystem, das eine Vielzahl von Wörtern zur Speicherung von Steuerinformationen und eines dekodierten Werts für jeden der Code variabler Länge bekannt.

**[0021]** Aus EP-A-0 50356 sind eine Vorrichtung und ein Verfahren zum Dekomprimieren kodierter komprimierter Videosignale mit Mitteln zum Abrufen komprimierter Videosignale aus dem Speicher und zum Bereitstellen der komprimierten Videosignale auf einem Bus bekannt.

**[0022]** Eine der vorliegenden Erfindung zugrunde liegende Aufgabe ist die Bereitstellung eines Videodekompressionsprozessors, bei dem die DRAM-E/A-Bandbreite verbessert ist.

**[0023]** Diese Aufgabe wird erfindungsgemäß gelöst durch den Gegenstand der Ansprüche 1 und 9.

#### KURZE DARSTELLUNG DER ERFINDUNG

**[0024]** Gemäß der vorliegenden Erfindung wird ein Verfahren zum Interpolieren von Pixeldaten, die in Mehrpixel-Wörtern, die in Reihen eines Video-Frames angeordnet sind, bereitgestellt. Ein erstes Set von mindestens einem der Wörter, die von einer Reihe des Video-Frames interpoliert werden sollen, wird ausgewählt. Die Pixeldaten für jedes Wort in dem Set werden in eine Interpolationsrichtung (z. B. horizontal), die durch die Reihe definiert wird, interpoliert. Die interpolierten Pixeldaten für das erste Set Wörter werden vorübergehend in einem Speicher gespeichert. Ein nachfolgendes Set von mindestens einem der Wörter, die aus einer nächsten sukzessiven Reihe des Video-Frames interpoliert werden sollen, wird dann ausgewählt. Das nachfolgende Set Wörter befindet sich angrenzend an das erste Set Wörter, so dass jedes Wort des nachfolgenden Sets ein Gegenstück im ersten Set aufweist. Die Pixeldaten für ein Wort im nachfolgenden Set werden in derselben Interpolationsrichtung (z. B. horizontal), die für die Pixeldaten in dem ersten Set verwendet wird, interpoliert. Die interpolierten Pixeldaten für das Wort im ersten Set, das das Gegenstück des interpolierten Wortes aus dem nachfolgenden Set ist, werden vom Speicher gelesen und (z. B. vertikal) mit ihrem interpolierten Pixeldatengegenstück aus dem nachfolgenden Set gemittelt. Während die Gegenstücke gemittelt werden, werden die interpolierten Pixeldatengegenstücke aus dem nachfolgenden Set an Stelle der interpolierten Pixeldatengegenstücke des ersten

Sets in dem Speicher gespeichert. Auf diese Weise kann ein kleiner Speicher (wie zum Beispiel ein einfaches Doppelport-Registerfile) zur Speicherung eines Teils horizontal interpolierter Pixeldaten aus einer ersten Reihe in dem Prädiktionsbereich verwendet werden, während die Pixeldatengegenstücke aus der nächsten Reihe horizontal interpoliert werden, und die beiden horizontal interpolierten Teile können dann wortweise vertikal interpoliert werden, während die jüngsten horizontal interpolierten Daten die gerade verwendeten vorherigen horizontal interpolierten Daten überschreiben.

**[0025]** Die Interpolation von Pixeldaten aus jeder sukzessiven Reihe in dem Prädiktionsbereich wird auf diese Weise fortgesetzt, bis die Pixeldaten in allen Reihen eines gewünschten Teils des Prädiktionsbereichs interpoliert worden sind.

**[0026]** Das Verfahren der vorliegenden Erfindung kann so durchgeführt werden, dass weniger oft die Reihenadresse geändert werden muss, wenn Pixeldaten aus einem Frame-Speicher, in dem die Pixeldaten gespeichert sind, abgerufen werden. Insbesondere wird ein Prädiktionsbereich eines Video-Frames, aus dem Pixeldaten interpoliert werden sollen, definiert. Es wird bestimmt, ob der Prädiktionsbereich Pixeldaten aus mehr als einer Seite eines Frame-Speichers, in dem der Video-Frame gespeichert ist, umfasst. Falls der Prädiktionsbereich Pixeldaten aus mehr als einer Seite des Frame-Speichers umfasst, werden sukzessive Sets Wörter, die interpoliert werden sollen, ausgewählt, so dass alle Pixeldaten in dem Prädiktionsbereich aus einer ersten Seite des Frame-Speichers interpoliert werden, bevor die Pixeldaten aus einer nachfolgenden Seite des Frame-Speichers interpoliert werden.

**[0027]** Um Pixeldaten aus einer ersten Seite bereitzustellen, die beim Interpolieren von Pixeldaten in einer angrenzenden zweiten Seite notwendig sind, ohne zu der ersten Seite in dem Rahmen-Speicher zurückzuschreiten, werden die Pixeldaten, die an eine Seitenabgrenzung der ersten Seite angrenzen, vorübergehend in einem separaten Speicher gespeichert. Dies ermöglicht die Interpolation an die Seitenabgrenzung angrenzender Pixeldaten in einer Seite des Rahmen-Speichers, die gerade interpoliert wird, ohne den Rahmen-Speicher neu zu adressieren, um in die erste Seite zurückzuschreiten.

**[0028]** Das Verfahren der vorliegenden Erfindung ist auch nützlich bei der Bereitstellung horizontal und vertikal interpolierter Pixeldaten zur Verwendung bei der Rekonstruktion bidirektionaler (B)-Rahmen. Pixeldaten werden für einen Prädiktionsbereich eines Innen-Frame-(I)- oder Prädiktions-Frame-(P)-Bildes (d. h. eines „Ankerbildes“) und möglicherweise für einen Prädiktionsbereich eines zweiten I- oder P-Bildes zur bidirektionalen Interpolation interpoliert. Die

interpolierten Pixeldaten für eines der I- oder P-Bilder werden vorübergehend in einem bidirektionalen Datenspeicher gespeichert, während die Pixeldaten für das andere der I- und P-Bilder interpoliert werden. Die vorübergehend gespeicherten interpolierten Pixeldaten aus dem bidirektionalen Datenspeicher werden mit den interpolierten Pixeldaten für das andere der I- und P-Bilder Bemittelt, während letztere interpoliert werden. Die gemittelten interpolierten I- und P-Pixeldaten werden dann zur nachfolgenden Verwendung bei der Dekodierung eines bidirektionalen Bildes in den bidirektionalen Datenspeicher geschrieben. Die gemittelten interpolierten I- und P-Pixeldaten können an Stelle der daraus gelesenen interpolierten I- oder P-Daten in den bidirektionalen Datenspeicher geschrieben werden.

**[0029]** Durch das Verfahren der vorliegenden Erfindung erhältene horizontal und vertikal interpolierte Pixeldaten können abgeschnitten werden. Ein solches Abschneiden ist besonders zum Runden der Subpixelergebnisse von Null weg nützlich.

**[0030]** Außerdem wird ein Subpixelinterpolationsfilter für einen Videobewegungsschätzprozessor offen gelegt. Es wird ein Mittel zur Erzeugung eines ersten Sets von N Subpels durch Bilden des Mittelwerts von angrenzenden Pixeln, die in einem ersten Vektor von  $(N + 1)$  Pixeln innerhalb einer Reihe eines Prädiktionsbereichs enthalten sind, bereitgestellt. Das erste Set Subpels wird vorübergehend gespeichert. Es wird ein Mittel zur Erzeugung eines zweiten Sets von N Subpels durch Bilden des Mittelwerts von angrenzenden Pixeln, die in einem zweiten Vektor von  $(N + 1)$  Pixeln aus einer nachfolgenden Reihe des Prädiktionsbereichs enthalten sind, bereitgestellt. Der zweite Vektor in der nachfolgenden Reihe ist das Gegenstück des ersten Vektors in der ersten Reihe. Von jedem Subpel aus dem gespeicherten ersten Set Subpels wird der Mittelwert mit einem Subpelgegenstück aus dem zweiten Set gebildet, wenn die Subpelgegenstücke im zweiten Set erzeugt werden. Diese Mittelwertbildung liefert ein drittes Set Subpels, das die Pixeldaten, die in zwei Dimensionen interpoliert wurden, z. B. horizontal und vertikal, repräsentiert.

**[0031]** Das Subpixelinterpolationsfilter kann weiterhin ein Mittel zum Speichern jedes Subpels aus dem zweiten Set im Speichermittel an Stelle seines Subpelgegenstücks aus dem ersten Set, wobei zur gleichen Zeit der Mittelwert des zweiten Sets Subpels seit seinem Gegenstück des ersten Sets Subpels gebildet wird, umfassen. Auf diese Weise kann eine kleine Menge Speicher für das Speichermittel verwendet werden, da Subpels aus einer ersten Reihe des Prädiktionsbereichs unmittelbar durch Subpels aus der nächsten Reihe überschrieben werden, während die Subpel-Gegenstücke aus den beiden Reihen gemittelt werden.

**[0032]** Es kann ein Mittel zum Bestimmen, wann der Prädiktionsbereich Pixeldaten aus mehr als einer Seite eines Frame-Speichers, in dem die Pixel, die interpoliert werden, gespeichert sind, umfasst, bereitgestellt werden. Ein auf das Bestimmungsmittel reagierendes Mittel wählt sukzessive Pixelvektoren, die aus den sukzessiven Reihen innerhalb des Prädiktionsbereichs interpoliert werden sollen, so aus, dass alle Pixeldaten in dem Prädiktionsbereich aus einer ersten Seite des Frame-Speichers interpoliert werden, bevor die Pixeldaten aus einer nachfolgenden Seite des Frame-Speichers interpoliert werden. Auf diese Weise wird die Anzahl von Seitenabgrenzungen, die beim Interpolieren aller Pixel aus dem Prädiktionsbereich überschritten werden, minimiert. Pixeldaten, die an eine Seitenabgrenzung einer vorher interpolierten Seite des Frame-Speichers angrenzen, können vorübergehend gespeichert werden, damit die Pixeldaten, die angrenzend an die Seitenabgrenzung in einer Seite des Frame-Speichers liegen, die gegenwärtig interpoliert wird, interpoliert werden können, ohne die Seitenabgrenzung zu überschreiten.

**[0033]** Um die Rekonstruktion bidirektionaler Frames zu ermöglichen, können Pixeldaten aus einem Prädiktionsbereich eines I-Bildes und aus einem Prädiktionsbereich eines P-Bildes interpoliert werden. Die interpolierten Pixeldaten für das I- oder das P-Bild werden vorübergehend in einem bidirektionalen Datenspeicher gespeichert und mit den interpolierten Pixeldaten für das jeweils andere I- oder P-Bild gemittelt, während letzteres interpoliert wird. Die Bemittelten interpolierten I- und P-Pixeldaten werden zur nachfolgenden Verwendung bei der Prädiktion des B-Frames in den bidirektionalen Datenspeicher geschrieben. Um Speicher zu sparen, können die gemittelten interpolierten I- und P-Pixeldaten an Stelle der gerade daraus gelesenen interpolierten I- oder P-Daten in den bidirektionalen Datenspeicher geschrieben werden.

**[0034]** Nachfolgend werden Ausführungsformen von Interpolationsfiltern beschrieben, die nicht in den Schutzmfang der Ansprüche fallen. Es wird ein Horizontalpixelinterpolationsfilter für einen Videodekompresionsprozessor bereitgestellt, bei dem ein Eingang aufeinander folgende Wörter empfängt, die jeweils N Pixel aus einem Prädiktionsbereich eines Videoprädiktions-Frames enthalten. Die aufeinander folgenden Wörter werden während sukzessiver Taktzyklen empfangen. Es wird ein Mittel zum Verzögern der aufeinander folgenden Pixeldatenwörter um einen Taktzyklus bereitgestellt. Registermittel halten einen ersten Pixel aus jedem aufeinander folgenden Wort, das am Eingang empfangen wird, zusammen mit dem unmittelbar vorangehenden Wort aus dem Verzögerungsmittel. Mittel sind gekoppelt, um die in dem Registermittel in jedem Taktzyklus gehaltenen angrenzenden Pixel zu empfangen und zu mitteln, um sukzessive Sets horizontal interpolierter Subpels

zur Verwendung durch den Videodekompressionsprozessor bereitzustellen.

**[0035]** Es kann ein Vertikalinterpolationsfilter zum vertikalen Interpolieren der horizontal interpolierten Subpels, die durch das Horizontalpixelinterpolationsfilter bereitgestellt werden, bereitgestellt werden. Das Vertikalinterpolationsfilter enthält einen Eingang zum Empfangen sukzessiver Sets horizontal interpolierter Subpels während sukzessiver Taktzyklen. Es werden Mittel zum Puffern der sukzessiven Sets der horizontal interpolierten Subpels zum Mitteln mit vertikal angrenzenden Setgegenstücken horizontal interpolierter Subpels in dem Prädiktionsbereich bereitgestellt. Die gepufferten Sets horizontal interpolierter Subpels werden mit den Setgegenstücken Bemittelt, um sukzessive Sets horizontal und vertikal interpolierter Subpels zur Verwendung durch den Videodekompressionsprozessor bereitzustellen. Ein bidirektionales Interpolationsfilter wird zur Verwendung in Kombination mit den Horizontal- und Vertikalinterpolationsfiltern bereitgestellt. Die Horizontal- und Vertikalinterpolationsfilter werden zur Interpolation von Pixeldaten aus einem Prädiktionsbereich eines I-Bildes und aus einem Prädiktionsbereich eines P-Bildes verwendet. Es werden Mittel zum vorübergehenden Speichern interpolierter Pixeldaten für das I- oder das P-Bild in einem bidirektionalen Datenspeicher und zum Mitteln dieser mit den interpolierten Pixeldaten für das andere I- oder P-Bild, während letztere durch die Horizontal- und Vertikalinterpolationsfilter interpoliert werden, bereitgestellt. Die Bemittelten interpolierten I- und P-Pixeldaten werden zur nachfolgenden Verwendung bei der Dekodierung eines B-Frames in den bidirektionalen Datenspeicher geschrieben.

**[0036]** Die Mittelwertbildungsmittel des Horizontalpixelinterpolationsfilters können eine Vielzahl an die Registermittel angekoppelter Addieren umfassen. Jeder Addieren addiert zwei angrenzende Pixel aus dem Registermittel, um eine Summe zu bilden. Jeder Addieren enthält einen Carry-Eingang, der so auslegt ist, dass er eine binäre „1“ zu der Summe addiert. Es werden Mittel zum Abschneiden der Summe um ihr niedrigstwertiges Bit, so dass ein Subpel entsteht, das von Null weg gerundet ist, bereitgestellt.

**[0037]** Das Horizontalinterpolationsfilter kann in Verbindung mit einem Direktzugangsspeicher (RAM) verwendet werden, in dem den Videoprädiktions-Frame als ein Set Tiles gespeichert wird. Jedes Tile enthält Pixeldaten und nimmt eine Seite (d. h. Reihe) des RAM ein. Das Interpolationsfilter umfasst weiterhin Speichermittel zum Speichern einer Teilmenge von Wörtern aus dem Prädiktionsbereich, der an eine erste Seite eines Vertikalseitenumbruchs angrenzt. Es werden Mittel zum Eingeben eines Worts aus den Speichermitteln in das Registermittel an Stelle des Eingabens des unmittelbar vorausgehenden Worts aus den Verzögerungsmitteln, wenn Pixel, die an

eine zweite Seite des Vertikalseitenumbruchs angrenzen, interpoliert werden, bereitgestellt. Auf diese Weise kann die notwendige Anzahl von Adressierungen verschiedener Seiten in dem Direktzugriffsspeicher minimiert werden.

**[0038]** Es werden Horizontal-, Vertikal- und bidirektionale Interpolationsfilter offen gelegt. Diese Filter können abhängig von den Anforderungen des Videodekompressionsprozessors allein oder in Kombination verwendet werden. Obwohl der Ausdruck „horizontal“ im Allgemeinen als Bezug auf Reihen einer Matrix und der Ausdruck „vertikal“ im Allgemeinen als Bezug auf die Spalten einer Matrix verwendet wird, ist nicht beabsichtigt, diese Ausdrücke in der vorliegenden Offenlegung und in den Ansprüchen derartig zu begrenzen, und es versteht sich, dass diese Ausdrücke soweit notwendig, um die allgemeinste mögliche Interpretation der hier offen gelegten Erfindung bereitzustellen, vertauschbar sind.

#### KURZE BESCHREIBUNG DER ZEICHNUNGEN

**[0039]** **Fig. 1** ist ein Blockschaltbild eines Videodekompressionsprozessors gemäß der vorliegenden Erfindung;

**[0040]** **Fig. 2** zeigt als Diagramm, wie ein Luminanz-Video-Frame gemäß der vorliegenden Erfindung in Seiten aufgeteilt wird;

**[0041]** **Fig. 3** ist eine Diagrammdarstellung einer Luminanzseite;

**[0042]** **Fig. 4** ist eine Diagrammdarstellung einer Chrominanzseite;

**[0043]** **Fig. 5** ist eine Diagrammdarstellung eines ungünstigsten Falls, in dem das Lesen von Daten aus einem Prädiktionsbereich den Zugriff auf vier verschiedene Speicherreihen erfordert;

**[0044]** **Fig. 6** ist ein Blockschaltbild eines Horizontalinterpolationsfilters gemäß der vorliegenden Erfindung;

**[0045]** **Fig. 7** ist ein Blockschaltbild eines Filters zur Bereitstellung einer vertikalen und bidirektionalen Interpolation gemäß der vorliegenden Erfindung;

**[0046]** **Fig. 8** ist eine Diagrammdarstellung der Funktionsweise des Horizontal-Subpelfilters von **Fig. 6** in einer Situation, in der der Prädiktionsbereich keinen vertikalen Seitenumbruch aufweist;

**[0047]** **Fig. 9** und **10** sind Diagrammdarstellungen der Funktionsweise des Horizontal-Subpelfilters von **Fig. 6** in einer Situation, in der ein Prädiktionsbereich einen vertikalen Seitenumbruch enthält; und

**[0048]** **Fig. 11** ist eine Diagrammdarstellung der Funktionsweise des Vertikal-Subpelfilters von **Fig. 7**.

#### AUSFÜHLICHE BESCHREIBUNG DER ERFINDUNG

**[0049]** **Fig. 1** ist ein Blockschaltbild eines Videodekompressionsprozessors, der einen Speichermanager **30** enthält, der externen DRAM **22** so adressiert, dass die Anzahl von Reihenänderungen minimiert wird, während Prädiktionsdaten aus dem DRAM gelesen werden. Der Prozessor mit der allgemeinen Kennzeichnung **20** ist ein Pipeline-Prozessor, der dafür ausgelegt ist, sowohl die Transportschicht (d. h. Steuer- und andere Nicht-Video-Informationen) als auch die Videoschicht der über den Anschluss **10** eingegebenen komprimierten Bitstromeingaben, die manchmal als „Transportpaketschnittstelle“ des Videoprozessors bezeichnet werden, zu dekodieren.

**[0050]** An dem Anschluss **14** ist eine Benutzerprozessorschnittstelle zur Steuerung des Videodatenprozessors über eine M-Bus-Steuerung **50**, die verschiedene Register in dem Prozessor **20** konfiguriert, vorgesehen. Der M-Bus ist ein bidirektonaler serieller Zweileitungsbus, der ein einfaches und effizientes Mittel zum Datenaustausch zwischen Einrichtungen bereitstellt und vollständig mit dem Busstandard I<sup>2</sup>C kompatibel ist.

**[0051]** Über die Adressenleitungen **24** und die Datenleitungen **26** wird eine Schnittstelle zu dem DRAM **22** bereitgestellt. Bei der in **Fig. 1** dargestellten spezifischen Ausführungsform enthält der DRAM **22** einen Neun-Bit-Adressenport und einen Zweiunddreißig-Bit-Datenport.

**[0052]** Eine Videoausgabeschnittstelle **38** wird für die dekomprimierten rekonstruierten Videosignale bereitgestellt, die zum Beispiel als gemultiplextes Luminanz-(Y-) und Chrominanz-(Cr-, Cb-)Signal gemäß dem Standard CCIR (International Radio Consultive Committee) 656 mit acht Bit und siebenundzwanzig Mhz ausgegeben werden können.

**[0053]** Über den Anschluss **62** kann eine Prüfschnittstelle zu einer JTAG-Steuerung (Joint Test Action Group) **60** bereitgestellt werden. JTAG ist eine standardisierte Boundary-Scan-Nlethodologie, die für das Prüfen auf Leiterplattenebene zur Erkennung von Fehlern bei Gehäuse- und Leiterplattenverbindungen sowie internen Schaltkreisen verwendet wird.

**[0054]** Der Videodekompressionsprozessor **20** empfängt über den Anschluss **12** ein Taktsignal. Der Takt liefert Zeitsteuerungsinformationen, um z. B. es einem Transportsyntaxparser **32** zu ermöglichen, Zeitsteuerungsinformationen und Videoinformationen aus Transportpaketen in dem komprimierten Bit-

stroneingangssignal über den Anschluss **10** wiederherzustellen. Eine Erfassungs- und Fehlermanagementschaltung **34** verwendet eine Programmtaktreferenz (PCR) und einen Dekodierungszeitstempel (DTS), der von einem Videosyntaxparser **40** erkannt wird, um den Start der Bilddekodierung zu synchronisieren. Diese Schaltung stellt die Vertikalsynchronisation ein und liefert globale Synchronisation für alle Videodekodierungs- und Anzeigefunktionen.

**[0055]** Die Videoschicht wird in einem Eingangspuffer (FIFO) gepuffert, der durch den Speichermanager **30** in dem externen DRAM **22** konfiguriert wird. Der Videosyntaxparser **40** empfängt die aus dem DRAM-FIFO ausgegebenen komprimierten Videodaten über den Speichermanager **30** und trennt die Bewegungsvektorinformationen von den Koeffizienten, die die Videoinformationen beschreiben. Die Koeffizienten werden durch einen Huffman-Dekodierer **52**, einen Invers-Quantisierer **54** und einen Prozessor **56** für inverse diskrete Kosinustransformation (IDCT) verarbeitet.

**[0056]** Bewegungsvektoren werden wiederhergestellt und zum Adressieren zuvor dekodierter Video-Frames benutzt, die zum Rekonstruieren eines gegenwärtigen Video-Frames erforderlich sind. Insbesondere dekodiert ein Bewegungsvektordekodierer **42** die aus dem Videosyntaxparser **40** empfangenen Bewegungsvektoren und leitet sie zu einem Prädiktionsadressengenerator **44** weiter. Der Prädiktionsadressengenerator liefert Adresseninformationen, die notwendig sind, um über den Speichermanager **30** die notwendigen Daten des Ankerframes (d. h. I- oder P-Frames) abzurufen, um es dem Prädiktionskalkulator **46** zu ermöglichen, ein Prädiktionssignal bereitzustellen, das zum Rekonstruieren eines gegenwärtigen Frame-Blocks notwendig ist. Der Differenzdekodierer **48** kombiniert die Prädiktionsdaten mit den dekodierten Koeffizientendaten, um dekomprimierte Videodaten bereitzustellen. Die dekomprimierten Daten werden über den Speichermanager **30** in entsprechenden Puffern des DRAM **22** gespeichert.

**[0057]** Es versteht sich, dass, obwohl die durch den Bewegungsvektordekodierer **42**, den Prädiktionsadressengenerator **44**, den Prädiktionskalkulator **46**, den Differenzdekodierer **48**, den Huffman-Dekodierer **52**, den Invers-Quantisierer **54** und die IDCT **56** ausgeführten Videodekompressionsprozesse im Allgemeinen üblich sind und von Fachleuten wohl verstanden werden, die spezifische Filterungsvorrichtung bzw. das spezifische Filterungsverfahren, die von dem Prädiktionskalkulator **46** zur Interpolation von Pixeldaten verwendet werden, neu sind und die Grundlage der vorliegenden Erfindung bilden. Die einzigartigen Aspekte des Prädiktionskalkulators werden nachfolgend ausführlicher beschrieben.

**[0058]** Der Speichermanager **30** plant alle Aktivitäten an der externen DRAM-Adresse und den Datenbussen **24**, **26** ein und adressiert effizient den DRAM **22**. Der Speichermanager stellt sicher, dass die Datentransferanforderungen des Eingangs-FIFO-Teils des DRAM **22**, des Videosyntaxparsers **40** und der Videorekonstruktionsschaltung **36** (sowie des Prädiktionskalkulators **46** und des Differenzdekodierers **48**) alle erfüllt sind. Die Videorekonstruktionsschaltung **36** berechnet ein gegenwärtiges Bild und fügt Einblendungen, ein Vertikalintervalltestsignal (VITS) und Testmusterdaten zur Ausgabe auf der Videoausgangsleitung **38** ein. Die Ausgangsanzeige wird durch Vergleichen des PCR mit einem Präsentationszeitstempel (PTS) synchronisiert. Eine Bestimmung, wann das Dekodieren und Anzeigen eines Video-Frames beginnen muss, erfolgt durch Vergleichen eines Dekodierungszeitstempels (DTS) mit dem PTS.

**[0059]** Der Speichermanager stellt außerdem abhängig von dem Dekodierungsmodus, der zum Beispiel NTSC oder PAL mit oder ohne bidirektionale Prädiktions-Frames (B-Frames) sein kann, eine variable Größe für den FIFO-Teil des DRAM **22** bereit. Die Videopuffersteuerung stellt sicher, dass der von dem DRAM **22** bereitgestellte FIFO nicht über- oder unterläuft. Die Puffersteuerung ist eine Funktion von Systemzeitsteuerungsparametern wie zum Beispiel des PCR und DTS.

**[0060]** Der DRAM **22** ist als ein externer Speicher dargestellt und kann von mehreren DRAM-Chips, wie z. B. zwei Vier-Mbit- (Megabit, d. h.  $2^{20}$  Bit) DRAMs bei einer Acht-Mbit-Implementierung oder durch vier Vier-Mbit-DRAMs bei einer Sechzehn-Mbit-Implementierung, bereitgestellt werden. Es versteht sich, dass bei zukünftigen Implementierungen und mit fortschreitender Speichertechnologie der DRAM **22** als interner Speicher in dem Videodekompressionsprozessor bereitgestellt werden kann. Der DRAM wird so abgebildet, dass verschiedene Dekodierungs- und Ausgangsvideopuffer sowie ein zirkularer FIFO für den komprimierten Eingangsvideobitstrom bereitgestellt werden. Der DRAM kann außerdem zur Bereitstellung eines Testmusterpuffers, eines VITS-Puffers und eines Einblendungsanzeigemordnungspuffers sowie zur Speicherung verschiedener Bildstrukturdaten, die zur ordnungsgemäßen Anzeige der dekodierten Video-Frames notwendig sind, dienen. Der DRAM kann über den Speichermanager **30** neu initialisiert werden, um verschiedene Speicherpläne bereitzustellen, die erforderlich sind, wenn Variablen modifiziert werden, wie zum Beispiel die Video-Frame-Vertikalgröße, PAL- oder NTSC-Video, das Vorhandensein von Testmustern, Acht- oder Sechzehn-Mbit-Speicherkonfiguration und ob B-Frames vorliegen.

**[0061]** Der Speichermanager **30** plant alle Aktivität

auf den externen DRAM-Bussen, einschließlich der Datentransferanforderungen des Eingangs-FIFO, des Videoparsers und der Videorekonstruktionsschaltung. Der Speichermanager führt außerdem auf herkömmliche Weise die erforderliche DRAM-Auffrischung durch. Zum Beispiel kann gleichzeitig dieselbe Reihe in jedem von zwei oder vier externen DRAMs aufgefrischt werden.

**[0062]** Wenn ein komprimierter Bitstrom in den Anschluss **10** des Videodekompressionsprozessors **20** eingegeben wird, werden durch den Bitstrom dargestellte Video-Frames einzeln rekonstruiert. Zu Anfang muss ein voller Frame Videodaten empfangen und in dem DRAM **22** gespeichert werden sein. Informationen für nachfolgende Video-Frames können eine Teilmenge des vollen Video-Frames umfassen, die, wenn sie zu Prädiktionsdaten aus dem vorherigen Video-Frame (der in DRAM **22** gespeichert ist) hinzugefügt wird, zu der Rekonstruktion eines vollen Frames führt. Während jeder neue Video-Frame eines Innenbildes (I) oder eines Prädiktionsbildes (P) rekonstruiert wird, wird er in dem DRAM **22** zur Verwendung als Ankerframe gespeichert, der notwendig ist, um einen nachfolgenden Frame, der durch Daten in dem komprimierten Bitstrom dargestellt wird, zu rekonstruieren, und zur Anzeige an dem entsprechenden Zeitpunkt in der Videosequenz. Während jeder bidirektional vorhergesagte Frame (B-Frame) rekonstruiert wird, wird er zur Anzeige am entsprechenden Zeitpunkt gespeichert. Der hier beschriebene Videodekompressionsprozessor nutzt den Umstand aus, dass der größte Teil (d. h. ungefähr 80%) der Daten, auf die aus dem DRAM **22** zugegriffen wird, auf rechteckigen „Prädiktionsbereichen“ von Pixeln hergestellt wird. Bei der dargestellten Ausführungsform ist jeder Prädiktionsbereich neun Pixel hoch und siebzehn Pixel breit. Die Verwendung eines Prädiktionsbereichs dieser Größe (an Stelle eines Bereichs von  $8 \times 16$  Pixeln, der zwei  $8 \times 8$ -Datenblöcken entspricht) ermöglicht eine Erzielung der Subpixelinterpolation, da eine Reihe und eine Spalte von Pixeln zu einem Set von zwei  $8 \times 8$ -Pixelblöcken hinzugefügt wird. Das Verhältnis von Höhe zu Breite (9 : 17) des Prädiktionsbereichs beträgt ungefähr 0,53. Um die Anzahl notwendiger Reihenüberschreitungen beim Adressieren des DRAM **22** zu minimieren, werden vorherige Frame-Daten in dem DRAM auf der Grundlage von Tiles mit einem ähnlichen Verhältnis von Höhe zu Breite (z. B. etwa 0,50) gespeichert. **Fig. 2** zeigt eine Zuteilung von Tiles in einem Video-Frame, die gemäß der vorliegenden Erfindung verwendet werden kann.

**[0063]** Wie in dem Beispiel von **Fig. 2** gezeigt, wird ein Luminanz-NTSC-Video-Frame **70** in 165 „Tiles“ gleicher Größe unterteilt. Jedes Tile **72** enthält acht Makroblöcke (MB) in zwei Slices von jeweils vier Makroblöcken. Die 165 Tiles werden in einem Array von fünfzehn Reihen organisiert, wobei jede Reihe elf Tiles enthält. Jede Reihe enthält zwei Slices, so dass

sich insgesamt dreißig Slices in jedem Video-Frame ergeben. In jedem Slice sind 44 Makroblöcke enthalten. Da es 165 Tiles gibt, die jeweils 8 Makroblöcke enthalten, gibt es insgesamt 1320 Makroblöcke in jedem NTSC-Video-Frame. Jeder Luminanz-Makroblock enthält vier Blöcke von  $8 \times 8$  Pixeln. Jeder Pixel umfasst 8 Datenbit. Der Video-Frame wird so aufgeteilt, dass jedes Tile die Datenmenge enthält, die in einer Reihe des DRAM **22** gehalten werden kann. Es versteht sich, dass für andere Fernsehformate (z. B. PAL oder SECAM) unter Verwendung derselben Grundprinzipien etwas unterschiedliche Spezifikationen resultieren.

**[0064]** Bei der gezeigten Ausführungsform hält jede Reihe des DRAM **22** 512 Zweiunddreißig-Bit-Wörter, so dass sich insgesamt 16.384 Bit ergeben. Jedes Tile **72** des Video-Frames sollte also 16.384 Bit enthalten. Da jeder Makroblock vier  $8 \times 8$ -Luminanzblöcke enthält, umfasst der Makroblock 256 Acht-Bit-Pixel, so dass sich insgesamt 2048 Luminanzbit pro Makroblock ergeben. Jedes Tile enthält acht solche Makroblöcke, so dass sich insgesamt 16.384 Luminanzbit ergeben. Jedes Tile **72** des in **Fig. 2** dargestellten Luminanz-Video-Frames wird deshalb auf einer Reihe des DRAM **22** abgebildet.

**[0065]** **Fig. 3** zeigt ein Luminanz-Tile **72** ausführlicher. Wie gezeigt, umfasst das Luminanz-Tile Teile zweier verschiedener Slices des Video-Frames. In jedem Slice-Teil werden vier Makroblöcke (die jeweils vier  $8 \times 8$ -Luminanzblöcke enthalten) bereitgestellt. Da jede DRAM-Speicherstelle ein Zweiunddreißig-Bit-Wort speichert, können an jeder DRAM-Speicherstelle vier Acht-Bit-Luminanzpixel (Y) gespeichert werden, wie durch das Vier-Pixel-Wort **80** in **Fig. 3** dargestellt. Wie bereits erwähnt, befinden sich in jeder Reihe des DRAM 512 Spalten, die jeweils durch eine der hexadezimalen RAM-Adressen 0–FF adressierbar sind.

**[0066]** Die Chrominanzdaten werden ähnlich behandelt, mit der Ausnahme, dass jeder Video-Frame halb so viele Chrominanzinformationen wie Luminanzinformationen enthält. Wie in **Fig. 4** dargestellt, enthält deshalb jedes Chrominanz-Tile **82** Teile aus vier Slices des Video-Frames, wobei jeder Slice-Teil vier Chrominanz-Makroblöcke enthält. Die Chrominanz-Makroblöcke sind halb so groß wie die Luminanz-Makroblöcke, und an Stelle von acht Makroblöcken pro Tile enthält ein Chrominanz-Tile deshalb sechzehn Makroblöcke. Wie die Luminanz-Tiles wird jedes Chrominanz-Tile auf eine Reihe des DRAM **22** abgebildet. Bei einer bevorzugten Implementierung wird der DRAM **22** in separate Bereiche zum Speichern der Luminanzdaten und der Chrominanzdaten aufgeteilt, wobei die Chromaabschnitte im Allgemeinen halb so groß wie die Luminanzabschnitte sind. Wie durch das Vier-Pixel-Wort **84** dargestellt, werden die Chrominanzdaten als abwechselnde cb- und

cr-Chrominanzkomponenten gespeichert.

**[0067]** **Fig. 5** zeigt, wie die Ein-Tile-Pro-Reihe-Abbildung, die gemäß der vorliegenden Erfindung verwendet wird, die Anzahl der Speicherreihenadressenänderungen, die notwendig sind, um eine Gruppe von Pixeln in einem Prädiktionsbereich aus dem DRAM zu lesen, minimieren kann. Teile der vier angrenzenden Tiles **74**, **75**, **77** und **79** (von links oben in dem Video-Frame **70** von **Fig. 2** aus) sind in **Fig. 5** dargestellt. Der Einfachheit halber ist das gesamte Tile **74** in **Fig. 5** dargestellt, aber nur Teile der Tiles **75**, **77** und **79**. Die vertikale Schnittmenge der Tiles **74**, **75**, **77** und **79** liegt entlang einer vertikalen „Seiten“-Abgrenzung **92** des DRAM **22**. Die horizontale Schnittmenge dieser Tiles verläuft entlang einer horizontalen „Seiten“-Abgrenzung **94** des DRAM. Da in jeder Reihe (d. h. „Seite“) des DRAM **22** ein Tile gespeichert ist, muss jedesmal, wenn eine vertikale oder horizontale Seitenabgrenzung überschritten wird, eine neue Reihe des DRAM adressiert werden.

**[0068]** Um die Pixeldaten für einen aktuellen Frame aus Pixeldaten eines in dem DRAM **22** gespeicherten vorherigen Frames vorherzusagen, werden die Pixel des vorherigen Frames für jede in dem gegenwärtigen Frame vorherzusagende Gruppe von Pixeln aus einem vorbestimmten Prädiktionsbereich gelesen. Der Prädiktionsbereich für jede Gruppe von Pixeln wird unter Verwendung von Bewegungsvektoren gefunden, die auf herkömmliche Weise mit den komprimierten Videodaten übertragen werden. Ein Prädiktionsbereich **90** im ungünstigsten Fall ist in **Fig. 5** dargestellt. Der Prädiktionsbereich **90** stellt ein ungünstigstes Szenario dar, da er Pixel aus vier verschiedenen Tiles abdeckt. Wenn dieser Prädiktionsbereich gelesen werden müsste, indem zum Beispiel einfach aufeinander folgende Reihen darin gescannt werden, wäre es notwendig, wiederholt die vertikale Seitenabgrenzung **92** zu überschreiten, wodurch auf verschiedene Reihen des DRAM zugegriffen würde. Dadurch würde die Geschwindigkeit, mit der Daten aus dem DRAM gelesen werden könnten, signifikant verringert.

**[0069]** Bei der hier dargestellten Ausführungsform wird der Prädiktionsbereich **90** auf eine Weise gescannt, die nur vier DRAM-Reihenadressierungsschritte erfordert, d. h., eine neue Reihenadresse ist nur einmal für jedes der Tiles **74**, **75**, **77** und **79**, die in dem Prädiktionsbereich enthalten sind, erforderlich. Um dies zu erreichen, werden alle Daten aus jedem Tile gelesen, bevor zu dem nächsten Tile übergegangen wird. Die dargestellte konkrete Ausführungsform verwendet ein Zickzack-Scan-Muster, das in der oberen linken Ecke des Prädiktionsbereichs beginnt, um alle Daten aus dem Tile **74**, das in dem Prädiktionsbereich **90** verankert ist, abzurufen. Dann wird die Reihe des DRAM **22**, die das Tile **75** des Video-Frames enthält, gescannt, bis alle Daten in dem

Prädiktionsbereich aus diesem Tile abgerufen sind. Dann wird auf die dem Tile **77** entsprechende DRAM-Reihe zugegriffen, und alle Daten, die in dem Tile **77** in dem Prädiktionsbereich **90** verankert sind, werden abgerufen. Als letztes wird auf die das Tile **79** enthaltende DRAM-Reihe zugegriffen, und alle Daten in dem Prädiktionsbereich **90** für dieses Tile werden abgerufen. Anstatt wiederholt auf verschiedene Reihen in dem DRAM zuzugreifen, müssen also im ungünstigsten Fall nur vier DRAM-Reihenzugriffe vorgenommen werden, um die Daten aus einem gesamten Prädiktionsbereich wiederherzustellen.

**[0070]** **Fig. 6** ist ein Blockschaltbild eines Horizontal-Subpelfilters gemäß der vorliegenden Erfindung. Das Filter von **Fig. 6** empfängt Pixel aus dem DRAM, die durch den Prädiktionsbereich begrenzt werden. Die horizontale Interpolation der Pixel wird unter Verwendung einer In-Place-Berechnung zum größtmöglichen Ausmaß bereitgestellt, um die Hardwareanforderungen zu verringern. Addieren sind vorgesehen, um angrenzende Pixel zusammenzufügen und sie danach durch Dividieren durch die Anzahl summierter Pixel (z. B. vier bei diagonaler Interpolation und zwei bei horizontalen oder vertikaler Interpolation) zu mitteln. Die tatsächliche Division erfolgt in den nachfolgend in Verbindung mit **Fig. 7** beschriebenen Abschneideschaltung **150**, wobei in den Addierern **114**, **116**, **118**, **120** und **148** eine Vorbetonung erfolgt. Insbesondere wird vor der Division ein Carry-Eingang jedes Addierers verwendet, um eine binäre „1“ zu der Summe zu addieren. Das niedrigstwertige Bit ( $\div 2$ ) oder die beiden niedrigstwertigen Bit ( $\div 4$ ) der Summe werden dann abgeschnitten, um die Division durchzuführen und gleichzeitig das Subpixelergebnis von Null weg zu runden. Diese neuartige Verwendung des Carry-Eingangs spart Hardware und vergrößert die Verarbeitungsgeschwindigkeit der Subpelfilterung.

**[0071]** Im ungünstigsten Fall lautet die MPEG-Gleichung für einen Prädiktionspixel folgendermaßen:

$$p = (((a + b) + (c + d)) \div 4) + (((e + f) + (g + h)) \div 4) \div 2$$

wobei  $\div$  eine Division mit Rundung auf die nächste ganze Zahl bedeutet und halb ganzzahlige Werte von Null weg gerundet werden.

**[0072]** Diese Gleichung kann folgendermaßen umgeschrieben werden:

$$p = (((a + b) + (c + d) + 2) \gg 2) + (((e + f) + (g + h) + 2) \gg 2) + 1 \gg 1$$

**[0073]** Die letzte Gleichung kann folgendermaßen umgeschrieben werden:

$$p = (((a + b + 1) + (c + d + 1)) \gg 2) + (((e + f + 1) + (g + h + 1)) \gg 2) + 1 \gg 1$$

**[0074]** Das Subpixelfilter von **Fig. 6** verwendet die obige Endgleichung, da das Addieren von 1 viel einfacher als das Addieren von größeren Zahlen als 1 ist. Bei der Berechnung der Prädiktion ist es notwendig, Zwischen- und Endergebnisse von Null weg zu runden. Das Runden einer vorzeichenbehafteten Zweierkomplementzahl von Null weg erfordert in der Regel die Durchführung einer Vorzeichenprüfung, gefolgt durch eine zusätzliche Addition mit anschließendem Abschneiden. Sei dem Filter von **Fig. 6** wird die Rundungsarithmetik jedoch mit der Mittelwertbildungs-/Filterungsarithmetik gefaltet, um eine Struktur zu erhalten, die eine minimale Anzahl von Hardwareaddierern verwendet. Da die Pixeldaten, die Beimittelt werden, immer positiv sind, ist es überflüssig geworden, das Runden negativer Zahlen zu steuern. Somit müssen nur die positiven Zwischenergebnisse gerundet werden. Wie bereits erwähnt, dient der Carry-Eingang der Addierer (d. h. der Addieren **114**, **116**, **118**, **120** und **148**) zur Implementierung der durch die obige Gleichung erforderten Additionen von Eins.

**[0075]** Daten aus dem DRAM **22** von **Fig. 1** werden über den Speichermanager **30** (mm\_data) zu dem Eingangsanschluss **100** des Horizontal-Subpelfilters von **Fig. 6** geroutet. Im einfachsten Fall (d. h., wenn keine Seitenbrüche bewältigt werden müssen und der Prädiktionsbereich am Anfang eines neuen Worts beginnt) werden die Daten wortweise eingelesen. Jedes Wort umfasst vier Pixel. Zum Beispiel kann ein in **Fig. 3** dargestelltes Vier-Pixel-Luminanzwort **80** in den Anschluss **100** eingegeben werden, oder es kann ein in **Fig. 4** dargestelltes Vier-Pixel-Chromianzwort **84** eingegeben werden. Die eingegebenen Wörter werden über ein Register **134**, das über den Weg **102** und den Multiplexer **132** das mm\_data empfängt, um einen Taktzyklus verzögert. Das verzögerte Wort wird dann in einen Schieber und Pixelselektor **106** eingegeben. Die Schaltung **106** gibt die vier Pixel aus dem verzögerten mm\_data-Wort (das als mm\_data1 bezeichnet wird) zusammen mit einem ersten Pixel aus dem gegenwärtigen, in den Anschluss **100** eingegebenen mm\_data-Wort aus. Bei einer bevorzugten Ausführungsform wird die Schiebefunktion der Schaltung **106** durch einen Barrel-Schieber bereitgestellt, der so konfiguriert ist, dass er wie dargestellt nach links schiebt (d. h. eine „Linksverschiebung“).

**[0076]** Wenn der Prädiktionsbereich keinen vertikalen Seitenbruch umfasst, werden die vier Pixel in jedem sukzessiven Wort in der richtigen Reihenfolge in dem Schieberteil der Schaltung **106** gespeichert. Die Leitungen **110** und **112** aus dem Schieber enthalten jeweils denselben Pixel. Diese Anordnung schließt jegliche Notwendigkeit, die Pixel umzuordnen, während sie aus dem Schieber **106** an die Addieren **114**, **116**, **118** und **120** ausgegeben werden, aus.

**[0077]** In dem in **Fig. 6** gezeigten Beispiel enthält ein erstes Wort Pixeldaten die Pixel  $A_1, A_2, A_3$  und  $A_4$  (mm\_data1). Gleichzeitig wird das verzögerte Wort mm\_data1 in den Schieber eingegeben, der erste Pixel „ $A_5$ “ des nächsten Worts (empfangen über den Eingangsanschluss 100) wird darin eingegeben. Die Linksverschiebungs- und Pixelselektorschaltung 106 speichert den Pixel  $A_5$  in der letzten Zelle ihres Schieberteils. Während jedes sukzessive Vier-Pixel-Wort am Anschluss 100 ankommt, wird der erste Pixel dieses Worts in der letzten Zelle des Schiebers gespeichert, während die vier Pixel aus dem vorherigen Wort (verzögert durch das Register 134) in die ersten vier Zellen des Schiebers geladen werden. Somit ist es möglich, sukzessive fünf Pixel auf einmal zu verarbeiten, um in jedem Taktzyklus vier interpolierte Subpels bereitzustellen, nachdem das erste Wort aus dem Prädiktionsbereich eingegeben worden ist.

**[0078]** Der erste Schritt, der notwendig ist, um die Pixel zu mitteln, die aus der Linksverschiebungs- und Pixelselektorschaltung 106 bereitgestellt werden, wird durch die Addieren 114–120 durchgeführt. Wenn die Schieberschaltung 106 die Pixel  $A_1$ – $A_5$  enthält, berechnet der Addierer 114 wie in **Fig. 6** dargestellt die Summe der Pixel  $A_1$  und  $A_2$ , wie bei 122 angegeben. Der Addierer 116 berechnet die Summe der Pixel  $A_2$  und  $A_3$ , wie bei 124 angegeben. Der Addierer 118 berechnet die Summe der Pixel  $A_3$  und  $A_4$ , wie bei 126 angegeben. Der Addieren 120 berechnet die Summe der Pixel  $A_4$  und  $A_5$ , wie bei 128 angegeben.

**[0079]** Jeder der Addierer 114–120 ist ein herkömmlicher digitaler Addierer mit einem (nicht gezeigten) Carry-Eingang, wie er in der Technik wohl bekannt ist. Die vorliegende Erfindung verwendet den Carry-Eingang auf neuartige Weise, indem er für jede Addition betätigt wird, um Eins zu der Summe der in den Addierer eingegebenen Pixel zu addieren. Um die Mittelwertbildung der beiden Pixel abzuschließen, wird das niedrigstwertige Bit der Summe abgeschnitten (nachdem der Carry-Eingang zum Addieren einer binären „1“ verwendet wurde). Die Abschneidung nach der Addition einer binären „1“ dient zum Runden des Mittelwerts von Null weg. Die Abschneidung kann direkt nach den Addierern 114–120 in **Fig. 6** durchgeführt werden oder anschließend, wie nachfolgend in Verbindung mit **Fig. 7** besprochen wird.

**[0080]** Die Funktionsweise des Horizontal-Subpelfilters von **Fig. 6** über vier sukzessive Zyklen hinweg ist in **Fig. 8** für den Fall dargestellt, dass kein Seitenumbruch in dem Prädiktionsbereich besteht. Wie in **Fig. 8** dargestellt, begrenzt ein Prädiktionsbereich 160 einen Bereich, der siebzehn Pixel breit und neun Pixel hoch ist. Dies entspricht zwei  $8 \times 8$ -Blöcken, wobei für Horizontal- und Vertikalinterpolation eine zusätzliche Reihe und eine zusätzliche Spalte hinzugefügt wird. Normalerweise beginnt das Lesen von Pixeln aus dem Prädiktionsbereich 160 in der oberen

linken Ecke des Prädiktionsbereichs (d. h. mit dem Pixel  $A_1$ ). Die Pixel werden dann in Wörtern von vier Pixeln horizontal gelesen. Wenn der Prädiktionsbereich wie dargestellt am Anfang einer Wortabgrenzung beginnt (d. h. alle Pixelwerte aus dem ersten in dem Prädiktionsbereich angetroffenen Wort sind gültig), wird das gesamte Wort in die Linksverschiebungs- und Pixelselektorschaltung 106 eingegeben. Wie in **Fig. 8** gezeigt, umfasst das erste gelesene Wort die Pixel  $A_1, A_2, A_3$  und  $A_4$ . Diese Pixel werden zum Zeitpunkt  $t_1$  wie dargestellt in den Selektor 106 eingegeben. Der Zeitpunkt  $t_1$  ist ein Zyklus nach dem Beginn des Lesens von Daten aus dem Prädiktionsbereich 160. Während des vorherigen Zyklus ( $t_0$ ) wäre das erste Wort ( $A_1, A_2, A_3, A_4$ ) eingelesen und durch das Register 134 um einen Zyklus verzögert worden. Zum Zeitpunkt  $t_1$  wird dieses erste Wort also als mm\_data1 dem Selektor 106 zugeführt. Das Wort, das gegenwärtig aus dem Prädiktionsbereich 160 gelesen wird, ist das Wort, das die Pixel  $A_5, A_6, A_7$  und  $A_8$  umfasst. Dieses Wort umfasst das gegenwärtige mm\_data, und der erste Pixel davon (d. h.  $A_5$ ) wird in die niedrigstwertigen Bit des Selektors 106 eingegeben. Die Addierer berechnen dann die Mittelwerte  $A_1 + A_2, A_2 + A_3, A_3 + A_4$  und  $A_4 + A_5$ . Nach Rundung und Abschneidung umfassen diese Summen die horizontal interpolierten Subpels, die aus dem Horizontal-Subpelfilter ausgegeben werden (unter der Annahme, dass die Abschneidung in dem Horizontal-Subpelfilter selbst durchgeführt wird).

**[0081]** Zum Zeitpunkt  $t_2$  umfasst mm\_data1 die Pixel  $A_5, A_6, A_7$  und  $A_8$ . mm\_data umfasst die Pixel  $A_9, A_{10}, A_{11}$  und  $A_{12}$ . An diesem Punkt wird nur der Pixel  $A_9$  von mm\_data in den Selektor 106 eingegeben. Die Addierer berechnen die Summen  $A_5 + A_6, A_6 + A_7, A_7 + A_8$  und  $A_8 + A_9$ . Ähnlich werden im Zyklus  $t_3$  die Pixel  $A_9, A_{10}, A_{11}, A_{12}$  und  $A_{13}$  horizontal interpoliert. Im Zyklus  $t_4$  werden die Pixel  $A_{13}, A_{14}, A_{15}, A_{16}$  und  $A_{17}$  horizontal interpoliert. Der Prozess wird fortgesetzt, bis alle Pixel in dem Prädiktionsbereich 120 interpoliert worden sind. Nachdem alle Pixel in der ersten Reihe ( $A_1$ – $A_{17}$ ) interpoliert worden sind, werden genauer gesagt die Pixel in der nächsten Reihe ( $B_1$ – $B_{17}$ ) interpoliert, dann die Pixel  $C, D, \dots, I$ , bis der gesamte Prädiktionsbereich verarbeitet worden ist.

**[0082]** **Fig. 9** und 10 zeigen eine kompliziertere Situation, in der der Prädiktionsbereich 160 einen vertikalen Seitenumbruch 162 umfasst. Wie in Verbindung mit der Besprechung von **Fig. 5** erwähnt, wird ein Prädiktionsbereich, der einen Seitenumbruch umfasst, seitenweise gescannt, um die Anzahl von Reihenänderungen zu minimieren, die bei der Adressierung des DRAM 22 vorgenommen werden müssen. In den **Fig. 9** und 10 werden also die Pixel  $A_1$ – $A_3$  gelesen, gefolgt durch die Pixel  $B_1$ – $B_3$ , und dann die Pixel  $C_1$ – $C_3, \dots, I_1$ – $I_3$ . Danach werden die Pixel  $A_4$ – $A_{17}$  gelesen, gefolgt durch die Pixel  $B_4$ – $B_{17}$  und schließlich die Pixel  $I_4$ – $I_{17}$ .

**[0083]** Das Interpolationsfilter von **Fig. 6** enthält ein  $9 \times 32$ -Registerfile (z. B. Doppelportspeicher) **130** zum Speichern der neun Wörter Pixeldaten, die einem Seitenumbruch unmittelbar vorangehen. In der Darstellung von **Fig. 9** und **10** liegen nur die letzten drei Pixel (z. B.  $A_1$ ,  $A_2$  und  $A_3$ ) jedes Worts unmittelbar vor dem Seitenumbruch in dem Prädiktionsbereich. Der Seitenumbruchspeicher **130** muss jedoch groß genug sein, um neun volle Wörter unterzubringen, falls dem Seitenumbruch vier volle Pixel in jeder Reihe des Prädiktionsbereichs vorangehen. Der Multiplexer **132** wählt entweder **mm\_data** aus Anschluss **100** oder die Daten, die dem Seitenumbruch unmittelbar vorangehen, aus dem Seitenumbruchspeicher **130**, abhängig davon, ob Pixel unmittelbar nach dem Seitenumbruch (z. B. Pixel  $A_4$ – $A_7$ ) gegenwärtig verarbeitet werden.

**[0084]** Nunmehr mit Bezug auf **Fig. 9** und **10** werden in dem Zyklus  $t_1$  Pixel  $A_1$ ,  $A_2$  und  $A_3$  zur nachfolgenden Verwendung, wenn Daten aus der rechten Seite des Seitenumbruchs **162** interpoliert werden, in den Seitenumbruchspeicher **130** geladen. Dadurch kann der Pixel  $A_1$  mit dem Pixel  $A_2$  und der Pixel  $A_2$  mit dem Pixel  $A_3$  Bemittelt werden. Außerdem kann dadurch Pixel  $A_3$  mit Pixel  $A_4$  gemittelt werden. Im Zyklus  $t_2$  werden die Pixel  $B_1$ ,  $B_2$  und  $B_3$  in dem Seitenumbruchspeicher **130** gespeichert. Im Zyklus  $t_3$  werden die Pixel  $C_1$ ,  $C_2$  und  $C_3$  in dem Seitenumbruchspeicher **130** gespeichert. Dieses faden des Seitenumbruchspeichers wird für alle relevanten Pixel in dem Prädiktionsbereich links des Seitenumbruchs **162** bis zum Zyklus  $t_9$  fortgesetzt, in dem die Pixel  $I_1$ ,  $I_2$  und  $I_3$  geladen werden.

**[0085]** Nachdem alle Pixel, die dem Seitenumbruch **162** vorangehen, vorübergehend gespeichert worden sind, kann die Verarbeitung der Pixel rechts des Seitenumbruchs beginnen. Im Zyklus  $t_{10}$  (siehe **Fig. 10**) werden die Pixel  $A_1$ – $A_3$  in den Schieber **106** eingegeben. Genauer gesagt werden die Pixel  $A_1$ – $A_3$  aus dem Seitenumbruchspeicher **130** (über den Multiplexer **132** und das Register **134**) erhalten und in die Linksverschiebungs- und Pixelselektorschaltung **106** als **mm\_data1** eingegeben. Diese Pixel werden dann zu den Addierern **114**, **116** und **118** ausgegeben. Gleichzeitig empfängt die Linksverschiebungs- und Pixelselektorschaltung **106** das gegenwärtige **mm\_data**, das die Pixel  $A_4$  und  $A_5$  umfasst, zur Ausgabe an die Addierer **118** und **120**, wie dargestellt. Im Zyklus  $t_{10}$  kann also der Pixel  $A_1$  horizontal mit dem Pixel  $A_2$  gemittelt werden, der Pixel  $A_2$  horizontal mit Pixel  $A_3$ , der Pixel  $A_3$  horizontal mit Pixel  $A_4$  und der Pixel  $A_4$  horizontal mit Pixel  $A_5$ .

**[0086]** Im Zyklus  $t_{11}$  werden die Pixel  $A_5$ ,  $A_6$  und  $A_7$  in die Verschiebungs- und Pixelselektorschaltung **106** als **mm\_data1** über das Register **134** eingegeben. Diese Pixel werden lediglich in dem Register **134** um einen Taktzyklus verzögert und werden nicht aus

dem Seitenumbruchspeicher **130** erhalten wie die Pixel  $A_1$ ,  $A_2$  und  $A_3$  von links des Seitenumbruchs. Gleichzeitig werden die ersten beiden Pixel des gegenwärtigen **mm\_data** (d. h. die Pixel  $A_8$  und  $A_9$ ) in die letzte Zelle des Schiebers **106** eingegeben. Dadurch können diese Pixel horizontal interpoliert werden.

**[0087]** In den Zyklen  $t_{12}$  und  $t_{13}$  werden die übrigen „A“ Pixel in der ersten Reihe des Prädiktionsbereichs **160** horizontal interpoliert. Danach werden die übrigen Pixel in jeder der Reihen „B“, „C“, ... und „I“ des Prädiktionsbereichs auf dieselbe Weise gemittelt.

**[0088]** Während jede Reihe horizontal durch das Filter von **Fig. 6** interpoliert wird, können deren interpolierte Subpels vertikal durch das Filter von **Fig. 7** interpoliert werden. Die Horizontal-Subpeldaten (**hsp\_data**) werden über den Anschluss **140** in das Vertikalinterpolationsfilter mit der allgemeinen Kennzeichnung **145** eingegeben, sobald sie erhalten wurden.

**[0089]** In einem  $4 \times 36$ -Registerfile (z. B. Doppelportspeicher) **142** werden die Horizontal-Subpeldaten zur Verwendung bei der Bildung des Mittelwerts jedes Worts dieser Reihe mit einem nachfolgenden vertikal verschobenen Wort Subpeldaten reihenweise gespeichert. Wenn ein Wort aus der nachfolgenden Reihe ankommt, wird es in dem Addieren **148** (der aus vier getrennten Addierern ähnlich den Addierern **114**, **116**, **118** und **120** von **Fig. 6** besteht) mit den Subpels des vorherigen Wortgegenstücks, das aus dem Horizontal-Subpelspeicher **142** ausgegeben wird, gemittelt und über das Register **146** um einen Taktzyklus verzögert.

**[0090]** Ein (als Selektor verwendeter) Multiplexer **144** ist vorgesehen, um eine spezielle Situation zu behandeln, in der einem vertikalen Seitenumbruch in dem Prädiktionsbereich nur ein Wort (d. h. vier Subpels oder weniger) in jeder Reihe des Prädiktionsbereichs vorangeht. Aus Zeitsteuerungsgründen wird ein Wort (z. B. die Subpels  $A_{1+2}$ ,  $A_{2+3}$ ,  $A_{3+4}$ ,  $A_{4+5}$ ) nicht in den Horizontal-Subpelspeicher **142** geschrieben, wenn das unmittelbar folgende Wort (z. B. Subpels  $B_{1+2}$ ,  $B_{2+3}$ ,  $B_{3+4}$ ,  $B_{4+5}$ ) mit ihm gemittelt werden soll. Stattdessen wird das erste Wort direkt über den Anschluss **140** und den Multiplexer **144** in dem Register **146** gespeichert, so dass es im nächsten Taktzyklus mit dem unmittelbar folgenden Wort in dem Addierer **148** summiert werden kann, ohne dass es durch den Horizontal-Subpelspeicher **142** laufen muss. Der Multiplexer **144** kann ein Subpel entweder wie oben angegeben aus dem ankommenden **hsp\_data** oder aus dem Horizontal-Subpelspeicher **142** auswählen. Das gewählte Subpel wird in dem Register **146** bis zum nächsten Taktzyklus gehalten und dann als Eingabe (**vsp\_m\_data**) den Addierern **148** zugeführt.

**[0091]** Die Funktionsweise des Vertikal-Subpelfilters über zwei Zyklen hinweg ist in **Fig. 11** als Diagramm dargestellt. Diese Figur zeigt den Horizontal-Subpelspeicher **142** zu zwei verschiedenen Zeitpunkten  $t_a$  und  $t_b$ . Zum Zeitpunkt  $t_a$  wurden die Subpels aus einer gesamten Reihe des Prädiktionsbereichs in den Horizontal-Subpelspeicher geladen. Das Subpel  $A_{1+2}$  gibt das Subpel an, das aus der Interpolation des Pixels  $A_1$  und des Pixels  $A_2$  wie dargestellt abgeleitet wird, z. B. zum Zeitpunkt  $t_1$  in **Fig. 8**. Während jedes der vier Subpels aus einem Wort in der nächsten Reihe (der B-Reihe) erzeugt wird, wird es gleichzeitig mit dem Subpelgegenstück aus der vorherigen Reihe (der A-Reihe) in dem Addierer **148** Bemittelt.

**[0092]** Zum Beispiel wurde zum Zeitpunkt  $t_b$  das erste Wort von B-Subpels ( $B_{1+2}$ ,  $B_{2+3}$ ,  $B_{3+4}$  und  $B_{4+5}$ ) durch das Horizontalinterpolationsfilter von **Fig. 6** berechnet und aus dem Register **136** an den Anschluss **140** des Vertikalinterpolationsfilters von **Fig. 7** ausgegeben. Diese vier Subpels werden sofort in den Horizontal-Subpelspeicher **142** eingegeben und überschreiben dort die Subpels  $A_{1+2}$ ,  $A_{2+3}$ ,  $A_{3+4}$  und  $A_{4+5}$ , die an diesem Punkt vorübergehend in dem Register **146** gespeichert worden sind. Zur selben Zeit werden die Subpels  $B_{1+2}$ ,  $B_{2+3}$ ,  $B_{3+4}$  und  $B_{4+5}$  direkt in zugeordnete Eingänge des Addierers **148** eingegeben, während die Subpels  $A_{1+2}$ ,  $A_{2+3}$ ,  $A_{3+4}$  und  $A_{4+5}$  aus dem Register **146** in ihre jeweiligen Eingänge des Addierers **148** eingegeben werden. Man erinnere sich, dass der Addierer **148** eigentlich vier separate Addierer umfasst; einen für jedes der vier Subpels (ein Wort), die auf einmal erzeugt werden sollen.

**[0093]** Der Addierer **148** erzeugt die Summe der Subpels  $A_{1+2}$  und  $B_{1+2}$ ;  $A_{2+3}$  und  $B_{2+3}$ ;  $A_{3+4}$  und  $B_{3+4}$ ; und  $A_{4+5}$  und  $B_{4+5}$ . Dies ist die resultierende Vertikalinterpolation der horizontal interpolierten A- und B-Subpels für ein Wort. Während eines nächsten Zyklus (nicht gezeigt) wird das nächste Wort von B-Subpels (d. h.  $B_{5+6}$ ,  $B_{6+7}$ ,  $B_{7+8}$  und  $B_{8+9}$ ) an dem Anschluss **140** ankommen und das entsprechende Wort von A-Subpels ( $A_{5+6}$ ,  $A_{6+7}$ ,  $A_{7+8}$  und  $A_{8+9}$ ) in dem Horizontal-Subpelspeicher **142** gleichzeitig überschreiben, während sie durch den Addierer **148** zu den entsprechenden A-Subpels addiert werden. Der Prozess wird fortgesetzt, wobei die Vertikal-interpolation gleichzeitig mit der Bereitstellung jedes Worts von B-Subpels erfolgt, das zur nachfolgenden Verwendung bei der Berechnung der Mittelwerte der B- und D-Subpels in den Horizontal-Subpelspeicher **142** geschrieben wird.

**[0094]** Derselbe Prozess wird über den gesamten Prädiktionsbereich hinweg fortgesetzt, um Horizontal- und Vertikalinterpolation an Gruppen sukzessiver Wörter bereitzustellen, wobei die erforderlichen Mittelwertbildungs- und zugeordneten Berechnungen an den Daten durchgeführt werden, während die Daten

aus dem DRAM **22** gelesen werden. Die dargestellte Implementierung ermöglicht die Erreichung einer Horizontal- und Vertikalinterpolation unter Verwendung eines Minimums an Hardware. Zum Beispiel muss der Horizontal-Subpelspeicher nur eine Reihe Subpels aus einem Prädiktionsbereich halten. Er kann deshalb wirtschaftlich in einem relativ kleinen  $4 \times 36$ -Registerfile implementiert werden.

**[0095]** Es ist zu beachten, dass bei der in **Fig. 6** und **7** dargestellten Implementierung das Abschneiden der Summen, die aus den Horizontaladdierern **114–120** und den Vertikaladdierern **148** ausgegeben werden, noch nicht erfolgt ist. Somit tragen die vier Acht-Bit-Subpels jedes Worts immer noch die binäre „1“, die durch die Carry-Eingänge der Addierer **114–120** und/oder **148** hinzugefügt wurde. Die Subpels sind daher jeweils neun Bit lang, und jedes Wort beträgt vor der Abschneidung sechsunddreißig Bit, wobei nur Horizontal- oder Vertikalinterpolation erfolgt ist. Wenn sowohl Horizontal- als auch Vertikalinterpolation bereitgestellt wurde, beträgt jedes Subpel zehn Bit, um insgesamt vierzig Bit pro Wort zu ergeben. Danach werden die Wörter durch eine herkömmliche Abschneideschaltung **150** abgeschnitten, um das eine oder die zwei niedrigstwertigen Bit jedes Subpels zu löschen. Die Abschneideschaltung **150** gibt dann Zweiunddreißig-Bit-Wörter horizontal- und/oder vertikal interpolierter Pixeldaten aus.

**[0096]** Die birektionale Interpolation kann unter Verwendung eines bidirektionalen Interpolationsfilters **155** bereitgestellt werden. Die aus dem Vertikalinterpolationsfilter für einen I- oder P-Frame-Makroblockteil (z. B. jeweils zwei Blöcke von  $8 \times 8$  Subpels) ausgegebenen Subpels werden in ein Register **152** des bidirektionalen Interpolationsfilters eingegeben und mit Subpelgegenstücken aus einem entsprechenden P- oder I-Frame in den Addierern **154** Bemittelt. Die resultierenden Ankerframe-Subpeldaten werden in einem bidirektionalen Datenspeicher **156** gespeichert, der zum Beispiel ein  $64 \times 32$ -Registerfile umfassen kann, das groß genug ist, um einen vollen Makroblock Luminanz- oder Chrominanzdaten (aber nicht beides) aufzunehmen. Die gespeicherten interpolierten Ankerframe-Daten können dann entweder für spezielle, Doppelprim- oder Bidirektionalmodusprädiktion gemäß dem Kompressionsstandard DCII und/oder MPEG-2 verwendet werden.

**[0097]** Im Betrieb werden die abgeschnittenen vertikal und horizontal interpolierten Subpels aus der Abschneideschaltung **150** vorübergehend einzeln in dem Register **152** zur nachfolgenden Mittelwertbildung der I- und P-Frame-Subpelgegenstücke in dem Addierer **154** gespeichert. Bei einem Implementierungsbeispiel kann ein  $8 \times 16$ -Block Subpels zum Beispiel aus einem I-Frame in dem bidirektionalen Datenspeicher **156** gespeichert werden. Wenn der  $8 \times 16$ -Block Subpelgegenstückdaten aus dem not-

wendigen P-Frame in dem Register **152** verfügbar ist, wird der I-Frame-Block aus dem bidirektionalen Datenspeicher **156** ausgelesen und in den Addieren **154** zurückgeführt, in dem die P- und I-Teile Bemittelt werden. Der resultierende Mittelwert (vsp.bi\_data) wird sofort in dem bidirektionalen Datenspeicher **156** an derselben Speicherstelle gespeichert, aus der die I-Daten (bi\_dout) gerade gelesen wurden. Dieser Prozess wird für einen ganzen  $8 \times 16$ -Block Subpel-daten auf einmal fortgesetzt, während die Mittelwerte für den vorherigen  $8 \times 16$ -Block in dem bidirektionalen Datenspeicher **156** gespeichert werden. Die Bemittelten I- und P-Subpels können dann als die zur Rekonstruktion eines B-Frames notwendigen Prädiktionsdaten aus dem bidirektionalen Datenspeicher **156** gelesen werden. Falls keine B-Frames verarbeitet werden sollen, können die horizontal und/oder vertikal interpolierten Daten aus dem Datenspeicher **156** ausgelesen werden, ohne irgendwelche bi\_dout-Daten zu dem Addieren **154** zurückzuführen.

**[0098]** Es sollte beachtet werden, dass bei der Verarbeitung von Chromianzdaten durch die Filter von **Fig. 6** und **7** ein zusätzlicher Schritt von der Linksverschiebungs- und Pixelselektorschaltung **106** durchgeführt werden muss. Insbesondere wird es bevorzugt, die Chromadaten als verschachtelte  $c_r$ - und  $c_b$ -Pixel zu speichern. Eine solche Speicherung ist optimal für das Abrufen der Chromadaten aus dem DRAM, wenn es Zeit zum Anzeigen des rekonstruierten Videosignals ist. Die Chromapixel können jedoch zum Zeitpunkt der Prädiktionsberechnung nicht verschachtelt werden. Daher entschachtelt die Linksverschiebung und Pixelauswahl **106** die Chromadaten dergestalt, dass sie dieselbe Horizontal-Subpixel-Mittelwertbildungshardware (**Fig. 6**) durchlaufen können, die für die Luminanzdaten verwendet wird.

**[0099]** Es sollte nun erkennbar sein, dass die vorliegende Erfindung ein Verfahren und eine Vorrichtung zum interpolieren zwischen Pixeln in einem Videodekompressionsprozessor bereitstellt. Die Interpolation wird unter Verwendung von In-Place-Berechnung durchgeführt, um die Notwendigkeit zu vermeiden, alle Pixel aus einem Prädiktionsbereich zur gemeinsamen Verarbeitung auf einmal zu lesen. Stattdessen werden die Pixel als Gruppen sukzessiver Wörter verarbeitet, wobei die erforderlichen Mittelwertbildungs- und zugeordneten Berechnungen an den Daten durchgeführt werden, während die Daten gelesen werden. Statt darauf zu warten, dass alle Daten eingelesen werden, lesen und filtern das Verfahren und die Vorrichtung der vorliegenden Erfindung die Daten gleichzeitig. Es werden alle verfügbaren Taktzyklen verwendet, so dass ein sehr wirtschaftliches Design resultiert.

**[0100]** Die notwendige Menge an Speicher zur Implementierung der vorliegenden Erfindung wird ebenfalls minimiert. Von besonderer Wichtigkeit ist die

Verwendung des Seitenumbruchspeichers **130** zum Halten einer Kopie z. B. der am weitesten rechts liegenden Spalte von Pixeln vor einem Seitenumbruch und die Bereitstellung des Horizontal-Subpelspeichers **142**, um eine Kopie der vorherigen Reihe Subpels oder Pixel, die für die Vertikalinterpolation notwendig sind, zu halten. Außerdem ist die Bereitstellung des bidirektionalen Datenspeichers **156** zum Halten der Prädiktion aus einer Richtung (bei der Durchführung der bidirektionalen Interpolation) wichtig, wonach sein Inhalt mit einer Prädiktion aus der anderen Richtung gemittelt wird.

## Patentansprüche

1. Ein Verfahren zum Interpolieren von Pixeldaten, die in Mehrpixel-Wörtern, die in Reihen eines Video-Frames angeordnet sind, bereitgestellt sind, das folgende Schritte beinhaltet:

- (a) Auswählen eines ersten Sets von mindestens einem der Wörter, die von einer Reihe des Video-Frames interpoliert werden sollen;
- (b) Interpolieren der Pixeldaten für jedes Wort in dem Set in eine erste Richtung, wobei die erste Richtung entweder eine senkrechte Richtung oder eine waagerechte Richtung beinhaltet;
- (c) vorübergehendes Speichern der interpolierten Pixeldaten für das erste Set Wörter in einem Speicher;
- (d) Auswählen eines nachfolgenden Sets von mindestens einem der Wörter, die aus einer nächsten sukzessiven Reihe des Video-Frames interpoliert werden sollen, wobei sich das nachfolgende Set Wörter angrenzend an das erste Set Wörter befindet, so dass jedes Wort des nachfolgenden Sets ein Wortgegenstück im ersten Set aufweist;
- (e) Interpolieren der Pixeldaten für ein Wort im nachfolgenden Set in der ersten Richtung;
- (f) Lesen der interpolierten Pixeldaten vom Speicher für das Wort im ersten Set, das das Gegenstück des interpolierten Wortes aus dem nachfolgenden Set ist; und
- (g) Bilden des Mittelwerts der interpolierten Pixeldaten, die vom Speicher mit seinem interpolierten Pixel-datengegenstück aus dem nachfolgenden Set gelesen wurden in eine zweite Richtung, während gleichzeitigem Speichern der interpolierten Pixeldaten aus nachfolgendem Set im Speicher an Stelle der vorübergehend gespeicherten interpolierten Pixeldaten aus dem ersten Set, wobei die zweite Richtung die andere der waagerechten oder senkrechten ersten Richtung beinhaltet, so dass die erste Richtung der interpolierten Pixeldaten des ersten Sets und des nachfolgenden Sets in der zweiten Richtung interpoliert werden, während die neueste erste Richtung der interpolierten Pixeldaten die schon verwendeten vorherigen interpolierten Pixeldaten der ersten Richtung überschreibt.

2. Verfahren gemäß Anspruch 1, wobei die Schritte (e), (f) und (g) für sukzessive Wörter im nach-

folgenden Set durchgeführt werden, bis alle Wörter im nachfolgenden Set interpoliert worden sind, deren Mittelwert gebildet worden ist und sie im Speicher gespeichert worden sind, um die interpolierten Daten für das erste Set Wörter zu ersetzen.

3. Verfahren gemäß Anspruch 1 oder 2, das ferner folgende Schritte beinhaltet:

Auswählen von sukzessiven Sets Wörter, die von sukzessiven Reihen des Video-Frames interpoliert werden sollen, wobei sich jedes sukzessive Set Wörter angrenzend an das Set Wörter, das in der vorherigen Reihe ausgewählt wurde, befindet, so dass jedes Wort von jedem sukzessiven Set ein Wortgegenstück im unmittelbar vorangehenden Set aufweist; Interpolieren der Pixeldaten für ein Wort in einem gegenwärtigen sukzessiven Set in der Interpolationsrichtung;

Lesen der interpolierten Pixeldaten vom Speicher für das Wort im unmittelbar vorangehenden Set, das das Gegenstück des interpolierten Wortes aus dem gegenwärtigen sukzessiven Set ist;

Bilden des Mittelwerts der interpolierten Pixeldaten, die vom Speicher mit seinem interpolierten Pixeldatengegenstück aus dem gegenwärtigen sukzessiven Set gelesen wurden, während des gleichzeitigen Speicherns der Letztgenannten im Speicher an Stelle der Erstgenannten; und

Durchführen der vorangehenden Schritte des Interpolierens, Lesens und Bildens des Mittelwerts für sukzessive Wörter in jedem sukzessiven Set, bis alle Wörter in einem gewünschten Bereich des Video-Frames interpoliert worden sind, ihr Mittelwert gebildet worden ist und sie durch den Speicher durchgelaufen sind.

4. Verfahren gemäß Anspruch 3, das ferner folgende Schritte beinhaltet:

Definieren eines Prädiktionsbereichs des Video-Frames, aus dem Pixeldaten interpoliert werden sollen; Bestimmen, ob der Prädiktionsbereich Pixeldaten von mehr als einer Seite eines Frame-Speichers, in dem der Video-Frame gespeichert ist, umfasst; und Auswählen der ersten nachfolgenden und sukzessiven Sets Wörter, die interpoliert werden sollen, so dass alle Pixeldaten im Prädiktionsbereich von einer ersten Seite des Frame-Speichers interpoliert werden, bevor die Pixeldaten von einer nachfolgenden Seite des Frame-Speichers interpoliert werden, für den Fall, dass der Prädiktionsbereich Pixeldaten von mehr als einer Seite des Frame-Speichers umfasst.

5. Verfahren gemäß Anspruch 4, das ferner folgende Schritte beinhaltet:

vorübergehendes Speichern von Pixeldaten, die an eine Seitenabgrenzung einer vorhergehend interpolierten Seite des Frame-Speichers angrenzen, in einem separaten Speicher, damit die Pixeldaten, die angrenzend an die Seitenabgrenzung in einer aktuellen Seite des Frame-Speichers liegen, ohne die Sei-

tenabgrenzung zu überschreiten, interpoliert werden können.

6. Verfahren gemäß Anspruch 5, wobei Pixeldaten für einen Prädiktionsbereich von mindestens zwei Ankerframes interpoliert werden, das ferner folgende Schritte beinhaltet:

vorübergehendes Speichern der interpolierten Pixeldaten für einen der Ankerframes in einem bidirektionalen Datenspeicher, während des Interpolierens der Pixeldaten für die anderen Ankerframes;

Bilden des Mittelwerts der vorübergehend gespeicherten interpolierten Pixeldaten aus dem bidirektionalen Datenspeicher mit den interpolierten Pixeldaten für die anderen Ankerframes, wenn die Letztgenannten interpoliert werden; und

Schreiben der interpolierten Pixeldaten des Ankerframes, von denen der Mittelwert gebildet wurde, in den bidirektionalen Datenspeicher für nachfolgende Verwendung beim Dekodieren eines bidirektionalen (B) Bilds.

7. Verfahren gemäß Anspruch 6, wobei die interpolierten Pixeldaten des Ankerframes, von denen der Mittelwert gebildet wurde, in den bidirektionalen Datenspeicher an Stelle der interpolierten Daten des Ankerframes, die daraus gelesen wurden, geschrieben wurden.

8. Verfahren gemäß einem der vorhergehenden Ansprüche, das ferner den Schritt des Abschneidens der interpolierten Pixeldaten, von denen der Mittelwert gebildet wurde, beinhaltet.

9. Ein Sub-Pixel-Interpolationsfilter für einen Videobewegungsschätzungsprozessor, der Folgendes beinhaltet:

ein Mittel (114, 116, 118, 120) zur Erzeugung eines ersten Sets N-Subpels durch Bilden des Mittelwerts in einer ersten Richtung, die an Pixel, die in einem ersten Vektor von (N + 1)Pixeln innerhalb einer Reihe (74, 75, 77, 79) eines Prädiktionsbereichs (79) enthalten sind, angrenzt, wobei die erste Richtung entweder eine senkrechte Richtung oder eine waagerechte Richtung beinhaltet;

ein Mittel (142) zum vorübergehenden Speichern des ersten Sets von Subpels;

ein Mittel (114, 116, 118, 120) zur Erzeugung eines zweiten Sets von N-Subpels durch Bilden des Mittelwerts in der ersten Richtung, die an Pexel, die in einem zweiten Vektor von ((N + 1)Pixeln aus einer nachfolgenden Reihe des Prädiktionsbereichs enthalten sind, angrenzt, wobei der zweite Vektor in der nachfolgenden Reihe das Gegenstück des ersten Vektors in der ersten Reihe ist;

ein Mittel (148) zur Bildung des Mittelwerts jeden Subpels aus dem gespeicherten ersten Set Subpels mit einem Subpelgegenstück aus dem zweiten Set, in eine zweite Richtung, wenn die Subpelgegenstücke im zweiten Set erzeugt werden, um ein drittes Set

Subpels zu erhalten, das die Pixeldaten, die in zwei Dimensionen interpoliert wurden, repräsentiert, wobei die zweite Richtung die andere waagerechte oder senkrechte Richtung beinhaltet; und ein Mittel zum Speichern jeden Subpels aus dem zweiten Set im Speichermittel an Stelle seines Subpelgegenstücks aus dem ersten Set, wobei zur gleichen Zeit der Mittelwert des zweiten Sets Subpels mit seinem Gegenstück des ersten Set Subpels durch ein Mittelwertsbildungsmittel gebildet wird.

10. Sub-Pixel-Interpolationsfilter gemäß Anspruch 9, der ferner Folgendes beinhaltet:  
 ein Mittel (**30**) zum Bestimmen, wann der Prädiktionsbereich Pixeldaten aus mehr als einer Seite eines Frame-Speichers, in dem die Pixel, die interpoliert werden, gespeichert sind, umfasst; und  
 ein Mittel (**30**), das auf das Bestimmungsmittel zum sukzessiven Auswählen von Pixelvektoren, die aus den sukzessiven Reihen innerhalb des Prädiktionsbereichs interpoliert werden sollen, reagiert, so dass alle Pixeldaten im Prädiktionsbereich von einer ersten Seite des Frame-Speichers interpoliert werden, bevor die Pixeldaten von einer nachfolgenden Seite des Frame-Speichers interpoliert werden, wodurch die Anzahl der Seitenabgrenzungen (**92**, **94**), die überschritten werden, wenn alle Pixel aus dem Prädiktionsbereich interpoliert werden, minimiert wird.

11. Sub-Pixel-Interpolationsfilter gemäß Anspruch 10, der ferner Folgendes beinhaltet:  
 ein Mittel (**130**) zum vorübergehenden Speichern von Pixeldaten, die an eine Seitenabgrenzung einer vorangehenden interpolierten Seite des Frame-Speichers angrenzen, damit die Pixeldaten, die angrenzend an die Seitenabgrenzung in einer Seite des Frame-Speichers liegen, die gegenwärtig interpoliert wird, interpoliert werden können, ohne die Seitenabgrenzung zu überschreiten.

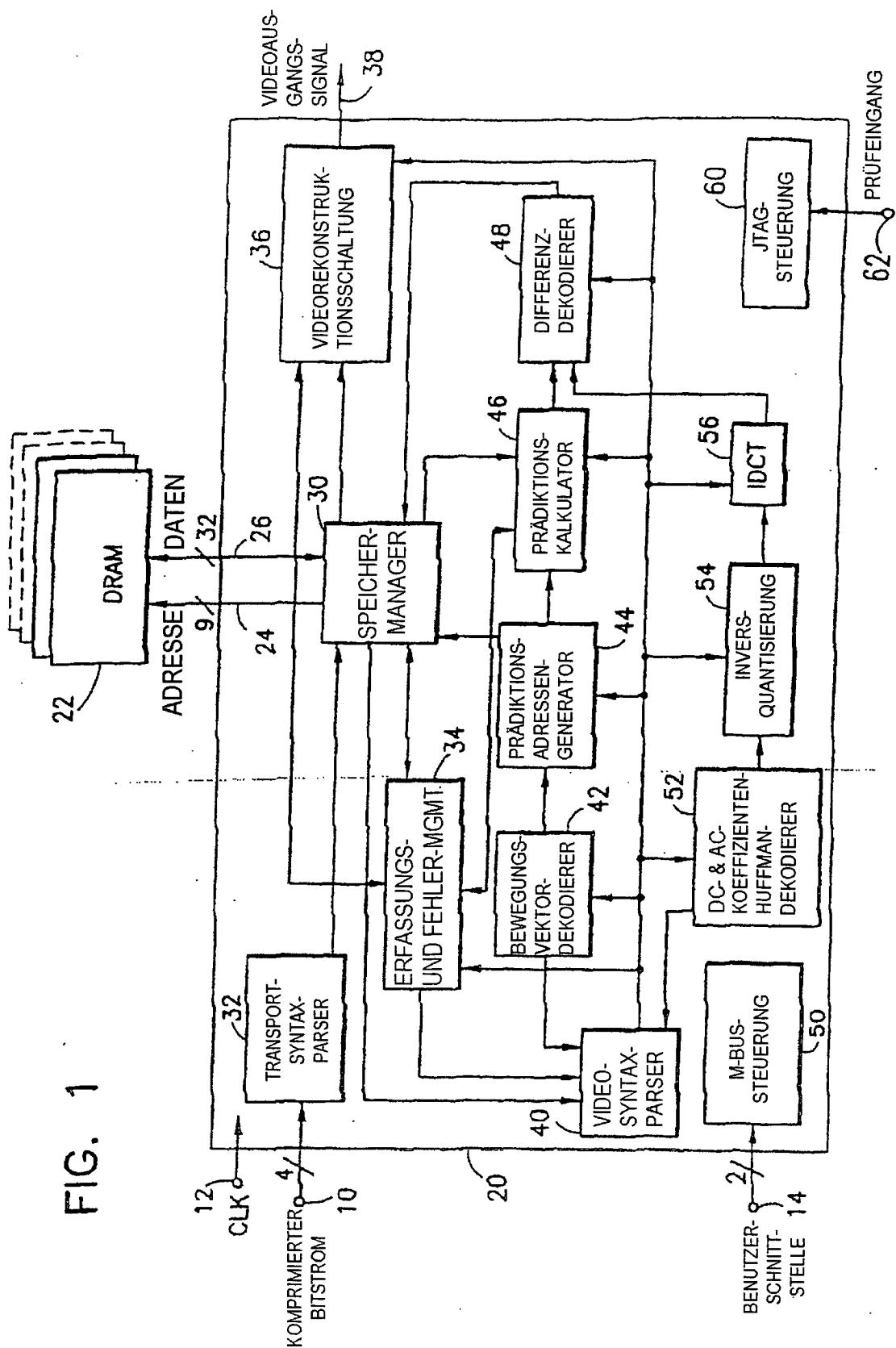
12. Sub-Pixel-Interpolationsfilter gemäß einem der Ansprüche 9 bis 11 zum Interpolieren von Pixeldaten aus einem Prädiktionsbereich eines Innen(I)-Bilds und aus einem Prädiktionsbereich eines Prädiktions-(P)-Bilds zur nachfolgenden Verwendung bei der Rekonstruktion eines bidirektionalen (B)-Frames, der ferner Folgendes beinhaltet:  
 ein Mittel (**154**, **156**) zum vorübergehenden Speichern von interpolierten Pixeldaten für eines der I- oder P-Bilder in einem bidirektionaien Datenspeicher und Bilden des Mittelwerts davon mit den interpolierten Pixeldaten für das andere der I- und P-Bilder während die Letztgenannten interpoliert werden; und ein Mittel zum Schreiben der interpolierten I- und P-Pixeldaten, deren Mittelwert gebildet wurde, in den bidirektionalen Datenspeicher zur nachfolgenden Verwendung beim Dekodieren des B-Frames.

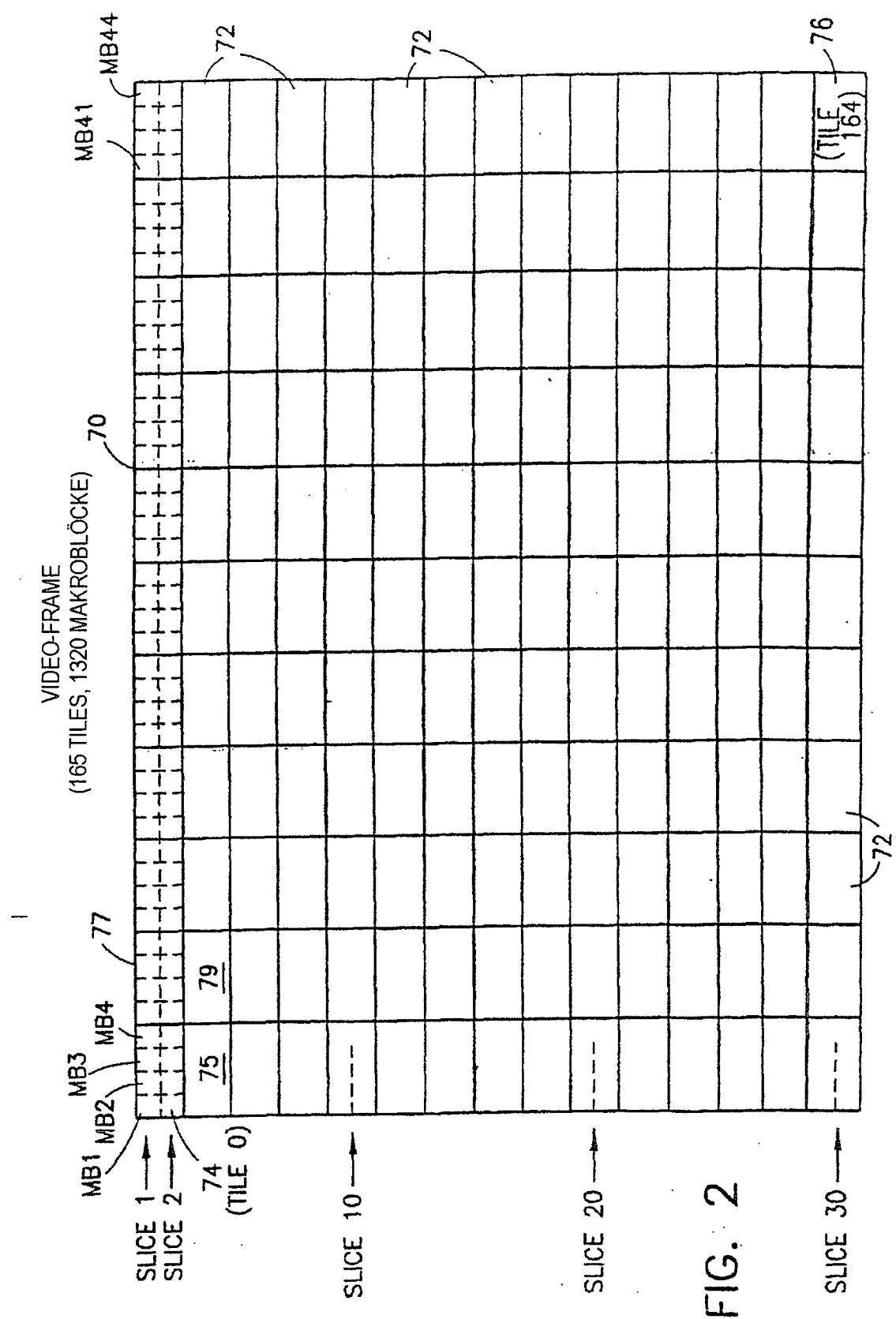
13. Sub-Pixel-Interpolationsfilter gemäß Anspruch 12, wobei die interpolierten I- und P-Pixeldaten,

ten, deren Mittelwert gebildet wurde, in den bidirektionalen Datenspeicher an Stelle der interpolierten I- oder P-Daten, die daraus gelesen wurden, geschrieben wird.

Es folgen 11 Blatt Zeichnungen

1  
FIG.





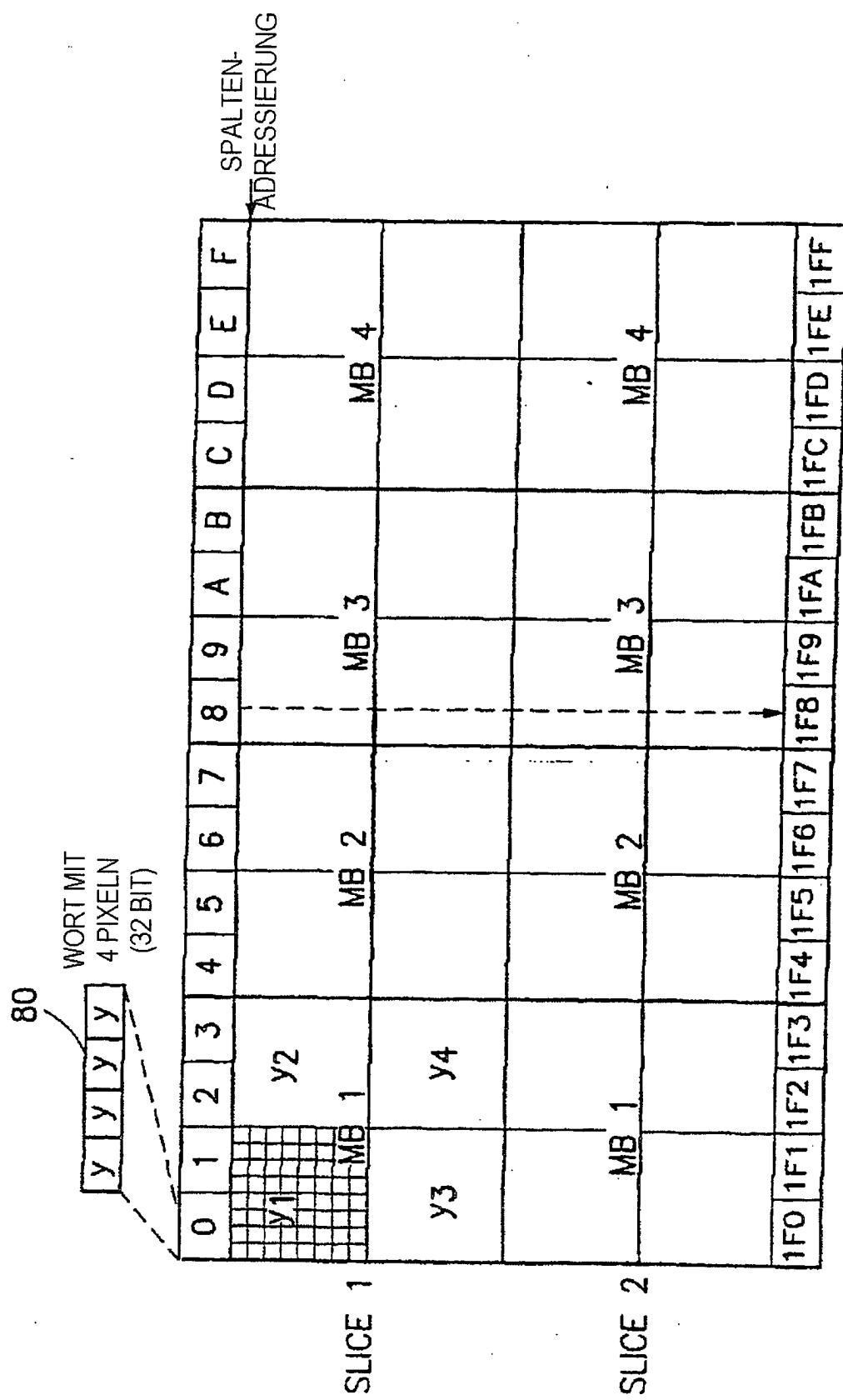


FIG. 3

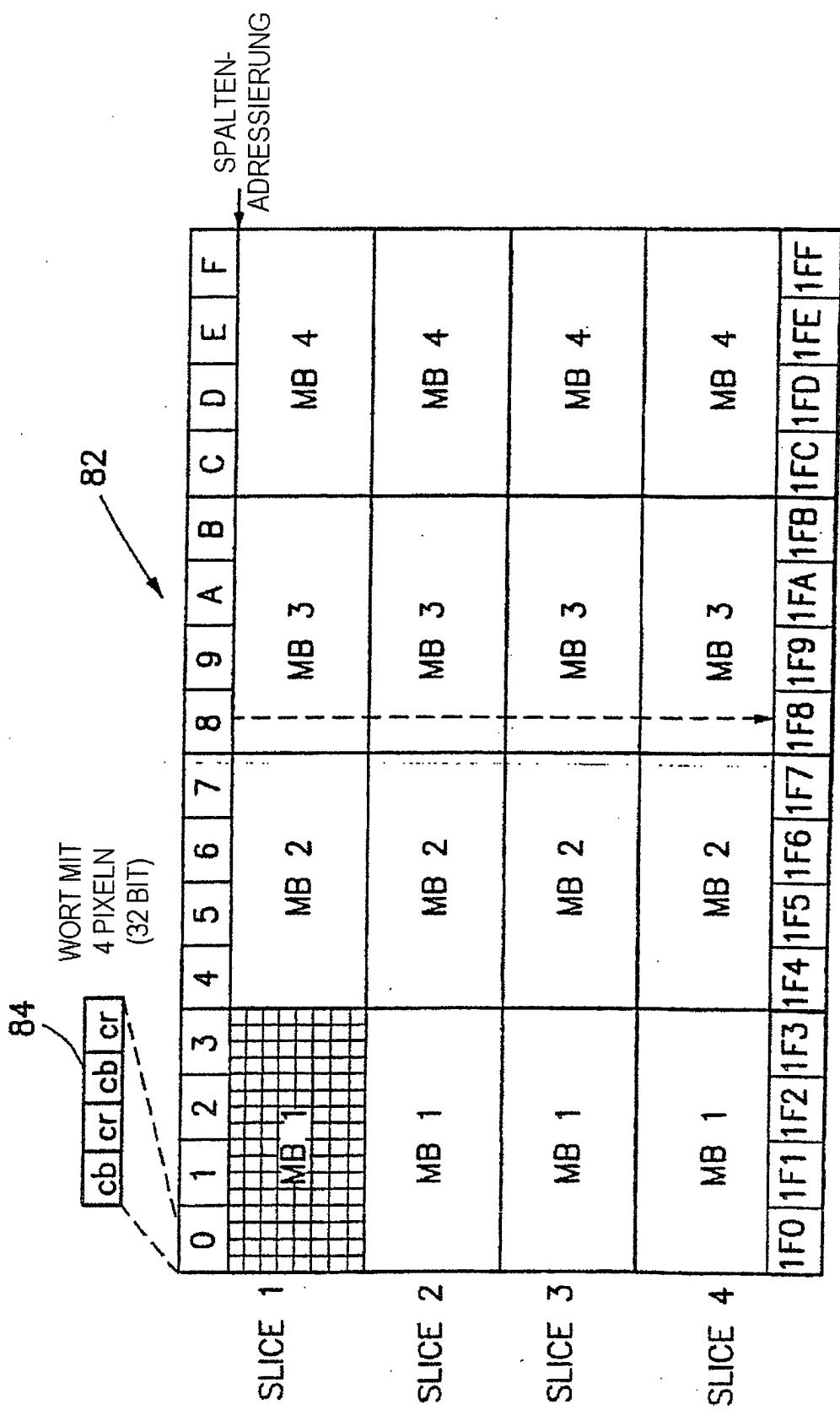
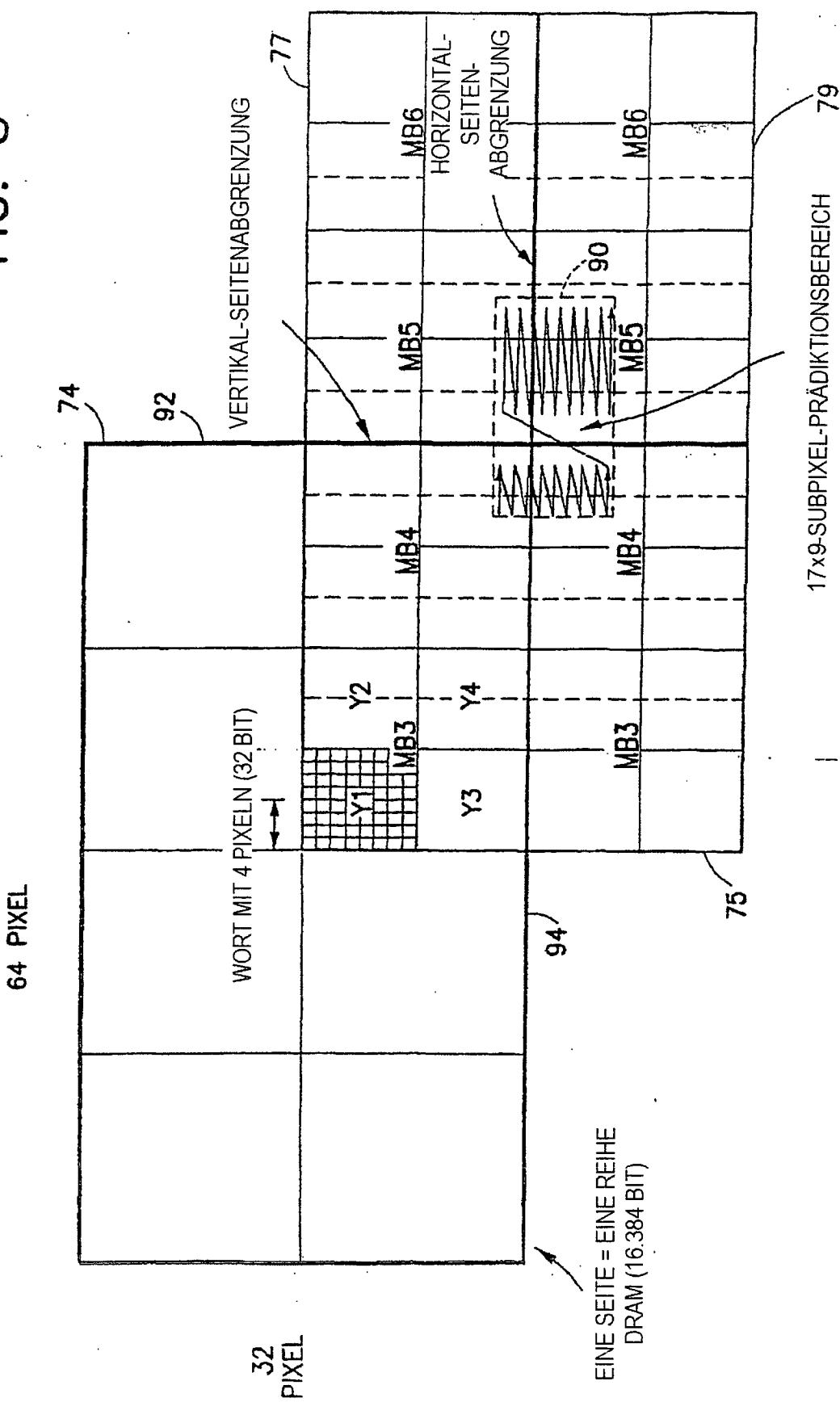


FIG. 4

FIG. 5



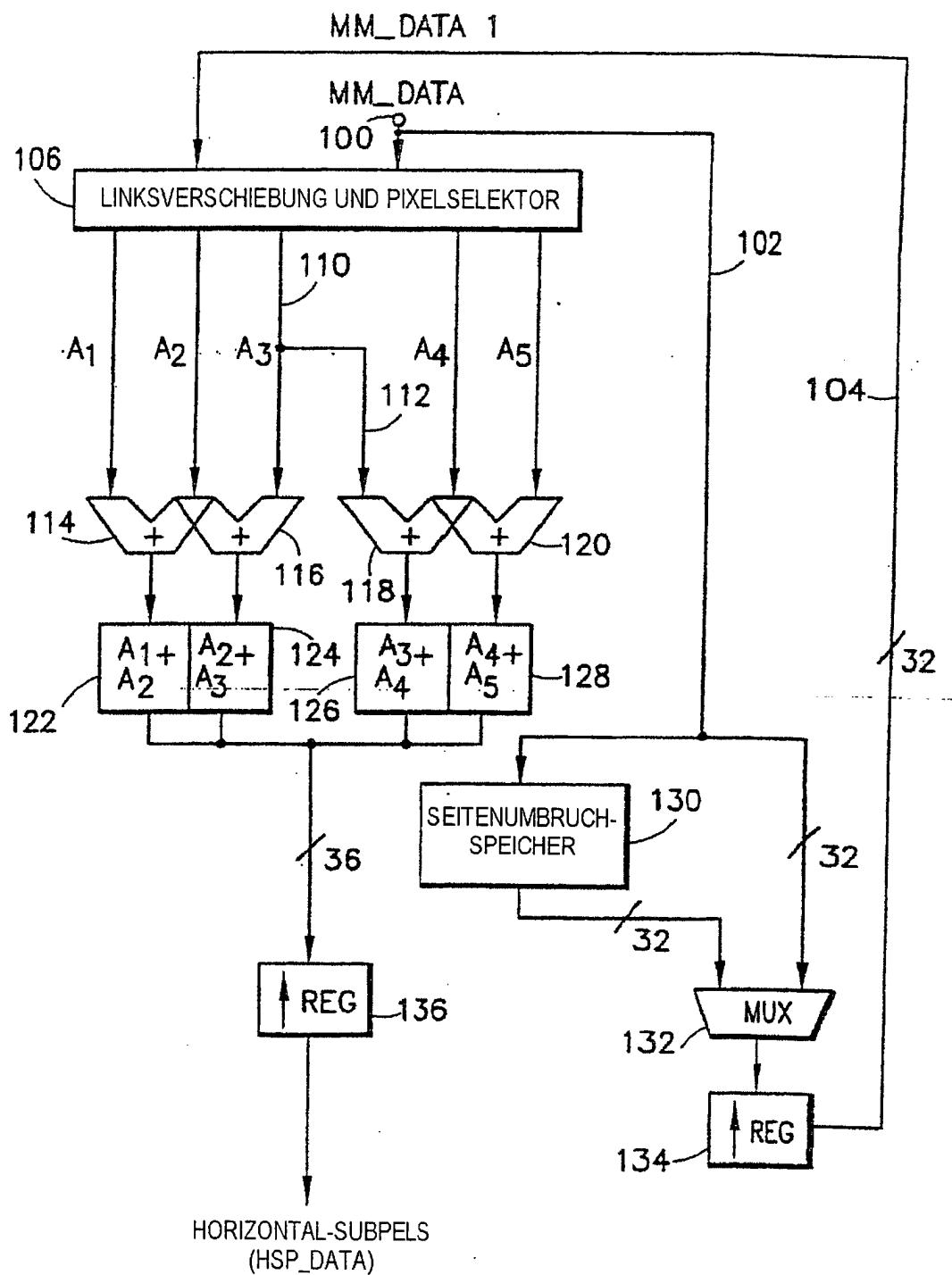


FIG. 6

FIG. 7

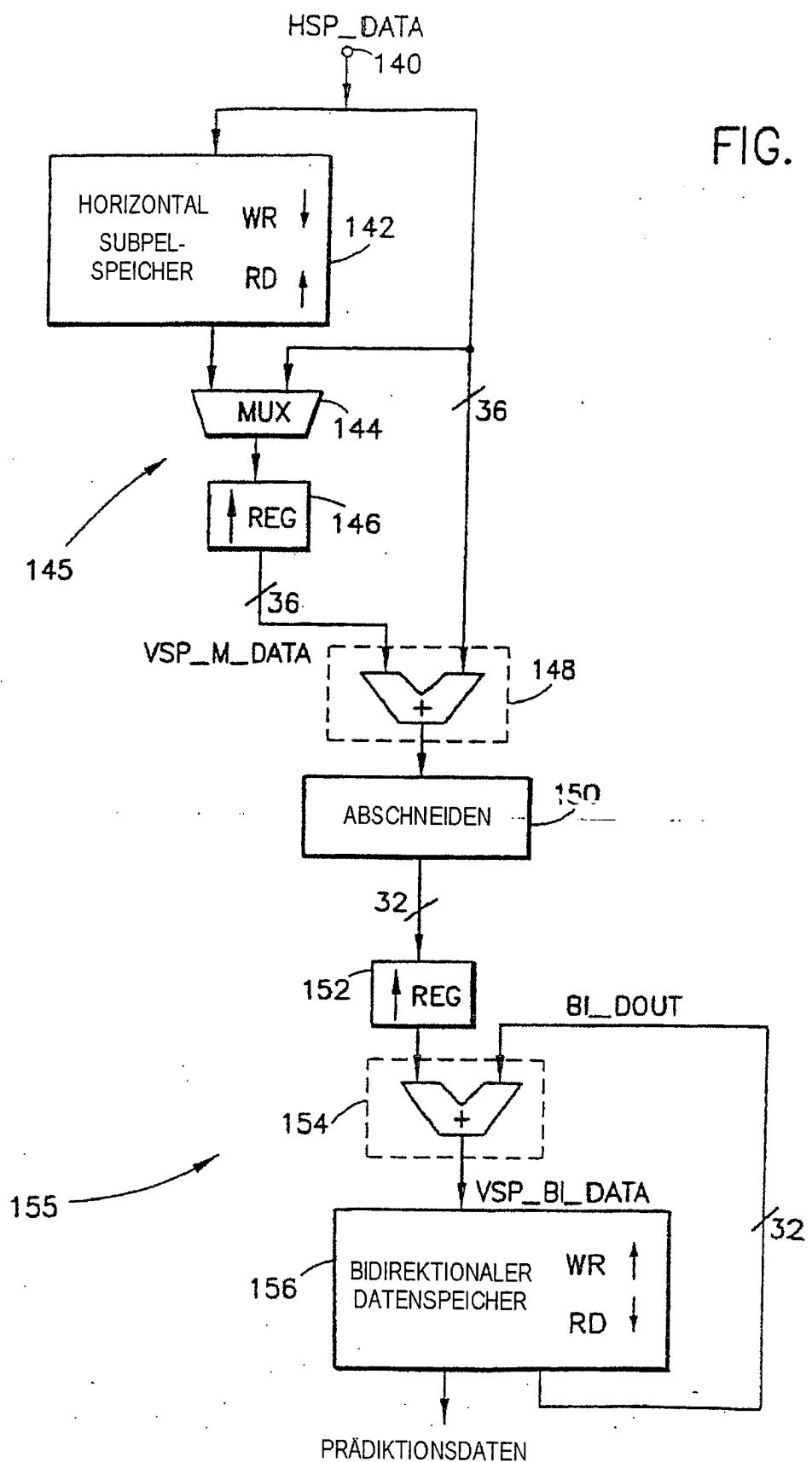
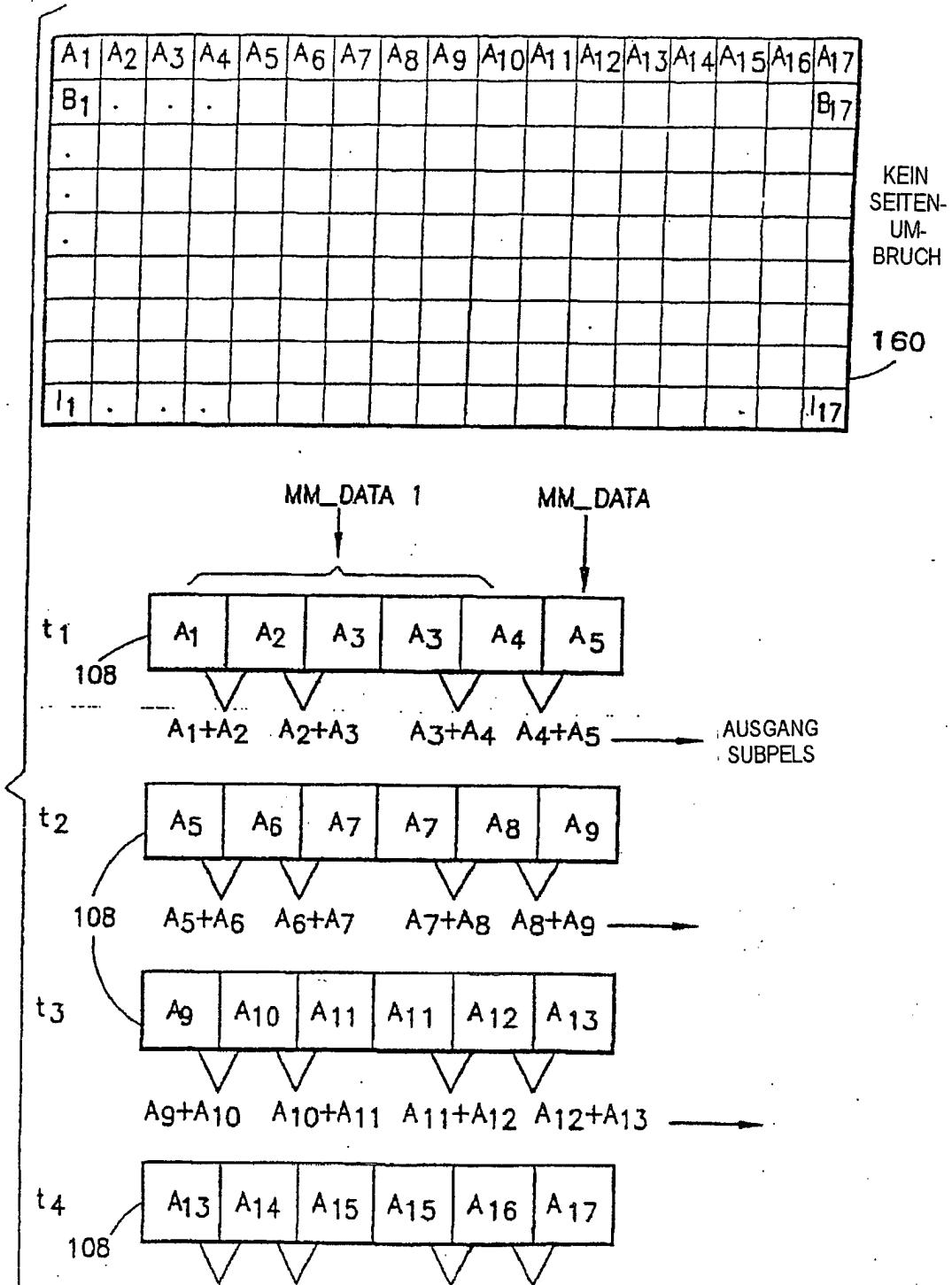


FIG. 8



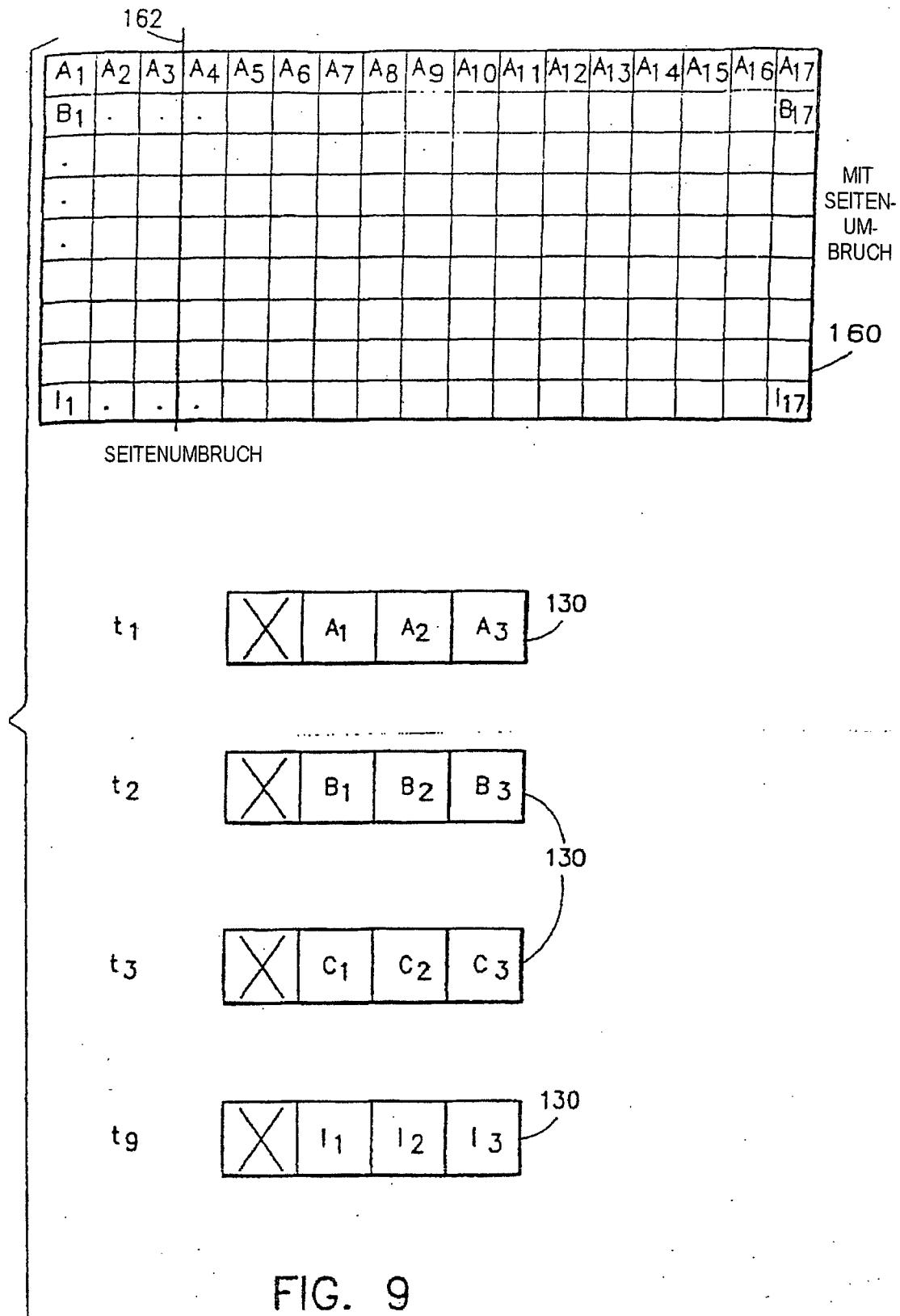


FIG. 9

