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

(10) **DE 196 20 263 B4** 2006.07.27

(12) **Patentschrift**

(21) Aktenzeichen: **196 20 263.9**
(22) Anmeldetag: **20.05.1996**
(43) Offenlegungstag: **12.12.1996**
(45) Veröffentlichungstag der Patenterteilung: **27.07.2006**

(51) Int Cl.[8]: ***G06T 1/00*** (2006.01)
G06T 15/00 (2006.01)
G06T 17/00 (2006.01)

Innerhalb von drei Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten(§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 2 Patentkostengesetz).

(30) Unionspriorität:
08/488,643 08.06.1995 US

(73) Patentinhaber:
Hewlett-Packard Development Co., L.P., Houston, Tex., US

(74) Vertreter:
Schoppe, F., Dipl.-Ing.Univ., Pat.-Anw., 82049 Pullach

(72) Erfinder:
Berry, Kyle R., Fort Collins, Col., US

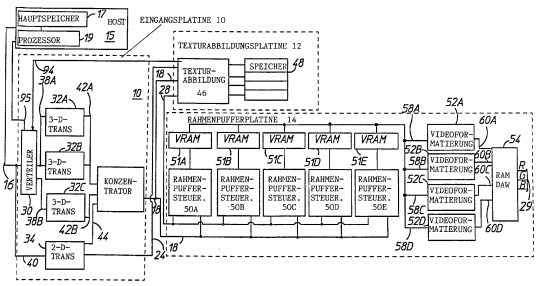
(56) Für die Beurteilung der Patentfähigkeit in Betracht gezogene Druckschriften:
DE 36 20 982 A1
US 53 88 206
FLIK,Th., LIEBIG,H.: 16-Bit Mikroprozessorsysteme, Springer-Verlag, 1982, S. 10-14;

(54) Bezeichnung: **Datensynchronisation zwischen einer Mehrzahl von asynchronen Datenaufbereitungen**

(57) Hauptanspruch: Datensynchronisator (200) in einem Graphiksystem, der Sätze von Datenworten für ein Dreieck-Grundelement synchronisiert, um synchronisierte Sätze von Datenworten für das Dreieck-Grundelement zu erzeugen, wobei jedes Datenwort in einem Satz von Datenworten durch eine einer Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen (46, 112) zu dem Datensynchronisator übergeben wird, wobei zumindest eine der asynchronen Datenaufbereitungsvorrichtungen (46, 112) Datenwörter für Vektor-Grundelemente, Punkt-Grundelemente und die Dreieck-Grundelemente aufbereitet, wobei der Datensynchronisator (200) folgende Merkmale aufweist:

eine Mehrzahl von Datenregistern (216, 218, 220, 222, 224, 226), wobei jedes Datenregister ein Datenwort in einem ersten Satz von Datenworten für das Dreieck-Grundelement von einer entsprechenden der Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen empfängt und temporär speichert, während es Datenwörter für die Vektor-Grundelemente und die Punkt-Grundelemente durchläßt, wobei jedes Datenregister (216, 218, 220, 222, 224, 226) bei der Aktivierung eines Steuersignals sein gespeichertes Datenwort als eines von einem ersten synchronisierten Satz von Datenworten ausgibt; und

eine Steuerung (209), die verhindert,...



Beschreibung

[0001] Die vorliegende Erfindung bezieht sich allgemein auf eine Farbdatenaufbereitung in einem Computergraphiksystem und insbesondere auf ein Verfahren und eine Vorrichtung zum Synchronisieren von Daten zwischen asynchronen Datenaufbereitungsvorrichtungen auf einer Pro-Pixel-Basis.

[0002] Computergrafiksysteme werden üblicherweise verwendet, um grafische Darstellungen von Objekten auf einem zweidimensionalen Anzeigeschirm anzuzeigen. Gegenwärtige Computergrafiksysteme können hochdetaillierte Darstellungen liefern und werden in einer Vielzahl von Anwendungen verwendet.

[0003] Bei typischen Computergrafiksystemen wird ein Objekt, das auf dem Anzeigebildschirm dargestellt werden soll, in eine Mehrzahl von Grafik-Grundelementen zerlegt. Grundelemente sind elementare Komponenten eines Grafikbildes und können Punkte, Linien, Vektoren und Vielecke, beispielsweise Dreiecke, einschließen. Typischerweise ist ein Hardware/Software-Schema implementiert, um die Grafik-Grundelemente, die die Ansicht eines oder mehrerer Objekte, die auf dem Bildschirm dargestellt werden, darstellen, auf dem zweidimensionalen Anzeigebildschirm aufzubereiten, oder zu zeichnen.

[0004] Typischerweise werden die Grundelemente, die das dreidimensionale Objekt, das aufbereitet werden soll, definieren, von einem Hostcomputer geliefert, der jedes Grundelement in Form von Grundelementdaten definiert. Wenn das Grundelement beispielsweise ein Dreieck ist, kann der Hostcomputer das Grundelement in der Form der x-, y-, z-Koordinaten seiner Scheitelpunkte ebenso wie der R-, G-, B-Farbwerte jedes Scheitelpunkts definieren. Eine Aufbereitungshardware interpoliert die Grundelementdaten, um die Anzeigebildschirmpixel zu berechnen, die eingeschaltet werden, um jedes Grundelement und die R-, G-, B-Werte für jedes Pixel darzustellen.

[0005] Frühe Grafiksysteme waren nicht in der Lage, Bilder auf eine ausreichend realistische Art und Weise anzuzeigen, um komplexe dreidimensionale Objekte darzustellen oder zu modellieren. Die Bilder, die durch derartige Systeme angezeigt werden, zeigten extrem glatte Oberflächen ohne Texturen, Hügel, Kratzer, Schatten und andere Oberflächendetails, die bei dem Objekt, das modelliert wird, vorliegen.

[0006] Folglich wurden Verfahren entwickelt, um Bilder mit verbesserten Oberflächendetails anzuzeigen. Die Texturabbildung ist ein solches Verfahren, das das Abbilden eines Quellenbilds, das als eine Textur bezeichnet wird, auf einer Oberfläche eines dreidimensionalen Objekts und nachfolgend das Abbilden des mit einer Texturabbildung versehenen, dreidimensionalen Objekts auf dem zweidimensionalen Grafikanzeigebildschirm, um das resultierende Bild anzuzeigen, einschließt. Oberflächendetail-Merkmale, die gewöhnlich mittels einer Textur abgebildet werden, schließen die Farbe, die Spiegelreflexion, die Vektorperturbation, das Spiegelvermögen (specularity), die Lichtdurchlässigkeit, Schatten, Oberflächenunregelmäßigkeiten und die Körnung ein.

[0007] Die Texturabbildung umfaßt das Anwenden eines oder mehrerer Punktelemente (Texel) einer Textur auf jedes Punktelement (Pixel) des angezeigten Abschnitts des Objekts, auf das die Textur abgebildet wird. Die Texturabbildungshardware ist üblicherweise mit Informationen versehen, die die Art und Weise anzeigen, auf die die Texel in einer Texturtabelle den Pixeln auf dem Anzeigebildschirm, der das Objekt darstellt, entsprechen. Jedes Texel in einer Texturtabelle ist durch S- und T-Koordinaten definiert, die den Standort desselben in der zweidimensionalen Texturtabelle identifizieren. Für jedes Pixel wird auf das entsprechende Texel oder die Texel, die auf dasselbe abbilden, aus der Texturtabelle zugegriffen und werden in die endgültigen R-, G-, B-Werte, die für das Pixel erzeugt werden, eingearbeitet, um das mit einer Textur versehene Objekt auf dem Anzeigebildschirm darzustellen.

[0008] Die Tiefenhinweisgebung (depth cuing) ist eine weitere Technik zum Erzeugen einer realistischeren Bildanzeige. Bei der Tiefenhinweisgebung wird die Farbe eines Objekts allmählich in die Hintergrundfarbe (die auch als die Tiefenhinweisfarbe bekannt ist) gemischt, basierend auf dem Abstand des angezeigten Objekts von dem Darstellungsbereich, der auf dem Anzeigebildschirm dargestellt wird. Die Tiefenhinweisgebung kann beispielsweise verwendet werden, um die atmosphärische Dämpfung der Lichtintensität zu simulieren. Folglich erscheinen Objekte schwächer, wenn sie weiter von dem Betrachter entfernt sind. Eine beliebige Farbe kann als die Tiefenhinweisfarbe verwendet werden, wobei jedoch am häufigsten schwarz verwendet ist.

[0009] In gleicher Weise kann die Tiefenhinweisgebung für interpolierte (schattierte) Polygone sowie für mit Texturabbildungen versehene Polygone verwendet werden. Zusätzlich zu dem Interpolieren der Werte R, G, B für jedes Pixel interpoliert die Aufbereitungshardware ferner einen Tiefenhinweis-Skalierfaktor für jedes Pixel des schattierten oder. mit Texturabbildungen versehenen Polygons. Die Pixelfarbparameter werden dann mit den entsprechenden Texeln von der Texturabbildungshardware gemischt, um Pixel-Zwischenfarbwerte zu liefern. Ferner werden der Zwischerfarbwert, der Tiefenhinweis-Farbwert und der Tiefenhinweis-Skalierfaktor auf einer Pro-Pixel-Basis kombiniert, um resultierende Pixelfarbwerte für die Anzeige zu ergeben.

[0010] Das Mischen der Pixel, der Farbwerte, die durch die Aufbereitungshardware erzeugt werden, und der Tiefenhinweis-Farbwerte erfordert eine Eins-zu-Eins-Entsprechung zwischen jedem Farbwert, derart, daß jeder der Farbwerte und der Texel einem gemeinsamen Pixel entspricht. In anderen Worten heißt das, daß die Werte geeignet gepaart werden, bevor die Mischoperation ausgeführt wird. Jedoch arbeiten die Texturabbildungshardware, die Rasterungsvorrichtung und die Tiefenhinweishardware typischerweise unabhängig und können ohne weiteres um einige zehn bis einige hundert von Taktzyklen desynchronisiert sein.

[0011] Ferner ist das Synchronisationsproblem verschlimmert, da die Texturabbildungshardware typischerweise nur Dreieck-Grundelemente aufbereitet, während die Aufbereitungshardware Punkte, Vektoren und Polygon-Grundelemente aufbereitet. Die Vektor- und Punkt-Grundelemente können zwischen Polygone, die durch die Aufbereitungshardware aufbereitet werden, eingestreut sein. Wenn jedoch keine Vektor- und Punkt-Grundelemente durch die Mischhardware verwendet werden, beispielsweise wenn eine Texturabbildung oder die Tiefenhinweisgebung von Vektoren und Punkten nicht durch das Computergraphiksystem unterstützt werden, existiert ein weiteres Problem beim Bestimmen einer Eins-zu-Eins-Entsprechung zwischen den Texeln, den Farbwerten, die durch die Aufbereitungshardware erzeugt werden, und den Tiefenhinweis-Farbwerten.

[0012] Außerdem sollte die Mischhardware in der Lage sein, eine Eins-zu-Eins-Entsprechung zu bestimmen, wenn die Texel, die Farbwerte, die durch die Aufbereitungshardware aufbereitet werden, und die Tiefenhinweiswerte in sich ändernden Sequenzen von Grundelement zu Grundelement ebenso wie innerhalb einer Grundelements ankommen können. Aus den vorher genannten Gründen besteht daher ein Bedarf danach, die Daten, die durch die asynchronen Datenaufbereitungsvorrichtungen in dem Computergraphiksystem geliefert werden, zu synchronisieren. Speziell besteht ein Bedarf danach, die Texel, die durch die Texturabbildungshardware aufbereitet werden, die Farbwerte, die durch die Aufbereitungshardware aufbereitet werden, und die Tiefenhinweis-Farbwerte, die durch die Aufbereitungshardware erzeugt werden, zu synchronisieren.

Stand der Technik

[0013] Die US 5,388,206 A beschreibt ein Bilderzeugungssystem, das eine Mehrzahl von Aufbereitungsvorrichtungen umfaßt, die parallel wirken und Grundelemente eines Bildes von einem Prozessor empfangen. Mittels eines Zusammensetzprozesses werden Pixelwerte und die Sichtbarkeit bestimmter Pixel bestimmt, die in einem Rahmenpuffer zur Anzeige auf einem Videomonitor gespeichert sind. Die Aufbereitungsvorrichtung umfaßt eine Rastervorrichtung, die aus einem Array aus Speicherbauteilen besteht. Jedes der Speicherbauteile umfaßt einen Pixelprozessor, um Pixelwerte zu berechnen, eine Speichereinrichtung, um Daten zur Verwendung durch den Pixelprozessor zu speichern, einen Zusammensetzpuffer und eine Zusammensetzschaltung. Diese führt die Bildzusammensetzung durch, wobei die Pixelwerte von jeder der Aufbereitungsvorrichtungen kombiniert werden, um ein abschließendes Bild für einen Abschnitt des Bildschirms zu erzeugen.

[0014] Die DE 36 20 982 A1 offenbart eine parallele Datenverarbeitungstechnik und ein Computersystem, das z. B. auf dem Gebiet der Computergrafikbearbeitung verwendet werden kann. Hier wird eine Technik und eine Schaltung gelehrt, um selektiv eine Mehrzahl von parallelen Prozessoren zu steuern, so daß die Mehrzahl der Prozessoren unterschiedliche Daten verarbeiten kann, und so daß bestimmte Prozessoren nicht wirksam sind, abhängig davon, ob die Daten in jedem der parallelen Prozessoren zu bearbeiten sind. Die getrennten Prozessoren erhalten unabhängige Daten auf Leitungen. Nachdem bestimmte Programmbefehle es erforderlich machen, daß einer oder mehrere Prozessoren eine bestimmte Programmanweisung, die an jeden der Prozessoren über den Bus angelegt wird, nicht ausführen, schließt die Architektur eine Steuerschaltung ein, die einem jeweiligen Verarbeitungselement zugeordnet ist. Eine Steuerungsleitung, die von jeder Steuerschaltung ausgeht, trägt im wesentlichen ein Signal zu jedem Verarbeitungselement, das das Verarbeitungselement anweist, ob ein Befehl ausgeführt werden soll, oder ob das Verarbeitungselement während der Zeitdauer, während der die verbleibenden Prozessoren, die bestimmte Anweisungen ausführen, inaktiv ist.

[0015] Flick, Th., und Liebig, H. beschreiben in "16-Bit Mikroprozessorsysteme", Springer Verlag, 1982, Seiten 10 bis 16 die grundsätzliche Funktionalität eines Mikroprozessors, was die Kombination von zwei Operanden einschließt, die in jeweilige Pufferregister geladen werden, um ein Ergebnis zu erzeugen.

Aufgabenstellung

[0016] Es ist die Aufgabe der vorliegenden Erfindung, eine Vorrichtung und ein Verfahren zum Synchronisieren von Datenworten, die von einer Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen geliefert werden, zu schaffen.

[0017] Diese Aufgabe wird durch einen Datensynchronisator gemäß Anspruch 1, ein Graphiksystem gemäß Anspruch 4, sowie ein Verfahren gemäß Anspruch 16 gelöst.

[0018] Gemäß einem veranschaulichenden Ausführungsbeispiel der vorliegenden Erfindung synchronisiert ein Datensynchronisator Sätze von Datenworten, um synchronisierte Sätze von Datenworten zu erzeugen, wobei jedes Datenwort in einem Satz von Datenworten durch eine einer Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen zu dem Datensynchronisator geliefert wird. Der Datensynchronisator weist eine Mehrzahl von Datenregistern auf. Jedes Datenregister empfängt ein Datenwort in einem ersten Satz von Datenworten von einer entsprechenden der Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen und speichert dasselbe temporär, wobei, wenn jedes der Datenregister sein Datenwort in dem ersten Satz von Datenworten empfangen hat, jedes Datenregister sein Datenwort als Teil eines synchronisierten Satzes von Datenworten ausgibt. Zusätzlich verhindert eine Steuerung, daß jede asynchrone Datenaufbereitungsvorrichtung ein Datenwort in einem zweiten Satz von Datenworten zu ihrem entsprechenden Datenregister sendet, wenn das Datenregister sein Datenwort in dem ersten Satz von Datenworten speichert.

[0019] Bei dieser Anordnung wird eine Mehrzahl von Datenworten, die durch eine Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen geliefert wird, in einen synchronisierten Satz von Datenworten synchronisiert. Der Datensynchronisator kann in einem Graphiksystem verwendet werden, um einen synchronisierten Satz von Datenworten für jedes Pixel eines Grundelements, das durch das Graphiksystem angezeigt werden soll, zu liefern. Zusätzlich kann jedes Datenwort für jedes Pixel eines Grundelements, das durch das Graphiksystem angezeigt werden soll, das nicht gemischt werden soll, zu einer nächsten Stufe in einer Verarbeitungspipeline des Graphiksystems geleitet werden, ohne die Verarbeitungspipeline zu räumen.

[0020] Gemäß einem weiteren veranschaulichenden Ausführungsbeispiel der vorliegenden Erfindung weist ein Verfahren zum Synchronisieren von Sätzen von Datenworten, um synchronisierte Sätze von Datenworten zu erzeugen, wobei jedes Datenwort in einem Satz von Datenworten durch eine einer Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen aufbereitet wird, das Empfangen eines ersten Datenworts von einer ersten Datenaufbereitungsvorrichtung und das temporäre Speichern des ersten Datenworts auf. Das Verfahren weist ferner das Warten auf den Empfang eines zweiten Datenworts und beim Empfangen des zweiten Datenworts das Ausgeben eines synchronisierten Satzes von Datenworten, der das erste Datenwort und das zweite Datenwort aufweist, auf.

[0021] Bei diesem Verfahren wird eine Mehrzahl von Datenworten in einen einzelnen synchronisierten Satz von Datenworten zusammengebracht. Zusätzlich kann dieses Verfahren in einem Graphiksystem verwendet werden, um einen einzelnen synchronisierten Satz von Datenworten für jedes Pixel jedes Grundelements, das durch das Graphiksystem angezeigt werden soll, zu liefern. Wenn ein Datenwort für jedes Pixel eines Grundelements des Graphiksystems nicht synchronisiert werden soll, wird das Datenwort ferner zu einer nächsten Stufe in einer Verarbeitungspipeline des Graphiksystems ausgegeben, ohne die Verarbeitungspipeline zu räumen.

Ausführungsbeispiel

[0022] Bevorzugte Ausführungsbeispiele der vorliegenden Erfindung werden nachfolgend bezugnehmend auf die beiliegenden Zeichnungen näher erläutert. Es zeigen:

[0023] Fig. 1 ein Blockdiagramm eines Ausführungsbeispiels des gesamten Computergraphiksystems der vorliegenden Erfindung;

[0024] Fig. 2 ein Blockdiagramm eines weiteren Ausführungsbeispiels des gesamten Computergraphiksystems der vorliegenden Erfindung;

[0025] Fig. 3 ein Blockdiagramm eines Abtastwandlers, der die Pro-Pixel-Datensynchronisation gemäß der Erfindung beinhaltet;

[0026] Fig. 4 ein Blockdiagramm eines Pixelgenerators, der ein Teil des Abtastwandlers, der in Fig. 3 gezeigt ist, ist;

[0027] Fig. 5 ein Blockdiagramm, das darstellend für jeden Parameterinterpolator, der in Fig. 4 gezeigt ist, ist;

[0028] Fig. 6 eine Darstellung des Dreieck-Aufbereitungsprozesses gemäß der Erfindung;

[0029] Fig. 7 ein Blockdiagramm einer Schnittstelle zwischen einem Datensynchronisator, einer Texturabbildungsplatine und einem Pixelgenerator gemäß der Erfindung;

[0030] Fig. 8 ein Blockdiagramm des Datensynchronisators, der in Fig. 7 gezeigt ist; und

[0031] Fig. 9a bis Fig. 9e ein Verfahren der vorliegenden Erfindung zum Steuern des Datensynchronisators, der in Fig. 7 gezeigt ist.

1. Systemüberblick

[0032] Fig. 1 ist ein Blockdiagramm eines Ausführungsbeispiels eines Grafiksystems der vorliegenden Erfindung, das eine Texturabbildungshardware mit einem Cache-Speicher zum lokalen Speichern von Texturdaten aufweist. Es sollte offensichtlich sein,

daß die veranschaulichende Implementierung nur exemplarisch bezüglich der Anzahl von Platinen und Chips, der Art und Weise, auf die dieselben partitioniert sind, der Busbreiten und der Datenübertragungsraten gezeigt ist. Zahlreiche andere Implementierungen können verwendet werden. Wie gezeigt ist, weist das System eine Eingangsplatine **10**, eine Texturabbildungsplatine **12** und eine Rahmenpufferplatine **14** auf. Die Eingangsplatine steht über einen 52-Bit-Bus **16** mit einem Hostcomputer **15** in Verbindung. Die Eingangsplatine empfängt Grundelemente, die aufbereitet werden sollen, über den Bus **16** von dem Hostcomputer. Die Grundelemente sind durch x-, y-, z-Vektorkoordinatendaten, R-, G-, B-Farbdaten und Texturkoordinaten S, T, alle für Abschnitte der Grundelemente, beispielsweise für die Scheitelpunkte, wenn das Grundelement ein Dreieck ist, spezifiziert. Daten, die die Grundelemente in drei Dimensionen darstellen, werden dann durch die Eingangsplatine **10** über einen 85-Bit-Bus **18** zu der Texturabbildungsplatine **12** und der Rahmenpufferplatine **14** geliefert. Die Texturabbildungsplatine interpoliert die empfangenen Grundelementdaten, um die Bildschirmanzeigepixel zu berechnen, die das Grundelement darstellen werden, und bestimmt die entsprechenden resultierenden Texturdaten für jedes Grundelementpixel. Die resultierenden Texturdaten werden über fünf 55-Bit-Busse **28**, die in Fig. 2 als ein einzelner Bus gezeigt sind, um die Figur deutlicher zu machen, zu der Rahmenpufferplatine geliefert.

[0033] Die Rahmenpufferplatine **14** interpoliert die Grundelementdaten, die von der Eingangsplatine **10** empfangen werden, ebenfalls, um die Pixel auf dem Anzeigebildschirm zu berechnen, die jedes Grundelement darstellen werden, und um die Objektfarbwerte für jedes Pixel zu bestimmen. Die Rahmenpufferplatine kombiniert dann auf einer Basis Pixel um Pixel die Objektfarbwerte mit den resultierenden Texturdaten, die von der Texturabbildungsplatine geliefert werden, um resultierende Werte R, G, B des Bilds für jedes Pixel zu erzeugen. R-, G-, B-Farb-Steuersignale für jedes Pixel werden jeweils über R-, G-, B-Leitungen **29** geliefert, um die Pixel des Anzeigebildschirms (nicht gezeigt) zu steuern, um auf dem Anzeigebildschirm ein resultierendes Bild anzuzeigen, das das mit einer Texturabbildung versehene Grundelement darstellt.

[0034] Die Eingangsplatine **10**, die Texturabbildungsplatine **12** und die Rahmenpufferplatine **14** sind jeweils pipelineartig aufgebaut und arbeiten auf mehreren Grundelementen gleichzeitig. Während die Texturabbildungs- und Rahmenpuffer-Platinen auf Grundelementen arbeiten, die vorher durch die Eingangsplatine geliefert werden, fährt die Eingangsplatine fort, auf neuen Grundelementen zu arbeiten und dieselben zu liefern, bis die Pipelines in den Platinen **12** und **14** voll werden.

[0035] Die Eingangsplatine **10** weist einen Verteilerchip **30**, drei Dreidimensionalgeometrie-Beschleunigerchips (3-D-Geometrie-Beschleunigerchips) **32A**, **32B** und **32C**, einen Zweidimensionalgeometrie-Beschleunigerchip (2-D-Geometrie-Beschleunigerchip) **34** und einen Konzentratorchip **36** auf. Der Verteilerchip **30** empfängt die x-, y-, z-Koordinaten- und Farb-Grundelementdaten über einen Bus **16** von dem Hostcomputer und verteilt 3-D-Grundelementdaten gleichmäßig unter den 3-D-Geometrie-Beschleunigerchips **32A**, **32B** und **32C**. Auf diese Art und Weise ist die Systembandbreite erhöht, da gleichzeitig drei Gruppen von Grundelementen bearbeitet werden. Daten werden über einen 40-Bit-Bus **38A** zu den 3-D-Geometrie-Beschleunigerchips **32A** und **32B** und über einen 40-Bit-Bus **38B** zu dem Chip **32C** geliefert. Beide Busse **38A** und **38B** übertragen Daten mit einer Rate von 60 MHz und liefern eine ausreichende Bandbreite, um zwei 3-D-Geometrie-Beschleunigerchips zu versorgen. 2-D-Grundelementdaten werden mit einer Rate von 40 MHz über einen 44-Bit-Bus **40** zu dem 2-D-Geometrie-Beschleunigerchip **34** geliefert.

[0036] Jeder 3-D-Geometrie-Beschleunigerchip transformiert die x-, y-, z-Koordinaten, die die empfangenen Grundelemente definieren, in entsprechende Bildschirmraumkoordinaten, bestimmt die R-, G-, B-Werte des Objekts und die S-, T-Werte der Textur für die Bildschirmraumkoordinaten, zerlegt Grundelementvierecke in Dreiecke und berechnet eine Dreieckflächengleichung, um jedes Dreieck zu definieren. Jeder 3-D-Geometrie-Beschleunigerchip führt ferner Sichtausschnittoperationen durch, um eine genaue Bildschirmanzeige des resultierenden Bilds sicherzustellen, wenn mehrere Fenster angezeigt werden, oder wenn sich ein Abschnitt eines Grundelements über das Sichtvolumen, das auf dem Anzeigebildschirm dargestellt ist, hinaus erstreckt. Ausgangsdaten von den 3-D-Geometrie-Beschleunigerchips **32A**, **32B** bzw. **32C** werden mit einer Rate von 60 MHz jeweils über 44-Bit-Busse **42A** und **42B** zu dem Konzentratorchip **36** geliefert.

[0037] Der Zweidimensionalgeometrie-Beschleunigerchip **34** liefert ferner Ausgangsdaten über einen 46-Bit-Bus **44** mit einer Rate von 45 MHz zu dem Konzentratorchip **36**. Der Konzentratorchip **36** kombiniert die 3-D-Grundelementausgangsdaten, die von den 3-D-Beschleunigerchips **32A** bis C empfangen werden, sortiert die Grundelemente in die ursprüngliche Reihenfolge, die dieselben vor der Verteilung durch den Verteilerchip **30** hatten, um und liefert die kombinierten Grundelementausgangsdaten über den Bus **18** zu der Texturabbildungs- und der Rahmenpuffer-Platine.

[0038] Die Texturabbildungsplatine **12** weist einen Texturabbildungschip **46** und eine Lokalspeicher **48** auf, der vorzugsweise als ein Cache-Speicher aus-

gebildet ist. Bei einem bevorzugten Ausführungsbeispiel der Erfindung ist der Lokalspeicher aus Gründen, die nachfolgend erläutert werden, aus einer Mehrzahl von SDRAM-Chips (SDRAM = synchronous dynamic random access memory = synchroner dynamischer Direktzugriffsspeicher) gebildet. Der Cache-Speicher **48** speichert Textur-MIP-Tabellendaten, die den Grundelementen zugeordnet sind, welche in der Rahmenpufferplatine aufbereitet werden. Die Textur-MIP-Tabellendaten werden über den Bus **40**, durch den 2-D-Geometrie-Beschleunigerchip **34** und über einen 24-Bit-Bus **24** aus einem Hauptspeicher **17** des Hostcomputers **15** heruntergeladen.

[0039] Der Texturabbildungschip **46** empfängt nacheinander Grundelementdaten über den Bus **18**, welche die Grundelemente darstellen, die auf dem Anzeigebildschirm aufbereitet werden sollen. Wie oben erläutert wurde, weisen die Grundelemente, die von den 3-D-Geometrie-Beschleunigerchips **32A** bis C geliefert werden, Punkte, Linien und Dreiecke auf. Die Texturabbildungsplatine führt keine Texturabbildung von Punkten oder Linien durch und bearbeitet nur Dreieck-Grundelemente. Die Daten, die die Dreieck-Grundelemente darstellen, enthalten die x-, y-, z-Pixelkoordinaten des Objekts für zumindest einen Scheitelpunkt, die R-, G-, B-Farbwerte des Objekts von zumindest einem Scheitelpunkt, die Koordinaten S, T der Abschnitte der Texturtabelle, die dem zumindest einem Scheitelpunkt entsprechen, und die Flächengleichung des Dreiecks. Der Texturabbildungschip **46** ignoriert die z-Koordinate des Objektpixels und die R-, G-, B-Farbwerte des Objekts. Der Chip **46** interpoliert die x-, y-Pixelkoordinaten und interpoliert die S- und T-Koordinaten, die jedem x-, y-Bildschirmanzeigepixel, das das Grundelement darstellt, entsprechen. Für jedes Pixel greift der Texturabbildungschip auf den Abschnitt der Textur-MIP-Tabelle, der demselben entspricht, aus dem Cache-Speicher zu und berechnet resultierende Texturdaten für das Pixel, welche einen gewichteten Durchschnitt mehrerer Texel einschließen können.

[0040] Bei einem exemplarischen Ausführungsbeispiel speichert der Cache vierundsechzig Blöcke von 256 × 256 Texeln. Anders als der Lokalspeicher, der in der Texturabbildungshardware bekannter Systeme verwendet ist, muß der Cache-Speicher der vorliegenden Erfindung nicht die gesamte Reihe von MIP-Tabellen der Textur speichern, die auf das Grundelement, das aufbereitet wird, abbildet, beispielsweise für große Texturen. Vielmehr speichert der Cache-Speicher zu jeder Zeit nur die speziellen Abschnitte der Reihe von MIP-Tabellen, die bei der gegenwärtigen Aufbereitung des Grundelements tatsächlich verwendet werden. Daher wird für die meisten Anwendungen nur ein Teil der gesamten Texturdaten für das Bild, das aufbereitet wird, zu jeder Zeit in dem Cache-Speicher gespeichert sein.

[0041] Die vollständige Reihe von MIP-Tabellen für jede Textur ist in dem Hauptspeicher **17** des Hostcomputers **15** angeordnet und gespeichert. Für jedes Pixel des Grundelements, das aufbereitet wird, greift der Texturabbildungschip **46** auf ein Verzeichnis des Cache-Speichers **48** zu, um zu bestimmen, ob das entsprechende Texel oder die Texel der Textur-MIP-Tabellen gegenwärtig in dem Cache vorliegen. Wenn die entsprechenden Texel zu der Zeit des Zugriffs in dem Cache-Speicher gespeichert sind, tritt ein Cache-Treffer auf, und die Texel werden aus dem Cache gelesen und durch den Texturabbildungschip **46** bearbeitet, um die resultierenden Texturdaten zu berechnen, die zu der Rahmenpufferplatine geleitet werden.

[0042] Wenn die entsprechenden Texel für das Grundelementpixel jedoch nicht in dem Cache-Speicher gespeichert sind, wenn ein Zugriff durch den Texturabbildungschip **46** stattfindet, tritt ein Cache-Fehlzugriff auf. Wenn ein Cache-Fehlzugriff auftritt, wird der Teil der Textur-MIP-Tabellendaten, der benötigt wird, um das Grundelement aufzubereiten, von dem Hauptspeicher **17** des Hostcomputers **15** in den Cache-Speicher **48** heruntergeladen, wobei möglicherweise einige Daten, die vorher in demselben gespeichert waren, ersetzt werden. Jedoch lädt anders als herkömmliche Texturabbildungssysteme, die die gesamte Reihe von MIP-Tabellen für jedes Grundelement, das aufbereitet wird, herunterladen, die vorliegende Erfindung nur den Teil der Reihe von MIP-Tabellen, der tatsächlich benötigt wird, um das Grundelement oder den momentan aufbereiteten Abschnitt desselben momentan aufzubereiten, herunter. Wenn ein Cache-Fehlzugriff stattfindet, wird ein Unterbrechungs-Steuersignal durch den Texturabbildungschip **46** erzeugt, um einen Texturunterbrechungsverwalter in dem Hostcomputer **15** zu initiieren. Das Unterbrechungs-Steuersignal wird über eine Leitung **94** zu dem Verteilerchip **30** geliefert, welcher wiederum ein Unterbrechungssignal über eine Leitung **95** zu dem Hostcomputer liefert.

[0043] Die angeforderten Texturdaten werden durch den Hostcomputer aus seinem Hauptspeicher wiedergewonnen und werden unter Umgehung der dreidimensionalen Grundelement-Aufbereitungspipeline durch die Eingangsplatine und den Texturabbildungschip über den Bus **24** zu der Texturabbildungsplatine **48** heruntergeladen. Wenn folglich eine Cache-Fehlzugriffs-Unterbrechung auftritt, kann die Eingangsplatine fortfahren, 3-D-Grundelemente zu bearbeiten und Ausgangsgrundelementdaten über den Bus **18** zu dem Texturabbildungschip und der Rahmenpufferplatine zu liefern, während die Texturdaten, die einem Grundelement zugeordnet sind, das den Cache-Fehlzugriff bewirkte, von dem Hauptspeicher **17** heruntergeladen werden. Im Gegensatz zu herkömmlichen Texturabbildungssystemen erfordert das Herunterladen von Texturdaten zu der Texturabbil-

dungshardware kein Räumen der 3-D-Grundelementpipeline, wodurch die Bandbreite und das Verhalten des Systems verbessert werden.

[0044] Die resultierenden Texturdaten für jedes Pixel werden über die fünf Busse **28** durch den Texturabbildungschip **46** zu der Rahmenpufferplatine geliefert. Die fünf Busse **28** sind jeweils mit fünf Rahmenpuffer-Steuerchips **50A**, **50B**, **50C**, **50D** und **50E**, die auf der Rahmenpufferplatine vorgesehen sind, gekoppelt und liefern parallel resultierende Texturdaten zu den Rahmenpuffer-Steuerchips. Die Rahmenpuffer-Steuerchips **50A** bis E sind jeweils mit Gruppen von zugeordneten VRAM-Chips **51A** bis E (VRAM = video random access memory = Video-Direktzugriffsspeicher) gekoppelt. Wie oben erläutert wurde, ist der 55-Bit-Bus **28** für jeden Rahmenpuffer-Steuerchip **50A** bis **50E** in elf Bits geteilt. Daher sind gemäß der vorliegenden Erfindung nur elf Signalleitungen notwendig, um Texeldaten zwischen der Texturabbildungsplatine **12** und jedem Rahmenpufferchip zu übertragen. Ferner werden gemäß der vorliegenden Erfindung keine Kennungen oder Identifikationsinformationen, um zu identifizieren, zu welchem Pixel die Texturdaten gehören, zwischen der Texturabbildungsplatine und dem jeweiligen Rahmenpufferchip gesendet.

[0045] Die Rahmenpufferplatine weist ferner vier Videoformatchips **52A**, **52B**, **52C** und **52D** und einen RAMDAW **54** auf (RAMDAW = Direktzugriffsspeicher-Digital/Analog-Wandler). Die Rahmenpuffer-Steuerchips steuern unterschiedliche, nicht überlappende Segmente des Anzeigebildschirms. Jeder Rahmenpuffer-Steuerchip empfängt Grundelementdaten über den Bus **18** von der Eingangsplatine und resultierende Texturabbildungsdaten über den Bus **28** von der Texturabbildungsplatine. Die Rahmenpuffer-Steuerchips interpolieren die Grundelementdaten, um die Bildschirmanzeige-Pixelkoordinaten in den jeweiligen Segmenten derselben, die das Grundelement darstellen, und die entsprechenden R-, G-, B-Farbwerte des Objekts für jede Pixelkoordinate zu berechnen. Für diejenigen Grundelemente (d.h. Dreiecke), für die resultierende Texturdaten von der Texturabbildungsplatine geliefert werden, kombinieren die Rahmenpuffer-Steuerchips auf einer Basis Pixel um Pixel die Objektfarbwerte und die resultierenden Texturdaten, um endgültige R-, G-, B-Werte für jedes Pixel, das auf dem Anzeigebildschirm angezeigt werden soll, zu erzeugen. Ein Mischmodus, der die Art und Weise definiert, auf die die Objekt- und Textur-Farbwerte kombiniert werden, wird durch ein Aufbereitungsmodus-Steuerwort gesteuert, das über den Bus **18** durch die Eingangsplatine **10** geliefert wird.

[0046] Die Art und Weise, auf die die Objekt- und Textur-Farbwerte kombiniert werden, kann auf eine Anzahl von unterschiedlichen Arten gesteuert werden. Beispielsweise können die Objektfarbwerte in einem Ersetzungsmodus einfach durch die Texturfarbwerte ersetzt werden, so daß nur die Texturfarbwerte beim Aufbereiten des Pixels verwendet werden. Alternativ können die Objekt- und Textur-Farbwerte in einem Modulationsmodus miteinander multipliziert werden, um die endgültigen R-, G-, B-Werte für das Pixel zu erzeugen. Außerdem kann ein Farbsteuerwort für jedes Texel gespeichert werden, das ein Verhältnis spezifiziert, das die Art und Weise definiert, auf die die entsprechenden Texturfarbwerte mit den Objektfarbwerten kombiniert werden sollen. Ein resultierendes Farbsteuerwort kann für die resultierenden Texeldaten, die jedem Pixel entsprechen, bestimmt werden und über den Bus **28** zu den Rahmenpuffer-Steuerchips geliefert werden, so daß die Steuerchips das Verhältnis verwenden können, das durch das entsprechende resultierende Steuerwort spezifiziert ist, um die endgültigen R-, G-, B-Werte für jedes Pixel zu bestimmen.

[0047] Die resultierenden Bildvideodaten, die durch die Rahmenpuffer-Steuerchips **50A** bis E erzeugt werden, einschließlich der R-, G-, B-Werte für jedes Pixel, werden in den entsprechenden VRAM-Chips **51A** bis E gespeichert. Jede Gruppe von VRAM-Chips **51A** bis E weist acht VRAM-Chips auf, derart, daß **40** VRAM-Chips auf der Rahmenpufferplatine angeordnet sind. Jeder der Videoformatchips **52A** bis D ist mit einem unterschiedlichen Satz von zehn VRAM-Chips verbunden und empfängt Daten von demselben. Die Videodaten werden seriell aus den VRAM-Chips geschoben und jeweils über 64-Bit-Busse **58A**, **58B**, **58C** und **58D** mit einer Rate von 33 MHz zu den vier Videoformatchips **52A**, **52B**, **52C** und **52D** geliefert. Die Videoformatchips formatieren die Videodaten, so daß dieselben durch den RAMDAW gehandhabt werden können, und liefern die formatierten Daten über 32-Bit-Busse **60A**, **60B**, **60C** und **60D** mit einer Rate von 33 MHz zu dem RAMDAW **54**. Der RAMDAW **54** wandelt die digitalen Farbdaten wiederum in analoge R-, G-, B-Farbsteuersignale um und liefert die R-, G-, B-Steuersignale für jedes Pixel entlang R-, G-, B-Steuerleitungen **29** zu einer Bildschirmanzeige (nicht gezeigt).

[0048] Bei einem Ausführungsbeispiel der Erfindung ist die Hardware auf der Texturabbildungsplatine **12** und der Rahmenpufferplatine **14** dupliziert, derart, daß bestimmte Grundelement-Aufbereitungsaufgaben auf mehreren Grundelementen parallel durchgeführt werden können, wodurch die Bandbreite des Systems erhöht wird. Ein Beispiel eines solchen alternativen Ausführungsbeispiels der vorliegenden Erfindung ist in Fig. 2 gezeigt, welche ein Blockdiagramm eines Computergrafiksystems der vorliegenden Erfindung ist, bei dem ein bestimmter Teil der Hardware dupliziert ist. Das System von Fig. 2 weist vier 3-D-Geometrie-Beschleunigerchips **32A**, **32B**, **32C** und **32D**, zwei Texturabbildungschips **46A** und

46B, denen jeweils Cache-Speicher **48A** und **48B** zugeordnet sind, und zehn Rahmenpufferchips **50A** bis **50J** auf, denen jeweils eine Gruppe von VRAM-Chips zugeordnet ist. Der Betrieb des Systems von Fig. 2 ist gleichartig dem des Systems von Fig. 1, der oben beschrieben ist. Die Duplizierung der Hardware bei dem Ausführungsbeispiel von Fig. 2 ermöglicht eine erhöhte Systembandbreite, da bestimmte Grundelement-Aufbereitungsoperationen parallel auf mehreren Grundelementen durchgeführt werden können.

[0049] Bei dem Graphiksystem der vorliegenden Erfindung wird ferner eine Pro-Pixel-Tiefenhinweisgebung geliefert und hierin bezugnehmend auf Fig. 3 beschrieben. Ein linker Abschnitt von Fig. 3, der mit "Geometriebeschleuniger" bezeichnet ist, stellt den Schaltungsaufbau in den 3-D-Geometriebeschleunigerchips **32A**, **32B** und **32C** (Fig. 1) dar. Der rechte Abschnitt von Fig. 3, der mit "Abtastwandler" bezeichnet ist, stellt den Schaltungsaufbau in den Rahmenpuffer-Steuerchips **50A**, **50B**, **50C**, **50D** und **50E** dar (Fig. 1). Obwohl das Graphiksystem, insbesondere der Abtastwandler, Objekte, die dargestellt werden sollen, in Dreiecke, Vektoren und Punkte zerlegt, wird die Tiefenhinweisgebung nur auf Dreiecke angewendet. Eine Kantenberechnungseinheit **110** in dem Geometriebeschleuniger empfängt Scheitelpunktparameter von dem Hostcomputer. Die Scheitelpunktparameter können für jeden Scheitelpunkt räumliche Koordinaten (x, y und z), Normalkoordinaten (Nx, Ny, Nz), die einen Einheitsvektor senkrecht zu der Oberfläche an dem Scheitelpunkt anzeigen, Farbwerte (R, G, B und α) und Texturabbildungskoordinaten (S und T) einschließen. Der Farbwert α stellt eine Undurchlässigkeit (Opazität) des Objekts dar, wenn eine Hintergrundmischung verwendet und aktiviert ist, wobei der Wert α bestimmt, wieviel des existierenden Hintergrunds verwendet ist, und wieviel der Grundelementfarbe verwendet ist. Die Kantenberechnungseinheit **110** empfängt ferner Tiefenhinweisparameter.

[0050] Im allgemeinen berechnet die Kantenberechnungseinheit alle Parameter, die durch den Abtastwandler angefordert werden. Beispielsweise weisen für Dreieck-Grundelemente einige der berechneten Parameter die Anfangsfarbwerte, R, G, B und α und partielle Ableitungen jedes Farbparameters bezüglich X und bezüglich einer Kante E_1 des Dreiecks (siehe Fig. 6) mit einer größten Abmessung in Y auf. Dieser Satz von berechneten Parametern wird häufig als die Ebenengleichungen für das Dreieck bezeichnet. Zusätzlich liefert die Kantenberechnungseinheit **110** für das Dreieck-Grundelement Tiefenhinweis-Aufbereitungsparameter zu dem Abtastwandler, einschließlich eines Tiefenhinweis-Skalierfaktorwerts für einen obersten Scheitelpunkt V_0 (siehe Fig. 6) in dem Dreieck und partieller Ableitungen des Tiefenhinweis-Skalierfaktors bezüglich X und der Kante E_1.

[0051] Der Abtastwandler weist einen Pixelgenerator **112** auf, der jeden der Aufbereitungsparameter unabhängig interpoliert, um x-, y- und z-Koordinaten und Farbwerte für jedes Pixel in dem Grundelement zu berechnen. Die Pro-Pixel-Farben weisen Diffusfarbwerte R, G, B und α auf, und können ferner Spiegelfarbwerte R_S, G_S und B_S aufweisen. Zusätzlich berechnet der Pixelgenerator **12** einen Tiefenhinweis-Skalierfaktor auf einer Pro-Pixel-Basis für jedes Dreieck-Grundelement. Die Pro-Pixel-Werte, die durch den Pixelgenerator **112** berechnet werden, werden einer Texturmischeinheit **114** zugeführt. Wie oben erläutert wurde, empfängt die Texturmischeinheit **114** auch Texel von dem Texturabbildungschip **46** (Fig. 1) über den 11-Bit-Bus **28**, der dem jeweiligen Rahmenpufferchip **50A** bis **50E** zugeordnet ist. Die Texel weisen Texturfarbwerte R_t, G_t, B_t und α_t für jedes Pixel auf.

[0052] Wie oben erläutert wurde, müssen die Pro-Pixel-Texturfarbwerte, die -Diffusfarbwerte und die -Spiegelfarbwerte durch die Texturmischeinheit **114** für jedes Pixel auf dem Anzeigebildschirm, das das Grundelement darstellt, kombiniert werden. Wie oben erläutert wurde, besteht ein Problem jedoch darin, daß die Texturabbildungsplatine **12** und der Abtastwandler unabhängig die Grundelemente bearbeiten, die durch die Eingangsplatine **10** aufbereitet werden. Folglich sind die Daten, die durch die Texturabbildungsplatine und den Abtastwandler aufbereitet werden, nicht synchronisiert, da diese Abschnitte des Graphiksystems asynchron arbeiten. Außerdem bereitet die Texturabbildungsplatine **12** nur Texturfarbwerte für Dreieck-Grundelemente auf. Im Gegensatz dazu bereitet der Abtastwandler Diffusfarbwerte auf und kann Spiegelfarbwerte für Dreieck-Grundelemente aufbereiten. Folglich schließen die Daten, die durch den Abtastwandler aufbereitet werden, Vektor- und Punkt-Daten ein, die mit Dreieckdaten durchsetzt sind. Da eine Eins-zu-Eins-Entsprechung für jedes Pixel zwischen den Texelfarbwerten, den Diffusfarbwerten und den Spiegelfarbwerten erforderlich ist, synchronisiert die Texturmischeinheit **114** diese Werte für jedes Pixel eines Dreieck-Grundelements. Wie nachfolgend detailliert beschrieben wird, weist die Texturmischeinheit **114** eine Datensynchronisationssteuerung **200** zum Synchronisieren der Texelfarbwerte, der Diffusfarbwerte und der Spiegelfarbwerte für Dreieck-Grundelemente auf einer Pixel-um-Pixel-Basis auf.

[0053] Sobald die Pro-Pixel-Diffus-, die -Spiegel-Farbwerte und die -Texel synchronisiert sind, werden sie gemäß einem voreingestellten Mischalgorithmus kombiniert, um resultierende Farbwerte zu liefern. Die resultierenden Farbwerte werden einer Tiefenhinweis-Mischeinheit **116** zugeführt. Der Pixel-Tiefenhinweis-Skalierfaktor, der durch den Pixelgenerator **112** erzeugt wird, wird durch die Texturmischeinheit **114** nicht verwendet, sondern wird durch die Texturmischeinheit **114** zu der Tiefenhin-

weis-Mischeinheit **116** geleitet, um sicherzustellen, daß die pipelinemäßig verarbeiteten Daten synchronisiert bleiben. Die Tiefenhinweis-Mischeinheit **116** kombiniert den Tiefenhinweis-Skalierfaktor und die Tiefenhinweis-Farbwerte für jedes Pixel, die von einem Tiefenhinweis-Farbdatenregister, welches durch den Hostcomputer beschrieben wird, empfangen werden, mit den resultierenden Farbwerten, um mit Tiefenhinweisen versehene Farben für jedes Pixel zu liefern.

[0054] Ein Blockdiagramm des Pixelgenerators **112** ist in Fig. 4 gezeigt. Die Aufbereitungsparameter von dem Geometriebeschleuniger werden in Aufbereitungsparameterregistern **130** plaziert. Die jeweiligen Aufbereitungsparameter werden von den Register **130** zu einem α-Interpolator **131**, einem Rot-Interpolator **132**, einem Grün-Interpolator **133**, einem Blau-Interpolator **134**, einem Tiefenhinweis-Skalierfaktor-Interpolator **135**, einem Rot-Spiegel-Interpolator **136**, einem Grün-Spiegel-Interpolator **137** und einem Blau-Spiegel-Interpolator **138** geliefert. Die Interpolatoren **131** bis **138** weisen einen identischen Schaltungsaufbau auf und bearbeiten für einen hohen Durchsatz unabhängig und gleichzeitig unterschiedliche Parameter.

[0055] Im allgemeinen wird jeder Parameter für jedes Pixel in einem Grundelement berechnet. Ein Dreieck mit Scheitelpunkten V_0, V_1 und V_2 ist in Fig. 6 gezeigt. Jeder der Interpolatoren **131** bis **138** bewegt sich zuerst schrittweise entlang der Kante E_1 des Dreiecks von dem Scheitelpunkt V_0 zu dem Scheitelpunkt V_2 und bestimmt den Anfangswert der Parameter für jede Pixelreihe in dem Dreieck. Pixelreihen **150**, **152**, **154**, usw., sind in Fig. 6 gezeigt. Die Anfangswerte der Parameter werden jeweils an Pixeln **160**, **162** und **164** für die Pixelreihen **150**, **152**, **154** bestimmt. Die Länge jeder Pixelreihe in dem Dreieck ist als seine Spanne bekannt. Folglich weist beispielsweise bezugnehmend auf Fig. 6 die Pixelreihe **154** eine Spanne von vier Pixeln von dem Pixel **164** zu einem Pixel **168** auf. Die Anfangsparameterwerte für jede Pixelreihe werden aus dem Anfangswert des Parameters an den Scheitelpunkt V_0 und der Steigung der Parameter entlang der Kante E_1 bestimmt. Nachdem die Anfangswerte für jede Pixelreihe bestimmt wurden, bewegt sich der Interpolator schrittweise entlang jeder Pixelreihe in die X-Richtung und bestimmt den Parameterwert an jedem Pixel unter Verwendung des Anfangswerts für die Pixelreihe und der Steigung des Parameterwerts als einer Funktion von X. Die Parameterwerte für jedes Pixel in dem Dreieck werden zu der Texturmischeinheit **114** (Fig. 3) geliefert.

[0056] Ein Blockdiagramm, das jeden Interpolator in dem Pixelgenerator **112** darstellt, ist in Fig. 5 gezeigt. Jeder Interpolator weist einen Kantenschrittgeber auf, der die Parameter entlang der Kanten jedes Dreiecks berechnet. Eine erste Kantenschrittgeberstufe **180** erzeugt x- und y-Werte für jedes Pixel entlang der Kante E_1, während eine zweite Kantenschrittgeberstufe **182** die Farb- und z-Werte berechnet. Für ein gutes Verhalten sind zwei Kantenschrittgeberstufen bevorzugt, da die Ergebnisse der x- und y-Berechnungen während der Berechnungen des Farb-Werts und z-Werts verwendet werden. Bei dem bevorzugten Ausführungsbeispiel der vorliegenden Erfindung sind die Kantenschrittgeberstufen **180** und **182** sowohl in der Texturabbildungsplatine als auch dem Abtastwandler verwendet, um sicherzustellen, daß die Grundelementdaten sowohl durch die Texturabbildungsplatine (Fig. 1) als auch den Abtastwandler in der gleichen Reihenfolge ausgegeben werden.

[0057] Ein Kachelschneider, der einen ersten Kachelschneider **184** und einen zweiten Kachelschneider **186** aufweist, beschränkt Pixel und Spannen auf innerhalb des Bereichs des verschachtelten VRAM-Chips (Fig. 1), der dem jeweiligen Abtastwandler zugeordnet ist, der den jeweiligen Pixelgenerator enthält. Wie oben gezeigt wurde, ist der Abtastwandler in getrennte Rahmenpuffer-Steuerchips unterteilt, die für einen hohen Durchsatz parallel zu zugeordneten VRAM-Chips arbeiten. Jeder VRAM-Chip handhabt vorzugsweise eine Kachel des Bildschirms, die sechzehn Pixel in x-Richtung mal zwei Pixel in y-Richtung aufweist. Der Kachelschneider begrenzt die Pixel, die durch den Interpolator verarbeitet werden, auf diejenigen in dem Bereich sechzehn mal zwei Pixel des entsprechenden VRAM. Dies verbessert das Verhalten durch das Beseitigen unnötiger Zyklen für Pixel, die außerhalb des Bereichs des entsprechenden VRAM-Chips liegen. Der Kachelschneider teilt jede Spanne des Dreiecks, das aufbereitet wird, in Teilspannen von sechzehn Pixeln oder weniger, die innerhalb des Bereichs des zugeordneten VRAM-Chips liegen. Durch das Verhindern dessen, daß unnötige Pixel in einem Spannenschrittgeber erzeugt werden, ist das Aufbereitungsverhalten signifikant verbessert.

[0058] Die Ausgabe des Kachelschneiders wird durch einen Spannen/Pixel-FIFO **190** zu einem Spannenschrittgeber **192** geliefert. Bei einem bevorzugten Ausführungsbeispiel ist der Spannen/Pixel-FIFO vierundsechzig Einträge tief. Der Spannenschrittgeber **192** erzeugt Pixelparameter basierend auf den Kanteninformationen für jede Spanne in dem Dreieck. Die Spannen weisen einen konstanten y-Wert auf und können abhängig von dem Dreiecktyp schrittweise entweder in die positive oder die negative x-Richtung durchlaufen werden. Speziell wird ein Dreieck eines ersten Typs schrittweise in eine positive x-Richtung durchlaufen, während ein Dreieck eines zweiten Typs schrittweise in eine negative x-Richtung durchlaufen wird. Bei einem bevorzugten Ausführungsbeispiel der vorliegenden Erfindung ist

der Spannenschrittgeber **192** sowohl in der Texturabbildungsplatine (Fig. 1) als auch in dem Abtastwandler verwendet, um sicherzustellen, daß die Parameterwerte für jedes Pixel in dem Dreieck, das aufbereitet wird, in der gleichen Reihenfolge von der Texturabbildungsplatine und dem Abtastwandler zu der Texturmischeinheit **11** ausgegeben werden. Die Ausgabe des Spannenschrittgebers **192** ist ein Satz von Parameterwerten für jedes Pixel in dem Dreieck, das aufbereitet wird. Wie oben in Verbindung mit Fig. 3 erläutert wurde, liefert der Pixelgenerator **112** für jedes Pixel interpolierte Parameterwerte für rot, grün, blau, α, den Tiefenhinweis-Skalierfaktor, Spiegel-rot, Spiegelgrün und Spiegel-blau. Die Interpolationsschaltungen zum Erzeugen von Pro-Pixel-Parameterwerten aus den Aufbereitungsparametern durch eine Kantenschrittgebung und eine Spannenschrittgebung sind Fachleuten bekannt.

[0059] Wie oben gezeigt wurde, mischt die Texturmischeinheit **114** (die in Fig. 3 gezeigt ist) die Diffusfarbwerte, die durch den Pixelgenerator **112** erzeugt werden, mit Texelfarbwerten, die von der Texturabbildungseinheit empfangen werden, um Zwischenfarbwerte zu liefern. Die Spiegelfarbwerte, die durch den Pixelgenerator **112** erzeugt werden, werden optional zu den Zwischenfarbwerten hinzugefügt, um resultierende Farbwerte zu liefern. Wie oben erläutert wurde, sollte eine Eins-zu-Eins-Entsprechung zwischen den Texelfarbwerten, den Diffusfarbwerten und den Spiegelfarbwerten (wenn eine Spiegelfarboption eingeschaltet ist) durchgeführt werden. Jedoch liefert die Texturabbildungsplatine Texelfarbwerte ohne irgendeine Identifikation oder Koordinateninformation zu der Texturmischeinheit **114**, wie oben erläutert wurde, während der Pixelgenerator **112** Diffusfarbwerte für alle Grundelemente liefert, die Punkte, Vektoren und Dreiecke einschließen. Wenn die Spiegelfarboption aktiviert ist, liefert der Pixelgenerator **112** ferner Spiegelfarbwerte für Dreieck-Grundelemente. Wie ferner oben erläutert wurde, werden die Daten von der Texturabbildungsplatine und dem Abtastwandler asynchron zu der Texturmischeinheit **114** aufbereitet. Außerdem kann die Reihenfolge der Ankunft der Texelfarbwerte, der Diffusfarbwerte und der Spiegelfarbwerte von Grundelement zu Grundelement, ebenso wie von Pixel zu Pixel in einem Grundelement, variieren.

[0060] Zusätzlich zu dem Vorhergehenden existiert eine Vielzahl von Datensynchronisationsmodi für die Texel-, die Diffus- und die Spiegel-Farbwerte. Bei einem bevorzugten Ausführungsbeispiel werden aufgrund der Pipeline-Beschränkungen in dem Computergraphiksystem (Fig. 1) die Spiegelfarbwerte stets nach den Diffusfarbwerten geliefert. Folglich können die Diffusfarbwerte und die Spiegelfarbwerte nicht während des gleichen Taktzyklusses an der Texturmischeinheit **114** ankommen. Diese Beschränkung ermöglicht jedoch noch eine Mehrzahl möglicher Aufbereitungsmodi für Dreieck-Grundelemente. Speziell weisen die möglichen Datensynchronisationsmodi Folgende auf: (1) nur Diffus-Farbwerte; (2) Diffus-Farbwerte, dann Spiegel-Farbwerte; (3) Diffus-Farbwerte, dann Texel-Farbwerte; (4) Texel-Farbwerte, dann Diffus-Farbwerte; (5) Diffus-Farbwerte und Texel-Farbwerte während eines gleichen Taktzyklusses; (6) Diffus-Farbwerte, dann Spiegel-Farbwerte, dann Texel-Farbwerte; (7) Diffus-Farbwerte, dann Spiegel-Farbwerte und Texel-Farbwerte während des gleichen Taktzyklusses; (8) Diffus-Farbwerte, dann Texel-Farbwerte, dann Spiegel-Farbwerte; (9) Texel-Farbwerte und Diffus-Farbwerte während des gleichen Taktzyklusses, dann Spiegel-Farbwerte; und (10) Texel-Farbwerte, dann Diffus-Farbwerte, dann Spiegel-Farbwerte. Folglich ist die Synchronisation der Farbwerte auf einer Pro-Pixel-Basis ferner durch die Mehrzahl von Datensynchronisations-Aufbereitungsmodi weiter erschwert.

[0061] Wie oben erläutert wurde, verwenden bei einem bevorzugten Ausführungsbeispiel der vorliegenden Erfindung die Texturabbildungsplatine und der Abtastwandler identische Kanten- und Spannen-Schrittgebereinheiten, wodurch sichergestellt ist, daß die Texelfarbwerte, die Diffusfarbwerte und die Spiegelfarbwerte für jedes Pixel in der gleichen Reihenfolge, jedoch nicht notwendigerweise zu der gleichen Zeit, erzeugt werden. Obwohl die Texelfarbwerte, die Diffusfarbwerte und die Spiegelfarbwerte in der gleichen Reihenfolge aufbereitet werden, werden die Daten, die den Farbwert darstellen, dennoch synchronisiert, bevor die Texturmischeinheit **114** mit diesen werten arbeiten kann, da die Interpolatoren auf Punkt- und Vektor-Grundelementen arbeiten, und da die Daten nicht synchron aufbereitet werden.

[0062] In Fig. 7 ist ein Blockdiagramm einer Schnittstelle zwischen einem Datensynchronisator **200** gemäß der vorliegenden Erfindung, der Texturabbildungsplatine **12** (Fig. 1) und dem Pixelgenerator **112** (Fig. 3) dargestellt. Wie gezeigt ist, umfaßt der Datensynchronisator **200** die Datenregister **203** und die Datensynchronisationssteuerung **209**. Die Schnittstelle weist eine Texelpackereinheit **202** zum Empfangen von Daten von der Texturabbildungsplatine **12**, zum Packen der Texturfarbdaten in ein einzelnes 32-Bit-Texel und zum Leiten der gepackten Texturfarbdaten zu einer Mehrzahl von Datenregistern **203** auf. Die Daten werden unter Verwendung von nur elf Signalleitungen zwischen der Texturabbildungsplatine **12** und der Texelpackereinheit **202** übertragen. Acht dieser Signalleitungen sind Eingangsdatenleitungen t data, die 8-Bit-Daten, die in vier Komponenten α, rot, grün und blau serialisiert sind, empfangen. Jede Komponente der serialisierten Daten wird getrennt quittiert, so daß vier Taktzyklen verwendet werden, um alle vier Komponenten der Texeldaten zu der Texelpackereinheit **202** zu übertragen. Ein Signal t_trans ist verwendet, um anzuzeigen, ob die mo-

mentanen Texeldaten transparent sind. Wenn es aktiviert ist, zeigt das Signal t_trans an, daß der Texelwert ignoriert werden sollte, da er nicht für dieses Pixel gilt. Ein Eingangssteuersignal t_valid_N zeigt an, ob die Texeldaten und t_trans gültig sind oder nicht. Schließlich quittiert ein Ausgangssteuersignal t_ack_N die Übertragung einer Komponente von Texeldaten.

[0063] Aus dem Vorhergehenden sollte offensichtlich sein, daß jede Komponente α, rot, grün und blau der Texturfarbwerte über die acht Signalleitungen t data des Busses **28** von der Texturabbildungsplatine **12** zu der Texelpackereinheit **202** gesendet wird. Die Texelpackereinheit **202** entpackt dann diese Komponenten in ein 32-Bit-Farbtexelwort tm_color, das zu dem Datenregister **203** geleitet wird. Eine vollständige Übertragung eines einzelnen 32-Bit-Texels benötigt vier Taktzyklen. Außerdem werden nur Texelfarbwerte zwischen der Texturabbildungsplatine und der Texelpackereinheit **202** übertragen. Keine räumlichen Koordinateninformationen (x, y oder z) oder weitere Identifikationsinformationen werden übertragen.

[0064] Die Schnittstelle zwischen der Texelpackereinheit **202** und dem Datensynchronisator **200** weist tm_trans auf, welches ein Signal ist, das anzeigt, ob das Texel transparent ist oder nicht; tm_tcolor_valid, welches ein Signal ist, um anzuzeigen, ob der 32-Bit-Texelfarbwert tm_color, der von der Texelpackereinheit **202** zu den Datenregistern **203** übertragen wird, gültig ist oder nicht; und ld_tcolor_ack, welches ein Signal von dem Datensynchronisator **200** ist, das quittiert, daß das Datenregister **203** die Übertragung des gepackten 32-Bit-Texelfarbwerts tm_color annehmen kann.

[0065] Der Pixelgenerator **112** (Fig. 3) weist den Spannenschrittgeber **192** (Fig. 5) auf, der ebenfalls schnittstellenmäßig mit der Datensynchronisationssteuerung **209** und den Datenregistern **203** verbunden ist. Die Datenregister **203** empfangen einen 13-Bit-x-Koordinatenwert sp_x, einen 13-Bit-y-Koordinatenwert sp_y und einen 23-Bit-z-Koordinatenwert sp_z von dem Spannenschrittgeber **192**. Außerdem empfangen die Datenregister über einen 32-Bit-Bus sp_color, einen 32-Bit-Spiegelfarbwert und einen 32-Bit-Diffusfarbwert in zwei Taktzyklen von dem Spannenschrittgeber **192**.

[0066] Die Schnittstelle zwischen dem Spannenschrittgeber **192** und der Datensynchronisationssteuerung **209** weist ein Wort sp_status auf, das aus einer Mehrzahl von Flags besteht, die beispielsweise den Typ des Grundelements (Punkt, Vektor oder Dreieck) anzeigt, dem die Pixeldaten zugeordnet sind. Speziell existiert eine Flag, die anzeigt, ob die Pixeldaten ein Punkt sind oder nicht; eine Flag, die anzeigt, ob die Pixeldaten ein Vektor sind oder nicht; eine Flag, die anzeigt, ob die Daten für ein Diffusfarb-Dreieckpixel stehen oder nicht; und eine Flag, die anzeigt, ob die Daten für ein Spiegelfarb-Dreieckpixel stehen oder nicht. Ferner weist das Wort sp_status eine Flag auf, die anzeigt, ob die 32-Bit-Farbdaten auf dem Bus sp_color gültig sind oder nicht. Die Schnittstelle weist ferner ein Signal ld_stall auf, das ein Blockiersignal von der Datensynchronisationssteuerung **209** zu dem Spannenschrittgeber **192** ist, welches anzeigt, daß die Datenregister **203** gegenwärtig Daten speichern und gegenwärtig keine weiteren Daten von dem Spannenschrittgeber **192** annehmen können.

[0067] Außerdem hält die Datensynchronisationssteuerung **209** ein Datensynchronisationsmodus-Signal, das anzeigt, welche Datenkomponenten (Diffus-, Spiegel-, Texel-Datenkomponenten) von der Texturmischeinheit **114** (Fig. 3) benötigt werden, um ein synchronisiertes Datenwort zu vollenden. Speziell zeigt das Datensynchronisationsmodus-Signal an, daß die Texturmischeinheit **114** (Fig. 3) nur eine Diffusfarbe, eine Diffus- und eine Spiegel-Farbe, eine Diffusfarbe und eine mit Texturabbildungen versehene Farbe oder eine Diffus-, Spiegel- und eine mit Texturabbildungen versehene Farbe benötigt. Das Modussignal weist ferner ein Bit auf, das anzeigt, ob die Tiefenhinweis-Mischeinheit **116** (Fig. 3) aktiviert oder deaktiviert ist.

[0068] Eine Schnittstelle zwischen der Datensynchronisationssteuerung **209** und den Datenregistern **203** weist ein Signal load_-pixel auf, das die x-, y-, und z- Koordinatendaten und die Diffusfarbdaten von dem Spannenschrittgeber in die jeweiligen x-, y-, z- und Diffus-Farbdaten-Register lädt, wie nachfolgend detaillierter in Verbindung mit Fig. 8 erläutert wird. In gleicher Weise lädt ein Signal load_specular die Spiegelfarbdaten von dem Spannenschrittgeber **192** in ein Spiegelfarbdatenregister (Fig. 8). Ferner lädt ein Signal load_texel die 32-Bit-Texelfarbdaten tm_color von der Texelpackereinheit **202** in ein Texelfarbdatenregister (Fig. 8). Noch ein weiteres Signal tri_xfer löscht die Spiegel- und Textur-Farbdaten-Wegregister (Fig. 8) auf Null.

[0069] Die Datensynchronisationssteuerung **209** gibt ferner ein Statuswort ld_status zu einer nächsten Stufe in der Verarbeitungspipeline aus. Das Wort ld_status weist ein Bit auf, das anzeigt, ob das abgehende synchronisierte Farb- und Koordinaten-Datenwort gültig ist oder nicht; ein Bit, das anzeigt, ob das abgehende synchronisierte Farb- und Koordinaten-Datenwort einem Pixel in einem Vektorgrundelement entspricht oder nicht; ein Bit, das anzeigt, ob das abgehende synchronisierte Farb- und Koordinaten-Datenwort einem Pixel eines Punktgrundelements entspricht oder nicht, und ein Bit, das anzeigt, ob das abgehende synchronisierte Farb- und Koordinaten-Datenwort einem Pixel eines Dreieck-Grundelements entspricht oder nicht. Außerdem empfängt

die Datensynchronisationssteuerung **209** ein Signal bc_stall, das von der Texturmischeinheit **114** ausgegeben wird, welches anzeigt, daß die Texturmischeinheit ein weiteres synchronisiertes Datenwort **201** von den Datenregistern **203** annehmen kann oder nicht.

[0070] In Fig. 8 ist detaillierter ein Blockdiagramm der Datenregister **203** gezeigt. Die Datenregister **203** bestehen aus einer Mehrzahl von Multiplexern **204** bis **214** und einer Mehrzahl von Registern **216** bis **226**. Die Datensynchronisationssteuerung **209** (Fig. 7) bestimmt, welche Register mit welchen Datenwerten beladen werden sollen. Sobald alle der Diffusfarbwerte, der Spiegelfarbwerte und der Texelfarbwerte, ebenso wie die x-Koordinaten-, die y-Koordinaten- und die z-Koordinaten-Werte in die entsprechenden Register **216** bis **226** geladen wurden, wird das synchronisierte Datenwort **201** (Fig. 7) für eine Farbzusammensetzung zu der Texturmischeinheit **114** übertragen.

[0071] Speziell wartet die Datensynchronisationssteuerung **209** auf die Texelfarbwerte, die Diffusfarbwerte und die Spiegelfarbwerte für jedes Pixel. Es sei angenommen, daß jeder dieser Werte diesem Pixel zugeordnet ist. Diese Annahme resultiert aus der Verwendung der identischen Spannenschrittgeber und Kantenberechnungseinheiten sowohl in der Texturabbildungsplatine **12** (Fig. 1) als auch dem Pixelgenerator **112** (Fig. 3). Wie oben erläutert wurde, liefert der Pixelgenerator **112** jedoch Diffus- und Spiegel-Farbwerte für Vektorgrundelemente und Punktgrundelemente, die in der Texturmischeinheit **114** nicht gemischt werden sollen. Folglich ermöglicht der Datensynchronisator **200**, daß die Diffus- und Spiegel-Farbwerte für Vektor- und Punkt-Pixel durch den Datensynchronisator **200** laufen. Der Datensynchronisator **200** bestimmt, ob die Diffus- und Spiegel-Farbwerte Pixeln eines Vektor- oder Punkt-Grundelements entsprechen, indem die Flag-Bits des Worts sp_status untersucht werden, welches zusammen mit den Daten durch die Pipeline des Abtastwandlers läuft.

[0072] In Fig. 8, in der ein Texelfarbwert tm_tcolor vor jedem der Spiegel- oder Diffus-Farbwerte sp_color ankommt, wird der Texelfarbwert beispielsweise in dem Register **226** gespeichert, und das Signal ld_tcolor_ack wird zu der Texelpackereinheit **202** ausgegeben, was anzeigt, daß das Datenregister **226** nicht für eine weitere Übertragung von Texeldaten bereit ist. Wie oben erläutert wurde, werden die Diffus- und Spiegel-Farbdaten auf der gleichen 32-Bit-Datenleitung sp_color zu den Datenregistern **222** und **224** übertragen, wobei zwei Taktzyklen benötigt werden, um sowohl den Diffus- als auch den Spiegel-Farbdatenwert von dem Spannenschrittgeber **192** zu den Datenregistern **222** und **224** zu übertragen. Wie ferner oben erläutert wurde, folgen die Spiegelfarbdaten stets den Diffusfarbdaten. Wenn jedoch die Spiegelfarboption nicht aktiviert ist, wird das Register **224** ausschließlich mit Nullen geladen, so daß die Datensynchronisationseinheit das synchronisierte Datenwort **201** ausgeben kann.

[0073] Nachfolgend wird der Betrieb der Multiplexer **204** bis **214** und der Datenregister **216** bis **226** beschrieben. Für jeden Taktzyklus wählt der Multiplexer zwischen Daten auf den Eingangsdatenleitungen, beispielsweise den Datenleitungen **228** und **240** für den Multiplexer **204** und gibt die gewählten Daten zu dem entsprechenden Datenregister **216** bis **226** aus. Das Signal load_pixel, das von der Datensynchronisationssteuerung **209** (Fig. 7) ausgegeben wird, weist den jeweiligen Multiplexer bezüglich dessen an, welche der Eingangsdatenleitungen als die Daten, die zu dem jeweiligen Datenregister ausgegeben werden sollen, ausgewählt werden sollen. Speziell wird, wenn das Signal load_pixel nicht aktiviert ist, die Eingangsdatenleitung, die mit dem Ausgang des jeweiligen Datenregisters **228** bis **238** verbunden ist, ausgewählt, so daß die Daten, die in dem jeweiligen Datenregister gespeichert sind, ausgewählt und zu dem Datenregister ausgegeben werden. Wenn das Signal load_pixel aktiviert ist, werden die Daten auf den jeweiligen Dateneingangsleitungen **240** bis **248** ausgewählt und in die jeweiligen Datenregister geladen.

[0074] Hinsichtlich der Multiplexer **212** und **214** wird, wenn die Texturmischeinheit **114** (Fig. 3) in einem Mischmodus ist, derart, daß entweder einer oder beide des Spiegelfarbwertes und des Texelfarbwertes für das entsprechende Pixel nicht durch die Mischeinheit verwendet werden soll, das Signal tri_xfer aktiviert, und 32 Nullen werden von einer oder beiden der Signalleitungen **245** und **247** ausgewählt und zu den jeweiligen Datenregistern **224** und **226** ausgegeben. Folglich sind die Multiplexer **204** bis **214** und die Datenregister **216** bis **226** in Koordination wirksam, um beständig den Datenwert zu speichern, der gegenwärtig in dem jeweiligen Datenregister ist, bis das Signal load_pixel für das jeweilige Datenregister aktiviert wird, wobei, wenn das Signal load_pixel aktiviert ist, die Daten auf der entsprechenden Datenleitung in das jeweilige Datenregister getaktet werden.

[0075] Es sollte offensichtlich sein, daß, obwohl bei dem oben beschriebenen Beispiel die Texelfarbdaten zuerst vor den x-, y-, z-Koordinatendaten und den Diffus-, Spiegel- und Farb-Werten an der Datensynchronisationseinheit ankommen, die Daten in einer beliebigen Reihenfolge an der Mehrzahl von Datenregistern **203** ankommen können, wobei die Datensynchronisationssteuerung **209** die Daten in das synchronisierte Datenwort **201** synchronisieren wird.

[0076] In den Fig. 9a bis Fig. 9e ist ein Verfahren

dargestellt, das durch die Datensynchronisationssteuerung **209** implementiert ist, um die Datensynchronisation gemäß der vorliegenden Erfindung zu steuern. Bei einem bevorzugten Ausführungsbeispiel der Erfindung ist die Synchronisationssteuerung eine Zustandsmaschine, die in einem Gatterarray implementiert ist. Es ist jedoch offensichtlich, daß die Steuerung in einem anderen Format, beispielsweise Software-mäßig, Hardwaremäßig oder dergleichen, implementiert sein kann, und daß derartige Ausführungsbeispiele laut Bestimmung durch die vorliegende Erfindung abgedeckt sein sollen.

[0077] Fig. 9a zeigt ein Verfahren zum Synchronisieren der Pixeldaten, wenn das Modussignal anzeigt, daß das Graphiksystem in einem Nur-Diffus-Modus ist. Die Datensynchronisationssteuerung **209** ist anfänglich in einem Leerlaufzustand **250**. Wenn das Modussignal den Nur-Diffus-Modus **251** anzeigt (Ja), springt das Verfahren entweder zu einem Zustand **252**, wenn die Daten einem Vektorpixel **252** entsprechen (Ja), einem Zustand **254**, wenn die Daten einem Punktpixel entsprechen (Ja), oder einem Zustand **256**, wenn die Daten einem Dreieck-Pixel in dem Nur-Diffus-Modus **256** entsprechen (Ja). Da keine zusätzlichen Daten benötigt werden, werden die Vektordaten **258**, die Punktdaten **260** und die Nur-Diffus-Dreieckdaten **262** in die jeweiligen Datenregister (Fig. 8) geladen. Die Vektordaten **264**, die Punktdaten **266** und die Nur-Diffus-Dreieckdaten **268** werden dann zu der nächsten Stufe in der Verarbeitungspipeline übertragen.

[0078] In Fig. 9b ist ein Verfahren gemäß der vorliegenden Erfindung zum Synchronisieren der Pixeldaten, wenn das Graphiksystem in dem Diffus- und Spiegel-Farbmodus ist, dargestellt. Wenn das Modussignal den Diffus- und Spiegel-Farbmodus **270** anzeigt (Ja), wartet die Datensynchronisationssteuerung **209** in einem Zustand **272** darauf, daß die Diffusfarbdaten an der Spannenschrittgeber-Schnittstelle ankommen. Beim Empfang der Diffusfarbdaten springt das Verfahren zu einem Zustand **274**, in dem die Diffusfarbdaten in ihr jeweiliges Register (Fig. 8) geladen werden **274**. Die Datensynchronisationssteuerung **209** wartet dann in einem Zustand **276** darauf, daß die Spiegelfarbdaten an der Spannenschrittgeber-Schnittstelle ankommen. Beim Empfang der Spiegelfarbdaten springt das Verfahren zu einem Zustand **278**, in dem die Spiegelfarbdaten in ihr entsprechendes Register (Fig. 8) geladen werden **278**. Die synchronisierten Daten **201** werden dann zu der nächsten Stufe in der Verarbeitungspipeline übertragen **280**, wobei die Steuerung in den Leerlaufzustand **250** zurückspringt.

[0079] In Fig. 9c ist ein Verfahren gemäß der vorliegenden Erfindung zum Synchronisieren der Pixeldaten, wenn das Graphiksystem in dem Diffus- und Texturtabellen-Farbmodus ist, dargestellt. Wenn das Modussignal anzeigt, daß die Pixeldaten in dem Diffus- und Texturtabellen-Modus sind **282** (Ja), können die Daten in einer von drei unterschiedlichen Reihenfolgen an dem Datenregister **203** ankommen, d.h. Texeldaten zuerst, Zustand **284**; Diffusdaten zuerst, Zustand **286**; oder Texel- und Diffus-Daten im gleichen Taktzyklus, Zustand **288**. Wenn in der Phase **284** bestimmt wird, daß die Texeldaten zuerst ankommen, springt das Verfahren zu einem Zustand **290**, in dem Texeldaten in ihr jeweiliges Datenregister (Fig. 8) geladen werden. Die Steuerung wartet dann in einem Zustand **290** darauf, daß die Diffuspixeldaten ankommen, wobei, wenn dies der Fall ist, dieselbe zu einem Zustand **294** springt, in dem die Diffusdaten in ihr jeweiliges Datenregister geladen werden. Das synchronisierte Diffus- und Texel-Farbdatenwort **201** wird dann in einer Phase **296** zu der nächsten Stufe in der Verarbeitungspipeline übertragen, und die Steuerung springt in den Leerlaufzustand **250** zurück.

[0080] Im Gegensatz dazu springt, wenn in der Phase **286** bestimmt wird, daß die Diffusfarbpixeldaten zuerst ankommen, die Datensynchronisationssteuerung **209** zu einer Phase **298**, in der die Diffusdaten in ihr jeweiliges Datenregister geladen werden. Die Steuerung wartet dann in einer Phase **300** darauf, daß die entsprechenden Texeldaten ankommen, wobei, wenn dies der Fall ist, die Steuerung zu einem Zustand **302** springt, in dem die Texeldaten in ihr jeweiliges Datenregister (Fig. 8) geladen werden. Das synchronisierte Diffus- und Texel-Farbdatenwort **201** wird dann zu der nächsten Stufe in der Verarbeitungspipeline übertragen **304**, wobei die Datensynchronisationssteuerung **209** in den Leerlaufzustand **250** zurückspringt.

[0081] Wenn als weitere Alternative im Schritt **280** bestimmt wird, daß die Texel- und Diffus-Daten zusammen ankommen, werden die Daten in einem Schritt **306** in die jeweiligen Datenregister (Fig. 8) geladen. Die synchronisierten Diffus- und Texel-Farbdaten werden dann in einem Zustand **308** zu der nächsten Stufe der Verarbeitungspipeline übertragen, wobei die Steuerung in den Leerlaufzustand **250** zurückspringt.

[0082] In Fig. 9d ist ein Verfahren gemäß der vorliegenden Erfindung dargestellt, um die Daten zu synchronisieren, wenn das Computergraphiksystem in dem Diffusfarb-, Spiegelfarb- und Texturabbildungs-Modus ist. Wenn die Mischeinheit **114** in dem Diffus-, Spiegel- und Texturabbildungs-Modus ist, kann die Steuerung einen von drei Modi annehmen, da die Daten in einer von drei Reihenfolgen ankommen können: Texeldaten zuerst **312**; Diffusdaten zuerst **314**; und Diffus- und Texel-Daten beim gleichen Taktsignal **318**.

[0083] Bei einem ersten Modus, wenn in einem Zu-

stand **312** bestimmt wird, daß die Texeldaten zuerst ankommen, werden die Texeldaten in einem Zustand **320** in ihr entsprechendes Datenregister (Fig. 8) geladen. Die Steuerung wartet dann in einem Zustand **322** auf die Ankunft der Diffusdaten und springt bei deren Ankunft in einen Zustand **324**, in dem die Diffusdaten in ihr entsprechendes Datenregister (Fig. 8) geladen werden. Die Steuerung wartet dann in einem Zustand **326** darauf, daß die Spiegeldaten ankommen, wobei, wenn dies der Fall ist, dieselbe zu einem Zustand **328** springt, in dem die Spiegeldaten in ihr jeweiliges Datenregister (Fig. 8) geladen werden. Das synchronisierte Spiegel-, Diffus- und Texel-Farbdatenwort **201** wird dann zu der nächsten Stufe in der Verarbeitungspipeline übertragen, und die Steuerung springt in den Leerlaufzustand **250**.

[0084] Bei einem zweiten Modus, wenn im Zustand **314** bestimmt wird, daß die Diffusfarbpixeldaten zuerst ankommen, werden die Diffusdaten in einem Zustand **332** in ihr jeweiliges Datenregister geladen. Wie in Fig. 9e gezeigt ist, kann die Steuerung dann einen von drei Untermodi betreten: Texel- und Spiegel-Daten kommen beim gleichen Taktzyklus an, Zustand **334**; Texeldaten als zweites, Zustand **336**; und Spiegeldaten als zweites, Zustand **338**. Wenn im Zustand **334** bestimmt wird, daß sowohl die Texel- als auch die Spiegel-Farbdaten gemeinsam ankommen, werden in einem ersten Untermodus die Texel- und Spiegel-Daten in einem Zustand **340** in die jeweiligen Datenregister (Fig. 8) geladen. Das synchronisierte Spiegel-, Diffus- und Texel-Farbdatenwort **201** wird dann in einem Zustand **342** zu der nächsten Stufe in der Verarbeitungspipeline übertragen, und die Steuerung springt in den Leerlaufzustand **250** zurück. Bei einem zweiten Untermodus, wenn im Zustand **336** bestimmt wird, daß die Texeldaten als nächstes ankommen, werden die Texeldaten in dem Zustand **334** in das jeweilige Datenregister geladen. Die Steuerung wartet dann in einem Zustand **346** auf den Empfang der Spektraldaten, wobei, beim Empfang derselben, die Spektraldaten in einem Zustand **348** in das jeweilige Datenregister (Fig. 8) geladen werden. Die synchronisierten Spiegel-, Diffus- und Texel-Farbdaten werden dann in einem Zustand **350** zu der nächsten Stufe in der Verarbeitungspipeline übertragen, und die Steuerung springt in den Leerlaufzustand **250** zurück. Bei einem dritten Untermodus, wenn im Zustand **338** bestimmt wird, daß die Spiegeldaten als nächstes ankommen, werden die Spiegeldaten in einem Zustand **352** in das jeweilige Datenregister (Fig. 8) geladen. Die Steuerung wartet dann in einem Zustand **354** auf die Texeldaten, wobei beim Empfang derselben die Texeldaten in einem Zustand **356** in das jeweilige Datenregister (Fig. 8) geladen werden. Die synchronisierten Spiegel-, Diffus- und Texel-Farbdaten werden dann in einem Zustand **358** zu der nächsten Stufe in der Verarbeitungspipeline übertragen, und die Steuerung springt in den Leerlaufzustand **250** zurück.

[0085] Bei einem dritten Modus, wenn im Zustand **318** bestimmt wird, daß die Diffus- und Texel-Daten zusammen ankommen, werden die Diffus- und Texel-Daten in einem Zustand **360** in die jeweiligen Datenregister (Fig. 8) geladen, wie in Fig. 9d gezeigt ist. Die Steuerung wartet dann in einem Zustand **362** auf die Spiegeldaten, wobei beim Empfang derselben die Spiegeldaten in einem Zustand **364** in das jeweilige Datenregister (Fig. 8) geladen werden. Das synchronisierte Spiegel-, Diffus- und Texel-Farbdatenwort **201** wird dann in einem Zustand **366** zu der nächsten Stufe in der Verarbeitungspipeline übertragen, und die Steuerung springt in den Leerlaufzustand **250** zurück.

[0086] Folglich ist es ein Vorteil der vorliegenden Erfindung, daß die Datensynchronisationssteuerung **209** keine spezielle Zeit benötigt, um sich nach einer Datenübertragung zu erholen. Insbesondere springt die Datensynchronisationssteuerung **209** unmittelbar in den Leerlaufzustand **250** zurück. In demselben kann dieselbe unmittelbar neue Daten an ihren Eingängen handhaben und möglicherweise die neuen Daten während eines einzelnen Taktzyklusses zu der nächsten Stufe übertragen.

[0087] Es ist ein weiterer Vorteil der vorliegenden Erfindung, daß die Pipeline, insbesondere die Anzahl von Signalleitungen, die Daten zwischen dem Texturabbildungschip **12** (Fig. 1) und den Rahmenpufferchips **14** übertragen müssen, gering ist. Ferner müssen keine Identifikations- oder Koordinaten-Informationen zwischen der Texturabbildungsplatine und den Rahmenpufferchips übertragen werden. Ferner ermöglicht die vorliegende Erfindung, daß andere Grundelementdaten, beispielsweise Vektoren und Punkte, mit Dreieck-Grundelementdaten durchsetzt sind, ohne die Pipeline räumen zu müssen, wenn die Texturabbildungsoption der Mischeinheit aktiviert ist. Folglich ist gemäß der vorliegenden Erfindung ein Computergraphiksystem mit einem verbesserten Verhalten geschaffen.

[0088] Sobald die Datensynchronisationssteuerung **209** die Texelfarbwerte, die Diffusfarbwerte und die Spiegelfarbwerte in ein einzelnes synchronisiertes Wort **201** für ein gegebenes Pixel zusammengebracht hat, werden die sychronisierten Farbinformationen durch die Texturmischeinheit **114** verwendet, um die Informationen in resultierende Farbdaten zusammen zu mischen. Bei einem bevorzugten Ausführungsbeispiel besitzt die Texturmischeinheit **114** mehrere Texturmischmodi, von denen einer zur Verwendung bei der Aufbereitung jedes Dreiecks verwendet wird. In einem "Keine-Änderung"-Modus werden die Diffusfarbwerte ohne Modifikation durch Texturfarben verwendet. Dies kann geeignet sein, wenn das Dreieck, das aufbereitet wird, keine Textur erfordert, oder wenn das Graphiksystem keine Texturabbildung aufweist. In diesem Fall werden die Diffus-

farbwerte nicht modifiziert. In dem "Ersetzungs"-Modus werden die Diffuswerte von dem Pixelgenerator durch die Texturtabellen-Farbwerte ersetzt. In dem "Modulations"-Modus werden die Diffusfarbwerte mit den Texturtabellen-Farbwerten multipliziert. In einem "Abziehbild"-Modus werden die Diffusfarbwerte von dem Pixelgenerator mit den Texturtabellen-Farbwerten gemäß einem Mischprozentsatz, der auf dem Alpha-Texel-Wert basiert, gemischt. Nach der Texturmischung werden die Spiegelfarbwerte optional zu den Zwischenfarbwerten addiert, um die resultierenden Farbwerte zu erzeugen. Die resultierenden Farbwerte, die durch die Texturmischeinheit **114** ausgegeben werden, werden dann zu der Tiefenhinweis-Mischeinheit **116** geliefert.

[0089] Der Schaltungsaufbau, der hierin gezeigt und beschrieben ist, ist nur als Beispiel angegeben. Der Schaltungsaufbau ist vorzugsweise unter Verwendung einer Logiksynthesesoftware, die kommerziell erhältlich ist, beispielsweise von Synopses, in einer kundenspezifischen integrierten Schaltung mit großer Stückzahl implementiert. Die logische Synthesesoftware optimiert und übersetzt Schaltungsbeschreibungen, die in Hochpegelsprachen geschrieben sind, beispielsweise Veralog, in logische Gatter. Der Schaltungsaufbau kann unter Verwendung eines CMOS-Verfahrens implementiert sein, das 1-Mikrometer-FETs erzeugt, die bei 5 Volt arbeiten, eines CMOS-Verfahrens, das Bauelemente mit einer gezogenen Gatelänge von 0,6 Mikrometer erzeugt, welche bei 3,3 Volt arbeiten, oder eines beliebigen anderen geeigneten Verfahrens zum Implementieren digitaler Schaltungen. Da die Eingabe in die logische Synthesesoftware funktionell und nicht strukturell ist, können sich die tatsächlichen Schaltungen, die durch die logische Synthesesoftware erzeugt werden, von denjenigen, die hierin offenbart sind, unterscheiden.

Patentansprüche

1. Datensynchronisator (**200**) in einem Graphiksystem, der Sätze von Datenworten für ein Dreieck-Grundelement synchronisiert, um synchronisierte Sätze von Datenworten für das Dreieck-Grundelement zu erzeugen, wobei jedes Datenwort in einem Satz von Datenworten durch eine einer Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen (**46**, **112**) zu dem Datensynchronisator übergeben wird, wobei zumindest eine der asynchronen Datenaufbereitungsvorrichtungen (**46**, **112**) Datenwörter für Vektor-Grundelemente, Punkt-Grundelemente und die Dreieck-Grundelemente aufbereitet, wobei der Datensynchronisator (**200**) folgende Merkmale aufweist:
eine Mehrzahl von Datenregistern (**216**, **218**, **220**, **222**, **224**, **226**), wobei jedes Datenregister ein Datenwort in einem ersten Satz von Datenworten für das Dreieck-Grundelement von einer entsprechenden der Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen empfängt und temporär speichert, während es Datenwörter für die Vektor-Grundelemente und die Punkt-Grundelemente durchläßt, wobei jedes Datenregister (**216**, **218**, **220**, **222**, **224**, **226**) bei der Aktivierung eines Steuersignals sein gespeichertes Datenwort als eines von einem ersten synchronisierten Satz von Datenworten ausgibt; und
eine Steuerung (**209**), die verhindert, daß jede asynchrone Datenaufbereitungsvorrichtung ein Datenwort in einen zweiten Satz von Datenworten für das Dreieck-Grundelement zu seinem entsprechenden Datenregister (**216**, **218**, **220**, **222**, **224**, **226**) überträgt, wenn das Datenregister sein entsprechendes Datenwort in dem ersten Satz von Datenworten speichert, und die das Steuersignal aktiviert, wenn jedes der Datenregister (**216**, **218**, **220**, **222**, **224**, **226**) sein Datenwort in dem ersten Satz von Datenworten empfangen hat.

2. Datensynchronisator gemäß Anspruch 1, der ferner eine Mehrzahl von Multiplexern (**204**, **206**, **208**, **210**, **212**, **214**) aufweist, wobei jeder der Mehrzahl von Multiplexern einen Ausgang aufweist, der mit einem Eingang eines entsprechenden der Mehrzahl von Datenregistern (**216**, **218**, **220**, **222**, **224**, **226**) gekoppelt ist, einen ersten Eingang (**240**, **242**, **244**, **246**, **248**), der mit einem Ausgang einer entsprechenden der Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen gekoppelt ist, und einen zweiten Eingang (**228**, **230**, **232**, **234**, **236**, **238**), der mit einem Ausgang des entsprechenden der Mehrzahl von Datenregistern gekoppelt ist.

3. Datensynchronisator gemäß Anspruch 2, bei dem der erste Eingang (**246**) eines ersten (**210**) der Mehrzahl von Multiplexern und der erste Eingang (**246**) eines zweiten (**212**) der Mehrzahl von Multiplexern mit dem Ausgang einer ersten der Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen verbunden sind, bei dem ein erstes Datenwort, das durch die erste der Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen geliefert wird, als Reaktion auf einen ersten Zyklus eines Taktsignals durch den ersten Multiplexer (**210**) empfangen und in einem ersten Datenregister (**222**) temporär gespeichert wird, und bei dem ein zweites Datenwort, das durch die erste der Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen geliefert wird, als Reaktion auf einen zweiten Zyklus des Taktsignals durch den zweiten Multiplexer (**212**) empfangen und in einem zweiten Datenregister (**224**) temporär gespeichert wird.

4. Graphiksystem mit folgenden Merkmalen:
einer Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen, wobei zumindest eine asynchrone Datenaufbereitungsvorrichtung Datenwörter für Vektor-Grundelemente, Punkt-Grundelemente und Dreieck-Grundelemente liefert; und
einem Datensynchronisator (**200**) nach einem der Ansprüche 1 bis 3.

5. Graphiksystem gemäß Anspruch 4, bei dem die Steuereinrichtung ferner eine Einrichtung zum Bestimmen, ob ein erstes Datenwort von einer ersten asynchronen Datenaufbereitungsvorrichtung mit einem zweiten Datenwort von einer zweiten asynchronen Datenaufbereitungsvorrichtung in entweder einem Nur-Diffus-Mischmodus, einem Diffus- und Spiegel-Mischmodus, einem Diffus- und Mit-Texturabbildung-Versehen-Mischmodus oder einem Diffus-, Spiegel- und Mit-Texturabbildung-Versehen-Mischmodus gemischt werden soll, aufweist.

6. Graphiksystem gemäß Anspruch 5, bei dem die Steuereinrichtung ferner eine Einrichtung zum Liefern eines voreingestellten Datenwortes als das zweite Datenwort zu einem entsprechenden Datenregister (**216**, **218**, **220**, **222**, **224**, **226**) aufweist, wenn die Einrichtung zum Bestimmen bestimmt, daß das erste Datenwort nicht mit dem zweiten Datenwort gemischt werden soll.

7. Graphiksystem gemäß Anspruch 5 oder 6, bei dem die Bestimmungseinrichtung ferner eine Einrichtung zum Bestimmen, ob das erste Datenwort einem Pixel entweder des Punkt-Grundelements, des Vektor-Grundelements oder des Dreieck-Grundelements in dem Nur-Diffus-Mischmodus entspricht, aufweist.

8. Graphiksystem gemäß einem der Ansprüche 5 bis 7, bei dem die Einrichtung zum Bestimmen eine Einrichtung zum Bestimmen, ob das erste Datenwort einem Pixel des Dreieck-Grundelements entweder in dem Diffus- und Spiegel-Mischmodus oder dem Diffus- und Mit-Texturabbildung-Versehen-Mischmodus entspricht, aufweist.

9. Graphiksystem gemäß einem der Ansprüche 5 bis 8, bei dem die Einrichtung zum Bestimmen eine Einrichtung zum Bestimmen, ob das erste Datenwort einem Pixel des Dreieck-Grundelements in dem Diffus-, Spiegel- und Mit-Texturabbildung-Versehen-Mischmodus entspricht, aufweist.

10. Graphiksystem gemäß einem der Ansprüche 4 bis 9, bei dem eine erste asynchrone Datenaufbereitungsvorrichtung der Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen eine Pixel-Erzeugungseinrichtung (**112**) zum Empfangen von Aufbereitungsparametern, zum Berechnen von x-, y-, z-Koordinaten und Farbwerten für jedes Pixel in den Vektor-Grundelementen, den Punkt-Grundelementen und den Dreieck-Grundelementen für die Aufbereitungsparameter und zum Ausgeben der x-, y-, z-Koordinaten und der Farbwerte als das Datenwort in das entsprechende Datenregister der Mehrzahl von Datenregistern (**216**, **218**, **220**, **222**, **224**, **226**) aufweist.

11. Graphiksystem gemäß Anspruch 10, bei dem eine zweite asynchrone Datenaufbereitungsvorrichtung der Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen eine Texelerzeugungseinrichtung (**46**) zum Empfangen der Aufbereitungsparameter, zum Berechnen von Texelfarbwerten für jedes Pixel in dem Dreieck-Grundelement aus den Aufbereitungsparametern und zum Ausgeben der Texelfarbwerte als das Datenwort zu dem entsprechenden Datenregister der Mehrzahl von Datenregistern (**216**, **218**, **220**, **222**, **224**, **226**) aufweist.

12. Graphiksystem gemäß Anspruch 11, bei dem sowohl die Pixelerzeugungseinrichtung (**112**) als auch die Texelerzeugungseinrichtung (**46**) eine identische Kantenschrittgebereinrichtung (**180**, **182**) zum Berechnen der Farbwerte für jedes Pixel entlang einer Kante des Dreieck-Grundelements und zum Liefern der Farbwerte in einer gleichen Reihenfolge aufweisen.

13. Graphiksystem gemäß Anspruch 11 oder 12, bei dem sowohl die Pixelerzeugungseinrichtung (**112**) als auch die Texelerzeugungseinrichtung (**46**) eine identische Spannenschrittgebereinrichtung (**192**) zum Berechnen der Farbwerte für jedes Pixel entlang einer Spanne des Dreieck-Grundelements und zum Liefern der Farbwerte in einer gleichen Reihenfolge aufweisen.

14. Graphiksystem gemäß einem der Ansprüche 11 bis 13, das ferner eine Texelpackereinrichtung (**202**) aufweist, die zwischen der Texelerzeugungseinrichtung (**46**) und dem entsprechenden Datenregister (**216**, **218**, **220**, **222**, **224**, **226**) angeordnet ist, zum Empfangen der Texelfarbwerte, zum Packen der Texelfarbwerte für jede einer Rot-Komponente, einer Grün-Komponente, einer Blau-Komponente und einer Alpha-Komponente in ein einzelnes Texelfarbdatenwort und zum Ausgeben des einzelnen Texelfarbdatenworts zu dem entsprechenden Datenregister.

15. Graphiksystem gemäß Anspruch 14, bei dem eine Schnittstelle zwischen der Texturabbildungsschaltung (**46**) und der Texelpackerschaltung (**202**) elf Datensignalleitungen aufweist.

16. Verfahren zum Synchronisieren von Sätzen von Datenworten für ein Dreieck-Grundelement in einem Graphiksystem, um synchronisierte Sätze von Datenworten für das Dreieck-Grundelement zu erzeugen, wobei jedes Datenwort in einem Satz von Datenworten durch eine einer Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen (**46**, **112**) aufbereitet wird, wobei zumindest eine der Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen Datenwörter für Vektor-Grundelemente, Punkt-Grundelemente und die Dreieck-Grundelemente aufbereitet, wobei das Verfahren folgende Schritte aufweist:
Durchlassen von Datenwörtern für die Vektor-Grundelemente und die Punkt-Grundelemente;
Empfangen eines ersten Datenworts in einem ersten

Satz von Datenworten von einer ersten asynchronen Datenaufbereitungsvorrichtung;
temporäres Speichern des ersten Datenworts;
Warten auf den Empfang eines zweiten Datenworts;
Empfangen des zweiten Datenworts für das Dreieck-Grundelement; und
Ausgeben eines synchronisierten Satzes von Datenworten, der das erste Datenwort und das zweite Datenwort aufweist.

17. Verfahren gemäß Anspruch 16, das ferner den Schritt des Bestimmens, ob das erste Datenwort mit dem zweiten Datenwort gemischt werden soll, aufweist.

18. Verfahren gemäß Anspruch 17, das ferner den Schritt des Lieferns eines voreingestellten Datenworts als das zweite Datenwort, wenn das erste Datenwort nicht mit dem zweiten Datenwort gemischt werden soll, aufweist.

19. Verfahren gemäß Anspruch 17 oder 18, das ferner den Schritt des Verhinderns, daß die erste asynchrone Datenaufbereitungsvorrichtung ein Datenwort in einem zweiten Satz von Datenworten sendet, bis der synchronisierte Satz von Datenworten ausgegeben wurde, aufweist.

20. Verfahren gemäß Anspruch 16, das ferner folgende Schritte aufweist:
temporäres Speichern des zweiten Datenworts; und
Warten auf den Empfang eines dritten Datenworts.

21. Verfahren zum Synchronisieren von Sätzen von Datenworten, gemäß einem der Ansprüche 16 bis 20 zum Erzeugen von synchronisierten Sätzen von Datenworten in einem Graphiksystem, das eine Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen aufweist, wobei eine der Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen ein Pixelgenerator (**112**) ist, der ein Datenwort für jedes Pixel in den Vektor-Grundelementen, den Punkt-Grundelementen und den Dreieck-Grundelementen aufbereitet, und eine weitere der Mehrzahl von asynchronen Datenaufbereitungsvorrichtungen eine Texturabbildungsschaltung (**46**) ist, die ein Datenwort für jedes Pixel in dem Dreieck-Grundelement aufbereitet.

22. Verfahren gemäß Anspruch 21, das ferner den Schritt des Bestimmens, ob das erste Datenwort mit dem zweiten Datenwort in entweder einem Nur-Diffus-Mischmodus, einem Diffus- und Spiegel-Mischmodus, einem Diffus- und Mit-Texturabbildung-Versehen-Mischmodus oder einem Diffus-, Spiegel- und Mit-Texturabbildung-Versehen-Mischmodus gemischt werden soll, aufweist.

23. Verfahren gemäß Anspruch 22, bei dem, wenn das erste Datenwort mit dem zweiten Datenwort gemischt werden soll, der Schritt des Wartens auf den Empfang des zweiten Datenworts das Warten auf den Empfang des zweiten Datenworts von entweder dem Pixelgenerator (**112**) oder der Texturabbildungsschaltung (**46**) aufweist.

24. Verfahren gemäß Anspruch 22 oder 23, das ferner den Schritt des Lieferns eines voreingestellten Datenwortes als das zweite Datenwort aufweist, wenn das erste Datenwort nicht mit dem zweiten Datenwort gemischt werden soll.

25. Verfahren gemäß Anspruch 23 oder 24, das ferner den Schritt des Verhinderns, daß entweder der Pixelgenerator (**112**) oder die Texturabbildungsschaltung (**46**) ein Datenwort in einem zweiten Satz von Datenworten senden, bis der synchronisierte Satz von Datenworten ausgegeben wurde, aufweist.

26. Verfahren gemäß einem der Ansprüche 22 bis 25, bei dem der Schritt des Bestimmens, ob das erste Datenwort gemischt werden soll, das Bestimmen, ob das erste Datenwort, das durch den Pixelgenerator (**112**) aufbereitet wird, einem Pixel entweder des Punkt-Grundelements, des Vektor-Grundelements oder des Dreieck-Grundelements in dem Nur-Diffus-Mischmodus entspricht, aufweist.

27. Verfahren gemäß einem der Ansprüche 22 bis 26, bei dem der Schritt des Bestimmens, ob das erste Datenwort gemischt werden soll, das Bestimmen, ob das erste Datenwort einem Pixel eines Dreieck-Grundelements entweder in dem Diffus- und Spiegel-Mischmodus oder dem Diffus- und Mit-Texturabbildung-Versehen-Mischmodus entspricht, aufweist.

28. Verfahren gemäß Anspruch 27, bei dem, wenn das erste Datenwort ein Diffusfarbdatenwort ist, das durch den Pixelgenerator (**112**) aufbereitet wird, in dem Diffus- und Spiegel-Mischmodus,
der Schritt des Speicherns des ersten Datenworts das Speichern des Diffusfarbdatenworts aufweist; und
der Schritt des Wartens auf das zweite Datenwort das Warten auf den Empfang eines Spiegelfarbdatenworts von dem Pixelgenerator (**112**) aufweist.

29. Verfahren gemäß Anspruch 27, bei dem, wenn das erste Datenwort in dem Diffus- und Mit-Texturabbildung-Versehen-Mischmodus ist,
der Schritt des Empfangens des ersten Datenworts das Empfangen entweder eines Texelfarbdatenworts von der Texturabbildungsschaltung (**46**), eines Diffusfarbdatenworts von dem Pixelgenerator (**46**) oder sowohl dem Texelfarbdatenwort als auch dem Diffusfarbdatenwort in einem einzelnen Taktzyklus entsprechend von der Texturabbildungsschaltung (**46**) und dem Pixelgenerator (**112**) aufweist;
der Schritt des Speicherns des ersten Datenworts

das Speichern entweder des Texelfarbdatenworts, des Diffusfarbdatenworts oder sowohl des Texelfarbdatenworts als auch des Diffusfarbdatenworts aufweist;
der Schritt des Wartens auf den Empfang des zweiten Datenworts das Warten auf den Empfang jeweils entweder des Diffusfarbdatenworts oder des Texelfarbdatenworts aufweist, wenn entweder das Texelfarbdatenwort oder das Diffusfarbdatenwort empfangen wird; und
der Schritt des Ausgebens das Ausgeben des synchronisierten Satzes von Datenworten, der das Texelfarbdatenwort und das Diffusfarbdatenwort aufweist, aufweist.

30. Verfahren gemäß Anspruch 21, das ferner folgende Schritte aufweist:
Speichern des zweiten Datenworts; und
Warten auf den Empfang eines dritten Datenworts.

31. Verfahren gemäß Anspruch 30, bei dem der Schritt des Bestimmens, ob das erste Datenwort gemischt werden soll, das Bestimmen, ob das erste Datenwort einem Pixel des Dreieck-Grundelements in dem Diffus-, Spiegel- und Mit-Texturabbildung-Versehen-Mischmodus entspricht, aufweist.

Es folgen 13 Blatt Zeichnungen

Anhängende Zeichnungen

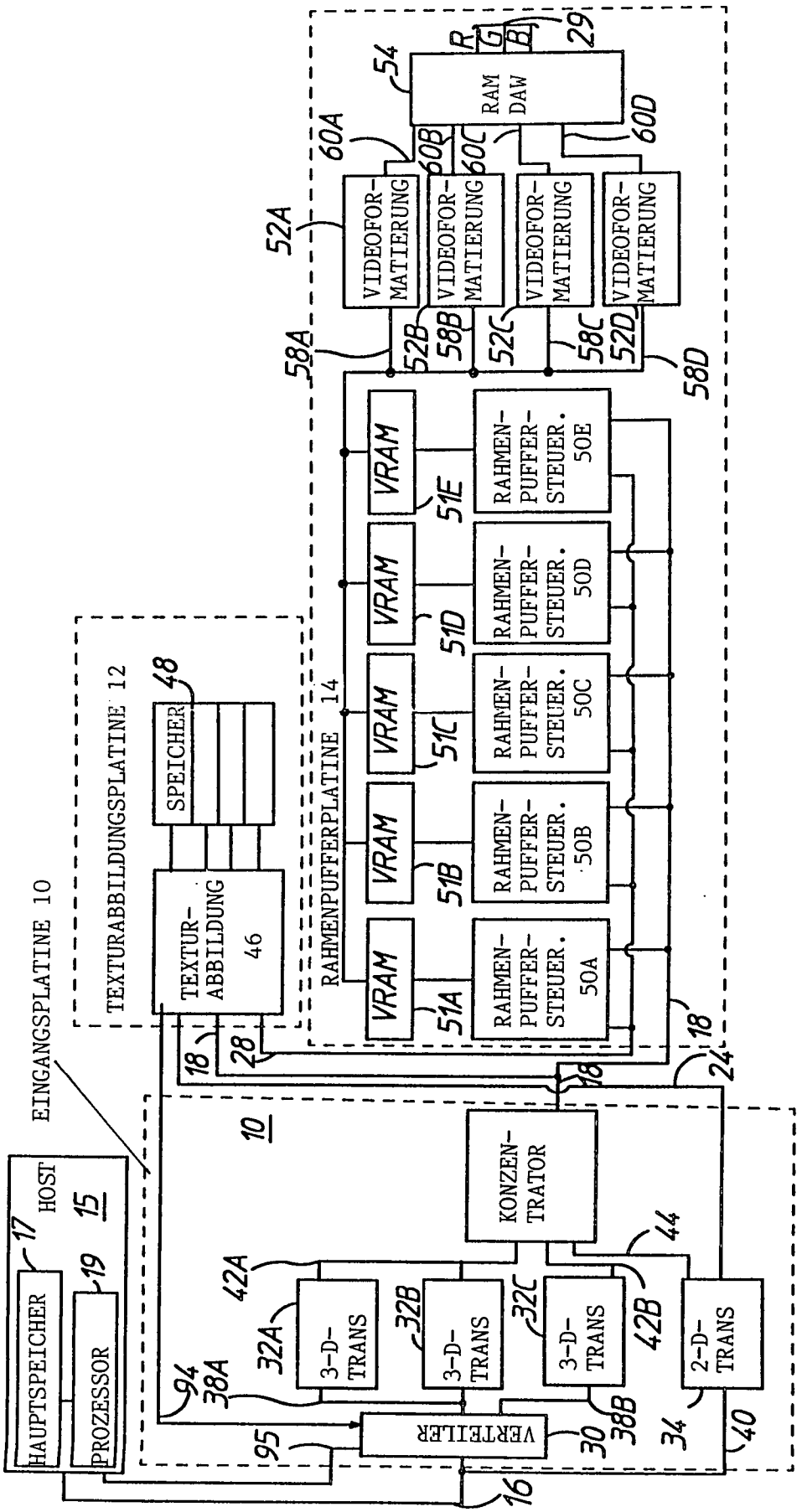


FIG 1

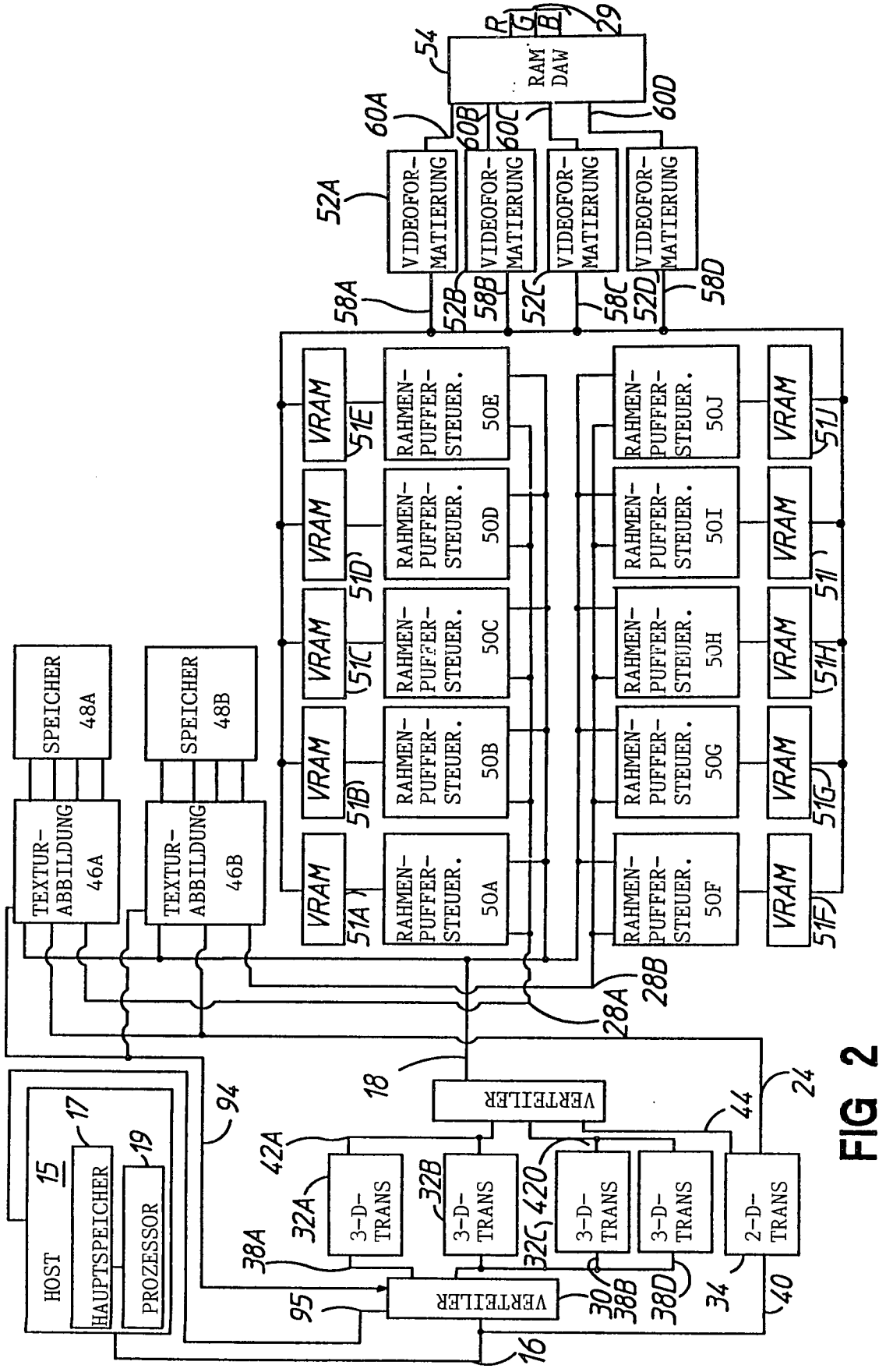

HOST
15
HAUPTSPEICHER
17
PROZESSOR
19
94
16
VERTEILER
18
VERTEILER
95
3-D-TRANS
32A
3-D-TRANS
32B
3-D-TRANS
32C
3-D-TRANS
420
2-D-TRANS
34
40
42A
44
24
38A
30
38B
38D
28A
28B
TEXTUR-ABBILDUNG 46A
TEXTUR-ABBILDUNG 46B
SPEICHER 48A
SPEICHER 48B
VRAM
51A
51B
51C
51D
51E
51F
51G
51H
51I
51J
RAHMEN-PUFFER-STEUER. 50A
RAHMEN-PUFFER-STEUER. 50B
RAHMEN-PUFFER-STEUER. 50C
RAHMEN-PUFFER-STEUER. 50D
RAHMEN-PUFFER-STEUER. 50E
RAHMEN-PUFFER-STEUER. 50F
RAHMEN-PUFFER-STEUER. 50G
RAHMEN-PUFFER-STEUER. 50H
RAHMEN-PUFFER-STEUER. 50I
RAHMEN-PUFFER-STEUER. 50J
VIDEOFOR-MATIERUNG
52A
52B
52C
52D
58A
58B
58C
58D
60A
60B
60C
60D
RAM DAW
54
R
G
B
29

FIG 2

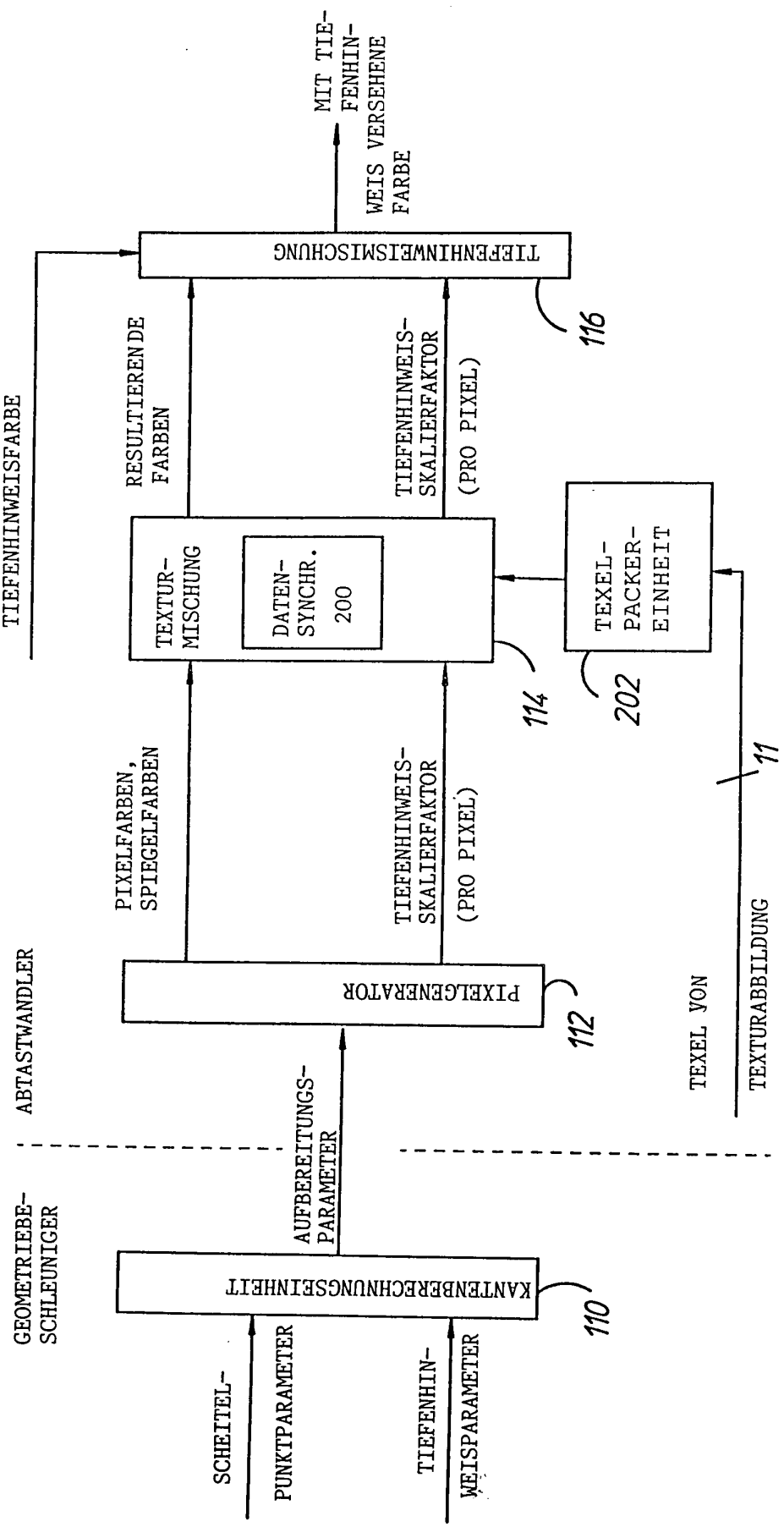

GEOMETRIEBE-SCHLEUNIGER
ABTASTWANDLER
SCHEITEL-PUNKTPARAMETER
TIEFENHIN-WEISPARAMETER
KANTENBERECHNUNGSEINHEIT
110
AUFBEREITUNGS-PARAMETER
PIXELGENERATOR
112
PIXELFARBEN, SPIEGELFARBEN
TIEFENHINWEIS-SKALIERFAKTOR (PRO PIXEL)
TEXEL VON TEXTURABBILDUNG
11
TEXEL-PACKER-EINHEIT
202
TEXTUR-MISCHUNG
DATEN-SYNCHR. 200
114
RESULTIERENDE FARBEN
TIEFENHINWEIS-SKALIERFAKTOR (PRO PIXEL)
TIEFENHINWEISFARBE
TIEFENHINWEISMISCHUNG
116
MIT TIEFENHINWEIS VERSEHENE FARBE

FIG 3

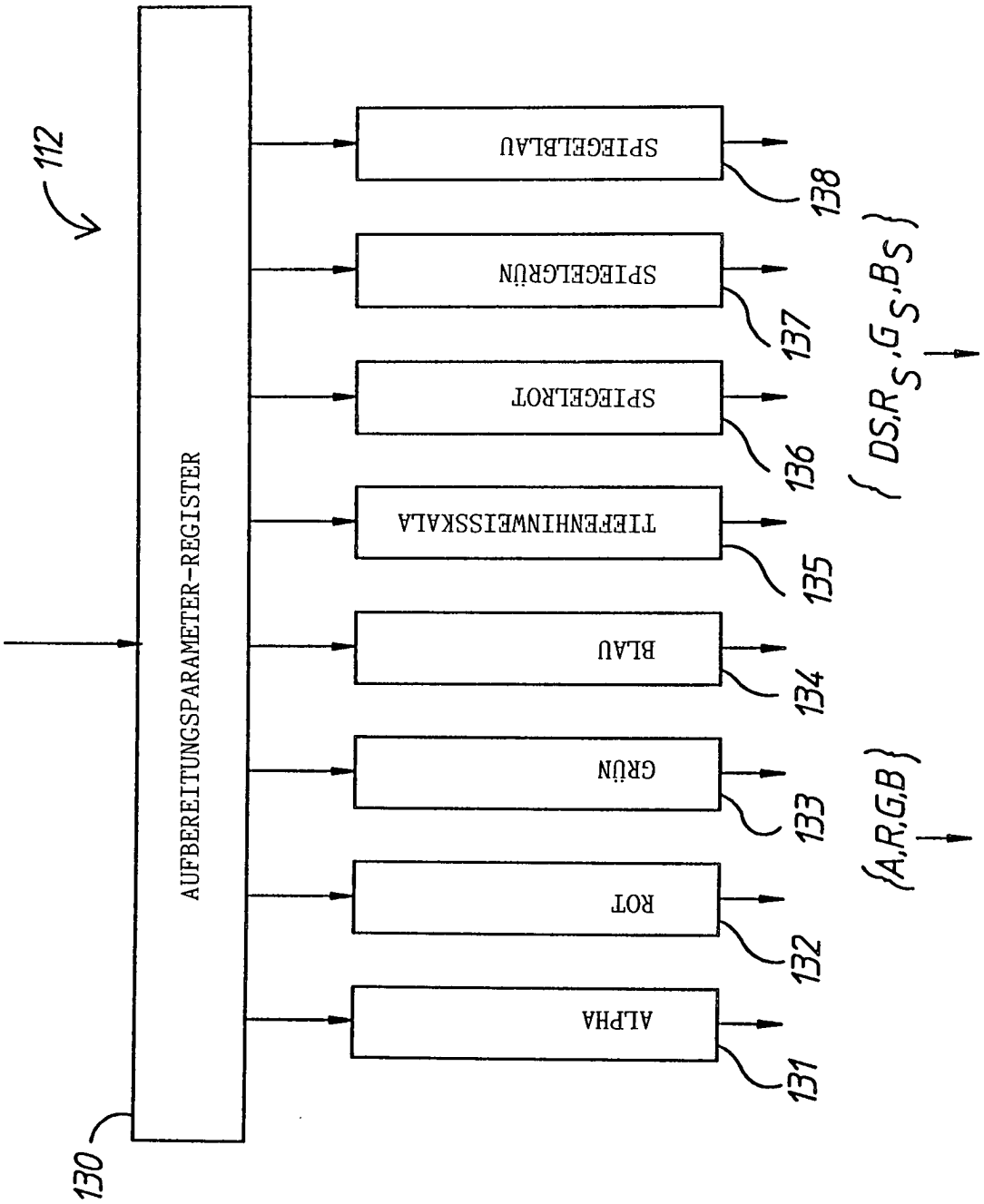

112
AUFBEREITUNGSPARAMETER-REGISTER
130
ALPHA
131
ROT
132
GRÜN
133
BLAU
134
TIEFENHINWEISSKALA
135
SPIEGELROT
136
SPIEGELGRÜN
137
SPIEGELBLAU
138
{A,R,G,B}
{DS,RS,GS,BS}

FIG 4

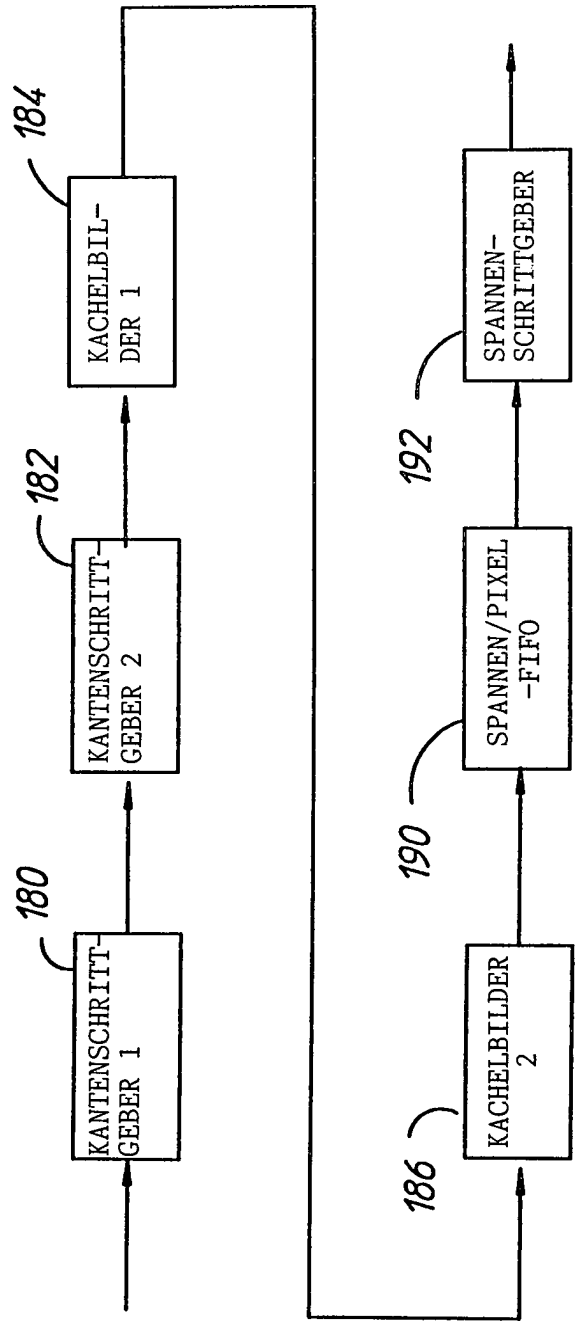

180
KANTENSCHRITT-GEBER 1
182
KANTENSCHRITT-GEBER 2
184
KACHELBIL-DER 1
186
KACHELBILDER 2
190
SPANNEN/PIXEL -FIFO
192
SPANNEN-SCHRITTGEBER

FIG 5

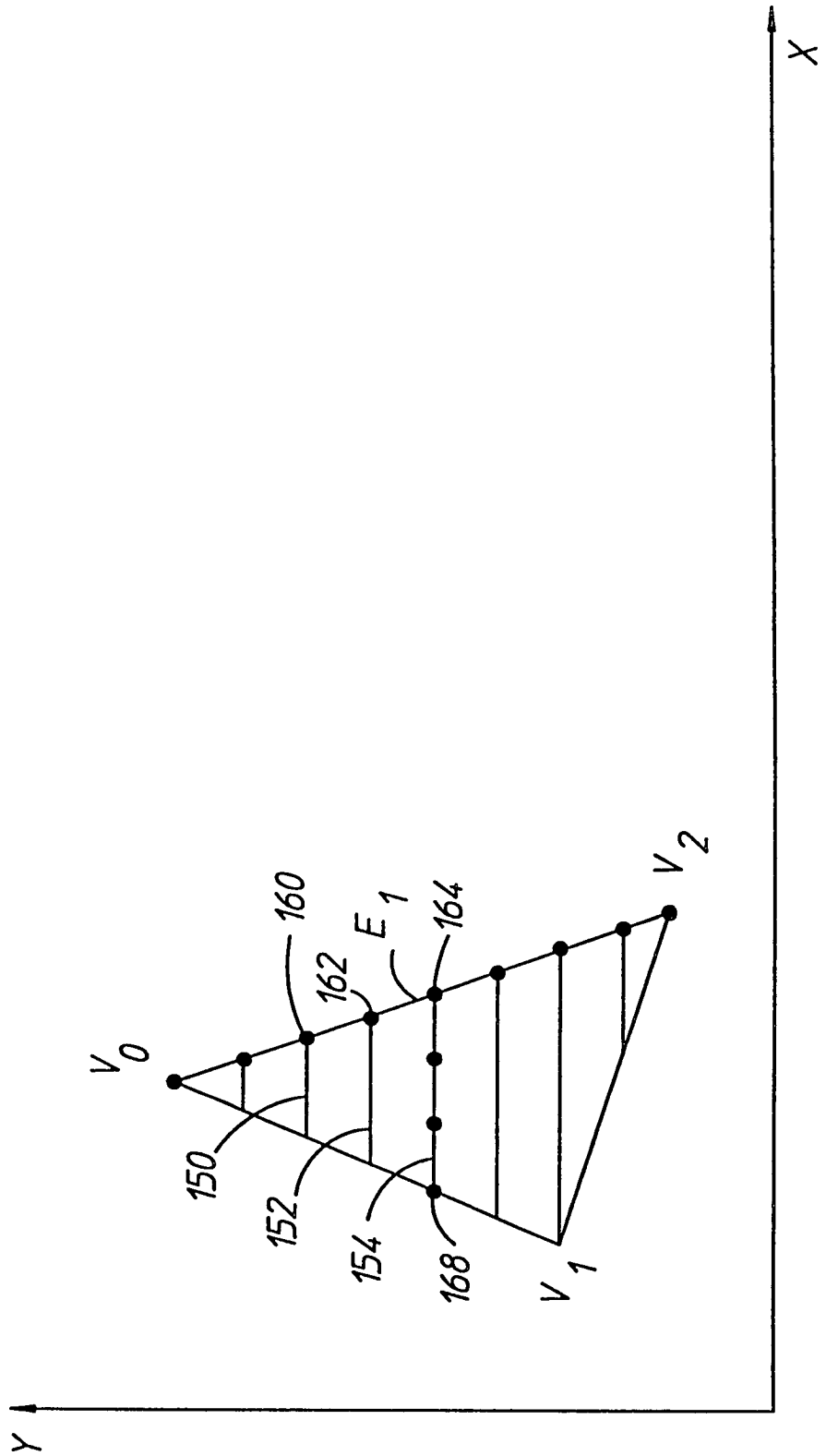


FIG 6

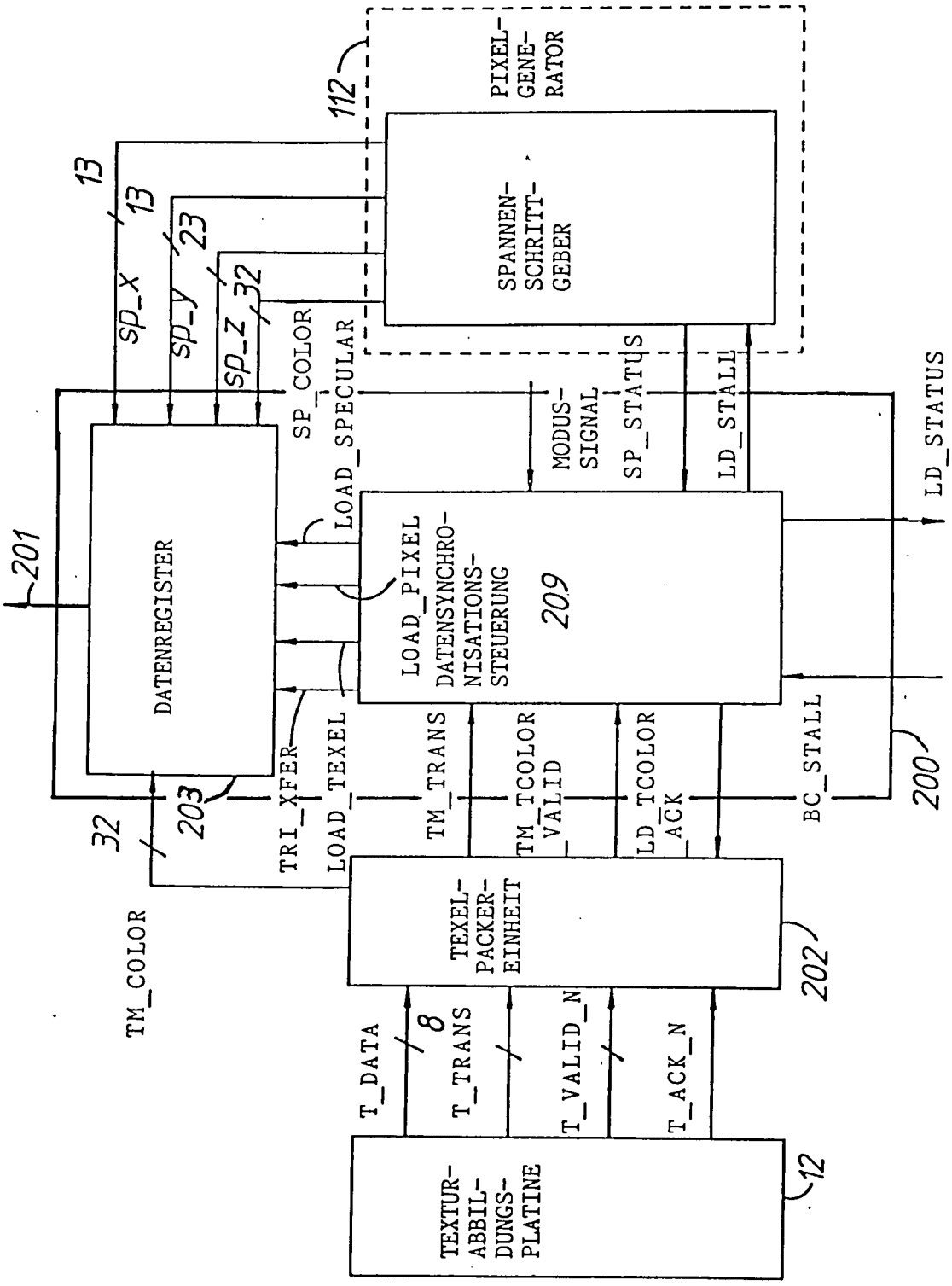
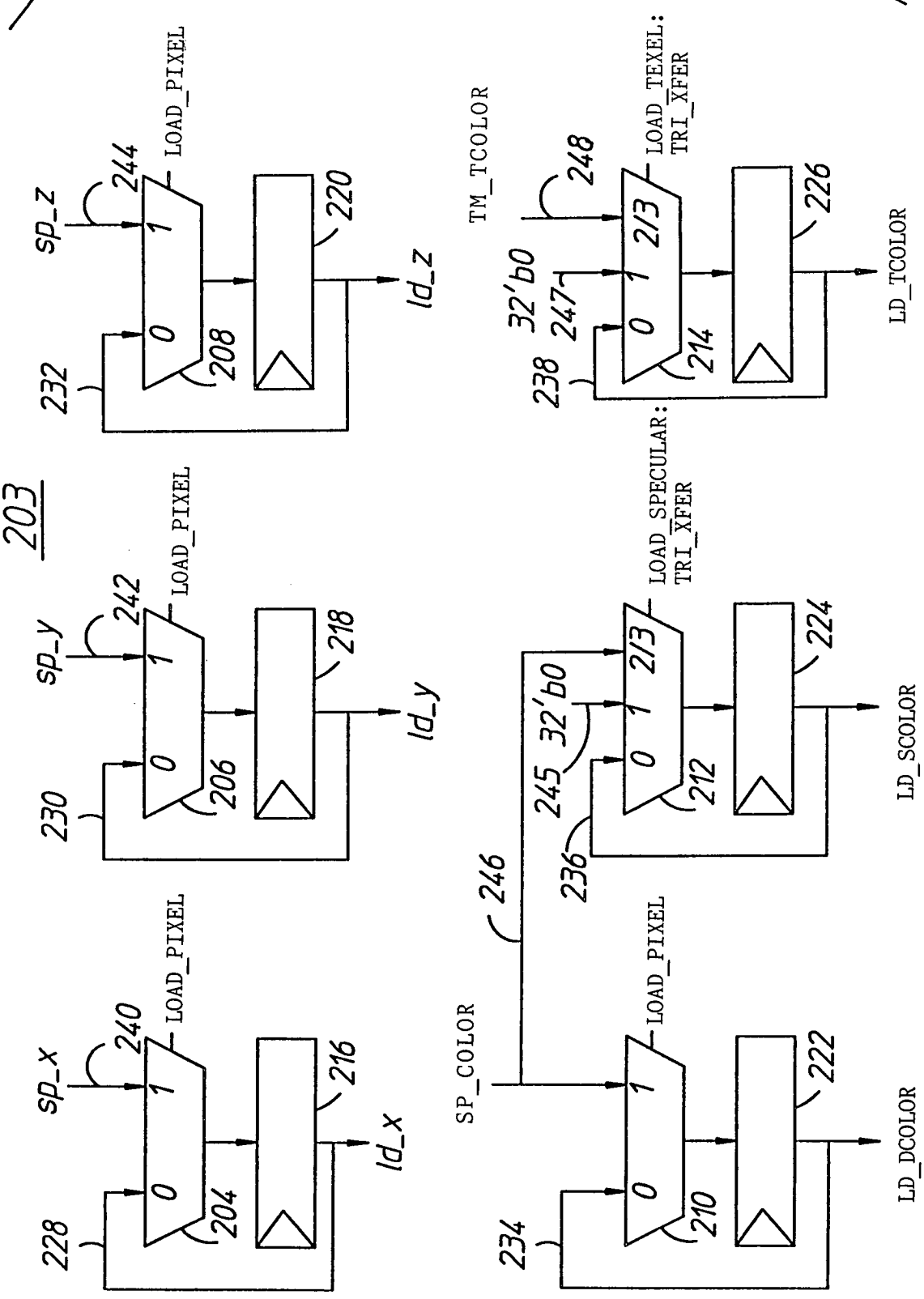

PIXEL-GENE-RATOR
112
SPANNEN-SCHRITT-GEBER
13
13
23
32
sp_x
sp_y
sp_z
SP_COLOR
LOAD_SPECULAR
MODUS-SIGNAL
SP_STATUS
LD_STALL
LD_STATUS
201
DATENREGISTER
LOAD_PIXEL
DATENSYNCHRO-NISATIONS-STEUERUNG
209
TRI_XFER
LOAD_TEXEL
TM_TRANS
TM_TCOLOR VALID
LD_TCOLOR ACK
BC_STALL
200
203
32
TM_COLOR
TEXEL-PACKER-EINHEIT
202
T_DATA
8
T_TRANS
T_VALID_N
T_ACK_N
TEXTUR-ABBIL-DUNGS-PLATINE
12

FIG 7

FIG 8


203
sp_x
240
LOAD_PIXEL
228
204
216
ld_x
sp_y
242
LOAD_PIXEL
230
206
218
ld_y
sp_z
244
LOAD_PIXEL
232
208
220
ld_z
SP_COLOR
246
LOAD_PIXEL
234
210
222
LD_DCOLOR
245
32'b0
LOAD_SPECULAR:
TRI_XFER
236
212
213
224
LD_SCOLOR
TM_TCOLOR
248
LOAD_TEXEL:
TRI_XFER
32'b0
247
238
214
213
226
LD_TCOLOR

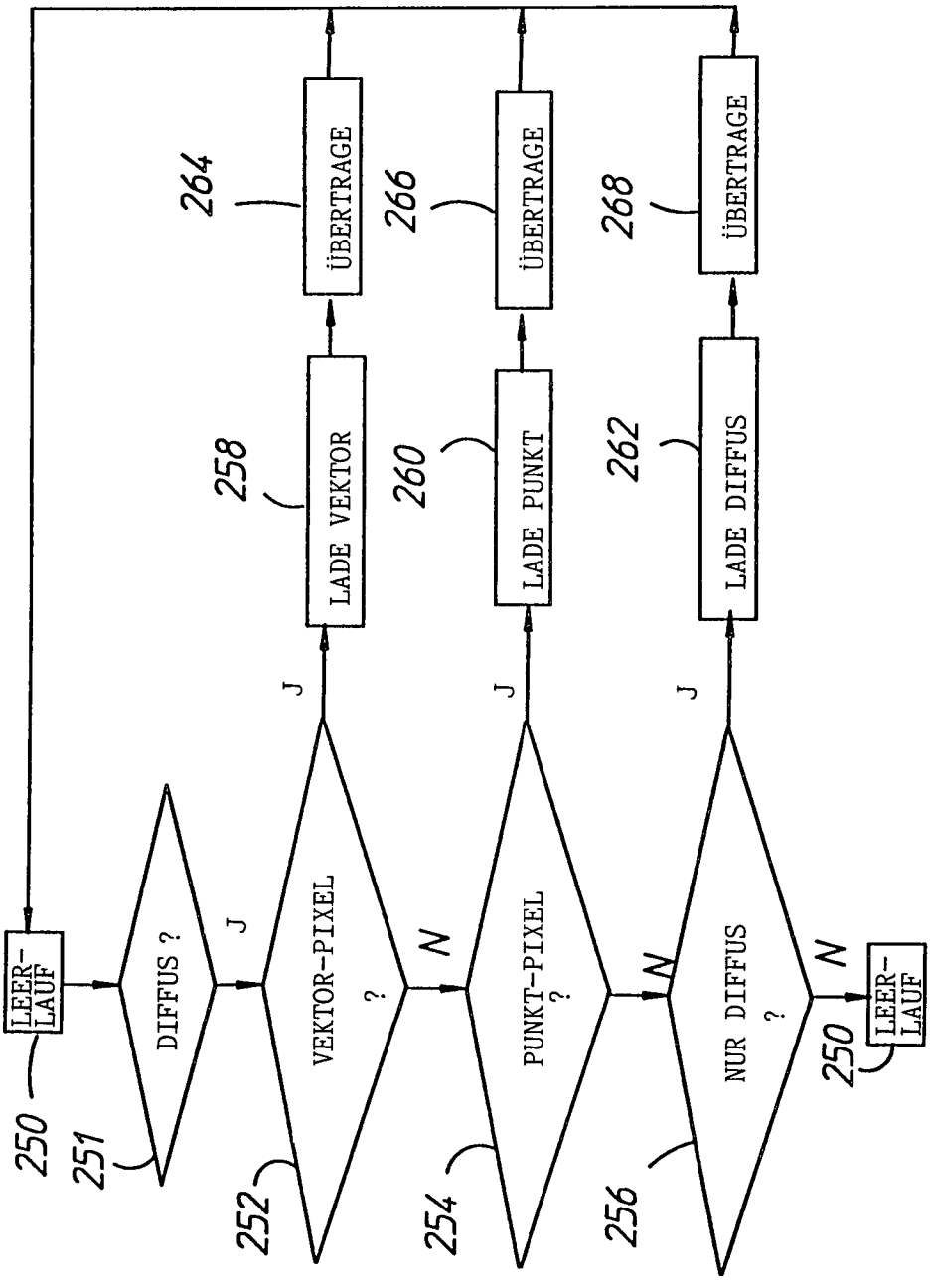

LEER-LAUF
250
DIFFUS ?
251
J
VEKTOR-PIXEL ?
252
N
PUNKT-PIXEL ?
254
N
NUR DIFFUS ?
256
N
LEER-LAUF
250
J
LADE VEKTOR
258
J
LADE PUNKT
260
J
LADE DIFFUS
262
ÜBERTRAGE
264
ÜBERTRAGE
266
ÜBERTRAGE
268

FIG 9A

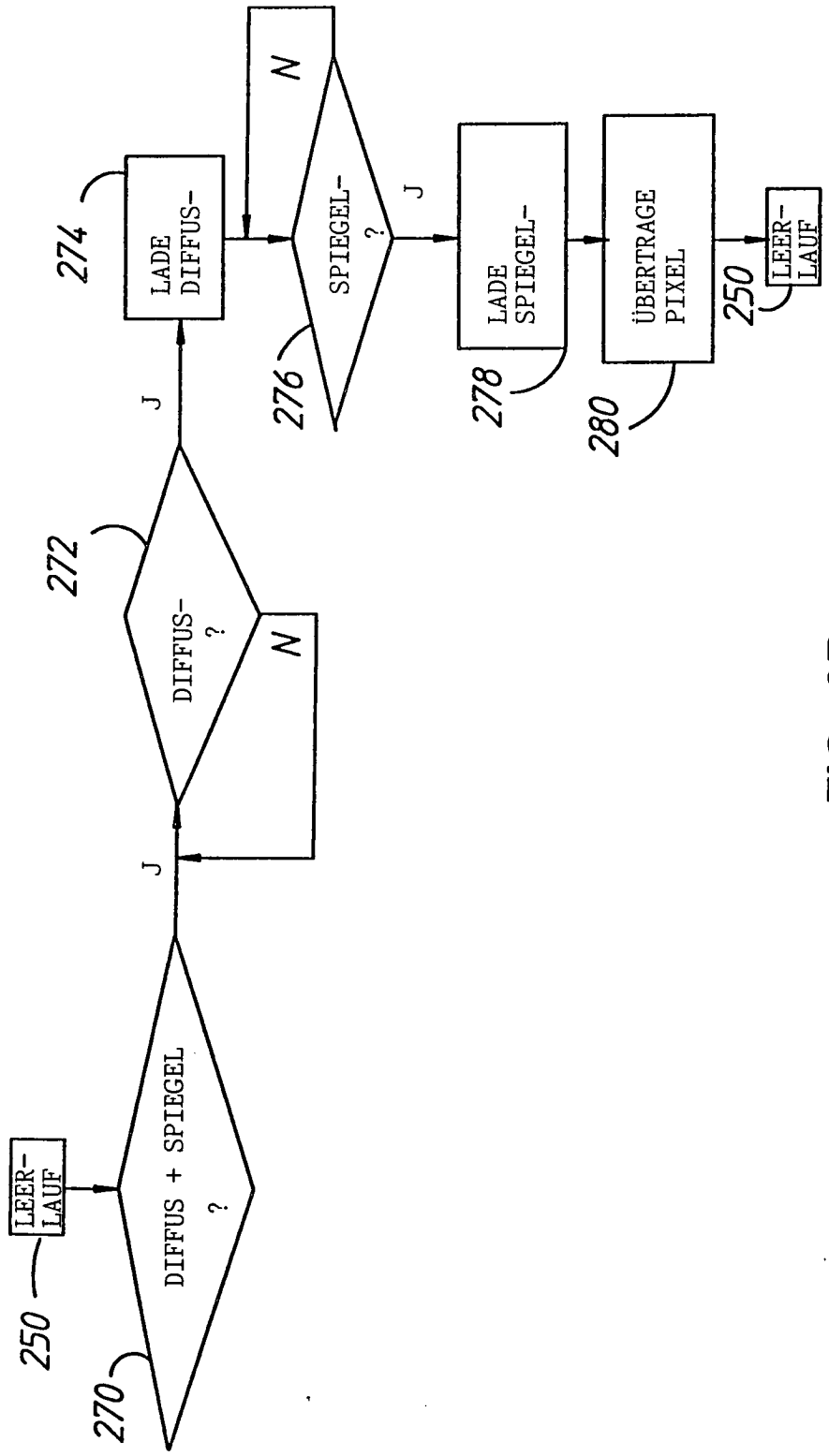

LEER-
LAUF
250
DIFFUS + SPIEGEL
?
270
J
DIFFUS-
?
272
N
J
LADE
DIFFUS-
274
N
SPIEGEL-
?
276
J
LADE
SPIEGEL-
278
ÜBERTRAGE
PIXEL
280
250
LEER-
LAUF

FIG 9B

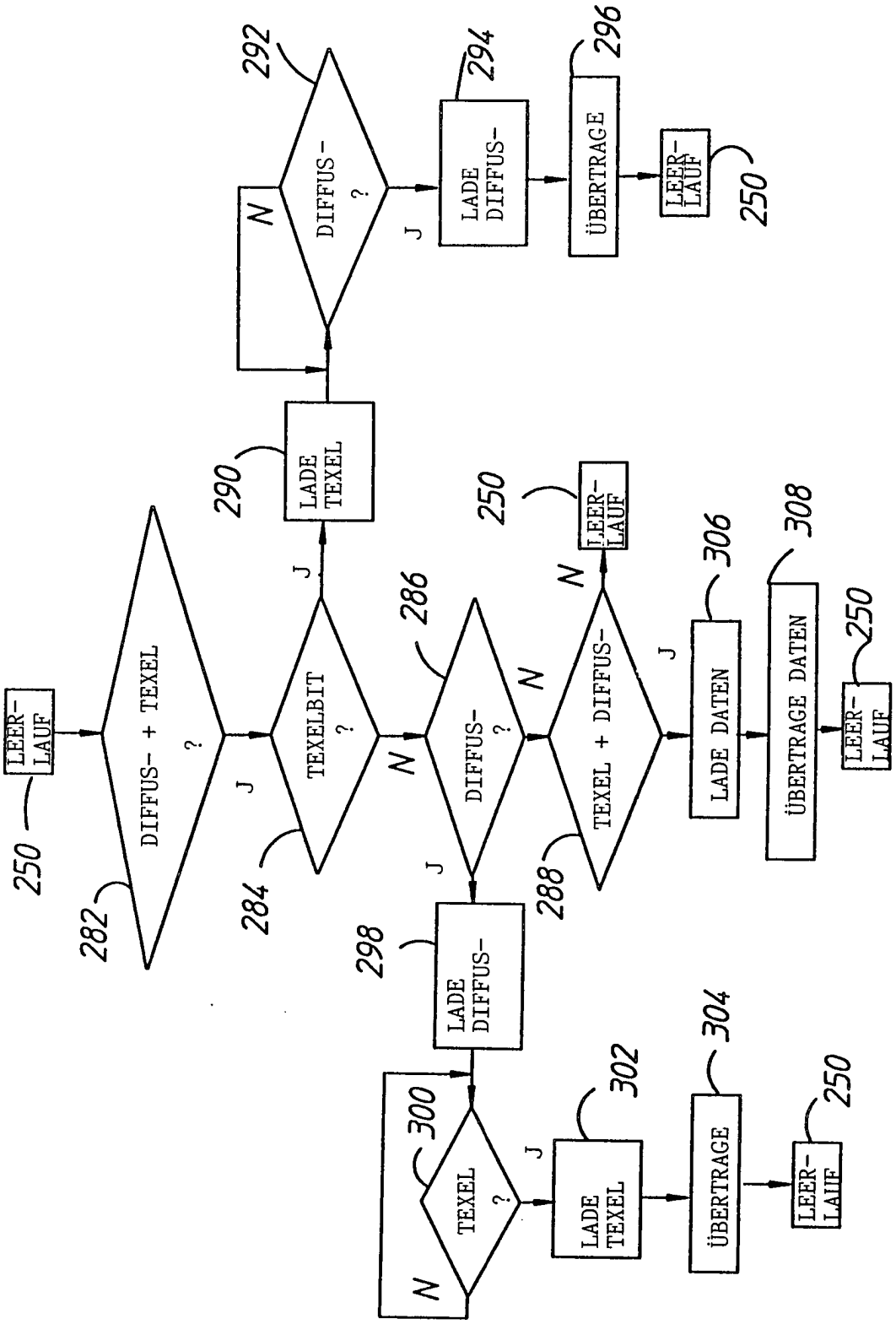

LEER-LAUF
250
DIFFUS- + TEXEL ?
282
J
TEXELBIT ?
284
J
LADE TEXEL
290
N
DIFFUS- ?
292
J
LADE DIFFUS-
294
ÜBERTRAGE
296
LEER-LAUF
250
N
DIFFUS- ?
286
J
LADE DIFFUS-
298
TEXEL ?
300
N
J
LADE TEXEL
302
ÜBERTRAGE
304
LEER-LAUF
250
N
TEXEL + DIFFUS-
288
J
LADE DATEN
306
ÜBERTRAGE DATEN
308
LEER-LAUF
250
N
LEER-LAUF
250

FIG 9C

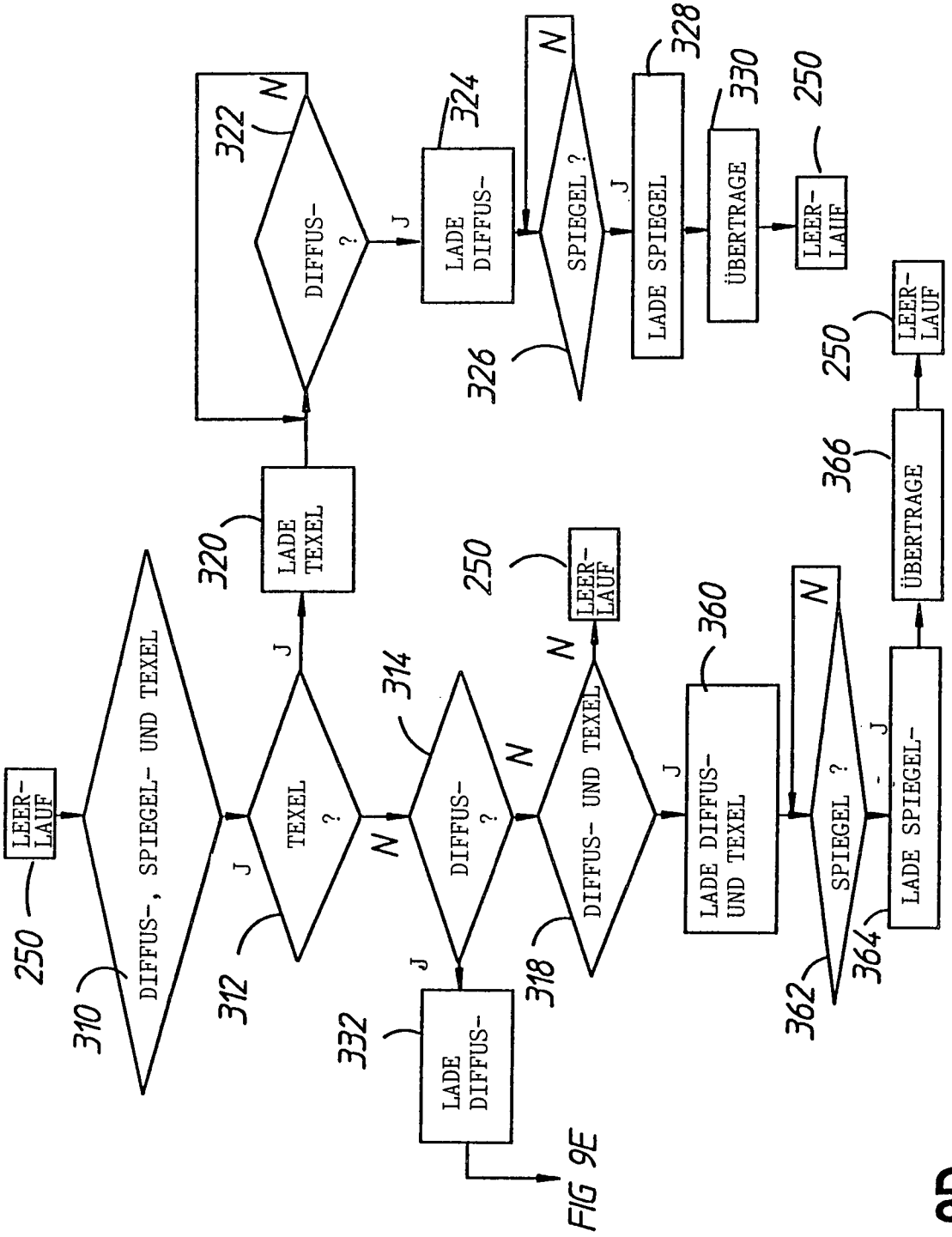

LEER-LAUF
250
DIFFUS-, SPIEGEL- UND TEXEL
310
TEXEL ?
312
J
N
LADE TEXEL
320
DIFFUS- ?
322
N
J
LADE DIFFUS-
324
SPIEGEL ?
326
N
J
LADE SPIEGEL
328
ÜBERTRAGE
330
LEER-LAUF
250
DIFFUS- ?
314
N
J
LADE DIFFUS-
332
FIG 9E
DIFFUS- UND TEXEL
318
N
J
LEER-LAUF
250
LADE DIFFUS-UND TEXEL
360
SPIEGEL ?
362
N
J
LADE SPIEGEL-
364
ÜBERTRAGE
366
LEER-LAUF
250

FIG 9D

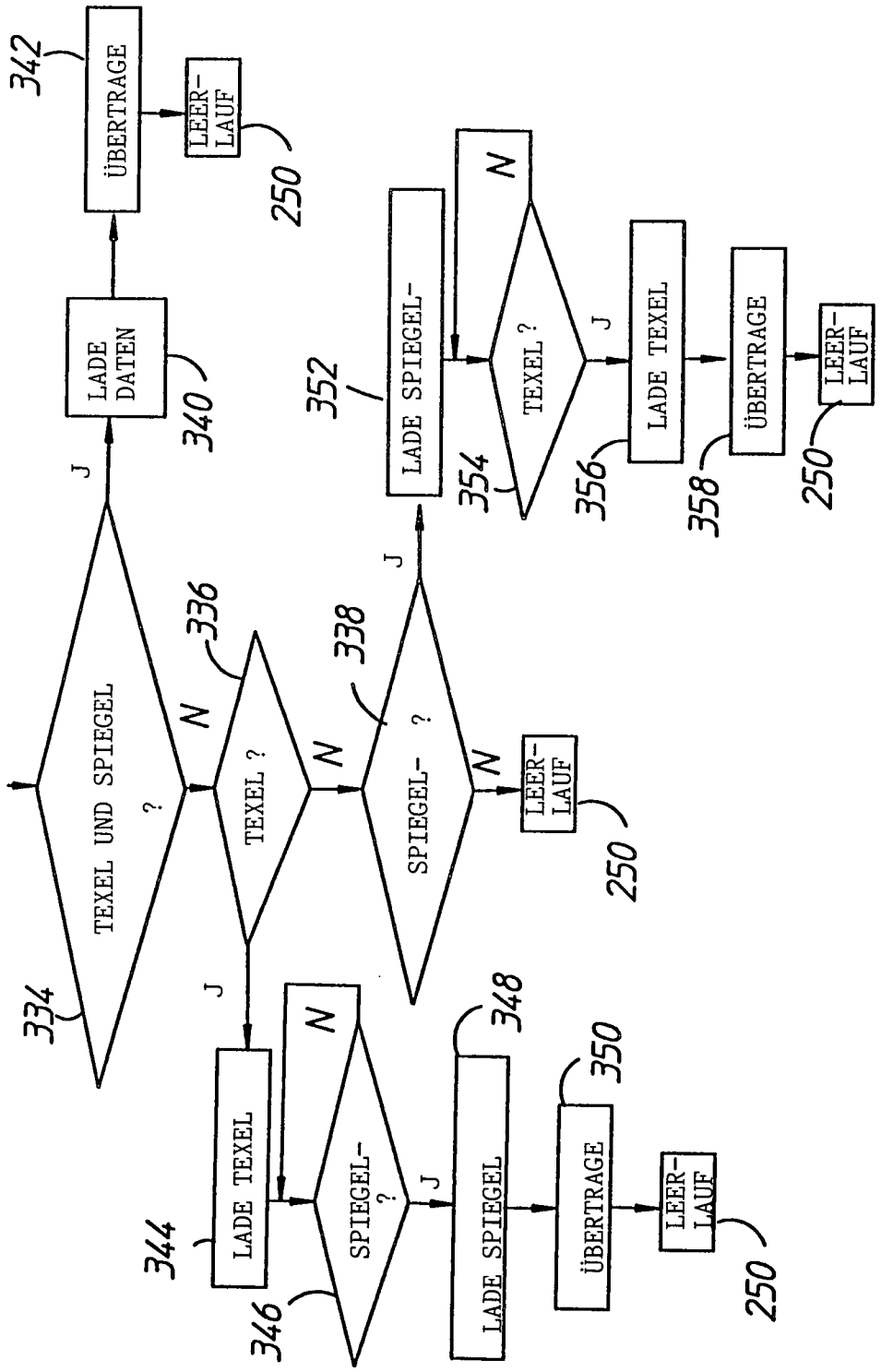

TEXEL UND SPIEGEL ?
J
N
TEXEL ?
N
SPIEGEL- ?
N
LEER-LAUF
LADE DATEN
ÜBERTRAGE
LEER-LAUF
LADE TEXEL
SPIEGEL- ?
LADE SPIEGEL
ÜBERTRAGE
LEER-LAUF
LADE SPIEGEL-
TEXEL ?
LADE TEXEL
ÜBERTRAGE
LEER-LAUF
334
336
338
340
342
344
346
348
350
352
354
356
358
250

FIG 9E